

# **New Analytics Paradigms in Online Advertising and Fantasy Sports**

Raghav Singal

Submitted in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy  
under the Executive Committee  
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2020

© 2020

Raghav Singal

All Rights Reserved

# Abstract

New Analytics Paradigms in Online Advertising and Fantasy Sports

Raghav Singal

Over the last two decades, digitization has been drastically shifting the way businesses operate and has provided access to high volume, variety, velocity, and veracity data. Naturally, access to such granular data has opened a wider range of possibilities than previously available. We leverage such data to develop application-driven models in order to evaluate current systems and make better decisions. We explore three application areas.

In Chapter 1, we develop models and algorithms to optimize portfolios in daily fantasy sports (DFS). We use opponent-level data to predict behavior of fantasy players via a Dirichlet-multinomial process, and our predictions feed into a novel portfolio construction model. The model is solved via a sequence of binary quadratic programs, motivated by its connection to outperforming stochastic benchmarks, the submodularity of the objective function, and the theory of order statistics. In addition to providing theoretical guarantees, we demonstrate the value of our framework by participating in DFS contests.

In Chapter 2, we develop an axiomatic framework for attribution in online advertising, i.e., assessing the contribution of individual ads to product purchase. Leveraging a user-level dataset, we propose a Markovian model to explain user behavior as a function of the ads she is exposed to. We use our model to illustrate limitations of existing heuristics and propose an original framework for attribution, which is motivated by causality and game theory. Furthermore, we establish that our framework coincides with an adjusted “unique-uniform” attribution scheme. This scheme is efficiently implementable and can be interpreted as a correction to the commonly used uniform attribution scheme. We supplement our theory with numerics using a real-world large-scale dataset.

In Chapter 3, we propose a decision-making algorithm for personalized sequential marketing. As in attribution, using a user-level dataset, we propose a state-based model to capture user behavior as a function of the ad interventions. In contrast with existing approaches that model only the myopic value of an intervention, we also model the long-run value. The objective of the firm

is to maximize the probability of purchase and a key challenge it faces is the lack of understanding of the state-specific effects of interventions. We propose a model-free learning algorithm for decision-making in such a setting. Our algorithm inherits the simplicity of Thompson sampling for a multi-armed bandit setting and we prove its asymptotic optimality. We supplement our theory with numerics on an email marketing dataset.

# Table of Contents

List of Figures . . . . .	iv
List of Tables . . . . .	ix
Acknowledgments . . . . .	x
Introduction . . . . .	1
Chapter 1: How to Play Fantasy Sports Strategically (and Win) . . . . .	4
1.1 Introduction . . . . .	4
1.2 Problem Formulation . . . . .	10
1.3 Modeling Opponents' Team Selections . . . . .	13
1.4 Solving the Double-Up Problem . . . . .	19
1.5 Solving the Top-Heavy Problem . . . . .	23
1.6 Numerical Experiments . . . . .	30
1.7 The Value of Modeling Opponents, Insider Trading, and Collusion . . . . .	37
1.8 Conclusions and Further Research . . . . .	45
Chapter 2: Shapley Meets Uniform: An Axiomatic Framework for Attribution in Online Advertising . . . . .	48
2.1 Introduction . . . . .	48
2.2 Model . . . . .	55
2.3 Current Attribution Approaches and Their Limitations . . . . .	59
2.4 Shapley Value (SV) . . . . .	64
2.5 CASV for the Markov Chain Model . . . . .	74
2.6 Numerical Experiments . . . . .	79
2.7 Value of State-Specific Attribution . . . . .	87

2.8	Conclusions and Further Research . . . . .	94
Chapter 3: Beyond Myopia: Model Free Approximate Bayesian Learning for Conversion		
	Funnel Optimization . . . . .	97
3.1	Introduction . . . . .	98
3.2	Consumer Model and Conversion Funnel Optimization . . . . .	107
3.3	Model-Free Approximate Bayesian Learning . . . . .	113
3.4	Properties of MFABL . . . . .	118
3.5	Model Extensions . . . . .	123
3.6	Numerical Experiments . . . . .	126
3.7	Value of Prior Information, Concept Shift, and Covariate Shift . . . . .	142
3.8	Conclusions and Further Research . . . . .	147
	Concluding Remarks . . . . .	151
	References . . . . .	153
Appendix A: Additional Details for Chapter 1 . . . . . 168		
A.1	Further Technical Details . . . . .	168
A.2	Parimutuel Betting Markets and Their Relation to DFS . . . . .	185
A.3	Further Details of Numerical Experiments . . . . .	194
Appendix B: Additional Details for Chapter 2 . . . . . 216		
B.1	Coalition Value . . . . .	216
B.2	Proof of Theorem 2.1 . . . . .	218
B.3	Sensitivity of CASV Due to Finite Size of Data . . . . .	221
B.4	Proofs of Theorem 2.3 and Proposition 2.2 . . . . .	224
B.5	Counterfactuals: Local, Global, and Forward-Looking . . . . .	225
Appendix C: Additional Details for Chapter 3 . . . . . 230		

C.1 Proof of Theorem 3.1 . . . . .	230
C.2 Algorithms for Model Extensions . . . . .	237

## List of Figures

1.1	Cumulative realized dollar P&L for the strategic and benchmark models across the three contest structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. . . . .	31
1.2	P&L distribution for the diversification strategy for the strategic and benchmark portfolios for week 10 contests of the 2017 NFL season. Recall $N = 50, 25,$ and $10$ for top-heavy, quintuple-up and double-up, respectively. The three metrics at the top of each image are the expected P&L, the standard deviation of the P&L and the probability of loss, that is, $\mathbb{P}(\text{P\&L} < 0)$ . . . . .	35
1.3	P&L distribution for the replication strategy for the strategic and benchmark portfolios for week 10 contests of the 2017 NFL season. Recall $N = 50, 25,$ and $10$ for top-heavy, quintuple-up and double-up, respectively. The three metrics at the top of each image are the expected P&L, the standard deviation of the P&L and the probability of loss, that is, $\mathbb{P}(\text{P\&L} < 0)$ . . . . .	35
1.4	Predicted and realized cumulative P&L for the strategic and benchmark models across the three contest structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. The realized cumulative P&Ls are displayed as points. . . . .	36
1.5	Weekly expected dollar P&L for the strategic model ( $N = 50$ ) with and without inside information $\mathbf{p}$ in the top-heavy series. . . . .	41
2.1	Network for Examples 2.1, 2.2, and 2.4. The action space consists of two actions: no-ad action ( $a = 1$ ) and ad action ( $a = 2$ ). The solid blue lines show transitions corresponding to the ad action. We assume the no-ad action in every state leads to a transition to the quit state w.p. 1. The advertiser takes the ad action at all states w.p. 1. . . . .	61



2.2	Network for Example 2.3. The action space consists of two actions: no-ad action ( $a = 1$ ) and ad action ( $a = 2$ ). The solid blue lines show transitions corresponding to the ad action. The no-ad action in every state leads to a transition to the quit state w.p. 1. The advertiser takes the ad action at all states w.p. 1. . . . .	62
2.3	Network for Example 2.5. The action space consists of two actions: no-ad action ( $a = 1$ ) and ad action ( $a = 2$ ). Solid blue (dashed red) lines denotes transitions for the ad action (no-ad action). The advertiser takes the ad action at state 1 w.p. 1, i.e., $\beta_1^2 = 1$ and $\beta_1^1 = 0$ . . . . .	68
2.4	Attributions to different actions under various schemes with $\tau = 10$ . We report the percentage attributions to each action by aggregating over states, i.e., $\sum_s \pi_s^a / \sum_{(s',a')} \pi_{s'}^{a'}$ for each $a \in \mathbb{A}$ . . . . .	85
2.5	Attributions to state-action pairs under various schemes with $\tau = 10$ . We report the attributions $\pi_s^a$ for all $(s, a) \in \mathbb{S} \times \mathbb{A}$ . The four colors correspond to the four states in $\mathbb{S}$ , arranged in the natural order (unaware, aware, interest, desire). . . . .	86
2.6	Attributions to different actions under various schemes with $\tau = 7$ . We report the percentage attributions to each action by aggregating over states, i.e., $\sum_s \pi_s^a / \sum_{(s',a')} \pi_{s'}^{a'}$ for each $a \in \mathbb{A}$ . . . . .	88
2.7	Attributions to different actions under various schemes with $\tau = 14$ . We report the percentage attributions to each action by aggregating over states, i.e., $\sum_s \pi_s^a / \sum_{(s',a')} \pi_{s'}^{a'}$ for each $a \in \mathbb{A}$ . . . . .	89
2.8	Value of state-specific attribution on the simulated dataset. For the state-specific model, we report the percentage attributions to each action by aggregating over states, i.e., $\sum_s \psi_s^{a, \text{Shap}} / \sum_{(s',a')} \psi_{s'}^{a', \text{Shap}}$ for each $a \in \mathbb{A}$ . For the aggregated model, we report $\bar{\psi}^{a, \text{Shap}}$ for each $a \in \mathbb{A}$ . . . . .	94

3.1	Sequence of emails received by one of the authors after providing his email to Netflix (but not subscribing to the membership). The emails were sent with a 15-20 days gap in between (May 3, May 18, and June 8) and the contents of each of the email were unique. The subject line of the three emails were “Movies & TV shows your way”, “Watch TV shows & movies anytime, anywhere”, and “Netflix - something for everyone”, respectively. . . . .	99
3.2	Consumer journey as a function of her interactions with the firm. On average, out of 100 consumers who sign-up, around 65, 30, and 2 receive, open, and click an email, respectively. Around 1.5 convert. As a consumer interacts more with the firm, her eventual conversion probability increases from 1.5% to 7%. . . . .	129
3.3	Benchmarking MFABL / MFABL2 with model-free algorithms. . . . .	138
3.4	Benchmarking MFABL2 with model-based algorithms. . . . .	139
3.5	Summarizing the various algorithms across two dimensions: (1) performance and (2) scalability. The scale of the y-axis (performance) is consistent with the performance ratios in Figures 3.3(a) and 3.4(a). The x-axis (scalability) is partitioned into 3 levels as follows. PSRL is least scalable since it stores the entire transition structure and solves an MDP per consumer (i.e., $N$ MDPs across all consumers) whereas ETO stores the entire transition structure but only solves one MDP (across all consumers). All other algorithms are model-free and exhibit similar scalability. .	140
3.6	Sensitivity of MFABL2 and QL with respect to $\epsilon$ . (Note that QL with $\epsilon = 0$ is equivalent to the random policy.) . . . . .	141
3.7	Sensitivity of mETO and ETO with respect to $N_1$ . . . . .	142
3.8	Performance ratios corresponding to perturbations in the transition probabilities. . .	143
3.9	Value of prior information corresponding to MFABL2. . . . .	144
3.10	Performance ratio for concept shift corresponding to various values of $N_1$ with $N = 10,000$ . . . . .	145
3.11	Regret for concept shift corresponding to various values of $N_1$ with $N = 10,000$ . . .	146

3.12	Illustration of concept shift for $N_1 = 50,000$ with $N = 100,000$ . . . . .	146
3.13	Covariate shift illustration. . . . .	148
A.1	Payoffs corresponding to ranks 1 to 8 of the week 10 top-heavy contest we participated in during the 2017 NFL season. We note the payoff structure of the top-heavy contests in other weeks were very similar and we do not show them here for the sake of brevity. . . . .	179
A.2	Predicted and realized QB ownerships ( $\mathbf{p}_{\text{QB}}$ ) for week 10 contests of the 2017 NFL season. . . . .	205
A.3	Highlighting instances where the Dirichlet regression either under-predicted or over-predicted ownerships of some athletes (“before”) and what would have happened (“after”) if we had (a) reacted to breaking news or (b) access to better quality features that accounted for historical factors such as Brady’s poor track record in Miami. . . . .	207
A.4	Predicted and realized portfolio points total of various opponent ranks for (a) the top-heavy contest in week 10 and (b) all weeks of the double-up series during the 2017 NFL season. For double-up, the rank of interest for each week was around 13,000 and the number of opponents was around 30,000. . . . .	208
A.5	Screenshot of a web-page from FanDuel.com when we click on an opponent lineup. The lineup has 9 athletes. For each athlete selected in the lineup we can observe the realized positional marginal of that athlete in the underlying contest. For example in the contest corresponding to this screenshot the realized positional marginal of Matthew Stafford equals 8.9%. . . . .	210
B.1	The action space consists of two actions: no-ad-action ( $a = 1$ ) and ad-action ( $a = 2$ ). Solid blue (dashed red) lines denote transitions if the ad-action (no-ad-action) is taken. The advertiser takes the ad-action at state 1 w.p. 1 and takes the no-ad-action in state 2 w.p. 1, i.e., $\beta_1^2 = \beta_2^1 = 1$ . . . . .	217

B.2 Summary of the sensitivity experiment. Each column corresponds to an ad action and each row corresponds to a different value of  $D$ . The dotted red line denotes the true CASV and the histogram denotes the distribution of the estimator. The solid black line equals the empirical mean of the distribution. (We report the percentage attributions to each action by aggregating over states, i.e.,  $\sum_s \psi_s^a / \sum_{(s', a')} \psi_{s'}^{a'}$  for each  $a \in \mathbb{A}$ .) . . . . . 223

B.3 Network for the proof of Proposition 2.2. The action space consists of three actions: show no-ad, show ad 1, and show ad 2. Solid blue (dashed brown) lines denote transitions if an ad 1 (ad 2) is shown. At state 1, taking action 1 moves the traffic to state 2 w.p. 1 whereas taking action 2 directs the traffic to quit state. At state  $s \in \{2, \dots, m\}$ , taking action 1 results in a transition to the quit state whereas action 2 moves the users to the “next” state. For brevity, we do not show the transitions if an ad is not shown (to quit state w.p. 1). The advertiser shows ad 1 at state 1 w.p. 1 and ad 2 at all other states w.p. 1. . . . . 225

B.4 There are two days (day A and day B) and the state equals (day, number of ads seen so far). For notational simplicity, we use “A0” to denote state (A, 0), “B0” to denote state (B, 0), and “B1” to denote state (B, 1). The action space consists of two actions: no-ad-action ( $a = 1$ ) and ad-action ( $a = 2$ ). Solid blue (dashed red) lines denote transitions if an ad-action (no-ad-action) is taken. The advertiser takes the ad-action on both the days w.p. 1. User’s decision to make a purchase is made after the second day and she converts if and only if she sees *at least* 1 ad in total. . . 227

## List of Tables

1.1	Total expected dollar P&L (over 17 weeks), average weekly Sortino ratio and average weekly probability of loss related to the top-heavy contests for both the non-colluding (“NC”) and colluding (“C”) portfolios with $E_{\max} = 50$ and $N_{\text{collude}} \in \{1, \dots, 5\}$ . The average weekly Sortino ratio is simply the average of the weekly Sortino ratios, $SR_i$ for $i = 1, \dots, 17$ . Specifically $SR_i := (\mathbb{E}[\text{P\&L}_i] - T) / DR_i$ where $\mathbb{E}[\text{P\&L}_i]$ denotes the expected P&L for week $i$ , $T$ denotes the target P&L which we set to 0, and $DR_i := \sqrt{\mathbb{E}[\text{P\&L}_i^2 \times \mathbb{1}_{\{\text{P\&L}_i \leq T\}]}$ denotes the downside risk for week $i$ . (The expected P&L is rounded to the nearest integer whereas the Sortino ratio and probability of loss are rounded to two decimal places.) . . . . .	44
3.1	Classification of the related marketing literature into four sets. Most of the traditional work belongs to set 1 and there has been noticeable progress in sets 2 and 3 over the last two decades. We focus on set 4, which is a relatively unexplored space in marketing. . . . .	102
3.2	Classification of the benchmark algorithms into four sets we introduced in our literature review. . . . .	135
A.1	Cumulative realized dollar P&L for the strategic and benchmark models across the three contest structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. We invested \$50 per week per model in top-heavy series with each entry costing \$1. In quintuple-up the numbers were \$50 per week per model with each entry costing \$2 and in double-up we invested \$20 per week per model with each entry costing \$2. (We were unable to participate in the quintuple-up contest in week 1 due to logistical reasons.) . . . . .	199

A.2	Performance of each week’s best realized entry for the strategic and benchmark models corresponding to the top-heavy contests for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. . . . .	200
A.3	Various statistics of interest for the ex-ante optimal entry $w_1^*$ of the strategic model across all three reward structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. Mean and StDev refer to the expected fantasy points and its standard deviation. (We were unable to participate in the quintuple-up contest in week 1 due to logistical reasons.) . . . . .	201
A.4	Mean fantasy points and its standard deviation for the first optimal entry $w_1^*$ of the benchmark model across all three reward structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. (We were unable to participate in the quintuple-up contest in week 1 due to logistical reasons.) . . . . .	202
A.5	Characteristics of the QB picked by the best performing entry of the strategic and benchmark models in the top-heavy contests for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. . . . .	203
A.6	Posterior predictive test summary statistic for each variation (denoted by V1, V2, and V3) of the Dirichlet regression model across all reward structures corresponding to the QB, RB, and WR positions. . . . .	212
A.7	Bayesian $p$ -values for the test statistic “most-picked athlete” for each variation of the Dirichlet regression model and each week corresponding to the QB, RB, and WR positions in the top-heavy reward structure. . . . .	214
A.8	Bayesian $p$ -values for the test statistic “most-picked athlete” for each variation of the Dirichlet regression model and each week corresponding to the QB, RB, and WR positions in the double-up reward structure. . . . .	214
A.9	Comparing the three variations of the Dirichlet regression model using normalized cross-validation scores for each position and each reward structure. . . . .	215

## Acknowledgements

The past few years have been some of the best in my academic life, thanks to the wonderful people I have had the pleasure to interact with. First, I would like to thank my advisors: Garud, Omar, and Vineet. Garud, thank you for supporting me throughout and giving me the freedom to pursue my varied interests. Omar, thank you for always pushing me to pursue research that matters. Your critical evaluation has been very valuable. Vineet, thank you for being patient with me and providing me useful advice whenever needed. I would also like to thank Martin for advising me in the initial years and teaching me the importance of rigorous thinking. Professor Bienstock, thank you for being on my dissertation committee and helping me to develop as a teacher. I consider myself fortunate to have had the opportunity to work with and learn from all of you.

A core part of my doctoral studies has been learning and I am grateful to all the instructors who taught me invaluable lessons. Thank you Dan for teaching me the fundamentals of reinforcement learning and guiding me on our work. Special thanks to Professors Andrew Gelman and John Paisley for their lectures on Bayesian thinking. I would also like to thank all the people who helped me prepare for the job market. Adam, Carri, Omar, Shipra, and Vineet, thank you for the mock interviews. Antoine, Auyon, and Tim, thank you for being there for me. Antoine, it was a pleasure working with you.

I very much enjoyed the vibrant community at Columbia and I would like to thank the IEOR staff for creating such an experience. Lizbeth, thank you for your open-door policy. Kristen, thank you for all the positivity. Jaya and Shi, thank you for keeping my bank balance positive. Carmen, Chinnell, Gerald, Jenny, Krupa, Lola, Mercedes, Mindi, Raven, and Yosimir, thank you for all your help and the hallway conversations. I appreciate the work all of you put behind in organizing events year-round – they added intangible value to my PhD life. I also thank the staff at CUIT for their support on cluster computing. Special thanks to the CKGSB fellowship for funding me.

Like most PhD journeys, mine has been a rollercoaster too. I thank all my friends for keeping me sane throughout the ride. Goutam, thank you for your friendship since day one. Jalaj, thank you for mentoring me. Allen, thank you for the study sessions. Omar and Rajan, thank you for the job

market gossip. Coming to the office always felt homely, thanks to the IEOR PhD family. Special thanks to Ana for being my study buddy, Bethany for being my running partner, undergraduate friends for the annual getaway, and friends in India for everlasting friendships. I also thank the I-House community for an invaluable experience.

Mom and Dad, thank you for being my backbone. Manu, thank you for being a role model. To my family in Canada and USA, thank you for making me feel at home since I moved to this continent. This work would not have been possible without you all.



# Introduction

This thesis focuses on application-driven developments in the data rich new economy. Over the last two decades, digitization has been drastically shifting the way businesses operate and has provided access to high volume, variety, velocity, and veracity data. Naturally, access to such granular data has opened a wider range of possibilities than previously available. In this thesis, we leverage such data to develop application-driven models in order to evaluate current systems and make better decisions.

In Chapter 1, we develop models and algorithms to optimize portfolios in daily fantasy sports (DFS). DFS is a multi-billion dollar industry with millions of annual users and widespread appeal among sports fans across a broad range of popular sports. In this chapter, building on the recent work of Hunter et al. 2016, we provide a coherent framework for constructing DFS portfolios where we explicitly model the behavior of other DFS players. We formulate an optimization problem that accurately describes the DFS problem for a risk-neutral decision-maker in both double-up and top-heavy payoff settings. Our formulation maximizes the expected reward subject to feasibility constraints and we relate this formulation to mean-variance optimization and the out-performance of stochastic benchmarks. Using this connection, we show how the problem can be reduced to the problem of solving a series of binary quadratic programs. We also propose an algorithm for solving the problem where the decision-maker can submit multiple entries to the DFS contest. This algorithm is motivated by submodularity properties of the objective function and by some new results on parimutuel betting. One of the contributions of our work is the introduction of a Dirichlet-multinomial data generating process for modeling opponents' team selections and we estimate the parameters of this model via Dirichlet regressions. A further benefit to modeling opponents' team selections is that it enables us to estimate the value in a DFS setting of both insider trading and collusion. We demonstrate the value of our framework by applying it to DFS contests during the 2017 NFL season. A preliminary version of this work was a finalist in the Sloan Sports Analytics Conference (2018) and the final version appeared as an article in Management Science (Haugh and Singal 2020).

In Chapter 2, we tackle one of the central challenges in online advertising: attribution. Simply put, we assess the contribution of individual advertiser actions such as e-mails, display ads and search ads to eventual conversion (purchase). Several heuristics are used for attribution in practice; however, most do not have any formal justification. The main contribution in this chapter is to propose an axiomatic framework for attribution in online advertising. We show that the most common heuristics can be cast under the framework and illustrate how these may fail. We propose a novel attribution metric, that we refer to as *counterfactual adjusted Shapley value* (CASV), which inherits the desirable properties of the traditional Shapley value while overcoming its shortcomings in the online advertising context. We also propose a Markovian model for the user journey through the conversion funnel, in which ad actions may have disparate impacts at different stages. We use the Markovian model to compare our metric with commonly used metrics. Furthermore, under the Markovian model, we establish that the CASV metric coincides with an adjusted “unique-uniform” attribution scheme. This scheme is efficiently implementable, and can be interpreted as a correction to the commonly used uniform attribution scheme. We supplement our theoretical developments with numerical experiments using a real-world large-scale dataset. A preliminary version of this work appeared in the WWW conference (Singal et al. 2019) and the current version is under revision at Management Science.

In Chapter 3, we study the problem of optimal sequential personalized interventions from the point-of-view of a firm promoting a product under the Markovian model proposed in Chapter 2. Our model captures the state of each consumer (interaction history with the firm for example) and allows the consumer behavior to vary as a function of both her state and firm’s interventions. In contrast with existing approaches that model only the myopic value of an intervention, we also model the long-run value by allowing the firm to make sequential interventions to the same consumer. The objective of the firm is to maximize the probability of conversion (consumer buying the product) and a key challenge is the firm does not know the state-specific effects of various interventions. To help make personalized intervention decisions, we propose a decision-making algorithm, which we call *model-free approximate Bayesian learning*. Our algorithm inherits the simplicity

of Thompson sampling for a multi-armed bandit setting and maintains an approximate belief over the value (myopic plus long-run) of each state-specific intervention. The belief is updated as the algorithm interacts with the consumers. Despite being an approximation to the Bayes update, we prove the asymptotic optimality of our algorithm. We supplement our theory with numerics on a real-world large-scale dataset, where we show the dominance of our algorithm over traditional approaches that are myopic or estimation-based. Furthermore, in contrast to the estimation-based approaches, our algorithm is able to adapt automatically to the underlying changes in consumer behavior (concept shift) and maintains a high level of uncertainty on the value of less explored consumer segments (covariate shift). Intuitively, one expects the value attributed to an advertising action (Chapter 2) to be connected to the decision-making problem we tackle in Chapter 3. We discuss this at the end of Chapter 3.

In addition to the application-driven work, we also pursued some theoretical research in reinforcement learning. Though not a part of this dissertation, an initial version of that work appeared in COLT (Bhandari et al. 2018) and the final version has been accepted to Operations Research.

# Chapter 1: How to Play Fantasy Sports Strategically (and Win)

Daily fantasy sports (DFS) is a multi-billion dollar industry with millions of annual users and widespread appeal among sports fans across a broad range of popular sports. In this chapter, building on the recent work of Hunter et al. 2016, we provide a coherent framework for constructing DFS portfolios where we explicitly model the behavior of other DFS players. We formulate an optimization problem that accurately describes the DFS problem for a risk-neutral decision-maker in both double-up and top-heavy payoff settings. Our formulation maximizes the expected reward subject to feasibility constraints and we relate this formulation to mean-variance optimization and the outperformance of stochastic benchmarks. Using this connection, we show how the problem can be reduced to the problem of solving a series of binary quadratic programs. We also propose an algorithm for solving the problem where the decision-maker can submit multiple entries to the DFS contest. This algorithm is motivated by submodularity properties of the objective function and by some new results on parimutuel betting. One of the contributions of our work is the introduction of a Dirichlet-multinomial data generating process for modeling opponents' team selections and we estimate the parameters of this model via Dirichlet regressions. A further benefit to modeling opponents' team selections is that it enables us to estimate the value in a DFS setting of both insider trading and collusion. We demonstrate the value of our framework by applying it to DFS contests during the 2017 NFL season. A preliminary version of this work was a finalist in the Sloan Sports Analytics Conference (2018) and the final version appeared as an article in *Management Science* (Haugh and Singal 2020).

## 1.1 Introduction

Daily fantasy sports (DFS) has become a multi-billion dollar industry (Anderton 2016; Kolodny 2015; O'Keeffe 2015; Wong 2015; Woodward 2016) with millions of annual users (FSTA 2015; Wong 2015). The pervasiveness of fantasy sports in modern popular culture is reflected by the

regular appearance of articles discussing fantasy sports issues in the mainstream media. Moreover, major industry developments and scandals are now capable of making headline news (Drape and Williams 2015b; Johnson 2016). The two major DFS websites are FanDuel and DraftKings and together they control approximately 95% of the U.S. market (Kolodny 2015; O’Keeffe 2015). Approximately 80% of DFS players have been classified as *minnows* (Pramuk 2015) as they are not believed to use sophisticated techniques for decision-making and portfolio construction. Accordingly, these users provide financial opportunities to the so-called *sharks* who do use sophisticated techniques (Harwell 2015; Mulshine 2015; Pramuk 2015; Woodward 2015) when constructing their fantasy sports portfolios. The goal of this chapter is to provide a coherent framework for constructing fantasy sports portfolios where we explicitly model the behavior of other DFS players. Our approach is therefore strategic and to the best of our knowledge, we are the first academic work to develop such an approach in the context of fantasy sports.

The number of competitors in a typical DFS contest might range from two to hundreds of thousands with each competitor constructing a fantasy team of real-world athletes, e.g. National Football League (NFL) players in a fantasy football contest, with each portfolio being subject to budget and possibly other constraints. The performance of each portfolio is determined by the performances of the real-world athletes in a series of actual games, e.g. the series of NFL games in a given week. The competitors with the best performing entries then earn a monetary reward, which depends on the specific payoff structure, e.g. *double-up* or *top-heavy*, of<sup>1</sup> the DFS contest.

Several papers have already been written on the topic of fantasy sports. For example, Fry et al. 2007 and Becker and Sun 2016 develop models for season-long fantasy contests while Bergman and Imbrogno 2017 propose strategies for the survivor pool contest, which is also a season long event. Multiple papers have been written of course on so-called office pools (which pre-date fantasy sports contests) where the goal is to predict the maximum number of game winners in an upcoming elimination tournament such as the *March Madness* NCAA college basketball tourna-

---

<sup>1</sup>Loosely speaking, in a double-up contest a player doubles her money if her entry is among the top 50% of submitted entries. In a top-heavy contest, the rewards are skewed towards the very best performing entries and often decrease rapidly in the rank of the entry. See Section 1.2 for further details.

ment. Examples of this work include Kaplan and Garstka 2001 and Clair and Letscher 2007. There has been relatively little work, however, on the problem of constructing portfolios for daily fantasy sports. One notable exception is the recent work of Hunter et al. 2016, which is closest to the work we present in this chapter. They consider a winner-takes-all payoff structure and aim to maximize the probability that one of their portfolios (out of a total of  $N$ ) wins. Their approach is a greedy heuristic that maximizes their portfolio means, that is, expected number of fantasy points, subject to constraints that lower bound their portfolio variances and upper bound their inter-portfolio correlations. Technically, their framework requires the solution of linear integer programs and they apply their methodology to fantasy sports contests which are *top-heavy* in their payoff structure as opposed to winner-takes-all. Their work has received considerable attention, e.g. Davis 2017, and the authors report earning<sup>2</sup> significant sums in real fantasy sports contests based on the National Hockey League (NHL) and Major League Baseball (MLB).

There are several directions for potential improvement, however, and they are the focus of the work in this chapter. First, Hunter et al. 2016 do not consider their opponents' behavior. In particular, they do not account for the fact that the payoff thresholds are stochastic and depend on both the performances of the real-world athletes as well as the unknown team selections of their fellow fantasy sports competitors. Second, their framework is only suitable for contests with the top-heavy payoff structure and is in general not suitable for the double-up payoff structure. Third, their approach is based on (approximately) optimizing for the winner-takes-all payoff, which is only a rough approximation to the top-heavy contests they ultimately target. In contrast, we directly model the true payoff structure (top-heavy or double-up) and seek to optimize our portfolios with this objective in mind.

Our work makes several contributions to the DFS literature. First, we formulate an optimization problem that accurately describes the DFS problem for a risk-neutral decision-maker in both double-up and top-heavy settings. Our formulation seeks to maximize the expected reward subject to portfolio feasibility constraints and we explicitly account for our opponents' unknown portfolio

---

<sup>2</sup>They donated their earnings to charity and we have done likewise with our earnings from playing DFS competitions during the 2017 NFL season. The results of these real-world numerical experiments are described in Section 1.6.

choices in our formulation. Second, we connect our problem formulation to the finance literature on mean-variance optimization and in particular, the mean-variance literature on outperforming stochastic benchmarks. Using this connection, we show how our problems can be reduced (via some simple assumptions and results from the theory of order statistics) to the problem of solving a series of binary quadratic programs. The third contribution of our work is the introduction of a Dirichlet-multinomial data generating process for modeling opponents' team selections. We estimate the parameters of this model via Dirichlet regressions and we demonstrate its value in predicting opponents' portfolio choices.

We also show the DFS objective function for the problem with multiple entries is monotone submodular under certain conditions that often apply approximately in practice. A classic result (Nemhauser et al. 1978) on submodular maximization then suggests that a greedy algorithm should perform very well on this problem. Unfortunately, it's not possible to implement this greedy algorithm and so instead we propose a modified version of it that we can implement. Further support for our algorithm is provided by some new results for the optimization of wagers in a parimutuel contest. Such a contest can be viewed as a special case of a DFS contest albeit with some important differences. Parimutuel betting in the horse-racing industry has long been a topic of independent interest in its own right, particularly in economics (Bayraktar and Munk 2017; Plott et al. 2003; Terrell and Farmer 1996; Thaler and Ziemba 1988), where it has been used to test theories related to market efficiency and information aggregation.

We demonstrate the value of our framework by applying it to both double-up and top-heavy DFS contests in the 2017 NFL season. Despite the fact that DFS contests have a negative net present value (NPV) on average (due to the substantial cut taken by the major DFS websites), we succeeded in earning a net profit over the course of the season. That said, model performance in DFS contests based on a single NFL season has an inherently high variance and so it is difficult to draw meaningful empirical conclusions from just one NFL season. Indeed other sports (baseball, ice hockey, basketball etc.) should have a much lower variance and we believe our approach is particularly suited to these sports.

We also use our model to estimate the value of “insider trading”, where an insider, e.g. an employee of the DFS contest organizers, gets to see information on opponents’ portfolio choices before making his own team selections. This has been a topic of considerable recent media interest (Drape and Williams 2015a; Drape and Williams 2015b) which was sparked by the case of a DraftKings employee who won \$350,000 in a FanDuel DFS contest by using data from similar DraftKings contests to construct his entries. This problem of insider trading is of course also related to the well known value-of-information concept from decision analysis. While insider trading does result in an increase in expected profits, the benefits of insider trading are mitigated by superior modeling of opponents’ team selections. This is not surprising: if we can accurately predict the distribution of opponents’ team selection, then insider information on the composition of these portfolios will be less valuable.

It is also straightforward in our framework to study the benefits of a stylized form of collusion in DFS contests. Specifically, we consider the case where a number  $N_{\text{collude}}$  of DFS players combine to construct a single portfolio of  $N_{\text{collude}} \times E_{\text{max}}$  entries for a given contest, where  $E_{\text{max}}$  is the maximum number of permitted entries per DFS player. In contrast, we assume that non-colluders choose identical portfolios of  $E_{\text{max}}$  entries. We show the benefits of this type of collusion can be surprisingly large in top-heavy contests. This benefit is actually twofold in that colluding can simultaneously result in a significant increase in the total expected payoff and a significant reduction in the downside risk of the payoff. In practice, however, it’s highly unlikely that non-colluding players will choose identical portfolios and so we argue that the benefits of collusion to a risk-neutral player are likely to be quite small.

Beyond proposing a modeling framework for identifying how to construct DFS portfolios, our work also has other implications. To begin with, it should be clear from our general problem formulation and solution approach that high levels of “skill” are required to play fantasy sports successfully. But this is not necessarily in the interest of the fantasy sports industry. In order to maintain popular interest (and resulting profit margins), the industry does not want the role of skill to be too great. Indeed a recent report from McKinsey & Company (Miller and Singer 2015)



on fantasy sports makes precisely this point arguing, for example, that chess is a high-skill and deterministic game, which is why it is rarely played for money. In contrast, while clearly a game of high skill, poker also has a high degree of randomness to the point that amateur players often beat professionals in poker tournaments. It is not surprising then that poker is very popular and typically played for money. The framework we have developed in this chapter can be used by the fantasy sports industry to determine whether the current DFS game structures achieve a suitable balance between luck and skill. One simple “lever” to adjust this balance, for example, would be to control the amount of data they release regarding the teams selected by the DFS players. By choosing to release no information whatsoever, it will become more difficult for skillful players to estimate their models and take advantage of their superior modeling skills. The industry can also use (as we do) our framework to estimate the value of insider trading and collusion and propose new rules / regulations or payoff structures to counter these concerns.

A recent relevant development occurred in May 2018 when the U.S. Supreme Court struck down a 1992 federal law – the Professional and Amateur Sports Protection Act – that prohibited states from authorizing sports gambling. As a result, some states are taking advantage of this ruling by passing their own sports betting laws and encouraging gambling with the goal of raising additional tax revenue. This remains a controversial development but would certainly appear to be a positive development for the fantasy sports industry. To the extent that individual states seek to regulate online gambling and DFS, the “skill-versus-luck” debate (referenced in the preceding paragraph) may continue to play a role as it has done historically in the federal regulation of gambling in the U.S.

The remainder of this chapter is organized as follows. In Section 1.2, we formulate both the double-up and top-heavy versions of the problem while we outline our Dirichlet regression approach to modeling our opponents’ team selections in Section 1.3. In Section 1.4, we use results from mean-variance optimization (that relate to maximizing the probability of outperforming a stochastic benchmark) to solve the double-up problem. We then extend this approach to solve the top-heavy problem in Section 1.5, where we prove the submodularity of the objective function.

We present numerical results based on the 2017 NFL season for both problem formulations in Section 1.6. In Section 1.7 we discuss the value of information and in particular, how much an insider can profit from having advance knowledge of his opponents’ team selections. We also consider the benefits of collusion there. We conclude in Section 1.8, where we also discuss some directions for ongoing and future research. Various technical details and additional results are deferred to the appendices.

## 1.2 Problem Formulation

We assume there are a total of  $P$  athletes / real-world players whose performance,  $\delta \in \mathbb{R}^P$ , in a given round of games is random. We assume that  $\delta$  has mean vector  $\mu_\delta$  and variance-covariance matrix  $\Sigma_\delta$ . Our decision in the fantasy sports competition is to choose a portfolio  $\mathbf{w} \in \{0, 1\}^P$  of athletes. Typically, there are many constraints on  $\mathbf{w}$ . For example, in a typical NFL DFS contest, we will only be allowed to select  $C = 9$  athletes out of a total of  $P \approx 100$  to 300 NFL players. Each athlete also has a certain “cost” and our portfolio cannot exceed a given budget  $B$ . These constraints on  $\mathbf{w}$  can then be formulated as

$$\begin{aligned} \sum_{p=1}^P w_p &= C \\ \sum_{p=1}^P c_p w_p &\leq B \\ w_p &\in \{0, 1\}, \quad p = 1, \dots, P \end{aligned}$$

where  $c_p$  denotes the cost of the  $p^{\text{th}}$  athlete. Other constraints are also typically imposed by the contest organizers. These constraints include positional constraints, e.g. exactly one quarterback can be chosen, diversity constraints, e.g. you can not select more than 4 athletes from any single NFL team, etc. These constraints can generally be modeled as linear constraints and we use  $\mathbb{W}$  to denote the set of binary vectors  $\mathbf{w} \in \{0, 1\}^P$  that satisfy these constraints.

A key aspect of our approach to constructing fantasy sports portfolios is in modeling our oppo-

nents, that is, other DFS players who also enter the same fantasy sports contest. We assume there are  $O$  such opponents and we use  $\mathbf{W}_{\text{op}} := \{\mathbf{w}_o\}_{o=1}^O$  to denote their portfolios with each  $\mathbf{w}_o \in \mathbb{W}$ .

Once the round of NFL games has taken place, we get to observe the realized performances  $\boldsymbol{\delta}$  of the  $P$  NFL athletes. Our portfolio then realizes a points total of  $F := \mathbf{w}^\top \boldsymbol{\delta}$  whereas our opponents' realized points totals are  $G_o := \mathbf{w}_o^\top \boldsymbol{\delta}$  for  $o = 1, \dots, O$ . All portfolios are then ranked according to their points total and the cash payoffs are determined. These payoffs take different forms depending on the structure of the contest. There are two contest structures that dominate in practice and we consider both of them. They are the so-called double-up and top-heavy payoff structures.

### 1.2.1 The Double-Up Problem Formulation

Under the double-up payoff structure, the top  $r$  portfolios (according to the ranking based on realized points total) each earn a payoff of  $R$  dollars. Suppose now that we enter  $N \ll O$  portfolios<sup>3</sup> to the contest. Then, typical values of  $r$  are  $r = (O + N)/2$  and  $r = (O + N)/5$  with corresponding payoffs of  $R = 2$  and  $R = 5$  assuming an entry fee of 1 per portfolio. The  $(r = (O + N)/2, R = 2)$  case is called a double-up competition whereas the  $(r = (O + N)/5, R = 5)$  is called a *quintuple-up* contest. We will refer to all such contests as “double-up” contests except when we wish to draw a distinction between different types of double-up contests, e.g. (true) double-up versus quintuple-up. In practice of course, the contest organizers take a cut and keep approximately 15% of the entry fees for themselves. This is reflected by reducing  $r$  appropriately and we note that this is easily accounted for in our problem formulations below. We also note that this means the average DFS player loses approximately 15% of her initial entry. In contrast to financial investments then, DFS investments are on average NPV-negative and so some skill is required in portfolio construction to overcome this handicap.

While it is possible and quite common for a fantasy sports player to submit multiple entries, that is, multiple portfolios, to a given contest, we will consider initially the case where we submit

---

<sup>3</sup>There is usually a cap on  $N$ , denoted by  $E_{\text{max}}$ , imposed by the contest organizer, however. Typical cap sizes we have observed can range from  $E_{\text{max}} = 1$  to  $E_{\text{max}} = 150$ .

just  $N = 1$  entry. Given the double-up payoff structure, our fantasy-sports portfolio optimization problem is to solve

$$\max_{\mathbf{w} \in \mathbb{W}} \mathbb{P} \left\{ \mathbf{w}^\top \boldsymbol{\delta} > G^{(r')}(\mathbf{W}_{\text{op}}, \boldsymbol{\delta}) \right\}, \quad (1.1)$$

where we use  $G^{(r)}$  to denote the  $r^{\text{th}}$  order statistic of  $\{G_o\}_{o=1}^O$  and we define  $r' := O + 1 - r$ . Note that we explicitly recognize the dependence of  $G^{(r)}$  on the portfolio selections  $\mathbf{W}_{\text{op}}$  of our  $O$  opponents and the performance vector  $\boldsymbol{\delta}$  of the NFL athletes.

### 1.2.2 The Top-Heavy Problem Formulation

The top-heavy payoff structure is more complicated than the double-up structure as the size of the cash payoff generally increases with the portfolio ranking. In particular, we first define payoffs

$$R_1 > \cdots > R_D > R_{D+1} := 0$$

and corresponding ranks

$$0 := r_0 < r_1 < \cdots < r_D.$$

Then, a portfolio whose rank lies in  $(r_{d-1}, r_d]$  wins  $R_d$  for  $d = 1, \dots, D$ . In contrast to the double-up structure, we now account for the possibility of submitting  $N > 1$  entries to the contest. We use  $\mathbf{W} := \{\mathbf{w}_i\}_{i=1}^N$  to denote these entries and  $F_i := \mathbf{w}_i^\top \boldsymbol{\delta}$  to denote the realized fantasy points total of our  $i^{\text{th}}$  entry. It is then easy to see that our portfolio optimization problem is to solve for<sup>4</sup>

$$\max_{\mathbf{W} \in \mathbb{W}^N} \sum_{i=1}^N \sum_{d=1}^D (R_d - R_{d+1}) \mathbb{P} \left\{ \mathbf{w}_i^\top \boldsymbol{\delta} > G_{-i}^{(r'_d)}(\mathbf{W}_{-i}, \mathbf{W}_{\text{op}}, \boldsymbol{\delta}) \right\} \quad (1.2)$$

where  $r'_d := O + N - r_d$ ,  $G_{-i}^{(r)}$  is the  $r^{\text{th}}$  order statistic of  $\{G_o\}_{o=1}^O \cup \{F_j\}_{j=1}^N \setminus F_i$  and  $\mathbf{W}_{-i} := \mathbf{W} \setminus \mathbf{w}_i$ . (We note there is a slight abuse of notation here since duplicate entries are possible and so the

---

<sup>4</sup>The probability term in (1.2) involves a strict inequality “>” but we note that the objective should also include an additional term for  $\mathbb{P}(\mathbf{w}_i^\top \boldsymbol{\delta} = G_{-i}^{(r'_d)}(\mathbf{W}_{-i}, \mathbf{W}_{\text{op}}, \boldsymbol{\delta}))$  in which case a *share* of the reward  $(R_d - R_{d+1})$  would be earned. To keep our expressions simple, we don’t include this term in (1.2) as it is generally negligible (except when replication is used) but we do account correctly for such ties in all of our numerical results.

union operator “ $\cup$ ” should be interpreted with this in mind.)

Later in Section 1.5, we will discuss our approach to solving (1.2) and we will argue (based on the submodularity of the objective function in (1.2) and our parimutuel betting formulation in Appendix A.2) that diversification, i.e., choosing  $N$  different entries, is a near-optimal strategy. For top-heavy payoffs where the reward  $R_d$  decreases rapidly in  $d$ , it should be clear why diversification might be a good thing to do. Consider the extreme case of a winner-takes-all structure, for example. Then, absent pathological instances<sup>5</sup>, replication of entries means you are only giving yourself one chance to win. This is accounted for in (1.2) by the fact that your  $w_i^{th}$  entry is “competing” with your other  $N - 1$  entries as they together comprise  $\mathbf{W}_{-i}$ . In contrast, when you fully diversify, you are giving yourself  $N$  separate chances to win the prize in total. (We are ignoring here the possibility of sharing the prize.)

We note that the top-heavy payoff structure is our main concern in this chapter. That said, it should be clear that the double-up formulation of (1.1) is a special case of the top-heavy formulation in (1.2). We will therefore address the double-up problem before taking on the top-heavy problem. Before doing this, however, we must discuss the modeling of our opponents’ portfolios  $\mathbf{W}_{op}$ .

### 1.3 Modeling Opponents’ Team Selections

A key aspect of our modeling approach is that there is value to modeling our opponents’ portfolio choices,  $\mathbf{W}_{op}$ . This is in direct contrast to the work of Hunter et al. 2016 who ignore this aspect of the problem and focus instead on constructing portfolios that maximize the expected number of fantasy points, subject to possible constraints<sup>6</sup> that encourage high-variance portfolios. Based on numerical simulations of DFS contests during the 2017 NFL season, we noted it is possible to

---

<sup>5</sup>For example, if the best team could be predicted in advance with perfect accuracy, then choosing and replicating this team would be optimal since by replicating this entry you will be (a) guaranteed to win and (b) gain a greater share of the reward if some of your competitors also chose it. If none of your competitors chose the team, you will earn the entire reward for yourself.

<sup>6</sup>They included constraints that encouraged high-variance portfolios because they too were focused on top-heavy contests where very few contestants earn substantial payoffs. It is intuitively clear that high-variance portfolios are desirable for such contests. We discuss this property in further detail in Sections 1.4 and 1.5 in light of the results from mean-variance optimization that we bring to bear on the problem.

obtain significant gains in expected dollar payoffs by explicitly modeling  $\mathbf{W}_{\text{op}}$ . This is partly due to the well-known fact that some athletes are (often considerably) more / less popular than other athletes and because there is some predictability in the team selections of DFS players who may be responding to weekly developments that contain more noise than genuine information. To the best of our knowledge, we are the first to explicitly model  $\mathbf{W}_{\text{op}}$  and embed it in our portfolio construction process. That said, we certainly acknowledge that some members of the fantasy sports community also attempt to be strategic in their attempted selection of less popular athletes and avoidance of more popular athletes, other things being equal; see for example Gibbs 2017.

If we are to exploit our opponents' team selections, then we must be able to estimate  $\mathbf{W}_{\text{op}}$  reasonably accurately. Indeed it is worth emphasizing that  $\mathbf{W}_{\text{op}}$  is not observed before the contest and so we must make do with predicting / simulating it, which amounts to being able to predict / simulate the  $\mathbf{w}_o$ 's. To make things clear, we will focus on the specific case of DFS in the NFL setting. Specifically, consider for example the following NFL contest organized by FanDuel (FanDuel 2016). Each fantasy team has  $C = 9$  positions which must consist of 1 quarterback (QB), 2 running backs (RB), 3 wide receivers (WR), 1 tight end (TE), 1 kicker (K) and 1 "defense" (D). We now write  $\mathbf{w}_o = (\mathbf{w}_o^{\text{QB}}, \mathbf{w}_o^{\text{RB}}, \dots, \mathbf{w}_o^{\text{D}})$  where  $\mathbf{w}_o^{\text{QB}}$  denotes the quarterback component of  $\mathbf{w}_o$ ,  $\mathbf{w}_o^{\text{RB}}$  denotes the running back component of  $\mathbf{w}_o$  etc. If there are  $P_{\text{QB}}$  QBs available for selection then  $\mathbf{w}_o^{\text{QB}} \in \{0, 1\}^{P_{\text{QB}}}$  and exactly one component of  $\mathbf{w}_o^{\text{QB}}$  will be 1 for any feasible  $\mathbf{w}_o$ . In contrast,  $\mathbf{w}_o^{\text{RB}}$  and  $\mathbf{w}_o^{\text{WR}}$  will have exactly two and three components, respectively, equal to 1 for any feasible  $\mathbf{w}_o$ . We refer to  $\mathbf{w}_o^{\text{QB}}$ ,  $\mathbf{w}_o^{\text{RB}}$  etc. as the *positional marginals* of  $\mathbf{w}_o$ . Moreover, it follows that  $P_{\text{QB}} + P_{\text{RB}} + \dots + P_{\text{D}} = P$  since there are  $P$  athletes in total available for selection.

In order to model the distribution of  $\mathbf{w}_o$ , we will use a classic result from copula theory (Nelsen 2007), namely Sklar's theorem (Sklar 1959). This theorem states that we can write

$$F_{\mathbf{w}_o}(\mathbf{w}_o^{\text{QB}}, \dots, \mathbf{w}_o^{\text{D}}) = C(F_{\text{QB}}(\mathbf{w}_o^{\text{QB}}), \dots, F_{\text{D}}(\mathbf{w}_o^{\text{D}})) \quad (1.3)$$

where  $F_{\mathbf{w}_o}$  denotes the CDF of  $\mathbf{w}_o$ ,  $F_{\text{QB}}$  denotes the marginal CDF of  $\mathbf{w}_o^{\text{QB}}$  etc., and  $C$  is the copula of

$\mathbf{w}_o^{\text{QB}}, \dots, \mathbf{w}_o^{\text{D}}$ . We note that  $C$ , which is only uniquely defined on  $\text{Range}(F_{\text{QB}}) \times \dots \times \text{Range}(F_{\text{D}})$ , models the dependence structure of the positional marginals. The representation in (1.3) is convenient as it allows us to break our problem down into two separate sub-problems:

1. Modeling and estimating the positional marginals  $F_{\text{QB}}, \dots, F_{\text{D}}$ .
2. Modeling and estimating the copula  $C$ .

Moreover, it turns out that the representation in (1.3) is particularly convenient from an estimation viewpoint as we will have sufficient data to estimate the positional marginals reasonably well whereas obtaining sufficient data to estimate the copula  $C$  is challenging. We note that this is often the case in copula modeling applications. For example, in the equity and credit derivatives world in finance, there is often plentiful data on the so-called marginal risk-neutral distributions but relatively little data on the copula  $C$ . We begin with the positional marginals.

### 1.3.1 The Positional Marginals

To simplify matters, we will focus here on the selection of the QB from the total of  $P_{\text{QB}}$  that are available. We assume a Dirichlet-multinomial data generating process for a random opponent's selection. Specifically, we assume:

- $\mathbf{p}_{\text{QB}} \sim \text{Dir}(\boldsymbol{\alpha}_{\text{QB}})$  where  $\text{Dir}(\boldsymbol{\alpha}_{\text{QB}})$  denotes the Dirichlet distribution with parameter vector  $\boldsymbol{\alpha}_{\text{QB}}$ .
- A random opponent then selects QB  $k$  with probability  $p_{\text{QB}}^k$  for  $k = 1, \dots, P_{\text{QB}}$ , i.e., the chosen QB follows a Multinomial( $1, \mathbf{p}_{\text{QB}}$ ) distribution.

Note  $\mathbf{p}_{\text{QB}} := \{p_{\text{QB}}^k\}_{k=1}^{P_{\text{QB}}}$  lies on the unit simplex in  $\mathbb{R}^{P_{\text{QB}}}$  and therefore defines a probability distribution over the available quarterbacks. It is important to note that  $\mathbf{p}_{\text{QB}}$  is not known in advance of the DFS contest. Moreover, they do not appear to be perfectly predictable and so we have to explicitly model<sup>7</sup> their randomness. Accordingly, it is very natural to model  $\mathbf{p}_{\text{QB}}$  as following a Dirichlet distribution.

---

<sup>7</sup>In initial unreported experiments, we assumed  $\mathbf{p}_{\text{QB}}$  was fixed and known but this led to over-certainty and poor performance of the resulting portfolios.

## Available Data

In most fantasy sports contests, it is possible to obtain some information regarding  $\mathbf{W}_{\text{op}}$  once the contest is over and the winners have been announced. In particular, it is often possible to observe the realized ownership proportions which (because  $O$  is assumed large) amounts to observing  $\mathbf{p}_{\text{QB}}, \mathbf{p}_{\text{RB}}, \dots, \mathbf{p}_{\text{D}}$  after each contest. We therefore assume we have such data from a series of historical contests. In practice, we will also have access to other observable features, e.g. expected NFL player performance  $\boldsymbol{\mu}_{\delta}$ , home or away indicators, quality of opposing teams etc. from these previous contests.

## Dirichlet Regression

We can then use this data to build a Dirichlet regression model for estimating the marginal distributions of  $\mathbf{w}_o$ . We do this by assuming that the parameter vector  $\boldsymbol{\alpha}_{\text{QB}} \in \mathbb{R}^{P_{\text{QB}}}$  is predictable. In particular, we assume

$$\boldsymbol{\alpha}_{\text{QB}} = \exp(\mathbf{X}_{\text{QB}}\boldsymbol{\beta}_{\text{QB}}) \quad (1.4)$$

where  $\boldsymbol{\beta}_{\text{QB}}$  is a vector of parameters that we must estimate and  $\mathbf{X}_{\text{QB}}$  is a matrix (containing  $P_{\text{QB}}$  rows) of observable independent variables that are related to the specific features of the NFL games and QBs underlying the DFS contest. To be clear, the exponential function in the r.h.s. of (1.4) is actually an  $P_{\text{QB}} \times 1$  vector of exponentials.

For example, in a DFS contest for week  $t$ , we might assume

$$\boldsymbol{\alpha}_{\text{QB},t} = \exp(\beta_{\text{QB}}^0 \mathbf{1} + \beta_{\text{QB}}^1 \mathbf{f}_{\text{QB},t} + \beta_{\text{QB}}^2 \mathbf{c}_{\text{QB},t} + \beta_{\text{QB}}^3 \boldsymbol{\mu}_{\text{QB},t}) \quad (1.5)$$

where  $\mathbf{f}_{\text{QB},t} \in \mathbb{R}^{P_{\text{QB}}}$  is an estimate of  $\mathbf{p}_{\text{QB}}$  for week  $t$  that we can obtain from the FantasyPros website (FantasyPros 2017),  $\mathbf{c}_{\text{QB},t} \in \mathbb{R}^{P_{\text{QB}}}$  are the (appropriately scaled) week  $t$  costs of the QBs in the contest, and  $\boldsymbol{\mu}_{\text{QB},t}$  is an (appropriately scaled) sub-vector of  $\boldsymbol{\mu}_{\delta}$  for week  $t$  whose components correspond to the QB positions in  $\boldsymbol{\mu}_{\delta}$ . Other features are of course also possible. For example,



we might also want to include expected returns  $\boldsymbol{\mu}_{\text{QB},t}/\mathbf{c}_{\text{QB},t}$  (where division is understood to be component-wise), home-away indicators, quality of opponents etc. as features.

We can estimate the  $\boldsymbol{\beta}_{\text{QB}}$  vector by fitting a Bayesian Dirichlet regression. Assuming we have data from weeks  $t = 1$  to  $t = T - 1$  and a flat prior on  $\boldsymbol{\beta}_{\text{QB}}$ , then the posterior satisfies

$$\begin{aligned}
p(\boldsymbol{\beta}_{\text{QB}} \mid \{\mathbf{p}_{\text{QB},t}\}_{t=1}^{T-1}) &\propto \underbrace{p(\boldsymbol{\beta}_{\text{QB}})}_{\propto 1} p(\{\mathbf{p}_{\text{QB},t}\}_{t=1}^{T-1} \mid \boldsymbol{\beta}_{\text{QB}}) \\
&\propto \prod_{t=1}^{T-1} \text{Dir}(\mathbf{p}_{\text{QB},t} \mid \underbrace{e^{\mathbf{X}_{\text{QB},t}\boldsymbol{\beta}_{\text{QB}}}}_{=\boldsymbol{\alpha}_{\text{QB},t}}) \\
&\propto \prod_{t=1}^{T-1} \frac{1}{\mathbf{B}(\boldsymbol{\alpha}_{\text{QB},t})} \prod_{k=1}^{P_{\text{QB}}} (p_{\text{QB},t}^k)^{\alpha_{\text{QB},t}^k - 1}
\end{aligned} \tag{1.6}$$

where  $\mathbf{B}(\boldsymbol{\alpha}_{\text{QB},t})$  is the normalization factor for the Dirichlet distribution. We fit this model using the Bayesian software package STAN (Team 2017).

### *The Other Positions*

It should be clear that we can handle the other positions in a similar fashion. In the case of the three selected WRs for example, we assume  $\mathbf{p}_{\text{WR}} \sim \text{Dir}(\boldsymbol{\alpha}_{\text{WR}})$  and that a random opponent then selects her three WRs according<sup>8</sup> to a Multinomial(3,  $\mathbf{p}_{\text{WR}}$ ) distribution. We can again use Dirichlet regression to estimate the parameter vector  $\boldsymbol{\beta}_{\text{WR}}$  where  $\boldsymbol{\alpha}_{\text{WR}} = \exp(\mathbf{X}_{\text{WR}}\boldsymbol{\beta}_{\text{WR}})$ .

### 1.3.2 The Copula

Returning to (1.3), the question arises as to what copula  $C$  should we use? For the sake of brevity, here we only summarize the copula that we use and defer the details to Appendix A.1.1.

Motivated by the prevalent QB-WR stacking practice among the fantasy players of NFL (Bales

---

<sup>8</sup>In fact, the rules of a DFS contest are likely to state that the same player can not be chosen more than once. In that case, we could simply repeatedly draw from the Multinomial(3,  $\mathbf{p}_{\text{WR}}$ ) distribution until 3 different WRs are selected. Alternatively (but equivalently), we could draw each WR sequentially adjusting the multinomial distribution each time so that once selected, a WR cannot be selected again.

2016), we allow an opponent to be a *stacker*<sup>9</sup> with probability  $q$  (*the stacking copula*) or sample all positions independently of each other with probability  $1 - q$  (*the independence copula*). We therefore use a *mixture* of the stacking and independence copulas and note that such a copula is itself a copula. Given the limitation on available data to model a more complicated copula, we believe this rather simple copula is a reasonable choice. We show how well this simple copula performs on real-world data when we present numerical results in Section 1.6 and Appendix A.3.3 in particular.

### 1.3.3 Generating Random Opponents' Portfolios

Suppose now that the Dirichlet regression model has been fit for each of the positional marginals and that we have also estimated the  $q$  parameter for the copula. It is then easy to simulate a candidate  $\mathbf{w}_o$ . We first generate  $\text{Stack} \sim \text{Bernoulli}(q)$  and if  $\text{Stack} = 0$ , we use the independence copula. For example, to generate the QB selection, we must:

1. First draw a sample  $\mathbf{p}_{\text{QB}}$  from the  $\text{Dir}(\boldsymbol{\alpha}_{\text{QB}})$  distribution.
2. Then draw a sample from the  $\text{Mult}(1, \mathbf{p}_{\text{QB}})$  distribution.
3. This draw then defines our chosen QB, i.e., it sets one component of  $\mathbf{w}_o^{\text{QB}}$  to 1 with the others being set to 0.

We repeat this for all positions. If  $\text{Stack} = 1$ , however, then we use the stacking copula and therefore follow the same steps except we must set the first WR to be the main WR from the selected QB's team. At this point, we only have a *candidate*  $\mathbf{w}_o$  as there is no guarantee that the resulting  $\mathbf{w}_o$  is feasible, i.e., that  $\mathbf{w}_o \in \mathbb{W}$ . We therefore use an accept-reject approach whereby candidate  $\mathbf{w}_o$ 's are generated according to the steps outlined above and are only accepted if they are feasible. In fact, we impose one further condition: we insist that an accepted  $\mathbf{w}_o$  uses up most of the available budget. We impose this condition because it is very unlikely in practice that a fantasy player in a

---

<sup>9</sup>By a “stacker”, we mean the opponent picks the QB and the “main” WR from the same NFL team and samples other positions independently of each other. By “main” WR of a team, we refer to the WR with the highest expected points among all the WRs in the same team.

DFS contest would leave much of her budget unspent. This is purely a behavioral requirement and so we insist the cost of an accepted  $\mathbf{w}_o$  satisfy  $\mathbf{c}^\top \mathbf{w}_o \geq B_{\text{lb}}$  for some lower bound  $B_{\text{lb}} \leq B$  that we get to choose. Algorithm 7 in Appendix A.1.1 describes how to generate  $O$  random opponents' portfolios  $\mathbf{W}_{\text{op}}$  and it therefore (implicitly) defines the distribution of  $\mathbf{W}_{\text{op}}$ .

## 1.4 Solving the Double-Up Problem

As mentioned earlier, we first tackle the double-up problem since our solution to this problem will help inform how we approach the top-heavy problem. We begin first by recalling a result from mean-variance optimization and in particular, the problem of maximizing the probability of exceeding a stochastic benchmark. Our approach to solving both double-up and top-heavy problems will be a mean-variance optimization based on this result.

### 1.4.1 Mean Variance Optimization and Outperforming Stochastic Benchmarks

We consider<sup>10</sup> a one-period problem where at time  $t = 0$  there are  $P$  financial securities available to invest in. At time  $t = 1$  the corresponding random return vector  $\boldsymbol{\xi} = (\xi_1, \dots, \xi_P)$  is realized. Let  $\boldsymbol{\mu}_\xi$  and  $\boldsymbol{\Sigma}_\xi$  denote the mean return vector and variance-covariance matrix, respectively, of  $\boldsymbol{\xi}$ . The goal is then to construct a portfolio  $\mathbf{w} = (w_1, \dots, w_P)$  with random return  $R_{\mathbf{w}} = \mathbf{w}^\top \boldsymbol{\xi}$  that maximizes the probability of exceeding a random benchmark  $R_b$ . Mathematically, we wish to solve

$$\max_{\mathbf{w} \in \mathbb{W}} \mathbb{P}(R_{\mathbf{w}} - R_b \geq 0) \quad (1.7)$$

where  $\mathbb{W}$  includes the budget constraint  $\mathbf{w}^\top \mathbf{1} = 1$  as well as any other linear constraints we wish to impose. If we assume  $R_{\mathbf{w}} - R_b$  has a normal distribution so that  $R_{\mathbf{w}} - R_b \sim \mathcal{N}(\mu_{\mathbf{w}}, \sigma_{\mathbf{w}}^2)$  for some<sup>11</sup>

<sup>10</sup>The material and results in this subsection follow Morton et al. 2003 and they should be consulted for further details and related results. In this subsection, we will sometimes use the same notation from earlier sections to make the connections between the financial problem of this subsection and the DFS problem more apparent.

<sup>11</sup>If the benchmark  $R_b$  is deterministic, then  $\mu_{\mathbf{w}} := \mathbf{w}^\top \boldsymbol{\mu}_\xi - R_b$  and  $\sigma_{\mathbf{w}}^2 := \mathbf{w}^\top \boldsymbol{\Sigma}_\xi \mathbf{w}$ .

$\mu_{\mathbf{w}}$  and  $\sigma_{\mathbf{w}}^2$  that depend on  $\mathbf{w}$ , then (1.7) amounts to solving

$$\max_{\mathbf{w} \in \mathbb{W}} 1 - \Phi \left( -\frac{\mu_{\mathbf{w}}}{\sigma_{\mathbf{w}}} \right) \quad (1.8)$$

where  $\Phi(\cdot)$  denotes the standard normal CDF. Let  $\mathbf{w}^*$  be the optimal solution to (1.8). The following result is adapted from Morton et al. 2003 and follows from the representation in (1.8).

**Proposition 1.1.** *Suppose  $R_{\mathbf{w}} - R_b \sim N(\mu_{\mathbf{w}}, \sigma_{\mathbf{w}}^2)$  for all  $\mathbf{w} \in \mathbb{W}$ .*

1. *Suppose  $\mu_{\mathbf{w}} < 0$  for all  $\mathbf{w} \in \mathbb{W}$ . Then*

$$\mathbf{w}^* \in \left\{ \mathbf{w}(\lambda) : \mathbf{w}(\lambda) \in \arg \max_{\mathbf{w} \in \mathbb{W}} (\mu_{\mathbf{w}} + \lambda \sigma_{\mathbf{w}}^2), \lambda \geq 0 \right\}. \quad (1.9)$$

2. *Suppose  $\mu_{\mathbf{w}} \geq 0$  for some  $\mathbf{w} \in \mathbb{W}$ . Then*

$$\mathbf{w}^* \in \left\{ \mathbf{w}(\lambda) : \mathbf{w}(\lambda) \in \arg \max_{\mathbf{w} \in \mathbb{W}, \mu_{\mathbf{w}} \geq 0} (\mu_{\mathbf{w}} - \lambda \sigma_{\mathbf{w}}^2), \lambda \geq 0 \right\} \quad (1.10)$$

*so that  $\mathbf{w}^*$  is mean-variance efficient.*

Proposition 1.1 is useful because it allows us to solve the problem in (1.8) efficiently. In particular, we determine which of the two cases from the proposition applies. This can be done when  $\mathbb{W}$  is polyhedral by simply solving a linear program that maximizes  $\mu_{\mathbf{w}}$  (which is affine in  $\mathbf{w}$ ) over  $\mathbf{w} \in \mathbb{W}$ . If the optimal mean is negative, then we are in case (i); otherwise we are in case (ii). We then form a grid  $\Lambda$  of possible  $\lambda$  values and for each  $\lambda \in \Lambda$ , we solve the appropriate quadratic optimization problem (defining  $\mathbf{w}(\lambda)$ ) from (1.9) or (1.10) and then choose the value of  $\lambda$  that yields the largest objective in (1.7) or (1.8). See Algorithm 1 in Section 1.4.2 below for when we apply these results to our double-up problem.

## 1.4.2 The Double-Up Problem

Recall now the double-up problem as formulated in (1.1). We define  $Y_{\mathbf{w}} := \mathbf{w}^\top \boldsymbol{\delta} - G^{(r')}$  and note that

$$\begin{aligned}\mu_{Y_{\mathbf{w}}} &:= \mathbf{w}^\top \boldsymbol{\mu}_{\boldsymbol{\delta}} - \mu_{G^{(r')}} \\ \sigma_{Y_{\mathbf{w}}}^2 &:= \mathbf{w}^\top \boldsymbol{\Sigma}_{\boldsymbol{\delta}} \mathbf{w} + \sigma_{G^{(r')}}^2 - 2\mathbf{w}^\top \boldsymbol{\sigma}_{\boldsymbol{\delta}, G^{(r')}}\end{aligned}\tag{1.11}$$

where  $\mu_{G^{(r')}} := \mathbb{E}[G^{(r')}]$ ,  $\sigma_{G^{(r')}}^2 := \text{Var}(G^{(r')})$  and  $\boldsymbol{\sigma}_{\boldsymbol{\delta}, G^{(r'')}}$  is a  $P \times 1$  vector with  $p^{\text{th}}$  component equal to  $\text{Cov}(\delta_p, G^{(r')})$ . Our approach to solving (1.1) is based on Proposition 1.1 and is presented in Algorithm 1 below. While this algorithm will deliver the optimal solution in the event that each  $Y_{\mathbf{w}} \sim \mathcal{N}(\mu_{Y_{\mathbf{w}}}, \sigma_{Y_{\mathbf{w}}}^2)$ , it should yield a good approximate solution even when the  $Y_{\mathbf{w}}$ 's are not normally distributed. Indeed the key insights yielded by Proposition 1.1 do not rely on the normality of the  $Y_{\mathbf{w}}$ 's. Specifically, if  $\mu_{Y_{\mathbf{w}}} < 0$  for all  $\mathbf{w} \in \mathbb{W}$ , then it seems intuitively clear that we need to select a team  $\mathbf{w}$  that simultaneously has a high mean and a high variance. The appropriate balance between mean and variance in the objective function will then be determined by  $\lambda$ . Similarly, if there is at least one  $\mathbf{w} \in \mathbb{W}$  such that  $\mu_{Y_{\mathbf{w}}} > 0$ , then intuition suggests we can search for a team  $\mathbf{w}$  with a large (and positive) mean and a small variance. Again, the appropriate balance between the two will be determined by  $\lambda$ . Not insisting on the normality of  $Y_{\mathbf{w}}$  also gives us the freedom to consider using non-normal distributions for  $\boldsymbol{\delta}$ . Indeed this parallels the situation in the asset allocation literature in finance where the mean-variance paradigm remains<sup>12</sup> very popular despite the well-known fact that asset returns have heavy tails and therefore are not normally distributed.

**Remark 1.1.** *Note that  $\lambda^*$  in line 10 can be computed using the Monte Carlo samples of  $(\boldsymbol{\delta}, G^{(r')})$  that are inputs to the algorithm. In this case, the computation of  $\mathbb{P}\{Y_{\mathbf{w}, \lambda} > 0\}$  does not rely on any normal approximation. Alternatively,  $\lambda^*$  could also be estimated via the assumption that each*

---

<sup>12</sup>To be clear, we are not claiming the original mean-variance approach of Markowitz is popular. Indeed it's well known that parameter estimation issues render Markowitz useless in practice. Developments which build on Markowitz such as Black-Litterman, robust mean-variance etc. are popular, however, and they too take a mean-variance perspective.

---

**Algorithm 1** Optimization for the Double-Up Problem with a Single Entry

---

**Require:**  $\mathbb{W}, \Lambda, \boldsymbol{\mu}_\delta, \boldsymbol{\Sigma}_\delta, \mu_{G^{(r')}}, \sigma_{G^{(r')}}^2, \boldsymbol{\sigma}_{\delta, G^{(r')}}$  and Monte Carlo samples of  $(\boldsymbol{\delta}, G^{(r')})$

```
1: if  $\exists \mathbf{w} \in \mathbb{W}$  with  $\mu_{Y_{\mathbf{w}}} \geq 0$ 
2:   for all  $\lambda \in \Lambda$ 
3:      $\mathbf{w}_\lambda = \operatorname{argmax}_{\mathbf{w} \in \mathbb{W}, \mu_{Y_{\mathbf{w}}} \geq 0} \left\{ \mu_{Y_{\mathbf{w}}} - \lambda \sigma_{Y_{\mathbf{w}}}^2 \right\}$ 
4:   end for
5: else
6:   for all  $\lambda \in \Lambda$ 
7:      $\mathbf{w}_\lambda = \operatorname{argmax}_{\mathbf{w} \in \mathbb{W}} \left\{ \mu_{Y_{\mathbf{w}}} + \lambda \sigma_{Y_{\mathbf{w}}}^2 \right\}$ 
8:   end for
9: end if
10:  $\lambda^* = \operatorname{argmax}_{\lambda \in \Lambda} \mathbb{P}\{Y_{\mathbf{w}_\lambda} > 0\}$ 
11: return  $\mathbf{w}_{\lambda^*}$ 
```

---

$Y_{\mathbf{w}_\lambda}$  is approximately normally distributed. We also note that if it turns out that  $\lambda^* \approx 0$ , then the optimization will basically seek to maximize  $\mathbf{w}^\top \boldsymbol{\mu}_\delta$ , thereby suggesting there is little value to be gained from modeling opponents.

One potential difficulty that might arise in practice with Algorithm 1 is if the distribution of the  $Y_{\mathbf{w}}$ 's display a significant skew. In this case it's possible that we end up seeking to increase variance when we should be decreasing it or vice-versa. While this was never an issue in any of our numerical experiments we discuss this possibility in Appendix A.1.3 and note that it's easy to adjust Algorithm 1 to allow for this possibility.

### Generating Monte Carlo Samples

In order to execute Algorithm 1, we must first compute the inputs  $\mu_{G^{(r')}}$ ,  $\sigma_{G^{(r')}}^2$  and  $\boldsymbol{\sigma}_{\delta, G^{(r')}}$  as defined above. These quantities can be estimated off-line via Monte Carlo simulation as they do not depend on our portfolio choice  $\mathbf{w}$ . We simply note here that the Monte Carlo can be performed relatively efficiently using results from the theory of order statistics. The specific details can be found in Appendix A.1.2.

## Solving the Binary Quadratic Programs

The optimization problems in lines 3 and 7 of Algorithm 1 require the solution of binary quadratic programs (BQPs). In our numerical experiments of Sections 1.6 and 1.7, we solved these BQPs using Gurobi’s (Gurobi Optimization 2016) default BQP solver although the specific algorithm used by Gurobi was not clear from the online documentation. (We do note in passing, however, that it is straightforward to transform a BQP into an equivalent binary program (BP) at the cost of adding  $O(P^2)$  binary variables and  $O(P^2)$  linear constraints.)

## The Double-Up Problem with Multiple Entries

Since the top-heavy payoff structure is our main focus, we defer the discussion of the double-up problem with multiple entries to Appendix A.1.3. We briefly note here, however, that we advocate for a *replication* strategy. In particular, after purchasing  $N$  entries in the contest, the DFS player should then submit  $N$  copies of  $\mathbf{w}^*$  where

$$\mathbf{w}^* := \operatorname{argmax}_{\mathbf{w} \in \mathbb{W}} \mathbb{P} \left\{ \mathbf{w}^\top \boldsymbol{\delta} > G^{(O+1-r)}(\mathbf{W}_{\text{op}}, \boldsymbol{\delta}) \right\}.$$

Appendix A.1.3 discusses the intuition behind this strategy and also provides (see Proposition A.1) a simple certificate-of-optimality that can be used to confirm its optimality or near-optimality.

## 1.5 Solving the Top-Heavy Problem

We can now extend the analysis we developed for the double-up problem in Section 1.4 to tackle the more interesting top-heavy problem. We consider first the single-entry case where  $N = 1$ . In that case, the problem in (1.2) simplifies to solving

$$\max_{\mathbf{w} \in \mathbb{W}} \sum_{d=1}^D (R_d - R_{d+1}) \mathbb{P} \left\{ \mathbf{w}^\top \boldsymbol{\delta} > G^{(r'_d)}(\mathbf{W}_{\text{op}}, \boldsymbol{\delta}) \right\}, \quad (1.12)$$

where  $r'_d := O+1-r_d$ . Following the development in Section 1.4.2, we can define  $Y_{\mathbf{w}}^d := \mathbf{w}^\top \boldsymbol{\delta} - G^{(r'_d)}$  and define

$$\mu_{Y_{\mathbf{w}}^d} := \mathbf{w}^\top \boldsymbol{\mu}_{\boldsymbol{\delta}} - \mu_{G^{(r'_d)}} \quad (1.13)$$

$$\sigma_{Y_{\mathbf{w}}^d}^2 := \mathbf{w}^\top \boldsymbol{\Sigma}_{\boldsymbol{\delta}} \mathbf{w} + \sigma_{G^{(r'_d)}}^2 - 2\mathbf{w}^\top \boldsymbol{\sigma}_{\boldsymbol{\delta}, G^{(r'_d)}} \quad (1.14)$$

where  $\mu_{G^{(r'_d)}} := \mathbb{E}[G^{(r'_d)}]$ ,  $\sigma_{G^{(r'_d)}}^2 := \text{Var}(G^{(r'_d)})$  and  $\boldsymbol{\sigma}_{\boldsymbol{\delta}, G^{(r'_d)}}$  is a  $P \times 1$  vector with  $p^{th}$  component equal to  $\text{Cov}(\delta_p, G^{(r'_d)})$ . Following our mean-variance approach, we can now approximate (1.12) as

$$\max_{\mathbf{w} \in \mathbb{W}} \sum_{d=1}^D (R_d - R_{d+1}) \left( 1 - \Phi \left( -\frac{\mu_{Y_{\mathbf{w}}^d}}{\sigma_{Y_{\mathbf{w}}^d}} \right) \right). \quad (1.15)$$

Before proceeding, we need to make two additional assumptions, which we will state formally.

**Assumption 1.1.**  $\mu_{Y_{\mathbf{w}}^d} < 0$  for  $d = 1, \dots, D$  and for all  $\mathbf{w} \in \mathbb{W}$ .

A justification for Assumption 1.1 can be found in Appendix A.1.4. Given Assumption 1.1, it follows that each of the arguments  $-\mu_{Y_{\mathbf{w}}^d}/\sigma_{Y_{\mathbf{w}}^d}$  to the normal CDF terms in (1.15) is *positive*. Given the objective in (1.15) is to maximize, it is also clear that for a fixed value of  $\mathbf{w}^\top \boldsymbol{\mu}_{\boldsymbol{\delta}}$  in (1.13), we would like the standard deviation  $\sigma_{Y_{\mathbf{w}}^d}$  to be as large as possible. Unfortunately, the third term,  $2\mathbf{w}^\top \boldsymbol{\sigma}_{\boldsymbol{\delta}, G^{(r'_d)}}$ , on the r.h.s. of (1.14) suggests that  $\mathbf{w}$  impacts the variance by a quantity that depends on  $d$ . Fortunately, however, we found this dependence on  $d$  to be very small in our numerical experiments with real-world DFS top-heavy contests. Specifically, we found these covariance terms to be very close to each other for values of  $d$  corresponding to the top 20 percentiles and in particular for the top few percentiles. We now formalize this observation via the following assumption.

**Assumption 1.2.**  $\text{Cov}(\delta_p, G^{(r'_d)}) = \text{Cov}(\delta_p, G^{(r'_{d'})})$  for all  $d, d' = 1, \dots, D$  and for all  $p \in \{1, \dots, P\}$ .

Further support for Assumption 1.2 is provided by the following proposition a proof of which may be found in Appendix A.1.4.



**Proposition 1.2.** *Suppose the  $\mathbf{w}_o$ 's are IID given  $\mathbf{p}$  and  $D$  is finite. Then, in the limit as  $O \rightarrow \infty$ , we have*

$$\text{Cov}\left(\delta_p, G^{(r'_d)}\right) = \text{Cov}\left(\delta_p, G^{(r'_{d'})}\right) \text{ for all } d, d' = 1, \dots, D \quad (1.16)$$

for any  $p \in \{1, \dots, P\}$ .

Given Assumption 1.2, it is clear from (1.14) that the impact of  $\mathbf{w}$  on  $\sigma_{Y_w^d}^2$  does not depend on  $d$  when  $d$  is finite and  $O$  is “large”. Given the preceding arguments, it follows that for any fixed value of  $\mathbf{w}^\top \boldsymbol{\mu}_\delta$ , we would like to make  $\mathbf{w}^\top \boldsymbol{\Sigma}_\delta \mathbf{w} - 2\mathbf{w}^\top \boldsymbol{\sigma}_{\delta, G^{(r'_d)}}$  (the terms from (1.14) that depend on  $\mathbf{w}$ ) as large as possible. We are therefore in the situation of part (i) of Proposition 1.1 and so we have a simple algorithm for approximately solving the top-heavy problem. This is given in Algorithm 2 below where we omit the dependence on  $d$  of those terms that are assumed (by Assumption 1.2) to not vary with  $d$ .

---

**Algorithm 2** Optimization for the Top-Heavy Problem with a Single Entry

---

**Require:**  $\mathbb{W}, \Lambda, \boldsymbol{\mu}_\delta, \boldsymbol{\Sigma}_\delta, \boldsymbol{\sigma}_{\delta, G^{(r' )}}$  and Monte Carlo samples of  $(\boldsymbol{\delta}, G^{(r'_d)})$  for all  $d = 1, \dots, D$

- 1: **for all**  $\lambda \in \Lambda$
  - 2:    $\mathbf{w}_\lambda = \underset{\mathbf{w} \in \mathbb{W}}{\text{argmax}} \left\{ \mathbf{w}^\top \boldsymbol{\mu}_\delta + \lambda \left( \mathbf{w}^\top \boldsymbol{\Sigma}_\delta \mathbf{w} - 2\mathbf{w}^\top \boldsymbol{\sigma}_{\delta, G^{(r' )}} \right) \right\}$
  - 3: **end for**
  - 4:  $\lambda^* = \underset{\lambda \in \Lambda}{\text{argmax}} \sum_{d=1}^D (R_d - R_{d+1}) \mathbb{P} \left\{ \mathbf{w}_\lambda^\top \boldsymbol{\delta} > G^{(r'_d)}(\mathbf{W}_{\text{op}}, \boldsymbol{\delta}) \right\}$
  - 5: **return**  $\mathbf{w}_{\lambda^*}$
- 

As with Algorithm 1, the  $\mathbf{w}_\lambda$ 's are computed by solving BQPs and the optimal  $\lambda^*$  from line 4 can then be determined via the Monte Carlo samples that were used as inputs.

### 1.5.1 The Top-Heavy DFS Problem with Multiple Entries

We now discuss the more general top-heavy DFS problem where we must submit  $N$  entries to the contest. Recalling the problem formulation from Section 1.2.2, we must solve for

$$\max_{\mathbf{W} \in \mathbb{W}^{|\mathbb{W}|}, |\mathbb{W}|=N} \mathcal{R}(\mathbf{W})$$

where

$$\mathcal{R}(\mathbf{W}) := \sum_{i=1}^{|\mathbf{W}|} \sum_{d=1}^D (R_d - R_{d+1}) \mathbb{P} \left\{ \mathbf{w}_i^\top \boldsymbol{\delta} > G_{-i}^{(r'_d)}(\mathbf{W}_{-i}, \mathbf{W}_{\text{ops}}, \boldsymbol{\delta}) \right\}, \quad (1.17)$$

$r'_d := O + |\mathbf{W}| - r_d$ ,  $G_{-i}^{(r)}$  is the  $r^{\text{th}}$  order statistic of  $\{G_o\}_{o=1}^O \cup \{F_j\}_{j=1}^{|\mathbf{W}|} \setminus F_i$  and  $\mathbf{W}_{-i} := \mathbf{W} \setminus \mathbf{w}_i$ . We propose a greedy algorithm motivated by (i) the submodularity of the objective function in (1.17) w.r.t. the decision variable  $\mathbf{W}$  and (ii) our analysis of parimutuel betting which can be viewed as a special case of our top-heavy DFS contests. For the sake of brevity, we defer our discussion of parimutuel betting to Appendix A.2 but note that it serves as a useful tool to gain intuition regarding the structure of an optimal portfolio in the multiple entries case for DFS top-heavy contests. In particular, we show in Appendix A.2 that a greedy algorithm that adds an entry with the highest “value-add” in each iteration returns an optimal portfolio in the parimutuel betting setup. We also highlight there the subtle difference between the reward structures of parimutuel betting and DFS contests and the implications this might have for good top-heavy DFS strategies.

We focus here on (i), i.e. the submodularity<sup>13</sup> of the top-heavy objective function in (1.17). We first state an assumption which will prove sufficient to guarantee the submodularity of the top-heavy objective.

**Assumption 1.3.** *Denote by  $V_k$  the payoff corresponding to rank  $k$  entry where  $V_1 \geq V_2 \geq \dots \geq V_K \geq 0$  where  $K := O + N$  corresponds to the last-ranked entry in the contest and define  $\Delta_k := V_k - V_{k+1}$  for all  $k = 1, \dots, K - 1$ . Then  $\Delta_k \geq \Delta_{k+1}$  for all  $k = 1, \dots, K - 2$ .*

We note there is a close relationship between the  $V_k$ 's of Assumption 1.3 and the  $R_d$ 's that we have been using to define the payoff function of the top-heavy contest, and in fact we could just as easily define the payoff function in terms of the  $V_k$ 's. Assumption 1.3 can be interpreted as a *convexity* assumption on the payoffs of the top-heavy contest. This convexity assumption is generally not satisfied in practice but is typically satisfied for the first several (and therefore most important) ranks which have the highest payoffs. As such, Assumption 1.3 may be viewed

---

<sup>13</sup>We note the greedy algorithm proposed by Hunter et al. 2016 was also motivated by submodularity considerations but their focus was on maximizing the probability of winning a WTA contest whereas our focus is on maximizing the expected reward in general top-heavy contests.



optimization to solve in the DFS context. In particular, in iteration  $i$  of Algorithm 3, given the current portfolio  $\mathbf{W}$  consisting of  $i - 1$  entries, it is a non-trivial task to identify the entry  $\widehat{\mathbf{w}}_i$  that will add the most to  $\mathbf{W}$  in terms of expected<sup>14</sup> reward. The reason is that it is not necessarily true that  $\widehat{\mathbf{w}}_i$  will lie on the “efficient frontier” constructed in lines 3 to 5 of Algorithm 3. Hence, even though our optimization in line 6 of Algorithm 3 identifies the highest value-add entry in the set  $\{\mathbf{w}_\lambda\}_{\lambda \in \Lambda}$ , it is possible and indeed very likely that  $\widehat{\mathbf{w}}_i$  does not belong to  $\{\mathbf{w}_\lambda\}_{\lambda \in \Lambda}$  to begin with. In fact, when  $\gamma = C$ , we expect that the candidate entry  $\mathbf{w}_{\lambda^*}$  will often coincide with a previously chosen entry, i.e., an entry from  $\{\mathbf{w}_1^*, \dots, \mathbf{w}_{i-1}^*\}$ . Indeed this is what we observed in our numerical experiments where we typically found just  $\approx 10$  unique entries when  $N = 50$ . But this is simply a reflection of our failure to find  $\widehat{\mathbf{w}}_i$ .

In order to find a better candidate  $\widehat{\mathbf{w}}_i$ , we introduce the parameter  $\gamma$  in line 8 of the algorithm. This line ensures that our  $i^{\text{th}}$  entry can not have more than  $\gamma$  athletes in common with each of the previous  $i - 1$  entries. Recalling that  $C$  is the number of athletes in a DFS entry, it therefore follows that if we set  $\gamma \geq C$ , then the constraint  $\mathbf{w}^\top \mathbf{w}_{\lambda^*} \leq \gamma$  on line 8 is never binding. But if we set  $\gamma < C$ , then the candidate entry  $\mathbf{w}_{\lambda^*}$  from iteration  $i$  will always be a new entry, i.e., an entry not represented in the current portfolio  $\{\mathbf{w}_1^*, \dots, \mathbf{w}_{i-1}^*\}$ . In particular, setting  $\gamma < C$  results in a completely diversified portfolio of  $N$  distinct entries. In our real-world numerical experiments we ultimately chose a value  $\gamma < C$ . It is important to note that diversification is imposed only to find a better choice of  $\widehat{\mathbf{w}}_i$ . More importantly, by setting  $\gamma < C$ , we observed a *much* higher expected reward for the final portfolio. In particular, the expected reward almost doubled. (In all of our numerical experiments, we found that  $\gamma = C - 3 = 6$  was an optimal choice in that it led to final portfolios with the highest expected reward.)

Though we defer our discussion of parimutuel betting markets to Appendix A.2, we emphasize that our results on parimutuel betting provide further support for Algorithm 3 and in particular,

---

<sup>14</sup>Note that the  $\mathcal{R}(\cdot)$  appearing in Algorithm 3 is really an estimated version of the expected reward since we can only evaluate it using the Monte-Carlo samples. On a related note we mention that samples of  $(\delta, G^{(r')})$  for some additional ranks  $r'$  besides the  $r'_d$ 's will be required in order to properly estimate  $\mathcal{R}$ . For example, suppose  $D = N = 2$  with  $r_1 = 1$  and  $r_2 = 20$ . Then we will also need samples corresponding to the  $19^{\text{th}}$  rank since if our first entry comes  $5^{\text{th}}$  say then our second entry will only be among the top 20 if it's among the top 19 of our opponents' entries.

the imposition of diversification by choosing a value of  $\gamma < C$  in Algorithm 3. Recognizing that Algorithm 3 does not enjoy the “ $1 - 1/e$ ” guarantee of Nemhauser et al. 1978, we now state a simple proposition which allows us to bound the degree of suboptimality of the portfolio of  $N$  entries returned by Algorithm 3, a proof of which is provided in Appendix A.1.4. (Note that this suboptimality bound does not require Assumption 1.3.)

**Proposition 1.3** (Suboptimality bound). *Let  $\mathbf{w}^* := \arg \max_{\mathbf{w} \in \mathbb{W}} \mathcal{R}(\mathbf{w})$  and  $\mathbf{W}^\# = \{\mathbf{w}_i^\#\}_{i=1}^N$  denote optimal single-entry and  $N$ -entry portfolios, respectively, for the top-heavy contest where  $\mathcal{R}(\cdot)$  is as defined in (1.17). Denote by  $\mathbf{W}$  any arbitrary feasible portfolio of  $N$  entries and define  $\nu_{\mathbf{W}} := \frac{\mathcal{R}(\mathbf{W})}{N\mathcal{R}(\mathbf{w}^*)}$ . Then the value of  $\mathbf{W}$  is within  $\nu_{\mathbf{W}}$  of the optimal value, i.e.,  $\nu_{\mathbf{W}} \leq \mathcal{R}(\mathbf{W})/\mathcal{R}(\mathbf{W}^\#) \leq 1$ .*

In the top-heavy contests calibrated to real-data experiments of Section 1.6, we observed that Algorithm 3 achieved a very satisfactory performance. In particular, during the 17 weeks of the 2017 NFL season, the lowest value of  $\nu_{\mathbf{W}^*}$  (for the portfolio  $\mathbf{W}^*$  returned by Algorithm 3) was 48.95%, the highest value was 80.21%, and the average<sup>15</sup> was 62.27%. We also emphasize that these are *lower bounds* on the algorithm’s suboptimality and it’s possible that the actual average performance was much higher than 62.27% of the unknown optimal portfolio’s performance. Finally, we acknowledge that these numbers are only legitimate to the extent that our model and the fitted parameters of our model are correct.

While we have taken  $N$  as given up to this point, it is perhaps worth mentioning that one can always use Algorithm 3 to determine an optimal value of  $N$ . Specifically, we can continue to increase  $N$  until the expected P&L contribution from the next entry goes negative or below some pre-specified threshold. We also note it is straightforward to add additional linear constraints to  $\mathbb{W}$  if further or different forms of diversification are desired. Finally, we note it’s easy to estimate the expected P&L of any portfolio of entries via Monte Carlo simulation.

---

<sup>15</sup>It is interesting but surely coincidental to see how close the average realized bound of 62.27% is to the guaranteed bound of  $1 - 1/e$  ( $\approx 63.2\%$ ) provided by the idealized greedy algorithm!

## 1.6 Numerical Experiments

We participated in real-world DFS contests on FanDuel during the 2017 NFL regular season, which consisted of 17 weeks. Each week, we participated in three contests: top-heavy, quintuple-up and double-up. The cost per entry was \$1 in top-heavy and \$2 in both quintuple-up and double-up contests. The number of opponents  $O$  was approximately 200,000, 10,000 and 30,000 for the three contests, respectively, with these numbers varying by around 10% from week-to-week. The payoff in the top-heavy contest<sup>16</sup> for rank 1 was approx. \$5,000, for rank 2 it was approx. \$2,500 and then it declined quickly to approx. \$100 for rank 30. The lowest winning rank was around 50,000, with a payoff of \$2.

We used two different models for each contest: our strategic model and a benchmark model. To be clear, for all top-heavy contests, our strategic model was Algorithm 3 with  $\gamma = 6$ . Our strategic model for the double-up and quintuple-up contests was also Algorithm 3 with<sup>17</sup>  $\gamma = 6$  but lines 3 to 5 replaced by lines 1 to 9 of Algorithm 1 and with the understanding that the expected reward function  $\mathcal{R}(\cdot)$  (used to determine  $\lambda^*$ ) corresponds to the double-up / quintuple-up contest. The second model is a *benchmark* model that does not model opponents and hence is not strategic; the details are provided in Appendix A.3.1. For each model, we submitted  $N = 50, 25$  and 10 entries to top-heavy, quintuple-up and double-up contests, respectively each week. Other details regarding our model inputs such as  $\mu_\delta, \Sigma_\delta$ , stacking probability  $q$ , diversification parameter  $\gamma$ , and the budget lower bound  $B_{\text{lb}}$  are discussed in Appendix A.3.2 along with the specifications of the hardware and software we use to solve the BQPs.

We now discuss the P&L-related results for the strategic and benchmark models across the three contest structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. Figure 1.1 displays the cumulative realized P&L for both models across the three contest

---

<sup>16</sup>We note that there are other top-heavy contests with even more competitors and payoff structures that are even more “top-heavy”. For example, a regular NFL contest on FanDuel often has approximately 400,000 entries with a top payoff of \$250,000 to \$1,000,000. Payoffs then decline quickly to approx. \$500 for the 50<sup>th</sup> rank. Top-heavy contests are therefore extremely popular and hence are our principal focus in this chapter.

<sup>17</sup>The reason for doing so in the double-up / quintuple-up contests was simply to reduce the variance of our P&L albeit at the cost of a (hopefully slightly) smaller expected P&L. This is discussed further later on in this section.

structures during the season. (Table A.1 in Appendix A.3.3 displays the actual numbers.) The strategic portfolio has outperformed the benchmark portfolio since inception in the top-heavy series of contests. The strategic portfolio has earned a cumulative profit of \$280.74, which is over 3 times the realized P&L of the benchmark portfolio. Moreover, the maximum cumulative loss, that is, the max shortfall, for the strategic portfolio is just \$18.5. In addition, the small initial investment of \$50 plus two additional investments of \$18.5 and \$7.26 (total of \$75.76) have been sufficient to fund the strategic portfolio throughout the season. This suggests a profit of \$280.74 on an investment of \$75.76, that is, a return of over 350% in just 17 weeks. In contrast, the benchmark portfolio needed much more capital than the initial investment of \$50. If we account for this additional required capital, then the benchmark portfolio has earned a return of less than 50% in 17 weeks. Note that given the so-called house-edge of approximately 15%, both models have performed considerably better than the average portfolio which would have lost  $\approx 17 \times 15\% \times 50 = \$127.5$  across the 17 weeks.

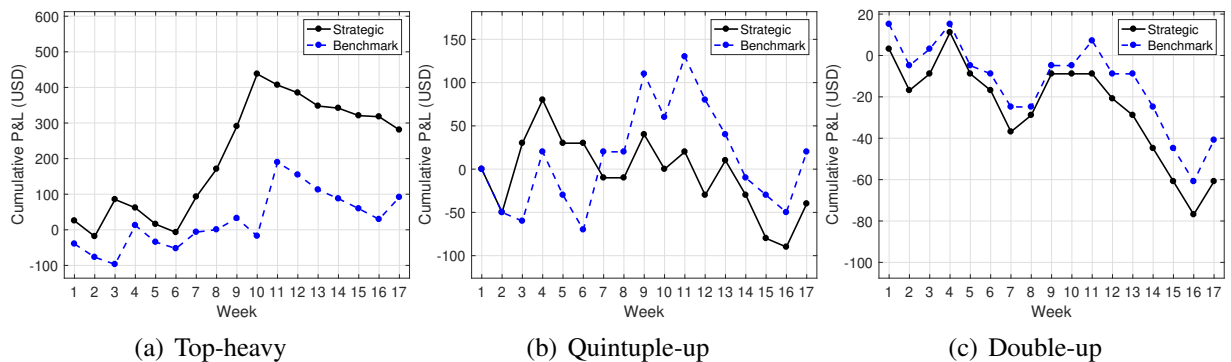


Figure 1.1: Cumulative realized dollar P&L for the strategic and benchmark models across the three contest structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season.

With regards to the quintuple-up series, the strategic model was better until the end of week 6 but since then the benchmark portfolio has outperformed it. We note, however, that the difference in the cumulative P&L between the two models at the end of the season ( $20 - (-40) = 60$ ) could easily be wiped out in just one week's contest as we can see when we look at the relative performances of the two strategies in week 7, for example.

We are confident that the realized P&L to-date for each contest series is actually conservative and that superior performance (in expectation) could easily be attained. There are at least three reasons for this. First, we used off-the-shelf estimates of the input parameters  $\mu_\delta$  and  $\Sigma_\delta$ , which are clearly vital to the optimization model. Moreover, we obtained the  $\mu_\delta$  estimate a day before the actual NFL games started and mostly ignored the developments in the last few hours preceding the games, which can be very important in football. For example, in week 7, the main RB of the Jacksonville Jaguars (Leonard Fournette) was questionable to play. Accordingly, their second main RB (Chris Ivory) was expected to play more time on the field. However, our  $\mu_\delta$  estimate did not reflect this new information. Our estimate projected 17.27 and 6.78 fantasy points for Fournette and Ivory, respectively. Moreover, since FanDuel sets the price of the athletes a few days before the games take place, Fournette was priced at 9000 and Ivory at 5900. There was a clear benefit of leveraging this information as Fournette was over-priced and Ivory was under-priced. In fact, our opponents exploited this opportunity as around 60% of them (in double-up) picked Ivory. A proactive user would have updated his  $\mu_\delta$  estimate following such news. In fact, the so-called sharks do react to such last-minute information (Nickish 2015), meaning that we were at a disadvantage by not doing so.

For another example, consider Devin Funchess, a wide-receiver (WR) for the Carolina Panthers. During the course of the season, Funchess was usually the main WR for Carolina but in week 16 he was expected to be only the second or third WR and in fact Damiere Byrd was expected to be the main WR. This was late developing news, however, and our  $\mu_\delta$  estimate did not reflect this. Moreover, Byrd was priced at 4900 while Funchess was priced at 7000 and so Byrd was clearly under-priced relative to Funchess. In the week 16 game itself, Byrd scored 9.6 points while Funchess scored only 2.6 points. Because of our failure to respond to this late developing news and update our parameters, it transpired that 52 of our entries picked Funchess. We observed (after the fact) many similar situations during the course of the season and there is no doubt that we could have constructed superior portfolios had we been more pro-active in monitoring these developments and updating parameters accordingly.



The second reason is simply a variance issue in that a large number of DFS contests (and certainly much greater than 17) will be required to fully establish the outperformance of the strategic model in general. In fact, we believe the variance of the cumulative P&L is particularly high for NFL DFS contests. There are several reasons for this. Certainly, the individual performance of an NFL player in a given week will have quite a high variance due to the large roster size<sup>18</sup> as well as the relatively high probability of injury. This is in contrast to other DFS sports where there is considerably more certainty over the playing time of each athlete. To give but one example, in week 5 we witnessed a series of injuries that impacted many of our submitted portfolios (both strategic and benchmark). Devante Parker (Miami Dolphins) was injured in the first quarter but was picked by 56 of our entries. Charles Clay (Buffalo Bills) and Sterling Shepard (NY Giants) were injured before halftime, affecting 70 and 4 entries, respectively. Bilal Powell (NY Jets) and Travis Kelce (Kansas City Chiefs) left the field close to the halftime, impacting 44 and 25 entries, respectively. Furthermore, the NFL season consists of just 16 games per team whereas teams in sports such as basketball, ice hockey and baseball play 82, 82 and 162 games, respectively, per season. As a result, the cumulative P&L from playing DFS contests over the course of an NFL season will have a very high variance relative to these other sports. This high variance of NFL-based fantasy sports has been noted by other researchers including for example Clair and Letscher 2007. We also suspect that Hunter et al. 2016 focused on ice hockey and baseball (and avoided NFL) for precisely this reason.

The third reason applies specifically to the quintuple-up contests. In our strategic model for quintuple-up, there is a possibility of incorrectly minimizing portfolio variance when we should in fact be maximizing it (along with expected number of points of course). Proposition 1.1 leads us to try and increase variance if  $\mu_{\mathbf{w}} < 0$  for all  $\mathbf{w} \in \mathbb{W}$  and to try and decrease variance otherwise. But  $\mu_{\mathbf{w}}$  must be estimated via Monte Carlo and is of course also model-dependent. As such, if we estimate a maximal value of  $\mu_{\mathbf{w}} \approx 0$ , it is quite possible we will err and increase variance when we should decrease it and vice versa. We suspect this may have occurred occasionally with the

---

<sup>18</sup>There are more than 45 athletes on a roster but only 11 on the field at any one time.

quintuple-up contests where we often obtained an estimate of  $\mu_w$  that was close to zero. This of course is also related to the median versus mean issue we mentioned immediately after Algorithm 1 and discuss in Appendix A.1.3. We note that one potential approach to solving this problem would have been to use Algorithm 8 instead of Algorithm 1. We note that the benchmark portfolio is always long expected points and variance of points.

Figure 1.2 displays the in-model P&L distribution for the diversification strategy from Section 1.5.1 for both strategic and benchmark portfolios in week 10<sup>19</sup> contests. For the strategic portfolio, we use Algorithm 3 as explained in the beginning of Section 1.6 and for the benchmark portfolio, we use the procedure outlined in Appendix A.3.1. We note this P&L distribution is as determined by our model with the continued assumption of the multivariate normal distribution for  $\delta$  as well as the Dirichlet-multinomial model for opponents' portfolio selections. The strategic model dominates the benchmark model in terms of expected profit. In the top-heavy contest, the expected profit of the strategic portfolio is over 5 times that of the benchmark portfolio. The gain is not as drastic in the quintuple-up and double-up contests. The substantial gain in top-heavy seems to come from the fact that the strategic portfolio has considerably more mass in the right-tail. Note this leads to the higher standard deviation of the top-heavy strategic portfolio<sup>20</sup>.

Figure 1.3 is similar to Figure 1.2 except it is based upon using the replication strategy from Appendix A.1.3 instead of the diversification strategy. We note the strategic model continues to have a higher expected P&L than the benchmark model. The main observation here is that the expected P&L drops considerably when we go from the diversification strategy to the replication strategy for top-heavy. This is consistent with our analysis from Appendix A.2 on parimutuel betting as well as our discussion surrounding Algorithms 3 and 10 in Section 1.5.1 and Appendix A.2 respectively. In contrast, the P&L increases for both quintuple-up and double-up when we employ the replication strategy. Again, this is consistent with our earlier argument in favor of replication for double-up style contests. In our numerical experiments, however, we used the diversification

---

<sup>19</sup>Other weeks have similar results as shown in Figure 1.4.

<sup>20</sup>The high standard deviation in the top-heavy strategic portfolio should be seen as a pro instead of a con, since it is mostly coming from the right-tail of the P&L distribution.

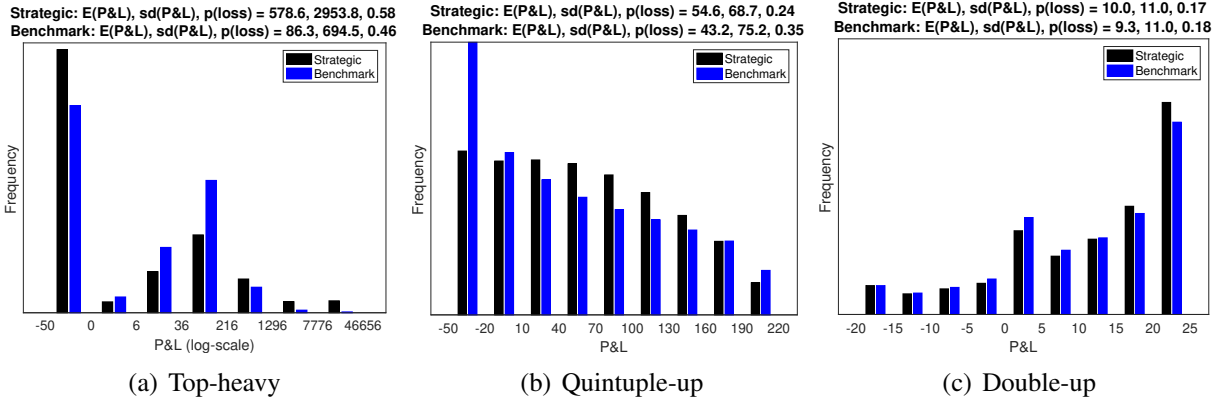


Figure 1.2: P&L distribution for the diversification strategy for the strategic and benchmark portfolios for week 10 contests of the 2017 NFL season. Recall  $N = 50, 25,$  and  $10$  for top-heavy, quintuple-up and double-up, respectively. The three metrics at the top of each image are the expected P&L, the standard deviation of the P&L and the probability of loss, that is,  $\mathbb{P}(P\&L < 0)$ .

strategy for both double-up and quintuple-up contests. This was only because of the variance issue highlighted earlier and our desire to use a strategy which had a considerably smaller standard deviation (while ceding only a small amount of expected P&L). As can be seen from Figures 1.2 and 1.3, the diversification strategy has (as expected) a smaller expected P&L as well as a smaller probability of loss.

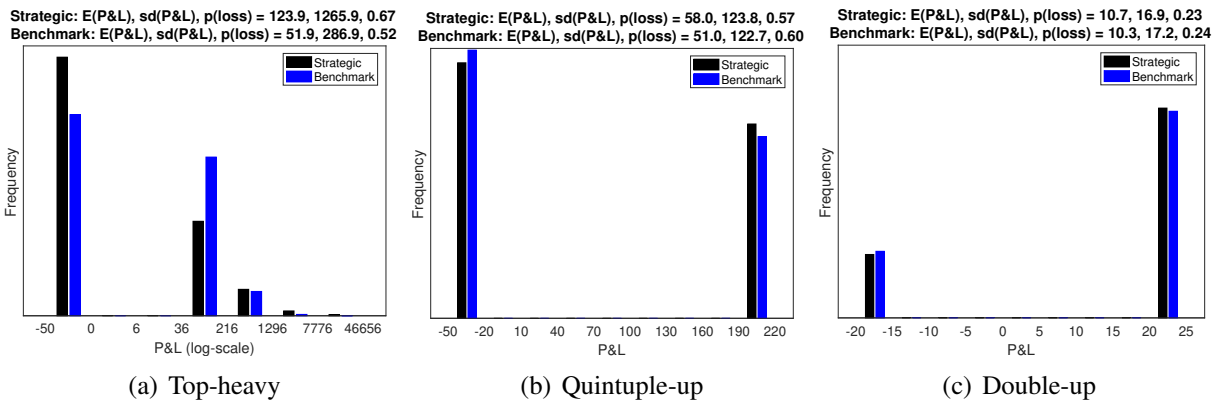


Figure 1.3: P&L distribution for the replication strategy for the strategic and benchmark portfolios for week 10 contests of the 2017 NFL season. Recall  $N = 50, 25,$  and  $10$  for top-heavy, quintuple-up and double-up, respectively. The three metrics at the top of each image are the expected P&L, the standard deviation of the P&L and the probability of loss, that is,  $\mathbb{P}(P\&L < 0)$ .

Figure 1.4 displays the realized and expected P&Ls. For both strategic and benchmark models

and all three contests, the expected profit is greater than the realized profit. This is perhaps not too surprising given the bias that results from optimizing within a model. In top-heavy, however, the realized P&L is within one standard deviation of the expected P&L although this is not the case for the quintuple- and double-up contests. As discussed above, we believe our realized results are conservative and that a more proactive user of these strategies who makes a more determined effort to estimate  $\mu_\delta$  and  $\Sigma_\delta$  and responds to relevant news breaking just before the games can do considerably better. Despite this potential for improvement, the strategic model has performed very well overall. The small losses from the double-up and quintuple-up contests have been comfortably offset by the gains in the top-heavy contests. As we noted earlier, the return on investment in top-heavy is over 350% for a seventeen week period although we do acknowledge there is considerable variance in this number as evidenced by Figure 1.4(a).

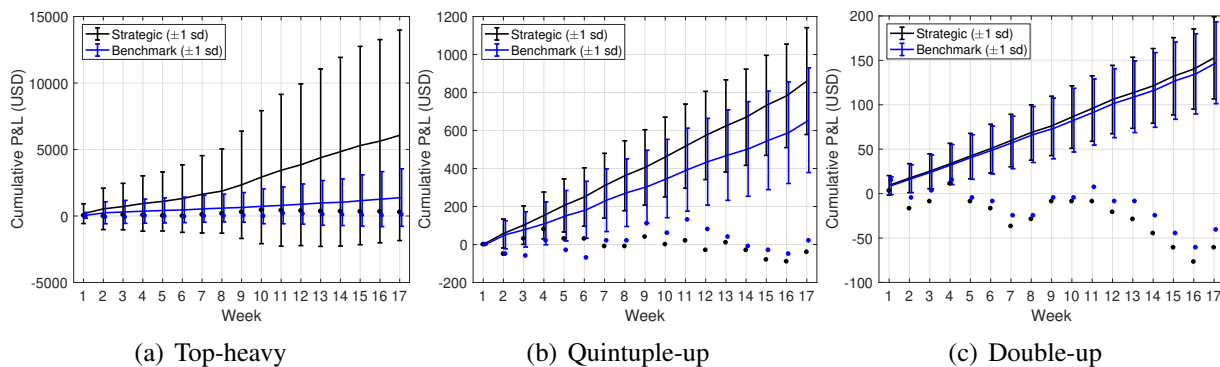


Figure 1.4: Predicted and realized cumulative P&L for the strategic and benchmark models across the three contest structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. The realized cumulative P&Ls are displayed as points.

More granular results are presented in Appendix A.3.3. For example, in that appendix we show the performance of each week's best entry for the strategic and benchmark models corresponding to the top-heavy contests for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. We also present various statistics of interest, e.g., expected fantasy points, standard deviation,  $\lambda^*$  for the first optimal entry in each week for the strategic and benchmark models. We observe there that  $\lambda^*$  is closer to zero for the double-up and quintuple-up contests thereby indicating (see Remark 1.1) that the value of modeling opponents is much greater for top-heavy contests.

In addition, Appendix A.3.3 also contains some additional anecdotes describing situations where our strategic model went against the “crowd” and was successful in doing so. Finally, Dirichlet regression results are also presented in Appendix A.3.3. Note that our Dirichlet regression models used the features described in (1.5) and we validated this choice of features by evaluating its goodness-of-fit and comparing its out-of-sample performance against two “simpler” variations of the Dirichlet regression model. Specific details are deferred to Appendix A.3.4.

## 1.7 The Value of Modeling Opponents, Insider Trading, and Collusion

In the numerical results of Section 1.6, we found that modeling opponents’ behavior can significantly increase the expected P&L from participating in top-heavy DFS contests and we explore it in more depth in Section 1.7.1. In Section 1.7.2, motivated by the issue of insider trading in fantasy sports we described in Section 1.1, we evaluate how much a fantasy player gains by having access to inside information. Finally, in Section 1.7.3, we analyze the value of collusion in fantasy sports, that is, how much does a fantasy player gain by strategically partnering with other fantasy players and submitting more portfolios than allowed. In each of these experiments, we employ the same algorithms as we did for the numerical experiments of Section 1.6.

### 1.7.1 The Value of Modeling Opponents

As we saw in Figures 1.2 and 1.3, the value of modeling opponents is clearly contest-dependent. Indeed our model, which explicitly models opponents, has a much bigger edge (in terms of expected P&L) over the benchmark model in the top-heavy contest<sup>21</sup> as compared to the double-up and quintuple-up contests. But the value of modeling opponents also depends on how *accurately* we model their behavior. On this latter point, it is of interest to consider:

- (a) How much do we gain (with respect to the benchmark model) if we use a deterministic

$\mathbf{p} = (\mathbf{p}_{QB}, \dots, \mathbf{p}_D)$ ? For example, in the NFL contests, we could set  $\mathbf{p}$  equal to the values

---

<sup>21</sup>This is discussed in more detail in Appendix A.1.4. The reason for the relative importance of modeling opponents is largely due to the importance of selecting entries with both a high variance and expectation in top-heavy contests.

predicted by the FantasyPros website.

- (b) How much additional value is there if instead we assume  $(\mathbf{p}_{QB}, \dots, \mathbf{p}_D) \sim (\text{Dir}(\boldsymbol{\alpha}_{QB}), \dots, \text{Dir}(\boldsymbol{\alpha}_D))$  as in Algorithm 7 but now  $\boldsymbol{\alpha}_{QB}, \dots, \boldsymbol{\alpha}_D$  only depend on the first two features stated in Equation (1.5), that is, the constant feature and the estimate of  $\mathbf{p}$  that we obtain from the FantasyPros website?
- (c) Finally, how much additional value is there to be gained by assuming the model of Algorithm 7 where  $\boldsymbol{\alpha}_{QB}, \dots, \boldsymbol{\alpha}_D$  is allowed to depend on any and all relevant features?

To answer these questions, we computed the optimal portfolios for each of the three cases described above (and for the benchmark model) and also the corresponding expected P&Ls by assuming case (c) to be the ground truth. We did this for all three contest structures for each of the 17 weeks in the 2017 NFL regular season. All the parameter values such as  $N$  and  $\gamma$  were as in Section 1.6. We found the value of modeling opponents accurately to be most valuable in the top-heavy contests. In particular, the total expected P&L (over 17 weeks) in the top-heavy series was approximately \$1,400, \$5,400, \$5,800, and \$6,000 for the benchmark model, case (a), case (b), and case (c), respectively. Accordingly, even though the deterministic model for  $\mathbf{p}$  (case (a)) explains most of the gain in expected P&L we reap by being strategic, there is approximately an additional 10% reward we receive by modeling the opponents more precisely (cases (b) and (c)). It is worth emphasizing, however, that this 10% additional gain depends on our “ground truth” model. For example, if we had assumed some other ground truth where  $\mathbf{p}$  was more predictable given additional and better chosen features, then there might be more to gain in moving from case (a) to case (c).

## 1.7.2 The Value of Insider Trading

A question that is somewhat dual to the first question concerns the issue of insider trading and the value of information. This question received considerable attention in 2015 (Drape and Williams 2015a; Drape and Williams 2015b) when a DraftKings employee was accused of using data from DraftKings contests to enter a FanDuel DFS contest in the same week and win \$350,000.

Without addressing the specific nature of insider trading in that case, we pose several questions:

1. How much does the insider gain if he knows the true positional marginals  $\mathbf{p} = (\mathbf{p}_{QB}, \dots, \mathbf{p}_D)$ ?
2. How much does the insider gain if he knows the entries of all contestants, that is,  $\mathbf{W}_{op}$ ? In that case, the only uncertainty in the system is the performance vector  $\boldsymbol{\delta}$  of the real-world athletes. (Note that the problem of computing an optimal portfolio given full knowledge of  $\mathbf{W}_{op}$  is straightforward in our framework.)

To state these questions more formally, we note that the optimal expected P&L for a portfolio consisting of  $N$  entries satisfies

$$\max_{\mathbf{W} \in \mathbb{W}^N} \left\{ \mathbb{E} \left[ \mathbb{E}_{\boldsymbol{\delta}, \mathbf{W}_{op}} [\text{Reward}(\mathbf{W}, \boldsymbol{\delta}, \mathbf{W}_{op}) \mid \mathbf{p}] \right] \right\} \quad (1.18)$$

where  $\text{Reward}(\mathbf{W}, \boldsymbol{\delta}, \mathbf{W}_{op})$  denotes the P&L function which is easy to compute given  $\mathbf{W}$ ,  $\boldsymbol{\delta}$ , and  $\mathbf{W}_{op}$ . The answer to question (i) is then given by the difference between (1.18) and

$$\mathbb{E} \left[ \max_{\mathbf{W} \in \mathbb{W}^N} \left\{ \mathbb{E}_{\boldsymbol{\delta}, \mathbf{W}_{op}} [\text{Reward}(\mathbf{W}, \boldsymbol{\delta}, \mathbf{W}_{op}) \mid \mathbf{p}] \right\} \right]. \quad (1.19)$$

Similarly, the answer to question (ii) is given by the difference between (1.18) and

$$\mathbb{E}_{\mathbf{p}, \mathbf{W}_{op}} \left[ \max_{\mathbf{W} \in \mathbb{W}^N} \left\{ \mathbb{E}_{\boldsymbol{\delta}} [\text{Reward}(\mathbf{W}, \boldsymbol{\delta}, \mathbf{W}_{op}) \mid \mathbf{p}, \mathbf{W}_{op}] \right\} \right]. \quad (1.20)$$

However, computing both (1.19) and (1.20) is computationally expensive since the optimization occurs inside the expectation over high-dimensional random variables and hence many expensive optimizations would be required. Though one could perform such computations on an HPC cluster over an extended period of time, we instead designed less demanding but nonetheless informative experiments to evaluate the value of insider trading. In particular, we ran the following two experiments for all three contest structures across all 17 weeks of the 2017 NFL season:

- **Experiment 1:** We first compute the optimal portfolio for each week conditional on know-

ing the realized  $\mathbf{p}$ . We call this portfolio the *insider portfolio*. We then compare the expected P&L of the insider portfolio with the optimal strategic non-insider portfolio that we submitted to the real-world contests. (We assume the ground truth in the P&L computations to be the realized marginals  $\mathbf{p}$  together with the same stacking parameters from Section 1.6.)

- **Experiment 2:** This is similar to Experiment 1 but we now replace  $\mathbf{p}$  with  $\mathbf{W}_{\text{op}}$ . However, we do not have access to the realized values of  $\mathbf{W}_{\text{op}}$  during the NFL season. Instead, for each week we sample one realization of  $\mathbf{W}_{\text{op}}$  using the realized  $\mathbf{p}$  (with the same stacking parameters from Section 1.6) and treat the sampled  $\mathbf{W}_{\text{op}}$  as the realized value. We then compute the optimal portfolio (the *insider portfolio*) for each week conditional on knowing the realized  $\mathbf{W}_{\text{op}}$  and compare the expected P&Ls of the insider portfolio with the strategic non-insider optimal portfolio assuming the ground truth in P&L computations to be the realized  $\mathbf{W}_{\text{op}}$ .

It is worth emphasizing that in both Experiments 1 and 2, we are taking expectations over  $\boldsymbol{\delta}$ , the performance vector of the underlying NFL athletes. As such, we are averaging over the largest source of uncertainty in the system.

In Experiment 1, we found the insider to have an edge (in terms of total expected P&L across the season) of around 20%, 1%, and 2% in top-heavy, quintuple-up<sup>22</sup>, and double-up contests respectively over the (strategic) non-insider. In Figure 1.5, we compare the weekly expected top-heavy P&L of the insider and (strategic) non-insider portfolios and observe that the weekly increase varies from 1% (week 6) to 50% (week 16). As one would expect, the insider portfolio’s P&L dominates that of the non-insider’s. Of course, the insider will have an even greater edge over a non-insider who is not strategic as we have already seen in Section 1.6 that the strategic non-insider has roughly five times the expected P&L of the non-strategic non-insider in top-heavy contests. Compared to this approximately 500% difference between the benchmark and strategic players, the

---

<sup>22</sup>As expected, the benefits of insider trading were much greater in top-heavy contests than in the double- and quintuple-up contests where we expected the benefits to be quite small. It is quite interesting, however, to see that the observed benefits in quintuple-up (1%) were less than the observed benefits in double-up (2%). We suspect this may be related to the same issue with quintuple-up that we identified earlier in Section 1.6, namely the issue that arises when the maximal value of  $\mu_{\mathbf{w}} \approx 0$ . In this case, the optimal value of  $\lambda$  in Algorithm 1 will be close to 0. Indeed this is what we observed in Table A.3. As a result (and this should be clear from the expression for  $\sigma_{Y_{\mathbf{w}}}^2$  in (1.11) together with lines 3 and 7 of Algorithm 1) the only benefit to inside information in this case is in estimating  $\mu_{Y_{\mathbf{w}}}$ .



additional 20% increase in expected P&L gained via insider trading seems modest. This modest increase is due in part to how well our Dirichlet regression model allows the (strategic) non-insider to estimate the positional marginals  $\mathbf{p}$ . Accordingly, the value of inside information depends on how well the non-insider can predict opponents' behavior. In particular, the more sophisticated the non-insider is, then the less value there is to having inside information.

In Experiment 2, we found the insider's edge to be similar to that of Experiment 1. Intuitively, one would expect the edge to be bigger in Experiment 2 due to the insider having the more granular information of  $\mathbf{W}_{\text{op}}$ . Noting that the variance of  $G^{(r')} \mid (\boldsymbol{\delta}, \mathbf{p})$  goes to zero as the number of opponents  $O$  goes to infinity, however, we can conclude that the additional value of seeing the realized  $\mathbf{W}_{\text{op}}$  over and beyond the value of seeing the realized  $\mathbf{p}$  should<sup>23</sup> be small when  $O$  is large. Given that the contests we participated in had large  $O$ , this observation supports our results from Experiment 2.

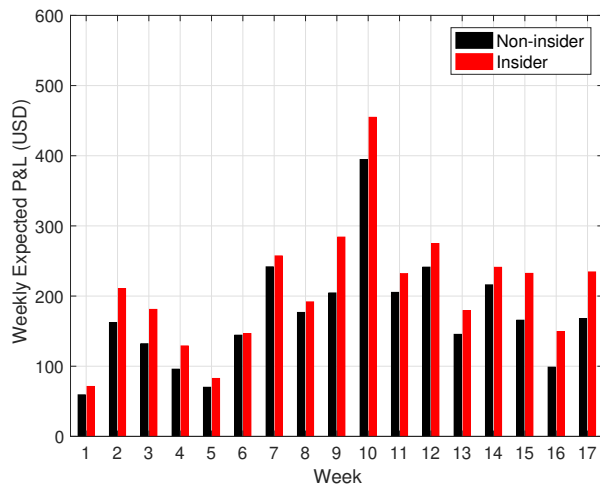


Figure 1.5: Weekly expected dollar P&L for the strategic model ( $N = 50$ ) with and without inside information  $\mathbf{p}$  in the top-heavy series.

<sup>23</sup>But note we are assuming here that the dependence structure between the positional marginals in  $\mathbf{p}$  is known regardless of whether we only see  $\mathbf{p}$  or  $\mathbf{W}_{\text{op}}$ .

### 1.7.3 The Value of Collusion

In addition to the insider trading controversy, the subject of collusion in fantasy sports contests has also received considerable attention. In one suspected case, two brothers were suspected of colluding when one of them won 1 million dollars (Brown 2016; Reagan 2016) in one of DraftKings’ “Fantasy Football Millionaire” contests, a particularly top-heavy contest where just the first few places earn most of the total payoff. Collusion refers to the situation where two or more DFS players form (unbeknownst to the contest organizers) a strategic partnership and agree to pool their winnings. Maintaining separate accounts allows the partnership to submit  $N_{\text{collude}} \times E_{\text{max}}$  entries to a given contest where  $N_{\text{collude}}$  is the number of players in the partnership and  $E_{\text{max}}$  is the maximum number of entries permitted per player. Collusion can be beneficial in top-heavy contests as it allows the colluding players to avoid substantial overlap (and therefore achieve greater diversification) in their portfolios thereby increasing the probability that the partnership will win a large payout.

We will assume that the  $N_{\text{collude}}$  players will construct a single portfolio of  $N_{\text{collude}} \times E_{\text{max}}$  entries when they collude. This portfolio can be constructed using Algorithm 3 from Section 1.5.1 with  $N = N_{\text{collude}} \times E_{\text{max}}$ . This portfolio can then be separated into  $N_{\text{collude}}$  separate sub-portfolios each consisting of  $E_{\text{max}}$  entries and each colluding player can then submit one of these sub-portfolios as his official submission.

In order to estimate the benefits of collusion, it is first necessary to understand the behavior of the colluding players when they are unable to collude. Many different behaviors are of course possible but it seems reasonable to assume that potentially colluding players are sophisticated and understand how to construct good portfolios. We therefore assume<sup>24</sup> that each of the potentially colluding players has access to the modeling framework outlined in this chapter and that as a result, each one submits *identical* portfolios of  $E_{\text{max}}$  entries. This portfolio is constructed using the same approach from Section 1.5.1. While this assumption is stylized and not realistic in practice, it does

---

<sup>24</sup>To the extent that our framework is a good framework for constructing DFS portfolios (which we believe to be the case!), then this might overstate the value of collusion as most colluding players will not have access to such a framework. Nonetheless, we can use this framework to consider just how beneficial colluding *might* be.

allow us to compute an upper bound on how beneficial colluding might be. Specifically, we can easily estimate and compare the expectations and standard deviations of the profits for the colluding and non-colluding portfolios in order to estimate the potential benefits of colluding. We would argue that the difference in expected values provides an upper bound on the value of colluding since in practice non-colluders are very unlikely to choose identical or even near-identical portfolios.

Before describing our numerical experiments, it is worthwhile noting that the results of Section 1.6 and specifically, Figures 1.2(a) and 1.3(a), can be used to estimate the benefits of colluding in week 10 top-heavy<sup>25</sup> contests of the 2017 NFL season if  $E_{\max} = 1$  and  $N_{\text{collude}} = 50$ . We see from Figure 1.2(a) that collusion in this case results in an estimated expected profit of 578.6 with a standard deviation of 2,953.8. In contrast, we can see from Figure 1.3(a) that the non-colluding portfolio has an expected profit of 123.9 with a standard deviation of 1,265.9. In this case, the colluding portfolio has an expected profit that is almost 5 times the expected profit of the non-colluding portfolio. It may appear this gain is coming at a cost, namely a higher standard deviation, but we note the higher standard deviation is entirely due to increased dispersion on the *right-hand-side* of the probability distribution. This is clear from Figures 1.2(a) and 1.3(a). Indeed we note that the probability of loss is 0.58 in Figure 1.2(a) (collusion) and increases to 0.67 in Figure 1.3(a) (non-collusion). This increased standard deviation can therefore hardly be considered a cost of collusion.

We also performed a more formal experiment to evaluate the value of collusion in top-heavy contests. We assumed the larger value of  $E_{\max} = 50$  which is quite common in practice and then varied the number of colluders so that  $N_{\text{collude}}$  ranged from 1 to 5. To be clear, the non-colluding portfolio comprised 50 strategic entries replicated  $N_{\text{collude}}$  times whereas the colluding portfolio consisted of  $N_{\text{collude}} \times 50$  strategic entries. In Table 1.1, we compare the performances of the colluding and non-colluding portfolios over the 2017 NFL season in terms of the total expected dollar P&L, the average weekly Sortino ratio, and the average weekly probability of loss over

---

<sup>25</sup>Not surprisingly we do not see any benefits to collusion in the double-up or quintuple-up contest here and indeed as pointed out earlier, we expect replication (which corresponds to non-collusion in the setting considered here) to be very close to optimal.

the 17 weeks of the 2017 NFL season. To be clear, both portfolios were constructed for each week using our calibrated model for that specific week. The expected P&L for the week was then computed by averaging (via Monte Carlo) over  $\delta$  and  $\mathbf{W}_{\text{op}}$  where samples of  $(\delta, \mathbf{W}_{\text{op}})$  were generated using the same<sup>26</sup> calibrated model. In particular, the *realized*  $(\delta, \mathbf{W}_{\text{op}})$ 's across the 17 weeks played no role in the experiment.

The colluding portfolio clearly dominates the non-colluding portfolio across the three metrics and for all values of  $N_{\text{collude}}$ . For example, collusion among 5 sophisticated fantasy players can increase the expected P&L for the 17-week season by 44%, increase the average weekly Sortino ratio by 63%, and decrease the average weekly loss probability by 8%. It is also clear from these numbers that collusion also results in a decreased downside risk (square root of  $\mathbb{E}[\text{P\&L}^2 \times \mathbb{1}_{\{\text{P\&L} \leq T\}}]$ ) since the percentage increase in the Sortino ratio is more than the percentage increase in the expected P&L. Accordingly, collusion results in a win-win situation by increasing the expected P&L and decreasing the downside risk simultaneously, which demonstrates that collusion can be surprisingly valuable in top-heavy DFS contests.

Table 1.1: Total expected dollar P&L (over 17 weeks), average weekly Sortino ratio and average weekly probability of loss related to the top-heavy contests for both the non-colluding (“NC”) and colluding (“C”) portfolios with  $E_{\text{max}} = 50$  and  $N_{\text{collude}} \in \{1, \dots, 5\}$ . The average weekly Sortino ratio is simply the average of the weekly Sortino ratios,  $\text{SR}_i$  for  $i = 1, \dots, 17$ . Specifically  $\text{SR}_i := (\mathbb{E}[\text{P\&L}_i] - T)/\text{DR}_i$  where  $\mathbb{E}[\text{P\&L}_i]$  denotes the expected P&L for week  $i$ ,  $T$  denotes the target P&L which we set to 0, and  $\text{DR}_i := \sqrt{\mathbb{E}[\text{P\&L}_i^2 \times \mathbb{1}_{\{\text{P\&L}_i \leq T\}}]}$  denotes the downside risk for week  $i$ . (The expected P&L is rounded to the nearest integer whereas the Sortino ratio and probability of loss are rounded to two decimal places.)

$N_{\text{collude}}$	Expected P&L (USD)			Sortino Ratio			Probability of Loss		
	NC	C	Increase	NC	C	Increase	NC	C	Decrease
1	6,053	6,053	0%	14.60	14.60	0%	0.49	0.49	0%
2	9,057	10,240	13%	11.02	13.24	20%	0.49	0.47	4%
3	10,975	13,776	26%	8.96	12.32	37%	0.49	0.46	6%
4	12,411	16,883	36%	7.64	11.56	51%	0.49	0.46	7%
5	13,632	19,677	44%	6.75	10.99	63%	0.49	0.45	8%

Of course the benefits from collusion are not as great as those from week 10 reported above

<sup>26</sup>Both colluding and non-colluding portfolios then benefitted in this experiment from the fact that the assumed model was indeed the correct model. We are interested in the *difference* in performances of the two portfolios, however, and so the bias that results from assuming the players know the true model should be relatively small.

when  $E_{\max} = 1$  and  $N_{\text{collude}} = 50$ . This is because it is intuitively clear that these benefits, while positive, are a decreasing function of  $E_{\max}$  all other things being equal. For example, in the extreme case where  $E_{\max} = \infty$ , there are clearly no benefits to colluding. In practice, we suspect the gains from collusion are much smaller for risk-neutral players since it is extremely unlikely that non-colluders would ever choose identical or near-identical portfolios as we have assumed here.

## 1.8 Conclusions and Further Research

In this chapter, we have developed a new framework for constructing portfolios for both double-up and top-heavy DFS contests. Our methodology explicitly accounts for the behavior of DFS opponents and leverages mean-variance theory (for the outperformance of stochastic benchmarks) to develop a tractable algorithm that requires solving a series of binary quadratic programs. Following Hunter et al. 2016, we also provide a tractable greedy algorithm for handling the multiple entry, i.e.,  $N > 1$ , case for top-heavy style contests. This is in contrast to the replication approach we advocate for double-up style contests. Moreover, our greedy algorithm (or simple variations of it) can be justified theoretically via the results we developed on parimutuel betting as well as the classic result of Nemhauser et al. 1978 on the performance of an idealized greedy algorithm for submodular maximization.

There are many potential directions for future research. We could back-test other benchmark strategies as well as refine our own preferred strategies. It would also be interesting to further develop our modeling and estimation approach for a random opponent's portfolio  $\mathbf{w}_o$ . We assumed in Section 1.3 that we had sufficient data to estimate the positional marginals of  $\mathbf{w}_o$  and we would like to explore other features that might be useful in the Dirichlet regression to better estimate these marginals. We would also like to explore other copula models for splicing these marginals together to construct the joint distribution of  $\mathbf{w}_o$ . It is not clear, however, whether we could ever obtain a rich enough data-set to estimate other copulas sufficiently accurately.

While NFL contests are among the most popular DFS contests, the season is quite short with only 17 rounds of games. Moreover, as mentioned in Section 1.6, the individual performance

of an NFL player in a given week has quite a high variance, potentially causing the cumulative P&L in football DFS contests to be high relative to other sports such as basketball, ice hockey and baseball. For these reasons and others, it would be interesting to apply our modeling framework to DFS contests in these other sports. It would also be interesting to use domain knowledge of these other sports to actively update estimates of  $\boldsymbol{\mu}_\delta$  and  $\boldsymbol{\Sigma}_\delta$  as the round of games approaches. This is something we did not do in the current NFL season. Indeed we recorded many instances when it would have been possible to avoid certain athletes in our DFS entries had we used up-to-date information that was available before the games in question and before our entries needed to be submitted. As a result, we believe the net positive P&L achieved by our models is very encouraging and can easily be improved (in expectation) by more active monitoring of the athletes.

Other directions for future research include the development of very fast re-optimization procedures / heuristics that could be performed on an already optimized portfolio of  $N$  entries when new information regarding player injuries, availability, weather etc. become known in the hours (and indeed minutes) before the portfolio of entries must be submitted to the DFS contest. As discussed in Section 1.6, such late-breaking developments are common-place and in order to extract the full benefit of the modeling framework presented here, it is important that such developments be reflected in updated parameter estimates which in turn calls for re-optimizing the entries. Of course, it would be desirable to re-optimize the entire portfolio in such circumstances but given time constraints, it may be necessary to make do with simple but fast heuristic updates. For the same reason, it would also be of interest to pursue more efficient Monte Carlo strategies for estimating the inputs  $\mu_{G(r')}$ ,  $\sigma_{G(r')}^2$  and  $\boldsymbol{\sigma}_{\delta, G(r')}$  that are required for the various algorithms we proposed. While we did make use of results from the theory of order statistics to develop our Monte Carlo algorithm, it should be possible to develop considerably more efficient algorithms to do this. In the case of top-heavy contests, for example, the moments corresponding to the top order statistics are particularly important and it may be possible to design importance-sampling or other variance reduction algorithms to quickly estimate them.

Finally, we briefly mention the area of mean-field games. In our modeling of opponents, we

did not assume they were strategic although we did note how some strategic modeling along the lines of stacking to increase portfolio variance could be accommodated. If we allowed some opponents to be fully strategic, then we are in a game-theoretic setting. Such games would most likely be impossible to solve. Even refinements such as mean-field games (where we let  $O \rightarrow \infty$  in some appropriate fashion) would still likely be intractable, especially given the discreteness of the problem (binary decision variables) and portfolio constraints. But it may be possible to solve very stylized versions of these DFS games where it is possible to purchase or sell short fractional amounts of athletes. There has been some success in solving mean-field games in the literature on parimutuel betting (Bayraktar and Munk 2017) in horse-racing and it may be possible to do likewise here for very stylized versions of DFS contests.

We hope to pursue some of these directions in future research.

## Chapter 2: Shapley Meets Uniform: An Axiomatic Framework for Attribution in Online Advertising

One of the central challenges in online advertising is attribution, namely, assessing the contribution of individual advertiser actions such as e-mails, display ads and search ads to eventual conversion. Several heuristics are used for attribution in practice; however, most do not have any formal justification. The main contribution in this work is to propose an axiomatic framework for attribution in online advertising. We show that the most common heuristics can be cast under the framework and illustrate how these may fail. We propose a novel attribution metric, that we refer to as *counterfactual adjusted Shapley value* (CASV), which inherits the desirable properties of the traditional Shapley value while overcoming its shortcomings in the online advertising context. We also propose a Markovian model for the user journey through the conversion funnel, in which ad actions may have disparate impacts at different stages. We use the Markovian model to compare our metric with commonly used metrics. Furthermore, under the Markovian model, we establish that the CASV metric coincides with an adjusted “unique-uniform” attribution scheme. This scheme is efficiently implementable, and can be interpreted as a correction to the commonly used uniform attribution scheme. We supplement our theoretical developments with numerical experiments using a real-world large-scale dataset. A preliminary version of this work appeared in the WWW conference (Singal et al. 2019) and the current version is under revision at Management Science.

### 2.1 Introduction

With the rise of the Internet, the digital economy has become a trillion dollar industry accounting for over 6% of the U.S. GDP (Hagan 2018). Retailers reach consumers through different online channels with the goal of acquiring new customers and managing relationships with the existing ones. The wide range and frequency of economic activities taking place in the digital world has



enabled data collection at a massive scale, allowing retailers to better understand customer behavior and improve service quality using data-driven decisions. With over a billion people with access to the Internet, it is not surprising that advertising has moved to the digital space. The global digital marketing sector witnessed a growth of 21% in 2017, which increased its market size to USD 88 billion (Sluis 2018). Key decisions in this space include allocating budget to various ad channels and media (e-mail, display media platforms, and paid search for instance) and optimizing tactical decisions in each channel. Such tactical decisions may be driven by the publishers; see, e.g., Balseiro et al. 2014; Hojjat et al. 2017 and Lejeune and Turner 2019 or by the advertisers themselves. For example, in display or search advertising, this would entail bidding for ads to push one's product towards the desired customer demographic at the right time; see, e.g., Iyer et al. 2014; Balseiro et al. 2015; Balseiro and Gur 2017 and Baardman et al. 2019. Furthermore, due to the digital nature of such online ad exchanges, advertisers can access user-level information before placing a bid, which motivates the importance of understanding the *state-specific* value of an advertiser action (ad action). All such tactical decisions require a deep understanding of the value of showing an ad via a specific channel at a given time. How can an advertiser assign or *attribute* value to the advertising actions taken across the different channels and media? *Attribution* is one of the central questions in online advertising. The value of each channel is an important input to media mix optimization, helps build an understanding of the customer journey, and also helps a company justify its marketing spend (United 2012; Priest 2017). Incorrect understanding of the effectiveness of online channels can result in suboptimal budget allocation, possibly resulting in significant lost revenue (Kireyev et al. 2016). Recognizing the importance of the attribution problem, the Marketing Science Institute has consistently identified attribution as a topmost research priority over the last few years (Institute 2016; Institute 2018; Institute 2020). Though multiple heuristics have been proposed and studied in this domain, there is lack of a systematic approach that has both theoretical foundations and is tractable.

Attribution is inextricably linked to causality since it involves quantifying the added value of showing an ad over the baseline value of what *would have* happened if no ad was shown (*counter-*

*factual*). We consider the following view of causality, which comes from the pioneering work of Rubin 1974:

“Intuitively, the causal effect of one treatment,  $E$ , over another,  $C$ , for a particular unit and an interval of time from  $t_1$  to  $t_2$  is the difference between what would have happened at time  $t_2$  if the unit had been exposed to  $E$  initiated at  $t_1$  and what would have happened at  $t_2$  if the unit had been exposed to  $C$  initiated at  $t_1$ .”

In the context of digital advertising, treatments  $E$  and  $C$  correspond to advertiser taking, and not taking an ad action, respectively. Accordingly, attribution involves capturing the causal effect of an ad where the baseline corresponds to not showing an ad.

Causality is an active research area, and there exist paradigms for capturing the causal effect of a treatment. See, e.g. Pearl 2009; Halpern and Pearl 2005; Chockler and Halpern 2004; Hitchcock 1997; Morgan and Winship 2014; Collins et al. 2004; Eells 1991; Hume 2003 and Rubin 1974. We comment on some of these alternative approaches in our concluding remarks. In addition to these foundational or axiomatic approaches to causality, there also exist data-driven approaches, see e.g., Bottou et al. 2013, where one uses importance sampling in a Bayesian network to estimate counterfactuals, without having to collect additional data. In particular, they get a handle on the “what would have happened” scenario using the data that one has *already* collected. In Section 2.5.2, we leverage this technique to construct data efficient algorithms.

As with the existing work in attribution, we also consider the setting where an advertiser is interested in understanding the contributions of various ads to a single product they are promoting. Even with a single product, attribution is a challenging problem. It involves distributing the value generated by a network of actions to each individual action. Such a network might have a mix of interaction effects that one needs to account for when decomposing the network value. Capturing such (possibly non-linear) interaction effects is a fundamental challenge.

### 2.1.1 Related Literature

The attribution problem has been studied widely and in diverse areas. We present a brief overview of existing approaches and refer the reader to existing surveys (Choi et al. 2017; Kannan et al. 2016) for a more complete treatment. Attribution methodologies can be classified into two broad classes: *rule-based* or *algorithmic*. We discuss both the classifications below.

#### *Rule-Based Heuristics.*

Rule-based heuristics include approaches such as *last touch attribution* (LTA), *uniform weights*, and *customized weights* (Arensman and Yeung 2016; Priest 2017; Quantcast 2013; Quantcast 2016). In LTA, an advertiser attributes all the value generated by a user to the last ad action, whereas under a uniform weights scheme, all the ad actions on a conversion path are allocated an equal credit. Ad actions receive tailored weights under a custom weights scheme. Although such heuristics are transparent and tractable, there is no rationale justifying their appropriateness as a measure for attribution. Metrics such as LTA can be unfair since they do not value the contribution of channels that build product awareness. Uniform or customized weights might appear to be a fix but there is no a priori reason to believe that attribution should be linear.

#### *Algorithmic Approaches.*

The algorithmic approaches can be classified as using either *incremental value heuristic* (IVH) (or *removal effect*) or *Shapley value* (SV) as a measure for attribution.

**IVH.** IVH computes the change in the eventual *conversion*<sup>1</sup> probability of a user when a specific ad is removed from her path. This is the most common metric for attribution (Abhishek et al. 2012; Anderl et al. 2016; Arava et al. 2018; Archak et al. 2010; Danaher and Heerde 2018; Kakalejčík et al. 2018; Li and Kannan 2014; Li et al. 2017). In this approach, one calibrates a model that predicts the conversion probability as a function of the ad actions and then, uses the estimated model to

---

<sup>1</sup>Conversion refers to the event in which a user buys the underlying product.

compute the incremental value of each action. The novelty comes from the model proposed to describe user behavior, e.g., a hidden Markov model (HMM) (Abhishek et al. 2012) or a neural network (Arava et al. 2018). There exists little (if any) formal justification for why IVH is a good attribution scheme. In Section 2.3.2, we show that IVH can result in unjustifiable attribution.

**SV.** SV (Shapley 1953) is a well-accepted concept for assigning credit to individual players in a cooperative game. The value generated by online advertising can be viewed as the outcome of a cooperative effect of the actions taken on various channels and media platforms. Dalessandro et al. 2012 pose attribution as a causal estimation problem and propose SV as an approximation scheme for the causally motivated problem. They also show that SV generalizes the probabilistic model of Shao and Li 2011. Using a stylized model, Berman 2018 shows the use of SV for attribution can be beneficial to the advertiser. Under a different stylized setting, Abhishek et al. 2017 analyze attribution contracts (including SV) used by an advertiser to incentivize two publishers that affect customer acquisition and they highlight an interesting tension between “fairness” to the publishers and “optimality” for the advertiser. Unlike in Berman 2018 and Abhishek et al. 2017, in our work, publishers are not strategic. Our work focuses on the “fairness” of the attribution to the various channels and avenues available to the advertiser. It is also worth mentioning that the online advertising industry is embracing using SV for attribution; see for example Google 2019.

**Other works.** Attribution has been tackled from various other angles. Jordan et al. 2011 use a Markovian model to motivate a payment scheme that satisfies incentive compatibility for the advertiser and is fair from the publisher’s point-of-view. Xu et al. 2014 propose a mutually exciting point process to capture dynamic interactions among various ads. Zhang et al. 2014, Ji et al. 2016, and Ji and Wang 2017 provide an interesting view of attribution via the lens of survival theory. Zhao et al. 2018 propose a regression-based relative importance method to compute the marginal contributions. From an empirical perspective, Blake et al. 2015 measure the return on investment of paid search and document the importance of accounting for the counterfactual.

### 2.1.2 Our Approach and Contributions

In spite of almost a decade of research, it remains unclear as to what is an “appropriate” or “best” attribution measure. IVH appears to be the most popular; but, to the best of our knowledge, there is no systematic framework to support it. In fact, as we later discuss, IVH suffers from serious drawbacks in certain settings. On the other hand, SV has a strong theoretical justification and a number of desirable properties such as *efficiency*, *symmetry*, *linearity*, and *null player*; in fact, SV is the unique solution to a cooperative game that has all these properties. However, estimating SV exactly is computationally intractable in general, and one has to resort to approximations (Avrachenkov et al. 2012; Castro et al. 2009; Fatima et al. 2008; Liben-Nowell et al. 2012; Littlechild and Owen 1973; Maleki et al. 2013; Michalak et al. 2013; Owen 1972). In addition, as we show in Section 2.4.2, SV is not counterfactual in nature. We seek a metric that has the desirable properties of SV, and yet is tractable and able to accommodate counterfactual reasoning.

Our contributions are as follows. First, we construct an abstract Markov chain model for the customer journey through the conversion funnel that generalizes most of the existing Markovian models in the attribution literature. In every period, the customer is in one of the finitely many *states*. The advertiser observes the state and takes an *action*. The customer transitions to a random state distributed according to a probability mass function that depends on the current state and the advertising action. We propose attributing value to each state-action pair, which is a generalization of the existing approaches that attribute only to advertising actions. This extension allows us to capture state-specific attribution for each action, the need for which is supported by the finding in Bleier and Eisenbeiss 2015. We analytically quantify the additional value of state-specific attribution and show that attributing at a state-action level allows one to capture a multitude of interactions that are missed otherwise.

Second, we show that the Markov chain model allows us to generate intuitive canonical Markovian networks that serve as a robustness check for an attribution scheme. Using these networks, we show that the current attribution metrics (LTA, IVH, and uniform) have serious limitations. Furthermore, we show how one can compute state-specific SV for each action in our Markovian

model and highlight that it does not adjust for the counterfactual, and hence, is not an appropriate metric for attribution in our setting. To the best of our knowledge, this is the first work in the literature to analyze the various existing attribution metrics using a common framework.

Third, and the main contribution of this work, is an axiomatic framework for attribution in online advertising that explicitly accounts for the counterfactual action. We propose a new metric for attribution that we call *counterfactual adjusted Shapley value* (CASV). We show that our proposed metric inherits the desirable axioms of the classical SV-based attribution. Note that we do not need the Markovian model to define CASV and establish the uniqueness of this attribution scheme. We show that this new metric leads to appropriate attribution in the Markovian networks that highlighted limitations of the existing metrics. In addition, we establish that CASV admits a crisp characterization under our Markovian model. It coincides with a *unique-uniform attribution scheme* that explicitly adjusts for the counterfactual; which, in turn, can be interpreted as a correction to the commonly used uniform attribution scheme. Furthermore, we exploit this characterization to develop simple algorithms to estimate our metric. We demonstrate the scalability of CASV for the Markovian framework by performing numerical experiments on a real-world large-scale dataset.

**Outline.** The remainder of this chapter is organized as follows. In Section 2.2, we introduce the Markovian model that describes the user journey as a function of ad exposure. In Section 2.3, we discuss how the existing attribution schemes (LTA, IVH, and uniform) apply to the Markovian model and we construct canonical Markovian networks that highlight the limitations of these attribution schemes. In Section 2.4, we introduce SV-based attribution and showcase its drawbacks. Next, we present our axiomatic framework for CASV. In Section 2.5, we show that the CASV-based attribution scheme is equivalent to the adjusted unique-uniform attribution scheme for the Markovian model, and propose simple algorithms to estimate it. In Section 2.6, we apply the proposed Markovian framework and the CASV-based attribution scheme to a large-scale real-world dataset. In Section 2.7, we discuss the value of state-specific attribution. We conclude in

Section 2.8 with some directions for ongoing and future research.

## 2.2 Model

We propose a Markovian model for user behavior where transitions in user’s state are stochastic, and are a function of only the current state and the advertiser action. This process ends when the user either quits (leaves the system) or converts (buys the product). In Section 2.2.1, we define the components of the Markov chain (denoted by  $\mathcal{M}$ ). When constructing the Markov chain, we keep most of its elements abstract to showcase its flexibility. In Section 2.2.2, we show how many practical settings can be modeled using the proposed Markovian framework. We conclude this section by defining the attribution problem for our Markovian model.

We note that our model builds upon existing works in this literature that have also used a Markovian model to describe user behavior (Abhishek et al. 2012; Anderl et al. 2016; Kakalejčík et al. 2018). We define our model in abstract terms and therefore, most of the existing models can be seen as special instances of our model.

### 2.2.1 Markovian Model of User Behavior

We first discuss the state space of the Markov chain, followed by the arrival process of the users, then the action space of the advertiser, and finally, we define the transition probabilities.

**State space.** We define  $\mathbb{S} := \{s\}_{s=1}^m$  as the set of states excluding the two absorbing states (quit  $q$  and conversion  $c$ ) and  $\mathbb{S}^+ := \mathbb{S} \cup \{q, c\}$ . In order to highlight the flexibility of our model, we do not give a concrete meaning to a state. We discuss some examples in Section 2.2.2.

**Arrival process.** External traffic arrives at state  $s \in \mathbb{S}$  w.p.  $\lambda_s$  (*initial state probability*). We define the vector  $\boldsymbol{\lambda} \in \mathbb{R}^m$  as  $[\lambda_s]_{s \in \mathbb{S}}$ . We assume no external traffic arrives at  $c$  and  $q$ .

**Action space.** We define  $\mathbb{A} := \{a\}_{a=1}^n$  as the set of advertiser actions, such as sending an e-mail or showing a display ad. We include a baseline or no-ad action ( $a = 1$ )<sup>2</sup> in  $\mathbb{A}$  that captures the impact of both the baseline online actions and any offline promotion efforts of the advertiser that have an impact on online transitions. Let  $\beta_s^a$  denote the probability that an advertiser takes action  $a \in \mathbb{A}$  at state  $s \in \mathbb{S}$ . Note that taking the baseline action  $a = 1$  in state  $s$  may still move the traffic along in the network. We denote by  $\boldsymbol{\beta}$  the collection of all  $\beta_s^a$  values and assume it is fixed.

**Transition probabilities.** We denote by  $p_{ss'}^a$ , the probability a user moves from  $s \in \mathbb{S}$  to  $s' \in \mathbb{S}$  in one transition given the advertiser takes action  $a \in \mathbb{A}$  at  $s$ . Also, for all  $(s, s') \in \mathbb{S}^2$ , we define  $p_{ss'}^\beta := \sum_{a \in \mathbb{A}} \beta_s^a p_{ss'}^a$ , which denotes the *average* transition probability. To keep the notation concise, for each  $a \in \mathbb{A}$ , we define the matrix  $\mathbf{P}^a := [p_{ss'}^a]_{(s,s') \in \mathbb{S}^2} \in \mathbb{R}^{m \times m}$ . Furthermore,  $\mathbf{P}^\beta := [p_{ss'}^\beta]_{(s,s') \in \mathbb{S}^2} \in \mathbb{R}^{m \times m}$  and  $\mathbf{B}^a := \text{diag}([\beta_s^a]_{s \in \mathbb{S}}) \in \mathbb{R}^{m \times m}$  for each  $a \in \mathbb{A}$  is a diagonal matrix. Clearly,  $\mathbf{P}^\beta$  represents the transition matrix over the *partial* state space  $\mathbb{S}$  and  $\mathbf{P}^\beta = \sum_{a \in \mathbb{A}} \mathbf{B}^a \mathbf{P}^a$ . For all  $s \in \mathbb{S}$ , we define the vector  $\mathbf{p}_s^a \in \mathbb{R}^m$  as the  $s$ -th row of  $\mathbf{P}^a$  for all  $a \in \mathbb{A}$  and  $\mathbf{p}_s^\beta \in \mathbb{R}^m$  as the  $s$ -th row of  $\mathbf{P}^\beta$ . Next, we state the only assumption we make on our problem primitives.

**Assumption 2.1** (Absorption). *The Markov chain corresponding to  $(\boldsymbol{\lambda}, \mathbf{P}^\beta)$  is absorbing, i.e., the probability each user will eventually either quit or convert from any state equals 1.*

Assumption 2.1 is equivalent to saying that from any state, there exists a positive probability path to one of the absorbing states. We discuss the precise definitions for the quit and convert states in the context of our numerical study in Section 2.6.

So far, we have not discussed the transitions to and from states  $q$  and  $c$ . We use the notation  $p_{sc}^a$  and  $p_{sc}^\beta$  to denote the action-specific and average one-step transition probabilities from  $s \in \mathbb{S}$  to  $c$  for all  $a \in \mathbb{A}$ . (For transitions from  $s \in \mathbb{S}$  to  $q$ , replace the index  $c$  by  $q$ .) Since both  $q$  and  $c$  are absorbing states, the transitions from them are self-loops w.p. 1. We define  $\mathbf{p}_c^a := [p_{sc}^a]_{s \in \mathbb{S}} \in \mathbb{R}^m$  for each  $a \in \mathbb{A}$  and  $\mathbf{p}_c^\beta := [p_{sc}^\beta]_{s \in \mathbb{S}} \in \mathbb{R}^m$ . (Note that  $\mathbf{p}_s^a$  and  $\mathbf{p}_s^\beta$  for  $s \in \mathbb{S}$  correspond to the probabilities of *leaving*  $s$  whereas  $\mathbf{p}_c^a$  and  $\mathbf{p}_c^\beta$  correspond to the probabilities of *entering*  $c$ .) Thus, a Markov

---

<sup>2</sup>We are reserving  $a = 0$  for future use as it will become transparent in Section 2.4.2.



chain  $\mathcal{M}$  is specified by  $(\lambda, \mathbf{P}^\beta, \mathbf{p}_c^\beta, \mathbf{p}_q^\beta)$ . Next, we shift our focus to three quantities of interest for the Markov chain  $\mathcal{M}$ .

**Expected number of visits.** Let  $\mathbf{F}^\beta \in \mathbb{R}^{m \times m}$  denote the *fundamental matrix* of the Markov chain  $\mathcal{M}$ , i.e., the  $(i, j)$ -th entry of  $\mathbf{F}^\beta$  equals the expected number of visits to state  $j$  when the initial state is  $i$ . Assumption 2.1 ensures that  $\mathbf{F}^\beta$  exists, and  $\mathbf{F}^\beta = (\mathbf{I} - \mathbf{P}^\beta)^{-1}$  (Grinstead and Snell 2012).

**Effective arrival rate.** Let  $\mu_s^\beta$  denote the *effective* arrival rate into state  $s \in \mathbb{S}$  and  $\boldsymbol{\mu}^\beta := [\mu_s^\beta]_{s \in \mathbb{S}} \in \mathbb{R}^m$ . It is easy to show that  $\boldsymbol{\lambda}^\top + (\boldsymbol{\mu}^\beta)^\top \mathbf{P}^\beta = (\boldsymbol{\mu}^\beta)^\top$ ; hence,  $(\boldsymbol{\mu}^\beta)^\top = \boldsymbol{\lambda}^\top \mathbf{F}^\beta$ .

**Eventual conversion probability.** We define  $h_s^\beta$  as the probability of eventually being absorbed in  $c$  from state  $s \in \mathbb{S}$  and the vector  $\mathbf{h}^\beta := [h_s^\beta]_{s \in \mathbb{S}} \in \mathbb{R}^m$ . It is easy to show that  $\mathbf{h}^\beta = \mathbf{P}^\beta \mathbf{h}^\beta + \mathbf{p}_c^\beta$ . Thus,  $\mathbf{h}^\beta = \mathbf{F}^\beta \mathbf{p}_c^\beta$ . We set  $h_q^\beta = 0$  and  $h_c^\beta = 1$ .

## 2.2.2 Model Discussion

We have defined the states and actions for the Markov chain  $\mathcal{M}$  in abstract terms. Next, we provide some possible settings for these quantities. These are, by no means, the only possible mappings. Indeed, we expect the specific definitions of states and actions to be context dependent.

One possibility for the state is a summary of past interactions with the customer, e.g., the number of visits to the product website, number of e-mails received, number of e-mails opened, number of display ads seen, number of display ads clicked, etc. These counts will need to be suitably quantized to control the size of the state space. The advertiser's actions could be to do nothing, or send an e-mail, or show a display ad, or bid for a search ad.

Another possibility for the state space is  $\mathbb{S} = \{\text{unaware, aware, interest, desire}\}$  associated with the conversion funnel used in the marketing literature (Strong 1925; Howard and Sheth 1969; Barry 1987; Bettman et al. 1998; Court 2009; Elzinga et al. 2009; Kotler and Armstrong 2010; Mulpuru 2011; Jansen and Schuster 2011; Bruce et al. 2012), which captures the journey of the user from being unaware of the product to becoming interested, and finally purchasing. Given the

fine granularity of data now available to advertisers, we believe the advertiser can infer such states using an appropriate statistical model. The action space remains the same as above.

The conversion state  $c$  refers to the user purchasing the underlying product, and is easily observable in data. The transition to the quit state  $q$  indicates that the user has decided not to buy the product<sup>3</sup>. In practice, the quit state is usually not explicitly observed in the data, but can be inferred from the customer’s inactivity level. In our experiments with real data detailed in Section 2.6, we present one simple heuristic to infer whether the customer has quit.

In order to contextualize the arrival process and the transition dynamics, it is useful to consider an advertising campaign that is run for a fixed time period. The initial state of a user is her state on the first day of the campaign. Different customers can have different initial states (depending on the level of prior engagement with the product for example) and such heterogeneity is captured by our arrival process. During the advertising campaign, the state of the customer evolves dynamically as a (stochastic) function of the ads she is exposed to, which is captured by our transition probabilities. (We provide an example based on real data when we present our numerical study in Section 2.6.)

We note that the definitions of state and action spaces can be customized as per the needs of the advertiser (since our attribution framework is developed for the abstract Markovian model). The state space is likely context-specific and driven by the user features that are relevant to a particular advertiser. For example, one can even encode the time dimension in the state if one wishes to explicitly model time. Learning context-specific state aggregation and Markov chain dynamics from data is an active area of research. We refer the reader to Hallak et al. 2013 and references therein.

### 2.2.3 The Attribution Problem

We assume that the sale of one unit of the product generates a value of 1. As a result, the total value generated by the network equals  $\lambda^\top h^p$ . The goal of an attribution model is to allocate this

---

<sup>3</sup>To tie it with the existing literature, it can be seen as the user “leaving the system” (Abhishek et al. 2017).

value to the underlying state-action pairs<sup>4</sup> in the system while accounting for the counterfactual of not taking the particular action in the state. Note that in our setting, the state-action probability  $\beta_s^a$  for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$  is fixed, i.e., we do not consider the change in  $\beta_s^a$  as a function of an attribution scheme. (Unlike the setting in Berman 2018 and Abhishek et al. 2017, the publishers here are *not* strategic.) Our focus here is on developing a framework to attribute the total generated value to each state-action pair in a manner that fairly accounts for the contribution of each state-action to the eventual conversion of the user. Let  $\pi_s^a$  denote the attribution to the pair  $(s, a) \in \mathbb{S} \times \mathbb{A}$ . Naturally, we require the total attribution across all state-action pairs to not exceed the total generated value, i.e.,  $\sum_{s \in \mathbb{S}} \sum_{a \in \mathbb{A}} \pi_s^a \leq \boldsymbol{\lambda}^\top \mathbf{h}^\beta$ . We will call an attribution scheme *budget-balanced* if it satisfies this property.

### 2.3 Current Attribution Approaches and Their Limitations

Next, we consider commonly used attribution schemes, e.g., LTA, IVH, and uniform, in the Markov chain setting, and illustrate their limitations. As we will see, our abstract Markov chain model is a useful tool for constructing examples that illustrate these limitations. At a high-level, both LTA and uniform are “backward looking” in the sense that they split the value of a path after observing its realization (from the initial state to the end). LTA allocates all the value to the last interacting state-action pair whereas uniform allocates it equally to all the state-action pairs that appeared in the path to conversion. In contrast, IVH is “forward looking” since it attributes to a given state-action pair based on what will happen in the future. It does not require the knowledge of how the actual path will unfold. All it requires is a model that can predict the change in eventual conversion probability that results if the state-action pair is removed from the path.

In many examples to come, we will sometimes assume that the no-ad action leads to the quit state with probability 1 (this could be seen as an extreme case in which the act of not advertising leads a customer to a competitor). We do so to find the simplest examples with the minimum number of moving parts. Note that this is not a limitation of our approach, as our theory does not

---

<sup>4</sup>One might want to attribute value to actions as opposed to state-action pairs. However, we attribute to state-action pairs because the impact of an action is likely to be state dependent.

require such an assumption.

### 2.3.1 Last Touch Attribution (LTA)

In LTA, all the value generated due to a purchase is attributed to the last state-action pair in the path. Denote by  $\mathcal{P}$  a random path (over state-action pairs) sampled uniformly from  $\mathcal{M}$ . For  $(s, a) \in \mathbb{S} \times \mathbb{A}$ , define

$$w_s^{a,LTA}(\mathcal{P}) := \begin{cases} 1 & \text{if } \mathcal{P} \text{ converts and } (s, a) \text{ is the last state-action pair in } \mathcal{P} \\ 0 & \text{otherwise.} \end{cases}$$

The attribution to  $(s, a) \in \mathbb{S} \times \mathbb{A}$  under the LTA scheme equals

$$\pi_s^{a,LTA} := \mathbb{E}_{\mathcal{P} \sim \mathcal{M}} [w_s^{a,LTA}(\mathcal{P})].$$

In other words,  $(s, a)$  receives complete credit if it is the last state-action pair before the convert state  $c$  on a path  $\mathcal{P}$ . Thus, LTA is clearly budget-balanced and easy to implement. However, LTA appears “unfair” since state-action pairs are not rewarded for moving the user up in the conversion funnel. We demonstrate this limitation next.

**Example 2.1** (LTA is unfair). *Consider the network in Figure 2.1 with self-loop probability  $p = 0$  (line). Under LTA, all the value goes to the ad action at state  $m$ , i.e.,  $\pi_m^{2,LTA} = 1$  and  $\pi_s^{a,LTA} = 0$  for all other pairs of  $(s, a)$ , even though the ad actions at the first  $m - 1$  states are essential to move the user to state  $m$ , because without them the user would have quit.*

### 2.3.2 Incremental Value Heuristic (IVH)

IVH allocates to each state-action pair  $(s, a)$  the increase in the eventual conversion probability by taking action  $a$  in state  $s$  as opposed to the no-ad action. For each  $a \in \mathbb{A}$ , we first define an auxiliary variable that captures the corresponding forward looking increment *conditioned* on the

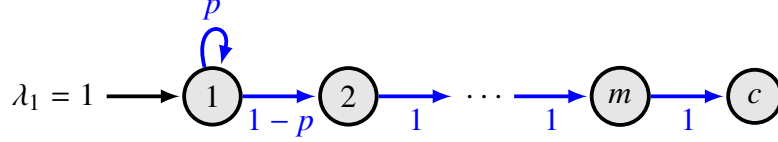


Figure 2.1: Network for Examples 2.1, 2.2, and 2.4. The action space consists of two actions: no-ad action ( $a = 1$ ) and ad action ( $a = 2$ ). The solid blue lines show transitions corresponding to the ad action. We assume the no-ad action in every state leads to a transition to the quit state w.p. 1. The advertiser takes the ad action at all states w.p. 1.

user being at a given state:

$$z^{a,IVH} := \underbrace{P^a h^\beta + p_c^a}_{\text{action } a} - \underbrace{(P^1 h^\beta + p_c^1)}_{\text{no-ad action}} = \underbrace{(P^a - P^1) h^\beta}_{\text{eventual}} + \underbrace{(p_c^a - p_c^1)}_{\text{immediate}}$$

where  $z^{a,IVH} = [z_s^{a,IVH}]_{s \in \mathcal{S}} \in \mathbb{R}^m$ . The scalar  $z_s^{a,IVH}$  can be interpreted as the allocation at a “trace” level, i.e., if the advertiser observes a user at state  $s$  and decides to take action  $a$ , the corresponding allocation would be  $z_s^{a,IVH}$ . However, we need to scale this metric for it to be seen as attribution over the entire population. In particular, given that at state  $s$ , the effective arrival rate is  $\mu_s^\beta$  and action  $a$  is taken w.p.  $\beta_s^a$ , the IVH attribution is given by

$$\pi_s^{a,IVH} := \mu_s^\beta \beta_s^a z_s^{a,IVH}. \quad (2.1)$$

Although IVH is tractable and somewhat adjusts for the counterfactual, a serious limitation is that it can distribute more value than the network generates because it pays for both eventual and immediate conversions; consequently, each conversion may be accounted for several times. We now show that this is, indeed, possible.

**Example 2.2** (IVH can over-allocate). *Consider the network in Figure 2.1 with  $p = 0$ . Under IVH, the ad action at all states receives an attribution of 1 (i.e.,  $\pi_s^{2,IVH} = 1$  for  $s \in \{1, \dots, m\}$ ) and the no-ad action at all states receives an attribution of 0 (i.e.,  $\pi_s^{1,IVH} = 0$  for  $s \in \{1, \dots, m\}$ ), resulting in a total allocation of  $m$  even though the value generated equals 1.*

A common workaround to ensure IVH is budget-balanced is to normalize the output by an

appropriate constant (*normalized IVH*). Normalizing the numbers in Example 2.2 results in the ad action at each state receiving  $1/m$ , which appears reasonable. However, as we illustrate in Example 2.3, even normalized IVH can be problematic.

**Example 2.3** (Normalized IVH can be unjustified). *Consider the network in Figure 2.2. IVH attributes 1 to the ad action at state 1 (i.e.,  $\pi_1^{2,IVH} = 1$ ) and  $1/2$  to the ad action at all other states (i.e.,  $\pi_s^{2,IVH} = 1/2$  for  $s \in \{2, \dots, m\}$ ). Thus, attribution under normalized IVH (denoted by  $\pi_s^{a,IVH}$ ) equals*

$$\pi_s^{a,IVH} = \begin{cases} \frac{1}{1+(m-1)/2} & \text{for } (s, a) = (1, 2) \\ \frac{1/2}{1+(m-1)/2} & \text{for } (s, a) \in \{(s, 2)\}_{s=2}^m \\ 0 & \text{for all other } (s, a) \text{ pairs.} \end{cases}$$

As  $m \rightarrow \infty$ , all the value goes to the ad action at states 2 to  $m$ , i.e.,  $\sum_{s=2}^m \pi_s^{2,IVH} = 1$  and  $\pi_1^{2,IVH} = 0$  as  $m \rightarrow \infty$ . This appears inappropriate as the ad action at state 1 “deserves” at least half the total value since 50% of the users convert immediately after seeing the ad at state 1.

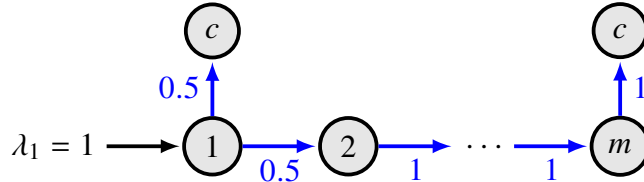


Figure 2.2: Network for Example 2.3. The action space consists of two actions: no-ad action ( $a = 1$ ) and ad action ( $a = 2$ ). The solid blue lines show transitions corresponding to the ad action. The no-ad action in every state leads to a transition to the quit state w.p. 1. The advertiser takes the ad action at all states w.p. 1.

### 2.3.3 Uniform Attribution

Under a uniform attribution scheme, the credit generated is attributed equally to all the state-action pairs that are encountered in a path. Denote by  $\mathcal{P}$  a random path (over state-action pairs)

sampled uniformly from  $\mathcal{M}$ . For  $(s, a) \in \mathbb{S} \times \mathbb{A}$ , define

$$w_s^{a,\text{uni}}(\mathcal{P}) := \begin{cases} \frac{n_s^a}{|\mathcal{P}|} & \text{if } \mathcal{P} \text{ converts and } (s, a) \in \mathcal{P} \\ 0 & \text{otherwise,} \end{cases}$$

where  $n_s^a$  equals the number of times  $(s, a)$  appears in  $\mathcal{P}$  and  $|\mathcal{P}|$  denotes the number of (not necessarily unique) state-action pairs in  $\mathcal{P}$ . The uniform attribution for  $(s, a) \in \mathbb{S} \times \mathbb{A}$  is

$$\pi_s^{a,\text{uni}} := \mathbb{E}_{\mathcal{P} \sim \mathcal{M}} [w_s^{a,\text{uni}}(\mathcal{P})]. \quad (2.2)$$

In other words,  $(s, a)$  receives an “equal cut” of each *converted* path it contributes to. Such a scheme is budget-balanced and scalable. On the line network (Figure 2.1 with  $p = 0$ ), it attributes  $1/m$  to the ad action at each state (i.e.,  $\pi_s^{2,\text{uni}} = 1/m$  for  $s \in \{1, \dots, m\}$ ), which seems reasonable. However, it does not account for the counterfactual. Furthermore, it can result in “unfair” attribution, which we show next.

**Example 2.4** (Uniform attribution can be unfair). *Consider the network in Figure 2.1 with<sup>5</sup>  $p \in (0, 1)$ . For a given path with  $n_1$  occurrences of state-action pair  $(1, 2)$ , the uniform attribution scheme attributes  $n_1/(n_1 + m - 1)$  to the state-action pair  $(1, 2)$ . As  $p$  increases, the state-action pair  $(1, 2)$  receives more credit. However, this seems inappropriate since more value is attributed to a state-action pair for “slowing down” the path to conversion. Ideally, an attribution scheme should reward the state-specific ads to push the user towards conversion.*

**Remark 2.1** (Unique-uniform). *A simple fix to the above drawback is to attribute based on the number of unique state-action pairs. For instance, in the example above, the unique-uniform scheme would attribute  $1/m$  to the ad action at each state. However, this scheme still does not account for the counterfactual, and there is no formal rationale for it. We show later that uniform allocation adjusted for uniqueness and the counterfactual does, in fact, have a strong mathematical*

---

<sup>5</sup>Note that a positive value of  $p$  can be interpreted as the ad action at state 1 not changing the state (e.g., awareness level) for a fraction of the user population, e.g. users who do not pay attention to promotional e-mails.

*rationale and will be part of our prescribed method.*

To summarize this section, the above examples highlight some limitations of existing heuristics but also point to the fact that no single heuristic “dominates the other”. In some way, IVH possesses a counterfactual form giving it some practical appeal. At the same time, it is only forward looking, ignoring all past actions and their potential contributions. Uniform, on the other hand, accounts for past actions, but is not counterfactual. Ideally, an attribution scheme would provide the best of both worlds, properly accounting for past actions while also accounting for the counterfactual associated with not advertising. We also note that our proposed Markov chain model can be used to generate simple canonical examples to test the robustness and intuitive appeal of attribution schemes.

## **2.4 Shapley Value (SV)**

In this section, we investigate SV-based attribution schemes. SV is a particular attractive metric for attribution in our context, since the value, i.e., customer conversion, is the result of the cooperative impact of actions taken during the course of the customer’s journey. We first present a primer on SV (Section 2.4.1) followed by a discussion on why a direct application to our context does not suffice (Section 2.4.2). Next, we propose a counterfactual adjusted SV (CASV) metric for attribution (Section 2.4.3) and evaluate the performance of CASV on all the motivating examples (Section 2.4.4). In this section, we hope to convince the reader that, leaving computational tractability aside, CASV is an attractive metric for attribution.

We emphasize that until Section 2.4.4, we do *not* assume that the customer journey is given by the Markovian model and hence, the axiomatic support of CASV (Theorem 2.1) does not rely on any Markovian assumption. However, we recycle the notation already introduced ( $s, a, \boldsymbol{\beta}, \mathcal{M}$ ) with the understanding that the underlying dynamics can be non-Markovian. In particular, we let the eventual conversion probability to be any arbitrary function of the state-action frequencies  $\boldsymbol{\beta}$ . For example, the transition probabilities can depend on the previous state-action pairs (as opposed to just the current state-action pair).



### 2.4.1 SV Primer

SV is solution concept in coalitional game theory (Shapley 1953). Consider a finite set  $\mathbb{P}$  of players with certain fixed strategies. Let the *characteristic function*  $v(\mathcal{X})$  denote the value generated by coalition  $\mathcal{X} \subseteq \mathbb{P}$ . The value  $v(\emptyset)$  of the empty coalition is normalized to 0. The goal is to “fairly” distribute the value  $v(\mathbb{P})$  to the individual players. Note that the players are not assumed to be strategic here, i.e., they do not choose strategies as a function of the distribution they are likely to receive; the strategies are fixed, ex-ante.

SV distributes the value  $v(\mathbb{P})$  of the grand coalition to a player  $r \in \mathbb{P}$  as follows:

$$\pi_r^{\text{Shap}} := \sum_{\mathcal{X} \subseteq \mathbb{P} \setminus \{r\}} w_{|\mathcal{X}|} \times \{v(\mathcal{X} \cup \{r\}) - v(\mathcal{X})\},$$

where

$$w_{|\mathcal{X}|} := \frac{|\mathcal{X}|!(|\mathbb{P}| - |\mathcal{X}| - 1)!}{|\mathbb{P}|!}.$$

The attractiveness of SV is rooted in the fact that it is the *unique* solution concept that satisfies the following four desirable properties:

1. **Efficiency:**  $\sum_{r \in \mathbb{P}} \pi_r = v(\mathbb{P})$ .
2. **Symmetry:** Consider players  $r, r' \in \mathbb{P}$  such that for any  $\mathcal{X} \subseteq \mathbb{P} \setminus \{r, r'\}$ ,  $r$  and  $r'$  are *equivalent*, i.e.,  $v(\mathcal{X} \cup \{r\}) = v(\mathcal{X} \cup \{r'\})$ . Then,  $\pi_r = \pi_{r'}$ .
3. **Linearity:** Consider two characteristic functions  $v_1(\cdot)$  and  $v_2(\cdot)$ . Linearity states that for all players  $r \in \mathbb{P}$ ,  $\pi_r(v_1 + v_2) = \pi_r(v_1) + \pi_r(v_2)$  and  $\pi_r(\alpha v_1) = \alpha \pi_r(v_1)$  for all  $\alpha \in \mathbb{R}$ .
4. **Null player:** Suppose player  $r \in \mathbb{P}$  does not add any value to any coalition, i.e., for all  $\mathcal{X} \subseteq \mathbb{P} \setminus \{r\}$ ,  $v(\mathcal{X} \cup \{r\}) = v(\mathcal{X})$ . Then,  $\pi_r = 0$ .

## 2.4.2 Direct Application

SV is a natural candidate for decomposing network value because the state-action pairs can be viewed as players participating in a cooperative game to achieve a common goal of converting the users. In particular, given the action intensities  $\boldsymbol{\beta}$ , the network generates a value of  $v(\boldsymbol{\beta})$  and the state-action-specific SV  $\pi_s^{a,\text{Shap}}$  represents the credit allocated to each state-action pair  $(s, a) \in \mathbb{S} \times \mathbb{A}$ . Note that we do not assume that the value function  $v(\boldsymbol{\beta})$  is given by the Markovian model proposed in Section 2.2. We first define the underlying components (players, coalitions, and the characteristic function) in our context and then discuss an important drawback of such a naïve application.

**Player.** We want to allow for the possibility that the value of an action  $a \in \mathbb{A}$  is state-dependent, and thus want to attribute to state-action pairs  $(s, a)$ . In order to make this possible in the SV setting, we must define the set of players  $\mathbb{P} = \mathbb{S} \times \mathbb{A}$ . Note that we want to allow attribution to the baseline no-ad action ( $a = 1$ ), i.e., we are explicitly interested in understanding the value of the baseline action.

**Coalition.** Next, we need to define the value associated with a coalition  $\mathcal{X} \subseteq \mathbb{P} \equiv \mathbb{S} \times \mathbb{A}$ . In the usual application of SV, one computes the value  $v(\mathcal{X})$  generated by the coalition  $\mathcal{X}$ , assuming that the rest of the players  $\mathbb{P} \setminus \mathcal{X}$  are *absent*, i.e., not contributing to the value generation process in any manner. In our setting, we accomplish this by postulating that the customer exits the network when an action  $a$  is employed in state  $s$  and the pair  $(s, a) \notin \mathcal{X}$ . We operationalize this by defining a “zero-value action” ( $a = 0$ ) that transitions customers to the quit state with probability 1, and setting the frequency  $\beta_s^0 = \sum_{a:(s,a) \notin \mathcal{X}} \beta_s^a$ , i.e., we take action 0 in state  $s$  whenever we intended to take action  $a$ , but  $(s, a) \notin \mathcal{X}$ . Note that the zero-value action is merely a construct to define the network behavior at  $(s, a) \notin \mathcal{X}$ , and is never an element of any coalition, and therefore, does not receive any attribution. Let  $\boldsymbol{\beta}^{\mathcal{X}}$  denote the state-action frequencies corresponding to the coalition  $\mathcal{X}$  where the effective set of actions is  $\mathbb{A} \cup \{0\}$ . Note that when  $\mathcal{X} = \emptyset$ , the zero-value action is

taken with probability 1 in all states, and therefore,  $v(\emptyset) = 0$ .

We note here that an alternative approach would be to attribute only to actions  $a \neq 1$ , i.e., actions other than the baseline no-ad action. In this setting, the set of players  $\mathbb{P} = \mathbb{S} \times (\mathbb{A} \setminus \{1\})$ , and we take the baseline action 1 in state  $s$  whenever we intended to take action  $a$ , but  $(s, a) \notin \mathcal{X}$ . In this setting, the value of the empty coalition corresponds to the value generated by taking only baseline no-ad action in all states, and this value may not be zero in our setting because of offline promotional activities, i.e., in this specification,  $v(\emptyset) \neq 0$  is possible. Since SV allocates the additional value  $v(\mathcal{X}) - v(\emptyset)$  to  $\mathcal{X} \subseteq \mathbb{P} = \mathbb{S} \times (\mathbb{A} \setminus \{1\})$ , the actions  $a \neq 1$  are allowed to “free ride” on the value generated by the baseline action. Our approach of introducing the zero-value action, and explicitly attributing value to the baseline no-ad action prevents this free riding. See Appendix B.1 for a more detailed discussion.

**Characteristic function.** Given  $\mathcal{X} \subseteq \mathbb{S} \times \mathbb{A}$ , we define  $v(\mathcal{X})$  to be the conversion probability when the state-action frequencies are given by  $\boldsymbol{\beta}^{\mathcal{X}}$ . For the Markov chain model,  $v(\mathcal{X}) := \boldsymbol{\lambda}^\top \mathbf{h}^{\boldsymbol{\beta}^{\mathcal{X}}}$ . Note that the key result in this section (Theorem 2.1) applies to any model where the conversion is a function of the state-action frequencies  $\boldsymbol{\beta}$ , i.e., we allow for the possibility of a non-Markovian model. Since the value depends on the (possibly non-Markovian) model  $\mathcal{M}$ , we will use the notation  $v_{\mathcal{M}}(\cdot)$  when the emphasis on  $\mathcal{M}$  is necessary.

**Shapley value.** The SV for each  $(s, a) \in \mathbb{S} \times \mathbb{A}$  is

$$\pi_s^{a, \text{Shap}}(v) := \sum_{\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s, a)\}} w_{|\mathcal{X}|} \times \{v(\mathcal{X} \cup \{(s, a)\}) - v(\mathcal{X})\}, \quad (2.3)$$

where

$$w_{|\mathcal{X}|} := \frac{|\mathcal{X}|!(mn - |\mathcal{X}| - 1)!}{(mn)!}. \quad (2.4)$$

SV depends on our choice of the characteristic function  $v(\cdot)$  and the (possibly non-Markovian) model  $\mathcal{M}$  and hence, we will use the notation  $\pi_s^{a, \text{Shap}}(v)$  and  $\pi_s^{a, \text{Shap}}(\mathcal{M})$  when the emphasis on  $v(\cdot)$  and  $\mathcal{M}$  is necessary. Though this view of attribution inherits the desirability of SV, it suffers from a critical flaw: it does not adjust for the counterfactual (Example 2.5).

**Example 2.5** (Need for counterfactual). *Consider the network in Figure 2.3 and suppose the customer behavior is described by a Markov chain model. Showing an ad in state 1 should not get any attribution since it provides no additional value over the counterfactual action (no ad). However, (2.3) attributes all the value to the ad action as shown below:*

$$\begin{aligned} \pi_1^{2, \text{Shap}} &= \underbrace{\frac{1}{2} \{v(\{(1, 2)\}) - v(\emptyset)\}}_{\text{"X}=\emptyset\text{"}} + \underbrace{\frac{1}{2} \{v(\{(1, 2), (1, 1)\}) - v(\{(1, 1)\})\}}_{\text{"X}=\{(1,1)\}\text{"}} = \frac{1}{2} \{1 - 0\} + \frac{1}{2} \{1 - 0\} = 1 \\ \pi_1^{1, \text{Shap}} &= \underbrace{\frac{1}{2} \{v(\{(1, 1)\}) - v(\emptyset)\}}_{\text{"X}=\emptyset\text{"}} + \underbrace{\frac{1}{2} \{v(\{(1, 1), (1, 2)\}) - v(\{(1, 2)\})\}}_{\text{"X}=\{(1,2)\}\text{"}} = \frac{1}{2} \{0 - 0\} + \frac{1}{2} \{1 - 1\} = 0. \end{aligned}$$

Note that  $v(\{(1, 1)\}) = 0$  since the action intensity of the no-ad action  $\beta_1^1$  is 0 in this example. Furthermore, even LTA and uniform attribution fail this sanity check.

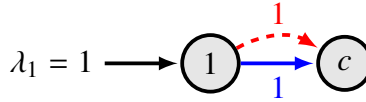


Figure 2.3: Network for Example 2.5. The action space consists of two actions: no-ad action ( $a = 1$ ) and ad action ( $a = 2$ ). Solid blue (dashed red) lines denotes transitions for the ad action (no-ad action). The advertiser takes the ad action at state 1 w.p. 1, i.e.,  $\beta_1^2 = 1$  and  $\beta_1^1 = 0$ .

### 2.4.3 Counterfactual Adjusted Shapley Value (CASV)

As we alluded to earlier, we seek a measure that provides the best of two worlds: (1) appropriately captures the contributions of the past actions (a property of, e.g., uniform and SV)<sup>6</sup> and (2) exhibits counterfactual reasoning (capturing a feature of IVH). To this end, we focus on adapting

<sup>6</sup>We will see in Section 2.5 that uniform and SV are closely linked.

SV to account for the counterfactual with respect to the baseline no-ad action and we do so by adhering to the causality framework of Rubin 1974. We first define a *counterfactual player* and then define the *counterfactual adjusted Shapley value* (CASV). Next, we motivate the desirability of CASV.

**Counterfactual player.** Following Rubin 1974, at state  $s$ , the counterfactual to taking action  $a$  w.p.  $\beta_s^a$  is to take the no-ad action (action 1) w.p.  $\beta_s^a$  (in addition to the default intensity of  $\beta_s^1$ ). Accordingly, for a given player  $(s, a) \in \mathbb{S} \times \mathbb{A}$ , we denote the counterfactual player as  $(s, 1)^a$ , where the “ $a$ ” in the superscript captures the dependence on  $\beta_s^a$ .

**Counterfactual adjusted Shapley value.** The game theoretic setup (player, coalition, characteristic function) is the same as in Section 2.4.2. For each  $(s, a) \in \mathbb{S} \times \mathbb{A}$ , we define CASV as

$$\psi_s^{a, \text{Shap}} := \sum_{\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s, a)\}} w_{|\mathcal{X}|} \times \{v(\mathcal{X} \cup \{(s, a)\}) - v(\mathcal{X} \cup \{(s, 1)^a\})\}, \quad (2.5)$$

where  $w_{|\mathcal{X}|}$  is the same as in (2.4). We use the symbol  $\psi$  instead of  $\pi$  to differentiate CASV from SV. The only change we make in going from  $\pi_s^{a, \text{Shap}}$  to  $\psi_s^{a, \text{Shap}}$  is that we replace  $v(\mathcal{X})$  by  $v(\mathcal{X} \cup \{(s, 1)^a\})$ , i.e., CASV captures the additional value added to the coalition  $\mathcal{X}$  by a player  $(s, a)$  as compared to the value added by its counterfactual player  $(s, 1)^a$ . Note that the definition of  $v(\mathcal{X})$  assumes that state-action pairs  $(s, a) \notin \mathcal{X}$  are replaced by the state-zero-action  $(s, 0)$ .

It is unclear whether CASV under the current cooperative game is equivalent to SV under a different cooperative game. However, CASV can be expressed as the difference between two SVs:

$$\psi_s^{a, \text{Shap}}(\mathcal{M}) = \pi_s^{a, \text{Shap}}(\mathcal{M}) - \pi_s^{a, \text{Shap}}(\mathcal{M}_{(s, a)}), \quad (2.6)$$

where  $\mathcal{M}$  denotes the original network and  $\mathcal{M}_{(s, a)}$  denotes a counterfactual network for  $(s, a)$  where we replace the state-action pair  $(s, a)$  by  $(s, 1)$ , i.e., take action 1 with probability  $\beta_s^1 + \beta_s^a$ . Here,  $\pi_s^{a, \text{Shap}}(\mathcal{M}_{(s, a)})$  is the *counterfactual value* of  $(s, a)$ , i.e., the value generated if no-ad action 1 was

taken instead of the ad action  $a$  at state  $s$ . (Note that the models  $\mathcal{M}$  and  $\mathcal{M}_{(s,a)}$  in (2.6) can be non-Markovian.)

Next, we present an axiomatic justification for CASV. However, one needs to be careful when defining the axioms in the counterfactual context. To be specific, efficiency should now pertain to the redistribution of additional value generated over the counterfactual value. Similarly, the definition of equivalent players should be adjusted, and the null player should correspond to a player with zero value-add. We give an axiomatic definition for CASV in Theorem 2.1 under a possibly non-Markovian model of customer behavior (see Appendix B.2 for the proof). The relevance of these axioms in the context of online advertising is discussed in Section 2.4.4, where we revisit the motivating examples.

**Theorem 2.1 (Axioms).** *CASV is the unique solution of the following four counterfactual axioms.*

1. **Counterfactual efficiency:** *The sum of CASVs equals the additional value generated over the counterfactual value, i.e.,*

$$\sum_{(s,a) \in \mathbb{S} \times \mathbb{A}} \psi_s^{a,Shap} = v(\mathbb{S} \times \mathbb{A}) - \sum_{(s,a) \in \mathbb{S} \times \mathbb{A}} \pi_s^{a,Shap}(\mathcal{M}_{(s,a)}).$$

2. **Counterfactual symmetry:** *If  $(s, a) \in \mathbb{S} \times \mathbb{A}$  and  $(s', a') \in \mathbb{S} \times \mathbb{A}$  are counterfactual equivalent, i.e.,*

$$(A) \ v(\mathcal{X} \cup \{(s, a)\}) - v(\mathcal{X} \cup \{(s, 1)^a\}) = v(\mathcal{X} \cup \{(s', a')\}) - v(\mathcal{X} \cup \{(s', 1)^{a'}\}) \text{ and}$$

$$(B) \ v(\mathcal{X} \cup \{(s, 1)^a, (s', a')\}) = v(\mathcal{X} \cup \{(s', 1)^{a'}, (s, a)\})$$

*for all  $\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s, a), (s', a')\}$ , then*

$$\psi_s^{a,Shap} = \psi_{s'}^{a',Shap}.$$

3. **Linearity:** *Consider two characteristic functions  $v_1(\cdot)$  and  $v_2(\cdot)$ . For all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ , we*

have

$$\psi_s^{a,Shap}(v_1 + v_2) = \psi_s^{a,Shap}(v_1) + \psi_s^{a,Shap}(v_2)$$

and for all  $\alpha \in \mathbb{R}$ ,

$$\psi_s^{a,Shap}(\alpha v_1) = \alpha \psi_s^{a,Shap}(v_1).$$

4. **Counterfactual null player:** Consider a player  $(s, a) \in \mathbb{S} \times \mathbb{A}$  that has a zero value-add to all coalitions that do not contain  $(s, a)$ , i.e., for all  $\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s, a)\}$ ,

$$v(\mathcal{X} \cup \{(s, a)\}) = v(\mathcal{X} \cup \{(s, 1)^a\}).$$

Then,  $\psi_s^{a,Shap} = 0$ .

In Theorem 2.1, we define  $(s, a) \in \mathbb{S} \times \mathbb{A}$  and  $(s', a') \in \mathbb{S} \times \mathbb{A}$  to be counterfactual equivalent if they satisfy (A) and (B). To better understand these conditions, consider the following three conditions for all  $\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s, a), (s', a')\}$ :

$$(A1) \quad v(\mathcal{X} \cup \{(s, a)\}) = v(\mathcal{X} \cup \{(s', a')\}),$$

$$(A2) \quad v(\mathcal{X} \cup \{(s, 1)^a\}) = v(\mathcal{X} \cup \{(s', 1)^{a'}\}), \text{ and}$$

$$(B) \quad v(\mathcal{X} \cup \{(s, 1)^a, (s', a')\}) = v(\mathcal{X} \cup \{(s', 1)^{a'}, (s, a)\}).$$

Condition (A1) states that adding either of the two players to any coalition  $\mathcal{X}$ , containing neither player, has the same effect on the network value, and condition (A2) states that the same is true for the corresponding counterfactual players. The only remaining case of interest is that we add the first player and the counterfactual of the second player, or the other way around. Condition (B) states that doing either is equivalent in terms of the characteristic function. Note that we only need conditions (A) and (B) for two players to be counterfactual equivalent, and (A1) and (A2) are sufficient conditions for (A). It is worth mentioning that our definition of counterfactual equivalent players is a generalization of equivalent players from Section 2.4.1. In particular, equivalent

players only need to satisfy condition (A1) and since the notion of a counterfactual player does not exist in the SV context, conditions (A2) and (B) are irrelevant.

Note that CASV is the unique solution to the counterfactual axioms for a (possibly non-Markovian) model where the characteristic function  $v(\cdot)$  is a function of state-action frequencies  $\beta$ . However, the Markovian customer model will be useful in gaining better insights into the rather involved and computationally intractable expression for CASV. We elaborate further in Section 2.5.

We conclude this subsection with a discussion regarding the attribution to the baseline no-ad action. By definition of CASV, the baseline no-ad action at each state receives zero counterfactual credit. However, one can attribute the counterfactual value  $\sum_{(s,a) \in \mathbb{S} \times \mathbb{A}} \pi_s^{a, \text{Shap}}(\mathcal{M}_{(s,a)})$  (“residual”) to the no-ad action in a post hoc fashion. In other words, the post hoc attribution to the no-ad action at state  $s \in \mathbb{S}$  equals

$$\sum_{a \in \mathbb{A}} \pi_s^{a, \text{Shap}}(\mathcal{M}_{(s,a)}) = \pi_s^{1, \text{Shap}}(\mathcal{M}) + \sum_{a=2}^n \pi_s^{a, \text{Shap}}(\mathcal{M}_{(s,a)}), \quad (2.7)$$

where we use the fact that  $\mathcal{M}_{(s,1)} = \mathcal{M}$ . The two terms in (2.7) have an intuitive interpretation. The first term captures the value generated by the no-ad action due to a positive value of  $\beta_s^1$  whereas the second term captures the value that *would have* been generated if the no-ad action was used in place of the other actions. It is also worth highlighting that this post hoc attribution to no-ad action also captures the value that could be attributed to any advertising action not being modeled (offline ads for example). We discuss this further when we revisit Example 2.5 in Section 2.4.4.

#### 2.4.4 Revisiting Canonical Examples

We now revisit the networks discussed so far and show that our CASV measure does not suffer from the same limitations as the other existing heuristics. We also use these relatively simple networks to illustrate the relevance of the counterfactual axioms (Theorem 2.1) in the online advertising context.



**Examples 2.1, 2.2, and 2.4.** For the network in Figure 2.1 with  $p < 1$ , all state-action pairs are counterfactual equivalent. Furthermore, since the counterfactual action, i.e., no-ad action, directs the user to the quit state with probability 1, there is no counterfactual value in the network. Therefore, it immediately follows that the CASV at every state equals  $1/m$  for the ad action and 0 otherwise.

**Example 2.3.** For the network in Figure 2.2, CASV attributes 0 to the no-ad action at all the states and the following to the ad action:

$$\begin{cases} \frac{1}{2} + \frac{1}{2m} & \text{if } s = 1 \\ \frac{1}{2m} & \text{if } s \in \{2, \dots, m\}. \end{cases}$$

This seems sensible as half of the paths convert just due to the ad action at state 1 whereas the other half of the paths convert with the help of the ad action at all the  $m$  states. This example illustrates the linearity axiom. The network in Figure 2.2 is the “sum” of two different Markov chains  $\mathcal{M}_i$ ,  $i = 1, 2$ . In chain  $\mathcal{M}_1$ , users convert directly after the ad at state 1, and in chain  $\mathcal{M}_2$ , users convert after the ad at state  $m$ , and a random user is equally likely to belong to either of the groups. The CASV on  $\mathcal{M}_1$  trivially attributes a value of 1 to the ad action at state 1 and 0 to all other players, whereas the CASV on  $\mathcal{M}_2$  attributes a value of  $1/m$  to the ad action at each state (since it can be seen as a “line” network). The CASV for the combined network follows from linearity.

**Example 2.5.** For the network in Figure 2.3, CASV allocates zero credit to the ad action, which is appropriate. However, CASV of the no-ad action is, by definition, zero. So, who gets credit for the value in the system? The answer, as discussed at the end of Section 2.4.3, is the baseline no-ad action. Recall that the attribution to the baseline no-ad action includes attribution to all “hidden” offline actions. This network exemplifies the counterfactual null player axiom since taking an ad action does not generate any additional value over the counterfactual action of taking the no-ad action.

In sum, CASV appears to be an appealing measure for attribution: it has a number of desirable properties and appears robust to various network structures. That said, similar to SV, computing CASV using (2.5) requires an exponential runtime in the number of underlying players. This is an important tractability concern that could render CASV impractical in settings with moderately-sized state and action spaces. In the next section, we show that CASV can be computed efficiently for the Markovian model.

## 2.5 CASV for the Markov Chain Model

In this section, we characterize CASV for the Markov chain model (Section 2.5.1) and use the characterization to develop simple algorithms for estimating CASV (Section 2.5.2).

### 2.5.1 Characterization

To characterize CASV for the Markov chain model, we use the fact that CASV can be expressed as a difference of two SVs (see (2.6)) and hence, analyze SV first. Proposition 2.1 connects the coalition-oriented game-theoretic construct of SV to the paths sampled from the Markov model  $\mathcal{M}$ . In particular, we show that, under our Markovian setup, SV is, in fact, identical to the unique-uniform attribution scheme (motivated in Remark 2.1).

**Proposition 2.1** (SV equals unique-uniform). *Consider  $(s, a) \in \mathbb{S} \times \mathbb{A}$  and the Markov chain  $\mathcal{M}$ . The SV of  $(s, a)$  as defined in (2.3) equals*

$$\pi_s^{a, \text{Shap}} = \mathbb{E}_{\mathcal{P} \sim \mathcal{M}} [w_s^a(\mathcal{P})],$$

where

$$w_s^a(\mathcal{P}) := \begin{cases} \frac{1}{u(\mathcal{P})} & \text{if } \mathcal{P} \text{ converts and } (s, a) \in \mathcal{P} \\ 0 & \text{otherwise.} \end{cases}$$

The function  $u(\cdot)$  returns the number of unique players and  $\mathcal{P}$  is a path over players, i.e., state-action pairs.

**Proof.** For convenience, we define  $r := (s, a)$  and  $\mathbb{P} := \mathbb{S} \times \mathbb{A}$  with the understanding that  $\pi_r^{\text{Shap}} := \pi_s^{a, \text{Shap}}$ , and  $w_r(\cdot) := w_s^a(\cdot)$ . The proof is split into three parts.

**Step 1:** Express  $v(\mathcal{X})$  as an expectation over paths:

$$\begin{aligned} v(\mathcal{X}) &= \boldsymbol{\lambda}^\top \mathbf{h}^{\beta^{\mathcal{X}}} \\ &= \boldsymbol{\lambda}^\top \left( \mathbf{I} + \mathbf{P}^{\beta^{\mathcal{X}}} + (\mathbf{P}^{\beta^{\mathcal{X}}})^2 + \dots \right) \mathbf{p}^{\beta^{\mathcal{X}}} \\ &= \mathbb{E}_{\mathcal{P} \sim \mathcal{M}(\mathcal{X})} \left[ \mathbb{I}_{\{\mathcal{P} \text{ converts}\}} \right] \\ &= \mathbb{E}_{\mathcal{P} \sim \mathcal{M}} \left[ \mathbb{I}_{\{\mathcal{P} \text{ converts}\}} \mathbb{I}_{\{\mathcal{X}(\mathcal{P}) \subseteq \mathcal{X}\}} \right], \end{aligned}$$

where  $\mathcal{M}(\mathcal{X})$  denotes the Markov chain in which only the players in coalition  $\mathcal{X}$  are “active” and  $\mathcal{X}(\mathcal{P})$  denotes the set of unique players in path  $\mathcal{P}$ .

**Step 2:** Use the representation from Step 1 in the definition of SV  $\pi_r^{\text{Shap}}$  as stated in (2.3):

$$\begin{aligned} \pi_r^{\text{Shap}} &= \sum_{\mathcal{X} \subseteq \mathbb{P} \setminus \{r\}} w_{|\mathcal{X}|} \times \{v(\mathcal{X} \cup \{r\}) - v(\mathcal{X})\} \\ &= \sum_{\mathcal{X} \subseteq \mathbb{P} \setminus \{r\}} w_{|\mathcal{X}|} \left\{ \mathbb{E}_{\mathcal{P} \sim \mathcal{M}} \left[ \mathbb{I}_{\{\mathcal{P} \text{ converts}\}} \left( \mathbb{I}_{\{\mathcal{X}(\mathcal{P}) \subseteq \mathcal{X} \cup \{r\}\}} - \mathbb{I}_{\{\mathcal{X}(\mathcal{P}) \subseteq \mathcal{X}\}} \right) \right] \right\} \\ &= \mathbb{E}_{\mathcal{P} \sim \mathcal{M}} \left[ \mathbb{I}_{\{\mathcal{P} \text{ converts}\}} \times f_r \right], \end{aligned}$$

where

$$f_r := \sum_{\mathcal{X} \subseteq \mathbb{P} \setminus \{r\}} w_{|\mathcal{X}|} \left( \mathbb{I}_{\{\mathcal{X}(\mathcal{P}) \subseteq \mathcal{X} \cup \{r\}\}} - \mathbb{I}_{\{\mathcal{X}(\mathcal{P}) \subseteq \mathcal{X}\}} \right).$$

**Step 3:** Observe that for a given path  $\mathcal{P}$ ,  $f_r$  can be seen as the SV of player  $r$  corresponding to

a different game with characteristic function defined as

$$v_{\mathcal{X}(\mathcal{P})}(\mathcal{X}) := \mathbb{I}_{\{\mathcal{X}(\mathcal{P}) \subseteq \mathcal{X}\}} \text{ for all } \mathcal{X} \subseteq \mathbb{P}.$$

A game with such a characteristic function is referred to as a *carrier game* and it is well-known (see Lemma 2 of Shapley 1953 for instance) that the SV of a player in a carrier game equals  $1/u(\mathcal{P})$  if the player is in  $\mathcal{P}$  and 0 otherwise. Therefore,

$$f_r = \begin{cases} \frac{1}{u(\mathcal{P})} & \text{if } r \in \mathcal{P} \\ 0 & \text{otherwise,} \end{cases}$$

which implies

$$\pi_r^{\text{Shap}} = \mathbb{E}_{\mathcal{P} \sim \mathcal{M}} [w_r(\mathcal{P})].$$

This completes the proof. ■

The above result enables one to characterize CASV, which constitutes the main result of this work.

**Theorem 2.2** (CASV equals unique-uniform minus counterfactual). *Consider  $(s, a) \in \mathbb{S} \times \mathbb{A}$ , the Markov chain  $\mathcal{M}$ , and the counterfactual Markov chain  $\mathcal{M}_{(s,a)}$ . The CASV of  $(s, a)$  as defined in (2.5) equals*

$$\psi_s^{a, \text{Shap}} = \mathbb{E}_{\mathcal{P} \sim \mathcal{M}} [w_s^a(\mathcal{P})] - \mathbb{E}_{\mathcal{P} \sim \mathcal{M}_{(s,a)}} [w_s^a(\mathcal{P})],$$

where  $w_s^a(\cdot)$  is as defined in Proposition 2.1.

There is a striking resemblance between this characterization of CASV and uniform attribution as defined in (2.2). In particular, the two expressions are identical except the definition of the

weight function and the counterfactual adjustment. The weight function in CASV only rewards based on whether a state-action pair appears in the path or not (as opposed to the number of times it appears). In other words, if the ad-action  $a$  was taken multiple times (say  $n_s^a$ ) at the *same* state  $s$  before the user moved to another state  $s'$ , the CASV weight function only rewards it once (as opposed to rewarding it  $n_s^a$  times). This seems very reasonable and, in fact, is the simple fix we proposed in Remark 2.1.

It is noteworthy that the coalition-oriented construct of CASV, which on the surface does not seem to be related to the paths of the underlying Markovian model, reduces to being expressed as a remarkably simple function of such paths. This connection is quite valuable as it helps to gain deeper insights regarding the structure of CASV and hence, build a better understanding. Next, we use this connection to develop simple algorithms to estimate CASV.

## 2.5.2 Algorithms

Given a set  $\mathcal{D}$  of user paths where each user path consists of various state-action-state  $(s, a, s')$  tuples, the action-specific transition probabilities and initial state probabilities can be estimated using empirical frequencies<sup>7</sup>. To estimate CASV for player  $(s, a)$ , one can sample paths from the estimated Markov chains  $\mathcal{M}$  and  $\mathcal{M}_{(s,a)}$  and use Theorem 2.2 directly. However, this involves sampling from a different Markov chain  $\mathcal{M}_{(s,a)}$  to estimate CASV for each player, which we address next.

A simple change of measure allows CASV to be expressed as

$$\psi_s^{a, \text{Shap}} = \mathbb{E}_{\mathcal{P} \sim \mathcal{M}} \left[ w_s^a(\mathcal{P}) \left( 1 - \frac{g_{(s,a)}(\mathcal{P})}{g(\mathcal{P})} \right) \right], \quad (2.8)$$

where  $g(\mathcal{P})$  and  $g_{(s,a)}(\mathcal{P})$  denote the probabilities of observing path  $\mathcal{P}$  under Markov chains  $\mathcal{M}$  and  $\mathcal{M}_{(s,a)}$ , respectively. The ratio  $g_{(s,a)}(\mathcal{P})/g(\mathcal{P})$  denotes the *importance weight* and it is easy to

---

<sup>7</sup>We are assuming the advertiser knows  $\beta$ . If not, then it can be estimated similarly.

show that

$$\frac{g_{(s,a)}(\mathcal{P})}{g(\mathcal{P})} = \prod_{s' \in \mathbb{S}^+} \left( \frac{p_{ss'}^1}{p_{ss'}^a} \right)^{n_{ss'}^a(\mathcal{P})},$$

where  $n_{ss'}^a(\mathcal{P})$  denotes the number of occurrences of the tuple  $(s, a, s')$  in  $\mathcal{P}$ . Given (2.8), it suffices to sample paths from just one Markov chain ( $\mathcal{M}$ ) to estimate CASV for all the players. In fact, each sample can be recycled for multiple players and the scheme can be implemented using a parallel (over both players and samples) architecture. We state this estimation procedure in Algorithm 4, where we assume we have access to the Markov chain  $\mathcal{M}$ .

In practice, the true model  $\mathcal{M}$  is not known, and one is forced to use a finite sample of data to compute an estimate  $\widehat{\mathcal{M}}$ , and attribute according to the CASV  $\psi_s^{a, \text{Shap}}(\widehat{\mathcal{M}})$  of the estimate model  $\widehat{\mathcal{M}}$ . We discuss sensitivity to model estimation when we report the result of our numerical experiments (Section 2.6). One can leverage the work of Bottou et al. 2013 to obtain confidence intervals of the estimate of CASV obtained using the importance weights technique in (2.8). Such confidence intervals are complementary to our sensitivity discussion and we refer the reader to Bottou et al. 2013 for more details.

Note that even if one has access to the true model  $\mathcal{M}$ , the output of Algorithm 4 is an *estimate* of the true CASV since it relies on Monte Carlo sampling. It is easy to show that the resulting estimate is unbiased, asymptotically consistent, and has a variance that decays at a rate of  $\mathcal{O}(1/N)$ , where  $N$  denotes the number of samples used in Monte Carlo. Furthermore, as we discuss in Section 2.6.3, we found  $N = 100,000$  to be large enough to obtain a stable estimate of CASV. Accordingly, moving forward, instead of focusing on the Monte Carlo error, we will focus on the sensitivity to model estimation.

It is also possible to employ a Bayesian estimation approach where one maintains a belief over the transition probability vector  $[p_{ss'}^a]_{s' \in \mathbb{S}^+}$  in the form of a Dirichlet distribution for each  $(s, a) \in \mathbb{S} \times \mathbb{A}$ . (One can also maintain a similar belief over the initial state probabilities vector  $\lambda$  and action intensities vector  $[\beta_s^a]_{a \in \mathbb{A}}$  for all  $s \in \mathbb{S}$  if required.) Since Dirichlet and multinomial

---

**Algorithm 4** Estimating CASV given Markov Chain  $\mathcal{M}$ 

---

**Require:** Markov chain  $\mathcal{M}$  and number of user paths  $N$

- 1: Initialize:  $\psi_s^{a,\text{alg}} = 0$  for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$
  - 2: **for**  $i = 1$  to  $N$
  - 3:   Sample path:  $\mathcal{P}_i \sim \mathcal{M}$
  - 4:   **if**  $\mathcal{P}_i$  converted
  - 5:     **for**  $(s, a) \in \mathcal{P}_i$
  - 6:        $\psi_s^{a,\text{alg}} = \psi_s^{a,\text{alg}} + \frac{1}{u(\mathcal{P}_i)} \left( 1 - \prod_{s' \in \mathbb{S}^+} (p_{ss'}^1 / p_{ss'}^a)^{n_{ss'}^a(\mathcal{P}_i)} \right)$
  - 7:     **end for**
  - 8:   **end if**
  - 9: **end for**
  - 10: Normalize:  $\psi_s^{a,\text{alg}} = \psi_s^{a,\text{alg}} / N$  for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$
  - 11: **return**  $\boldsymbol{\psi}^{\text{alg}} := \{\psi_s^{a,\text{alg}}\}_{(s,a) \in \mathbb{S} \times \mathbb{A}}$
- 

are conjugate distributions, the posterior belief can be computed easily. The posterior belief can be used to generate posterior samples for CASV as follows. Use the posterior belief to generate samples of the entire Markov chain  $\mathcal{M}$ , and for each sample of  $\mathcal{M}$ , use Algorithm 4 to compute an estimate for CASV, potentially in parallel. These posterior samples quantify the uncertainty in CASV that arises due to the noise in data, or lack of data.

## 2.6 Numerical Experiments

We now shift our focus to evaluating the performance of our framework of attribution on a large-scale real-world dataset and comparing it against competing metrics (LTA, IVH, uniform, and SV). We first discuss the dataset composed of several million real-world user paths (Section 2.6.1), and then the postulated Markovian model and its estimation using this real-world data (Section 2.6.2). Next, we present the attribution metrics (computed on data simulated from the estimated model) corresponding to various schemes along with some discussion (Section 2.6.3). We note that our numerics are for illustrative purposes and in particular, to see how various attribution schemes differ.

## 2.6.1 Dataset

Our real-world dataset corresponds to a single product (software) promoted and sold on the Internet by a Fortune 500 company<sup>8</sup>. The dataset contains several million user paths with a few hundred thousand conversions (purchases). Each path starts with a “sign-up”, i.e., the user creating an account on the company’s website. From the date of the sign-up, we have access to the user’s interaction with the company (*touchpoints*) for a period of 8 months<sup>9</sup>. The touchpoints can be classified based on the type of ad:

- **E-mail ad:** There are three related touchpoints: (1) advertiser sending an e-mail (ad action), (2) user opening an e-mail (user action), and (3) user clicking on a link in the e-mail (user action).
- **Display ad:** There are two related touchpoints: (1) advertiser showing a display ad (ad action) and (2) user clicking on a display ad (user action).
- **Paid search ad:** There is only one touchpoint corresponding to paid search impressions: user clicking on the paid search ad (user action). Due to the way data is collected, an advertiser usually does not know whether a paid search ad is shown to a user if the user does not click on the ad.

There are two additional touchpoints: (1) “sign-up” (user creates an account) and (2) “conversion” (user buys the product). If the user converted within 8 months, we truncate the path at the time of conversion (since we are interested in activity up to first conversion). The quit state is not explicitly observed in the data and we use the following rule to determine a transition to  $q$ : if there is no activity (user and advertiser) in the last 30 days and in the future, we mark the next state as  $q$ . Not every path ends in the conversion state or the quit state at the end of the 8 months period and we let it be that way (as opposed to forcing it to transition to the quit state).

---

<sup>8</sup>We do not disclose the name of the company and specific statistics related to real data for anonymity reasons. The company is the advertiser as opposed to a website / platform that serves ads for many advertisers.

<sup>9</sup>Our user paths data suffers from the common issues such as we lose track of a user if she clears cookies from the web browser.



We discarded certain dubious paths after discussing with our industry partner. In particular, paths with over 100 touchpoints were thrown away with the suspicion of possible bots and paths with any illogical sequence (for example, e-mail opened before being received) were also discarded (possible data recording errors). The occurrence of such paths was very low (less than 1%).

## 2.6.2 Markovian Model and Estimation

Our Markov chain in earlier sections is defined with abstract action and state spaces to showcase the flexibility of our framework. For the numerics, we use the following construction.

The action space  $\mathbb{A}$  consists of four actions: (1) no-ad, (2) e-mail, (3) display ad, and (4) paid search *click*<sup>10</sup>. In our dataset, the no-ad action is not observed explicitly and we use the following rule to “implant” no-ad actions in the user paths: if there has been no touchpoint activity (user and advertiser) in the past  $\tau$  days, we implant the no-ad action. We use  $\tau = 10$  in our numerics and perform sensitivity analysis with respect to  $\tau$  after we present the results (at the end of Section 2.6.3).

We assume that that a user goes through the following states during his decision process: (1) unaware, (2) aware, (3) interest, and (4) desire. This definition of the state space  $\mathbb{S}$  is motivated by the widely used conversion funnel in the marketing literature (Strong 1925; Howard and Sheth 1969; Barry 1987; Bettman et al. 1998; Court 2009; Elzinga et al. 2009; Kotler and Armstrong 2010; Mulpuru 2011; Jansen and Schuster 2011; Bruce et al. 2012). However, unlike most of this literature, we assume that our microscopic user level data allows us to objectively define the states using the touchpoints as follows:

- **Unaware:** User has received no e-mail, no display ad, and clicked on no paid search ad so far.
- **Aware:** At least one ad (e-mail or display) received by the user so far.

---

<sup>10</sup>Since we only observe a paid search impression when it is clicked, we use the “click” as an action even though it is a user action (as opposed to an ad action). This is a limitation of the data and affects all attribution metrics, i.e., it is not specific to the framework we propose.

- **Interest:** At least one e-mail opened by the user so far.
- **Desire:** At least one ad (e-mail, display, or paid search) clicked by the user so far.

Clearly, a “back” transition (for example, transition from aware to unaware) is impossible in our state space construction but we allow for the possibility of a “jump” (for example, transition from unaware to interest). Furthermore, transitions to quit and conversion states are possible from any of the four states in  $\mathbb{S}$ . (The state space can be made more granular depending on the needs of an advertiser and the type of data available. For instance, being “aware” as a result of receiving an e-mail might be different from being “aware” due to seeing a display impression. Furthermore, the “level of awareness” might depend on the number of ads seen and hence, it might be worthwhile to define multiple states capturing different levels of awareness. We kept our construction simple for illustrative purposes.)

Having defined the action and state spaces, we discuss the estimation of the corresponding Markov model  $\mathcal{M}$  using real data. In particular, we need to estimate  $\lambda$ ,  $\beta$ , and  $\{\mathbf{P}^a\}_{a \in \mathbb{A}}$ . The touchpoints in the dataset allow us to construct “ $(s, a, s')$ ” tuples. Since each user path starts with the “sign-up” touchpoint, the initial state is always “unaware” and hence,  $\lambda_1 = 1$ . To estimate  $\beta_s^a$  for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ , we simply compute the empirical frequencies, i.e., we count the number of times action  $a$  was taken at state  $s$  and divide it by the number of times we observe state  $s$  in the dataset. We employ the same technique<sup>11</sup> to estimate  $p_{ss'}^a$ , for all  $(s, a, s') \in \mathbb{S} \times \mathbb{A} \times \mathbb{S}^+$ . As mentioned in Section 2.5.2, it is important to understand the impact of finite data set on the CASV estimate. The CASV estimates reported here are robust because of the large number (several million) of user paths in our dataset (see Appendix B.3 for details). See also Footnote 11.

We acknowledge that our simple estimation scheme could potentially be improved. In particular, there might be undesired endogeneity effects in our data, which might result in biased estimates. For instance, a user who is more inclined to convert might be less likely to receive an ad, which is against the spirit of our  $\beta$ -randomized policy. However, we note that such estima-

---

<sup>11</sup>We also experimented with the Bayesian approach discussed at the end of Section 2.5.2. Due to the enormous number of user paths in our dataset, the posterior variance in the Bayesian approach was extremely small and hence, the results obtained were very similar. We do not present them to be concise.

tion issues are not peculiar to CASV but they also affect other attribution schemes that require the model to be estimated (IVH for instance) and hence, we place these challenges outside the scope of this work.

The summary<sup>12</sup> of the parameter estimates is as follows. In terms of actions, e-mail has the highest frequency ranging from 40% to 90% depending on the state followed by the no-ad action (10% to 50%). Combined, e-mail and no-ad actions account for over 85% of the actions at each state. The action intensity of display impression ranges from 2% to 10% and of paid search click ranges from 0.1% to 7%<sup>13</sup>. In terms of transition probabilities, we note that from each state in  $\mathbb{S}$ , the eventual conversion probability is positive under each action, which validates the absorption assumption (Assumption 2.1). Furthermore, the self-loop probability for each state-action pair (except for the no-ad action at the unaware state and the impossible self-transitions) is over 90%, indicating “slow-moving” users. For the no-ad action at the unaware state, it is around 50%. In addition, there are other patterns in the estimates that are consistent with a priori expectations. For example, a user is around 5 times more likely to move from the “aware” state to the “interest” state under the e-mail action as compared to the no-ad action. Moreover, the one-step conversion probabilities are highest for paid search click (5% to 10% depending on the state) followed by display impression (0.2% to 1.5%). As one would expect, the eventual conversion probability  $h_s^\beta$  increases monotonically with  $s \in \mathbb{S}$ , i.e., as a user goes deeper into the conversion funnel, she becomes more likely to buy the product. For instance, a user is around twice more likely to purchase if she is in the “desire” state as compared to the “unaware” state.

We also observe some estimates to be counterintuitive. For instance, our estimate of one-step conversion probability under the no-ad action is higher than under the e-mail action for all states except “unaware”. This might be a consequence of the endogeneity issue we alluded to earlier or advertising activity (such as offline ads) not captured in the online data. Recall that we use the baseline no-ad action as a proxy for all offline promotion efforts. Furthermore, for all three

---

<sup>12</sup>We do not report the exact numbers to protect the private data of our provider.

<sup>13</sup>Interestingly, the highest (among all the states) action intensity of paid search click was at the first state in the funnel (“unaware”). We believe this is a result of “unaware” users actually knowing the product (from offline channels for instance).

ad actions (e-mail, display, and paid search click), the one-step conversion probability from the “unaware” state is higher than in the “aware” state. We believe this is primarily a result of the user actually being acquainted with the product (via offline channels for instance).

### 2.6.3 Results and Discussion

Given the model estimate from real data, we estimate CASV using Algorithm 4 with  $N = 100,000$  (number of paths to be sampled)<sup>14</sup>. It took less than a second to compute the attribution for each scheme<sup>15</sup> on a single core of Intel Xeon E5-2650v2 2.6 GHz processor with 8 GB of RAM. This highlights the scalability of our approach. Figures 2.4 and 2.5 display the results. Interestingly, the five attribution schemes output quite different results and we discuss them next. We first discuss the attributions to each action (Figure 2.4), and then, comment on the state-specific attributions to each action (Figure 2.5).

In terms of attributions to actions (Figure 2.4), LTA allocates the majority of the share to e-mail and paid search click, which seems appropriate for LTA since e-mail ads are the most prevalent in our dataset and paid search click strongly hints a higher level of customer engagement (since it is a user action as opposed to an ad action) and hence, a relatively high one-step conversion probability. Since LTA does not adjust for the counterfactual, the no-ad action receives little credit. IVH allocates almost all the credit to e-mail, which aligns with the facts that IVH scales with action intensity (see (2.1)) and e-mail action occurs around 10 times more often than display and paid search click actions. We make a similar observation for uniform attribution. Due to its unique-uniform nature, SV accounts for the fact that players corresponding to e-mail action appear multiple times in certain paths and hence, the credit allocated to e-mail shrinks to roughly half (compared to uniform). This can be seen as a correction to other schemes (IVH and uniform in particular) allocating higher credit to e-mail than it deserves simply because it appears more often in the paths. Compared with LTA, SV allocates less value to paid search click, which aligns with the observation that LTA al-

---

<sup>14</sup>We repeated our computation using different seeds and obtained almost identical results, indicating  $N = 100,000$  is high enough.

<sup>15</sup>For LTA, IVH, and uniform, computing attribution is straightforward (see Section 2.3). To estimate SV, we used the unique-uniform characterization (Proposition 2.1).

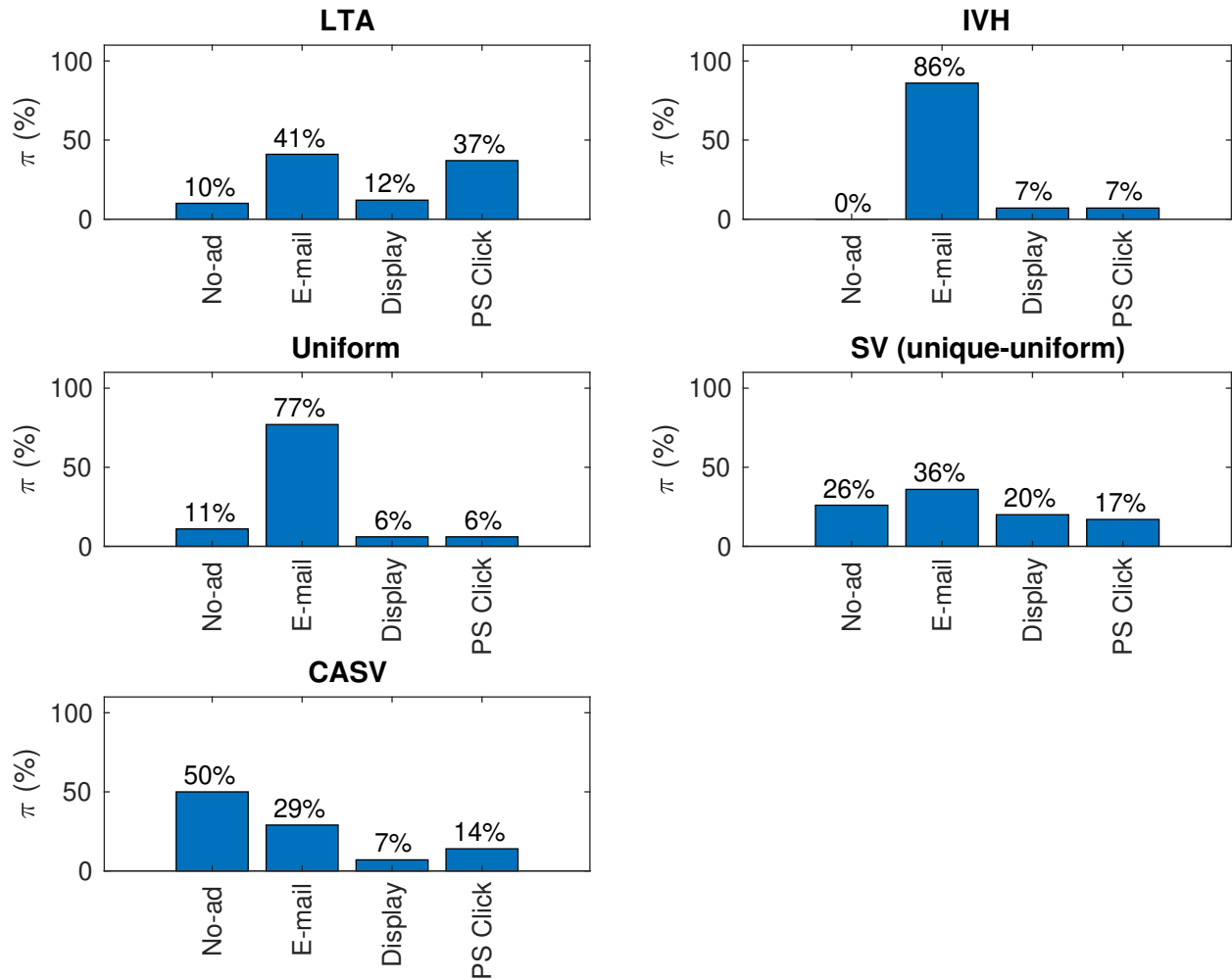


Figure 2.4: Attributions to different actions under various schemes with  $\tau = 10$ . We report the percentage attributions to each action by aggregating over states, i.e.,  $\sum_s \pi_s^a / \sum_{(s', a')} \pi_{s'}^{a'}$  for each  $a \in \mathbb{A}$ .

locates more credit than appropriate to channels appearing later in the conversion funnel. Finally, CASV corrects SV by adjusting for the counterfactual. As expected, CASV takes away a portion from e-mail, display, and paid search click and allocates it to the no-ad counterfactual. This results in no-ad receiving roughly half of the total network value<sup>16</sup>. The action that is least affected by the counterfactual adjustment is paid search click, which seems reasonable since replacing paid search click (a user action) by no-ad (an ad action) can create a big difference in total value of the network

<sup>16</sup>As noted earlier (around Equation (2.7) and when revisiting Example 2.5), the attribution to no-ad factors in the value generated by advertising actions that are not modeled (offline ads for instance) and hence, is not necessarily the value attributed *only* to no-ad.

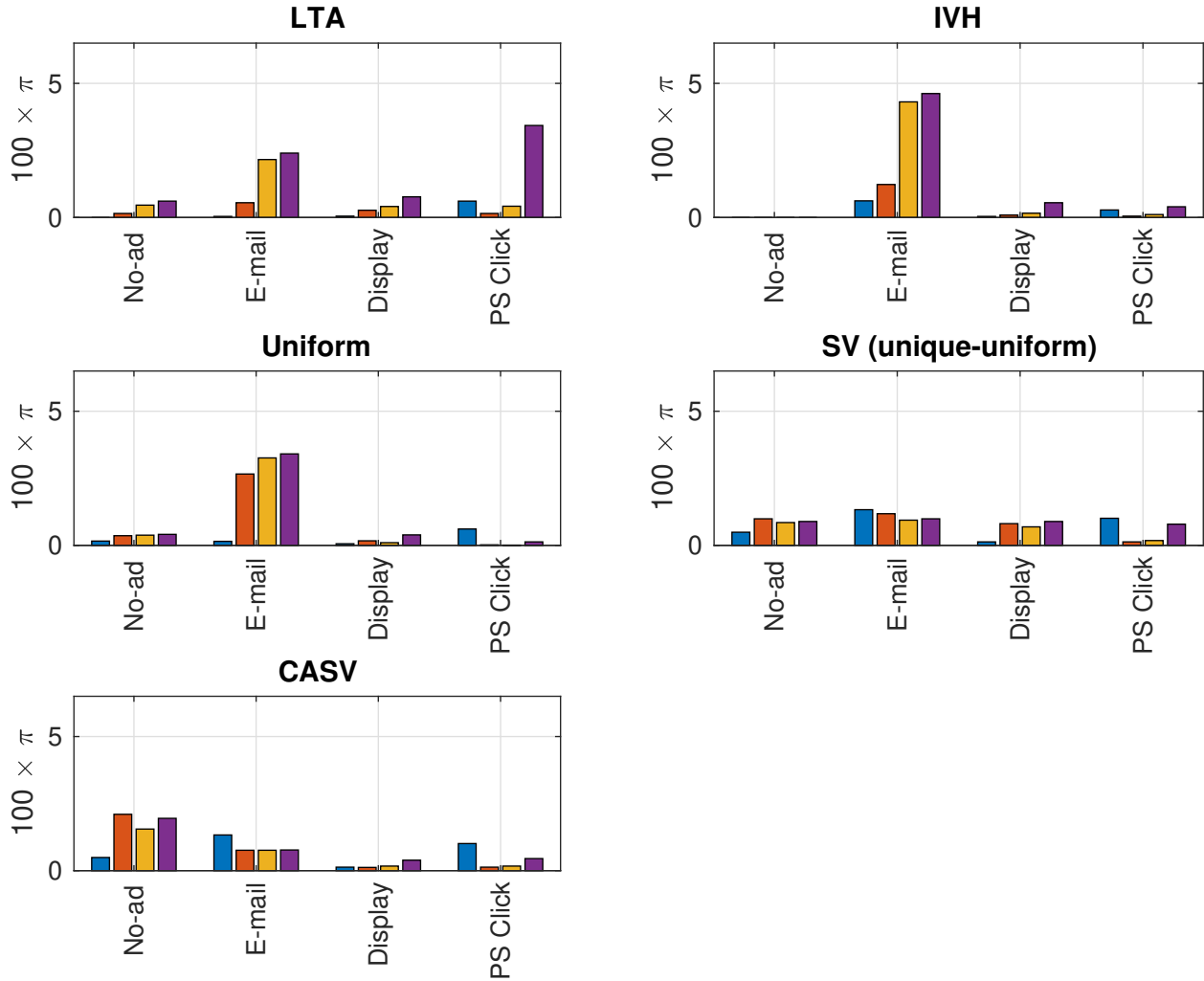


Figure 2.5: Attributions to state-action pairs under various schemes with  $\tau = 10$ . We report the attributions  $\pi_s^a$  for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ . The four colors correspond to the four states in  $\mathbb{S}$ , arranged in the natural order (unaware, aware, interest, desire).

and hence, the counterfactual value would be small.

Next, we briefly discuss the state-action-specific attributions (Figure 2.5), which are reported on an absolute scale (contrast with the percentages in Figure 2.4). As is evident, each attribution scheme can output quite different allocations to various states for a given action. LTA allocates the most to paid search click at the “desire” state. Similar to the empirical finding of Blake et al. 2015, it is possible that a user in a state of “desire” is already inclined to buy the product and performs a Google search, leading to a paid search click and a conversion. However, in the counterfactual

scenario, even if the paid search click did not occur, the user might have bought the product (by clicking on the “organic” link). In fact, the relatively low allocation to paid search click at the “desire” state under CASV provides empirical support to such a view. In terms of e-mail, existing heuristics such as LTA, IVH, and uniform have an increasing pattern in the attribution allocated with respect to the user state whereas CASV allocates the most to the first state. This also aligns with the observation above, suggesting that the existing heuristics fail to account for the counterfactual appropriately. Interestingly, all the schemes allocate a considerable amount to paid search click at the “unaware” state. This might be a result of “unaware” users actually being aware of the product (from offline sources for instance) and searching the product (on Google for example) just to make a purchase but clicking the paid search ad in the process. In theory, CASV should adjust for this phenomenon by allocating the “offline value” to the no-ad counterfactual. However, this correction is not reflected in our numerics due to the hidden nature of data corresponding to paid search impressions (recall we only observe them when they are clicked).

We conclude this section by performing sensitivity analysis on  $\tau$ . Recall that to implant the no-ad action in raw user paths, we used  $\tau = 10$ . We now experiment with a lower value of  $\tau = 7$  (Figure 2.6) and a higher value of  $\tau = 14$  (Figure 2.7) and analyze the resulting changes in the attribution numbers. It is easy to see that all the discussions we did in the case of  $\tau = 10$  (Figure 2.4) still apply, indicating robustness of high-level insights to the choice of  $\tau$ . We do see changes in the actual numbers but they align with our expectation. For example, with  $\tau = 7$ , we implant more instances of no-ad action (as compared to  $\tau = 10$ ) and hence, expect it to receive more attribution (and vice versa with  $\tau = 14$ ).

## 2.7 Value of State-Specific Attribution

In our cooperative game theory setup presented in Section 2.4, we defined each state-action pair  $(s, a) \in \mathbb{S} \times \mathbb{A}$  to be a player (*state-specific model*) and hence, had CASV  $\psi_s^{a, \text{Shap}}$  corresponding to each state-action pair. As we mentioned previously, a naïve implementation to *exactly* compute  $\{\psi_s^{a, \text{Shap}}\}_{(s,a) \in \mathbb{S} \times \mathbb{A}}$  using (2.5) would have an exponential runtime in the size of  $\mathbb{S} \times \mathbb{A}$ . Interestingly,

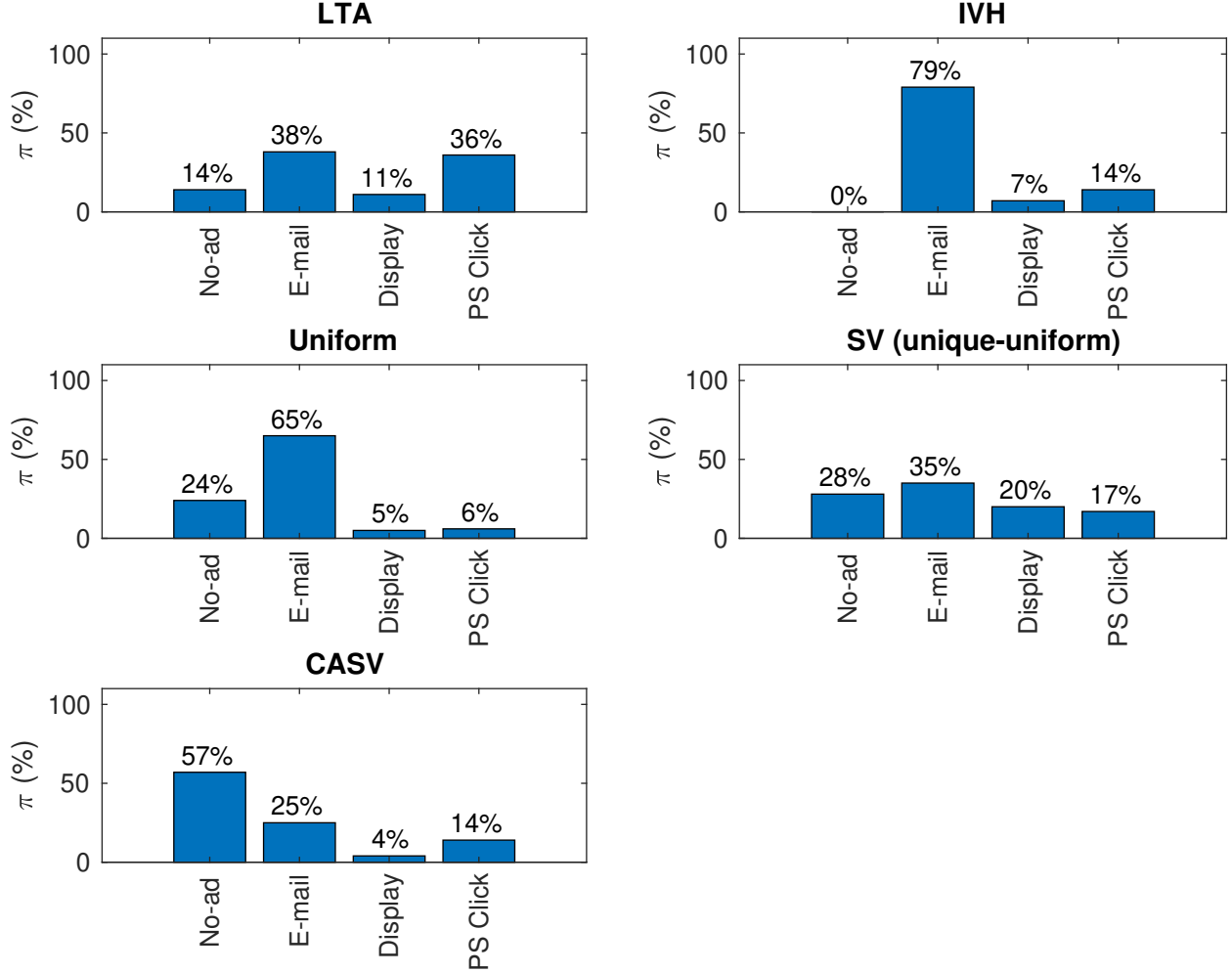


Figure 2.6: Attributions to different actions under various schemes with  $\tau = 7$ . We report the percentage attributions to each action by aggregating over states, i.e.,  $\sum_s \pi_s^a / \sum_{(s', a')} \pi_{s'}^{a'}$  for each  $a \in \mathbb{A}$ .

there is an alternative view that reduces the computational complexity of a naïve implementation to be exponential only in the size of  $\mathbb{A}$ , which might be tractable. In particular, one can define players in the cooperative game to be the actions in  $\mathbb{A}$  (*aggregated model*) and compute CASV  $\bar{\psi}^{a, \text{Shap}}$  for each action  $a \in \mathbb{A}$ . Under such a view, a coalition  $\bar{\mathcal{X}}$  corresponds to a collection of actions, i.e.,  $\bar{\mathcal{X}} \subseteq \mathbb{A}$ . To be more precise, we define the empty coalition in the same way as we defined earlier, i.e., it corresponds to the Markov chain containing all the states but action 0 being taken at all of them w.p. 1. If an action  $a \in \mathbb{A}$  is added to the empty coalition, then the advertiser takes action  $a$  at *each* state  $s \in \mathbb{S}$  w.p.  $\beta_s^a$  and so on. Under this aggregated model, each action  $a \in \mathbb{A}$  receives an



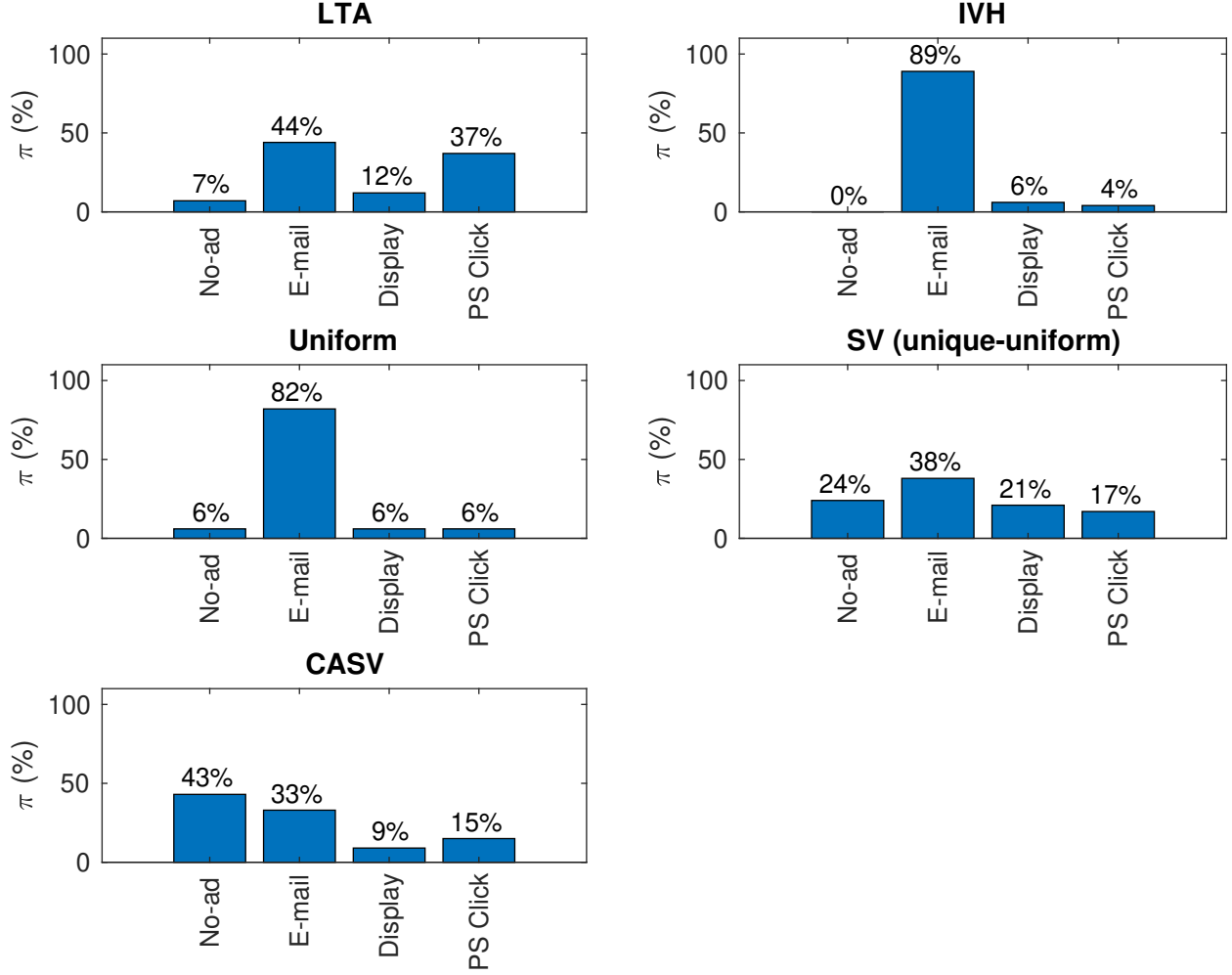


Figure 2.7: Attributions to different actions under various schemes with  $\tau = 14$ . We report the percentage attributions to each action by aggregating over states, i.e.,  $\sum_s \pi_s^a / \sum_{(s', a')} \pi_{s'}^{a'}$  for each  $a \in \mathbb{A}$ .

attribution of

$$\bar{\psi}^{a, \text{Shap}} := \sum_{\bar{\mathcal{X}} \subseteq \mathbb{A} \setminus \{a\}} \bar{w}_{|\bar{\mathcal{X}}|} \times \{v(\bar{\mathcal{X}} \cup \{a\}) - v(\bar{\mathcal{X}} \cup \{1^a\})\} \quad \text{where} \quad \bar{w}_{|\bar{\mathcal{X}}|} := \frac{|\bar{\mathcal{X}}|!(n - |\bar{\mathcal{X}}| - 1)!}{n!} \quad (2.9)$$

and  $1^a$  denotes the counterfactual player, i.e., the no-ad action (action 1) is taken at state  $s$  w.p.  $\beta_s^a$  (in addition to the default intensity  $\beta_s^1$ ) for all  $s \in \mathbb{S}$ . Similar to the state-specific model, CASV in

the aggregated model can be expressed as  $\bar{\psi}^{a,\text{Shap}}(\mathcal{M}) = \bar{\pi}^{a,\text{Shap}}(\mathcal{M}) - \bar{\pi}^{a,\text{Shap}}(\bar{\mathcal{M}}^a)$  where

$$\bar{\pi}^{a,\text{Shap}}(\mathcal{M}) := \sum_{\bar{X} \subseteq \mathbb{A} \setminus \{a\}} \bar{w}_{|\bar{X}|} \times \{v_{\mathcal{M}}(\bar{X} \cup \{a\}) - v_{\mathcal{M}}(\bar{X})\}, \quad (2.10)$$

$\mathcal{M}$  denotes the original Markov chain and  $\bar{\mathcal{M}}^a$  denotes the counterfactual network for  $a$  (in an aggregated sense), i.e., it is identical to  $\mathcal{M}$  except that we replace the transition probabilities of  $(s, a)$  by those of  $(s, 1)$  for all  $s \in \mathbb{S}$ . Furthermore, SV  $\bar{\pi}^{a,\text{Shap}}$  for  $a \in \mathbb{A}$  in the aggregated model can be characterized as a unique-uniform scheme using the same proof technique as for Proposition 2.1:

$$\bar{\pi}^{a,\text{Shap}} = \mathbb{E}_{\mathcal{P} \sim \mathcal{M}} [\bar{w}^a(\mathcal{P})] \quad \text{where} \quad \bar{w}^a(\mathcal{P}) := \begin{cases} \frac{1}{\bar{u}(\mathcal{P})} & \text{if } \mathcal{P} \text{ converts and } a \in \bar{\mathcal{P}} \\ 0 & \text{otherwise.} \end{cases}$$

The function  $\bar{u}(\cdot)$  returns the number of unique actions and  $\bar{\mathcal{P}}$  denotes the actions in the path  $\mathcal{P}$  (over state-action pairs), i.e.,  $\bar{\mathcal{P}} := \{a : (s, a) \in \mathcal{P}\}$ . Thus, the CASV  $\bar{\psi}^{a,\text{Shap}}$  for  $a \in \mathbb{A}$  in the aggregated model equals

$$\bar{\psi}^{a,\text{Shap}} = \mathbb{E}_{\mathcal{P} \sim \mathcal{M}} [\bar{w}^a(\mathcal{P})] - \mathbb{E}_{\mathcal{P} \sim \bar{\mathcal{M}}^a} [\bar{w}^a(\mathcal{P})],$$

which is analogous to Theorem 2.2.

If the size of  $\mathbb{A}$  is relatively small (which might be the case for certain retailers), CASV can be computed *exactly* in the aggregated model using (2.9). Accordingly, it is of interest to check whether using the granular state-specific model (which is computationally demanding) provides some quantifiable value over the aggregated model. However, one needs to be careful when comparing the attributions of two models as they are on different “scales”. The state-specific model outputs attribution at a state-action level whereas the aggregated model outputs at an action level. Furthermore, decomposing the action level allocation of the aggregated model to the state-action level seems non-trivial. On the other hand, aggregating the state-action level allocations of the state-specific model is straightforward ( $\sum_{s \in \mathbb{S}} \psi_s^{a,\text{Shap}}$ ). Therefore, we will compare  $\sum_{s \in \mathbb{S}} \psi_s^{a,\text{Shap}}$  to

$\bar{\psi}^{a,\text{Shap}}$  to answer the following question: does computing state-specific attribution and then aggregating over states provide a different answer than directly computing attribution that is not state-specific? Clearly, this is a stronger notion of comparison in the sense that if  $\sum_{s \in \mathbb{S}} \psi_s^{a,\text{Shap}}$  differs from  $\bar{\psi}^{a,\text{Shap}}$ , then any state-specific decomposition of  $\bar{\psi}^{a,\text{Shap}}$  would be “inappropriate”.

Our unique-uniform characterizations imply that

$$\sum_{s \in \mathbb{S}} \psi_s^{a,\text{Shap}} - \bar{\psi}^{a,\text{Shap}} = \mathbb{E}_{\mathcal{P} \sim \mathcal{M}} \left[ \sum_{s \in \mathbb{S}} w_s^a(\mathcal{P}) - \bar{w}^a(\mathcal{P}) \right] - \left[ \sum_{s \in \mathbb{S}} \mathbb{E}_{\mathcal{P} \sim \mathcal{M}_{(s,a)}} [w_s^a(\mathcal{P})] - \mathbb{E}_{\mathcal{P} \sim \bar{\mathcal{M}}^a} [\bar{w}^a(\mathcal{P})] \right],$$

which gives a path-based view of the difference between the two approaches. For instance, in a network with zero counterfactual value, if a path contains  $x$  unique state-action pairs with  $x - 1$  of them corresponding to the same action, the state-specific weight function allocates  $1/x$  to each state-action pair and hence, the repeated action receives an attribution of  $(x - 1)/x$  to account for its state-dependent impact. On the contrary, the aggregated model allocates  $1/2$  to both the unique actions. Though useful, such a path-based view does not build an understanding in terms of the type of interactions among players captured (and missed) by the two approaches. To facilitate such a fundamental understanding, we define two types of coalitions (assuming the underlying players are state-action pairs).

**Type 1 coalition.** In a type 1 coalition (say  $\mathcal{X}_1$ ), if a player  $(s, a)$  is in  $\mathcal{X}_1$ , then the player  $(s', a)$  is in  $\mathcal{X}_1$  for all  $s' \in \mathbb{S}$ . In other words, if an action is present at a single state, then it is present at all the states. We denote by  $\mathbb{T}_1$  the set of all type 1 coalitions. Type 1 coalitions correspond to the coalitions in the aggregated model. Intuitively, both the aggregated and the state-specific models capture the interactions present in type 1 coalitions. However, it is not clear whether the “weights” assigned to each of these interactions are the same in the two models. (We shed more light on this issue in Theorem 2.3.)

**Type 2 coalition.** Any coalition that is a subset of  $\mathbb{S} \times \mathbb{A}$  but not a type 1 coalition is defined as a type 2 coalition. In other words, type 2 coalitions have at least one action that is present at one or

more states but not at all the states. We define  $\mathbb{T}_2$  as the collection of all type 2 coalitions. It should be intuitively clear from (2.9) that the aggregated model will miss all the interactions present in type 2 coalitions. (We formalize this intuition in Theorem 2.3.)

We are now equipped to precisely quantify the value of state-specific attribution in terms of the interactions among the underlying players and we do so in Theorem 2.3, the proof of which is presented in Appendix B.4.

**Theorem 2.3** (Value of state-specific attribution). *For any action, computing state-specific CASV and then aggregating over states is not the same as computing CASV over the aggregated model. In particular, for all  $a \in \mathbb{A}$ ,*

$$\sum_{s \in \mathbb{S}} \psi_s^{a, \text{Shap}} - \bar{\psi}^{a, \text{Shap}} = \sum_{\mathcal{X} \in \mathbb{T}_1} \{c^a(\mathcal{X}) - \bar{c}^a(\mathcal{X})\} \{v_{\mathcal{M}}(\mathcal{X}) - v_{\bar{\mathcal{M}}^a}(\mathcal{X})\} + \sum_{\mathcal{X} \in \mathbb{T}_2} \sum_{s \in \mathbb{S}} c_s^a(\mathcal{X}) \{v_{\mathcal{M}}(\mathcal{X}) - v_{\mathcal{M}_{(s,a)}}(\mathcal{X})\},$$

where  $c^a(\mathcal{X}) := \sum_{s \in \mathbb{S}} c_s^a(\mathcal{X})$ ,

$$c_s^a(\mathcal{X}) := \begin{cases} w_{|\mathcal{X}|-1} & \text{if } (s, a) \in \mathcal{X} \\ -w_{|\mathcal{X}|} & \text{otherwise} \end{cases} \quad \text{and} \quad \bar{c}^a(\mathcal{X}) := \begin{cases} \bar{w}_{|\bar{\mathcal{X}}|-1} & \text{if } a \in \bar{\mathcal{X}} \\ -\bar{w}_{|\bar{\mathcal{X}}|} & \text{otherwise.} \end{cases}$$

The notation  $\bar{\mathcal{X}}$  denotes the actions in  $\mathcal{X}$ , i.e.,  $\bar{\mathcal{X}} := \{a : (s, a) \in \mathcal{X}\}$ .

We now parse the mathematical result presented above. The first term in the difference corresponds to type 1 coalitions. Such coalitions are captured by both the models. However, it is not true that the two models assign equal weights to them ( $c^a(\mathcal{X})$  vs.  $\bar{c}^a(\mathcal{X})$ ). On the other hand, the second term corresponds to type 2 coalitions, which the aggregated model completely disregards (note that there is no  $\bar{c}$  weight in the second term). Hence, the aggregated model misses a multitude of interactions that are captured by the state-specific model.

**Remark 2.2.** *It is easy to show that  $c^a(\mathcal{X})$  for any coalition  $\mathcal{X} \subseteq \mathbb{S} \times \mathbb{A}$  and action  $a \in \mathbb{A}$  equals*

$$c^a(\mathcal{X}) = m^a(\mathcal{X})w_{|\mathcal{X}|-1} - (m - m^a(\mathcal{X}))w_{|\mathcal{X}|},$$

where the scalar  $m^a(\mathcal{X})$  equals the number of players in the set  $\{(s, a)\}_{s \in \mathbb{S}}$  that exist in the coalition  $\mathcal{X}$  and  $m$  equals the number of states in  $\mathbb{S}$  (refer to the definition of  $\mathbb{S}$  in Section 2.2.1). Furthermore, for  $\mathcal{X} \in \mathbb{T}_1$ , we have

$$m^a(\mathcal{X}) = \begin{cases} m & \text{if } a \in \bar{\mathcal{X}} \\ 0 & \text{otherwise.} \end{cases}$$

As a result, the coefficient in the first term simplifies to

$$c^a(\mathcal{X}) - \bar{c}^a(\mathcal{X}) = \begin{cases} mw_{|\mathcal{X}|-1} - \bar{w}_{|\bar{\mathcal{X}}|-1} & \text{if } a \in \bar{\mathcal{X}} \\ \bar{w}_{|\bar{\mathcal{X}}|} - mw_{|\mathcal{X}|} & \text{otherwise.} \end{cases}$$

Having quantified the difference between the state-specific and aggregated models, we show that the ratio of the two models can be arbitrarily large, which accentuates the importance of state-specific attribution.

**Proposition 2.2.** *The ratio of the aggregated model's attribution to the aggregation of the state-specific model's attribution can be arbitrarily large. Mathematically,*

$$\sup_{\mathcal{M}} \sup_{a \in \mathbb{A}} \frac{\bar{\psi}^{a, \text{Shap}}(\mathcal{M})}{\sum_{s \in \mathbb{S}} \psi_s^{a, \text{Shap}}(\mathcal{M})} = \infty.$$

A proof of Proposition 2.2 is presented in Appendix B.4. We conclude this section by showing Figure 2.8, which displays the difference between the two models (state-specific and aggregated) on the simulated dataset from Section 2.6. The aggregated model underestimates the contributions of no-ad and e-mail actions by around 10%.

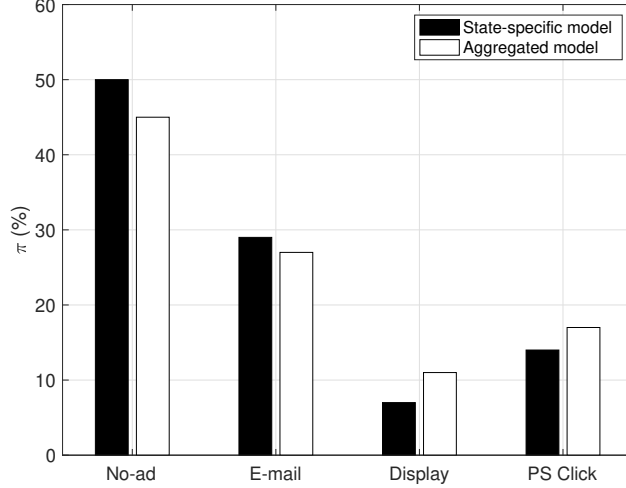


Figure 2.8: Value of state-specific attribution on the simulated dataset. For the state-specific model, we report the percentage attributions to each action by aggregating over states, i.e.,  $\sum_s \psi_s^{a, \text{Shap}} / \sum_{(s', a')} \psi_{s'}^{a', \text{Shap}}$  for each  $a \in \mathbb{A}$ . For the aggregated model, we report  $\bar{\psi}^{a, \text{Shap}}$  for each  $a \in \mathbb{A}$ .

## 2.8 Conclusions and Further Research

In this chapter, using a Markovian model for user behavior, we propose counterfactual adjusted Shapley value (CASV) as a new metric for attribution in online advertising. We establish its theoretical foundations and appropriateness in two ways. First, we show the robustness of the proposed measure over various canonical settings in which the existing metrics fail. Second, we provide an underlying axiomatic framework motivated by game theory and causality that supports our choice. We show that the CASV in our Markovian model is an adjustment to the unique-uniform attribution scheme. Finally, we propose multiple approaches to estimate our metric for the Markov chain model and show its scalability through numerical experiments on a real-world large-scale dataset, in addition to benchmarking it against existing metrics.

Our work proposes a very general Markovian model for the customer behavior and shows that the CASV metric can be efficiently computed for any such Markov chain. In Section 2.6, we propose a state space definition that is adequate for the particular application. Since the resulting attribution depends on the definition of state, defining an appropriate state space is important. However, we expect that the appropriate definition for the state is likely to be very context specific.

Learning context-specific state aggregation and Markov chain dynamics from data is an active area of research. We refer the reader to Hallak et al. 2013 and references therein where the authors posit a model selection procedure when the candidate models are generated by domain experts. Furthermore, since our axiomatic justification does not require the Markovian assumption, characterizing CASV under non-Markovian models is of interest.

Our work does not account for the cost of ad actions; a cost-aware attribution scheme is an interesting extension to explore since a relatively cheap but less effective ad action might be more valuable than an expensive but more effective ad action. Further, in this work, we implicitly assume that the various ad actions / channels cooperate with each other towards the common goal of maximizing network value instead of being strategic and maximizing their individual values. Motivated by the works of Berman 2018 and Abhishek et al. 2017, it would be interesting to analyze our framework when the individual channels are strategic. However, it is unclear if a *cooperative* game theory framework is even appropriate under this strategic setting. For example, the *unique*-uniform scheme provides each channel an incentive to pose as two smaller but distinct channels.

Our framework could be of interest in other domains. For instance, there has been a growing interest in the field of “interpretable machine learning”, where the goal is to attribute the output of a prediction model to the individual features of the input (Baehrens et al. 2010; Sundararajan et al. 2017; Lundberg and Lee 2017) or to the components of the prediction model (Dhamdhere et al. 2018; Leino et al. 2018). Extensions of the ideas presented in this work have implications to answering such questions.

In this work, we modify Shapley value using Rubin’s definition of causality (Rubin 1974) and the counterfactual we consider can be viewed as one of a family of possible counterfactual constructs. In Appendix B.5, we describe other possible definitions that are worthy of further exploration. In addition, it is of interest to explore alternative formulations for causality. See, e.g. Pearl 2009; Halpern and Pearl 2005; Chockler and Halpern 2004; Hitchcock 1997; Morgan and Winship 2014; Collins et al. 2004; Eells 1991 and Hume 2003 for approaches to causality in the computer science and philosophy literature. Chockler and Halpern 2004 proposed metrics such

as “degree of responsibility” and “blame” in order to *quantify* the causal effect of one variable on another. However, computing these metrics is computationally expensive. Furthermore, these metrics lack axiomatic support. Also, similar to IVH, these metrics are not budget balanced. In fact, Chockler and Halpern 2004 mention adjusting their notions using Shapley value as a potential future work. Accordingly, though we have unified the existing attribution literature in this work, there is more to explore in terms of the set of counterfactuals to consider.



## Chapter 3: Beyond Myopia: Model Free Approximate Bayesian Learning for Conversion Funnel Optimization

Deciding what ad to show based on consumer-level data is one of the most important decisions in modern-day marketing. We study the problem of optimal sequential personalized interventions from the point-of-view of a firm promoting a product under a fairly general conversion funnel model of consumer behavior. Our model captures the state of each consumer (interaction history with the firm for example) and allows the consumer behavior to vary as a function of both her state and firm's interventions. In contrast with existing approaches that model only the myopic value of an intervention, we also model the long-run value by allowing the firm to make sequential interventions to the same consumer. The objective of the firm is to maximize the probability of conversion (consumer buying the product) and a key challenge is the firm does not know the state-specific effects of various interventions. To help make personalized intervention decisions, we propose a decision-making algorithm, which we call *model-free approximate Bayesian learning*. Our algorithm inherits the simplicity of Thompson sampling for a multi-armed bandit setting and maintains an approximate belief over the value (myopic plus long-run) of each state-specific intervention. The belief is updated as the algorithm interacts with the consumers. Despite being an approximation to the Bayes update, we prove the asymptotic optimality of our algorithm. We supplement our theory with numerics on a real-world large-scale dataset, where we show the dominance of our algorithm over traditional approaches that are myopic or estimation-based. Furthermore, in contrast to the estimation-based approaches, our algorithm is able to adapt automatically to the underlying changes in consumer behavior (concept shift) and maintains a high level of uncertainty on the value of less explored consumer segments (covariate shift).

### 3.1 Introduction

Over the last two decades, digitization has been drastically shifting the way businesses operate. A significant portion of a modern-day organization’s operations (ranging from marketing to sales) happen over the Internet, boosting the size of the digital economy to be between 4.5 to 15.5 per cent of world GDP (UNCTAD 2019). This shift from offline to online nature of operations has provided businesses access to “big data” (IBM 2020), giving them an unprecedented opportunity to understand customer behavior at a *personalized* level and make data-driven decisions to enhance customer experience and ultimately, boost revenue (Mišić and Perakis 2020).

Consider a firm that promotes a product or service online (Netflix selling its paid membership service for example). A central goal of such a firm is to maximize its *conversion probability*, i.e., given a new customer lead (*consumer*), maximize the probability the consumer buys its product. To do so, the firm performs *sequential interventions* (Netflix sending promotional emails as shown in Figure 3.1 for example) as a function of the information it collects on the consumer (*state*). The intervention space of the firm includes multiple *actions* (type of email for example) and the consumer’s state evolves dynamically as a function of the firm’s actions (in an endogenous manner). For instance, a consumer who has interacted with a previous intervention (opening an email and clicking on one of the links in it for example) might exhibit a different behavior (and hence, might be in a different “state”) as compared to a consumer who has “avoided” previous interventions. Adding more to the complexity, the firm does not know apriori the effect of its state-specific interventions on the consumer’s behavior and hence, needs to *learn* or *estimate* as it interacts with various consumers.

In this chapter, we focus on the *conversion funnel optimization* problem (formally defined in Section 3.2). Informally, we consider a firm promoting a single product. The firm interacts with multiple consumers in parallel and its goal is to maximize the conversion probability. We model the consumer state by a conversion funnel, allowing different consumers to be in different states at each point of time. The firm observes the state of each consumer and performs *personalized*

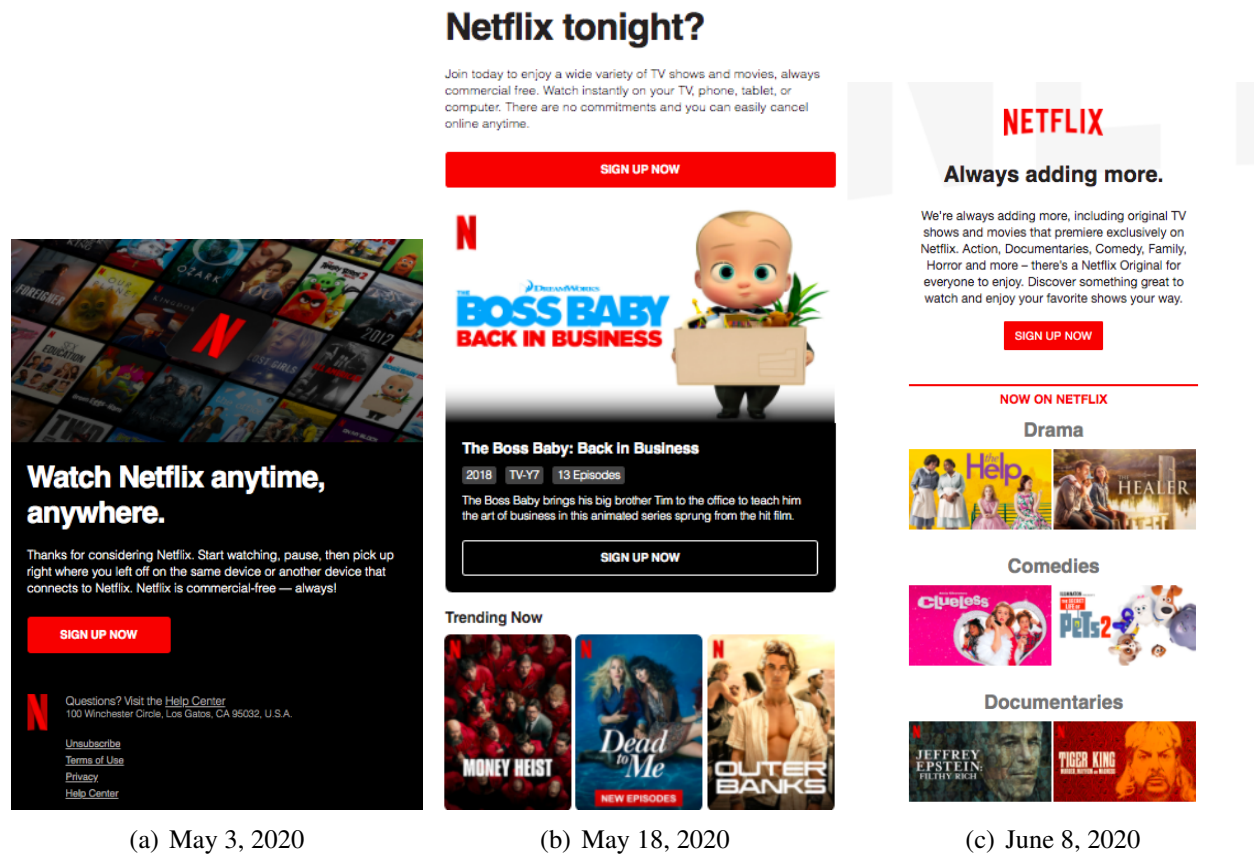


Figure 3.1: Sequence of emails received by one of the authors after providing his email to Netflix (but not subscribing to the membership). The emails were sent with a 15-20 days gap in between (May 3, May 18, and June 8) and the contents of each of the email were unique. The subject line of the three emails were “Movies & TV shows your way”, “Watch TV shows & movies anytime, anywhere”, and “Netflix - something for everyone”, respectively.

interventions, i.e., the intervention targeted at a consumer can depend on her state. As a function of the current state and the personalized intervention, the consumer state evolves. The process for each consumer continues until she buys the product (subscribing to Netflix’s paid membership for example) or leaves (unsubscribing from Netflix’s email list for example). For any consumer, we allow the firm to perform sequential interventions. We consider the setting where the firm a priori does not know the underlying parameters of the conversion funnel (transition probabilities between various states for example) and needs to learn / estimate the effects of state-specific interventions by interacting with the consumers. This maps to a real-world scenario in which the firm is promoting a new product, or an existing product to an unexplored consumer segment such as a new geography

(covariate shift), or perhaps there has been a change in the underlying consumer behavior possibly due to shifts in macroeconomic conditions, competitors' policies, or other marketing activities of the firm itself (concept shift). (For ease of exposition, we introduce our framework using a single product and costless interventions. However, our framework extends to multiple products and interventions with costs. We discuss these extensions in Section 3.5.)

Even with a single product and costless interventions, optimizing the conversion funnel is challenging, primarily due to complex consumer behavior. The consumer behavior can be affected by the earlier interventions (carryover and spillover effects) and the consumer's interactions with such interventions (Manchanda et al. 2006; Abhishek et al. 2012; Braun and Moe 2013; Xu et al. 2014; Li and Kannan 2014; Ghose and Todri 2015; Bleier and Eisenbeiss 2015; Kireyev et al. 2016; Bruce et al. 2017; Zantedeschi et al. 2017). For example, a consumer who interacts with an initial intervention (opening a promotional email) might be more likely to convert in the future than a consumer who does not. These effects can be non-linear (Chatterjee et al. 2003). There can also be temporal effects (Sahni 2015; Bleier and Eisenbeiss 2015; Sahni et al. 2019), including marketing fatigue (Sinha and Foscht 2007; Cheng et al. 2010; Byers et al. 2012; Abebe et al. 2018; Cao et al. 2019). For example, sending promotional emails to a consumer every day might annoy her, leading her to leave the system (by unsubscribing to firm's email list). The key challenge here is that the consumer behavior can be a complicated function of her state (past interactions, time since last intervention, etc.) and apriori, the firm does not know how the consumer behaves as a function of her state and the intervention. The firm needs to learn / estimate such state-specific effects of its various interventions. Doing so while optimizing the funnel at the same time is challenging.

It is worth mentioning that the problem of conversion funnel optimization is quite widespread in real-life marketing applications. For instance, there are multiple firms operating in the space of email campaign management<sup>1</sup> and the global market size for just email marketing is estimated to be around USD 7.5 billion, with a projected growth to USD 17.9 billion by 2027 (ReportLinker

---

<sup>1</sup>See <https://www.privvy.com/>, <https://www.klaviyo.com/>, <https://mailchimp.com/>, <https://www.constantcontact.com/>, <https://sendgrid.com/>, and <https://cordial.com/> for a sample of firms operating in this space.

2020). To define the focus of our work, we highlight an important characteristic that is omnipresent in the email marketing industry. In particular, when a consumer journey begins (for instance, consumer providing her email to Netflix), the firm has very limited data on the consumer. In particular, all the firm has is consumer’s email and perhaps her name. Hence, in this work, we focus on a *limited consumer features regime*. In particular, we do not assume the firm has access to the *features* of the consumers (age, sex, location, for example) but we let the firm learn about the consumers’ preferences as it interacts with them via sequential interventions. Nonetheless, we discuss an extension of our framework to a setting in which the firm has access to consumer features in Section 3.5.

### 3.1.1 Related Literature

Conversion funnel optimization relates to the existing work in the marketing literature focussed on determining optimal marketing actions. Motivated by Bertsimas and Mersereau 2007, we classify the related work into four sets as shown in Table 3.1: (1) estimation-based myopic approaches, (2) estimation-based non-myopic approaches, (3) learning-based myopic approaches, and (4) learning-based non-myopic approaches. We say an approach is *myopic* if it maximizes immediate / single-period reward (one-step conversion probability for example) and we say an approach is *non-myopic* if it maximizes the long-run reward (eventual conversion probability for example). We say an approach is *estimation-based* if it uses data from a fixed period of time to estimate the underlying parameters and make decisions using such fixed estimates. On the other hand, an approach is *learning-based* if it does not use a pre-specified period to estimate parameters but learns them while optimizing for rewards at the same time.

**Estimation-based myopic approaches.** Most of the traditional work in marketing falls in the first set, i.e., estimation-based myopic approaches. These approaches first propose a model that predicts immediate reward (one-step conversion probability for example) as a function of the intervention and possibly the consumer state / features and then, estimate the proposed model to make

	<b>myopic</b>	<b>non-myopic</b>
<b>estimation</b>	1	2
<b>learning</b>	3	4

Table 3.1: Classification of the related marketing literature into four sets. Most of the traditional work belongs to set 1 and there has been noticeable progress in sets 2 and 3 over the last two decades. We focus on set 4, which is a relatively unexplored space in marketing.

intervention decisions. Since such an approach is not our main focus, we do not discuss this literature extensively but briefly mention the recent work of Simester et al. 2020 and refer the reader to references therein. Simester et al. 2020 consider seven widely used machine learning methods to predict single-period reward and investigate how a firm can use experimentation to estimate their parameters and then, use such estimates to optimize marketing decisions when prospecting for new customers. A key limitation of such approaches is that even though the parameter estimates can be specific to the state / features of the consumer, the final decision is myopic (since it maximizes single-period reward) and hence, can result in suboptimal reward. In the words of Simester et al. 2006,

“As early as 1960, it was recognized that catalog companies may be able to profit by focusing on long-run rather than immediate profits when designing their mailing policies (Howard 2002).”

For example, in the Netflix example introduced earlier, an intervention that has a lower one-step conversion probability will never be picked under a myopic policy even though if it leads consumer to a state from where the conversion probability is higher than at the current state. We will demonstrate this in our numerics (Section 3.6). Some other related work includes Bult and Wansbeek 1995; Rossi et al. 1996 Ansari and Mela 2003, and Feit and Berman 2019.

**Estimation-based non-myopic approaches.** Relatively little existing work falls in the second set, i.e., estimation-based non-myopic approaches. Examples include Bitran and Mondschein

1996; Gönül and Shi 1998, Simester et al. 2006, Montoya et al. 2010; Schweidel et al. 2011; Ariely and Bitran 2013, Ma et al. 2016; Zhang et al. 2016, Zhang et al. 2017, and Liberali and Ferecatu 2019. These approaches first propose a model that predicts long-run reward (eventual conversion probability for example) as a function of the intervention and possibly the consumer state / features and then, estimate the proposed model to make intervention decisions. A key limitation of such approaches is that they might suffer from issues such as concept shift and covariate shift (Simester et al. 2020). Furthermore, arbitrarily setting aside an initial period of time to estimate the underlying parameters can introduce suboptimality. We further discuss these limitations in Sections 3.6 and 3.7. (We place Ariely and Bitran 2013 in this category since they assume customer behavior to be known conditional on the customer segment.)

**Learning-based myopic approaches.** In the last two decades, we have seen a noticeable progress here. As with estimation-based myopic approaches, these approaches propose a model that predicts immediate reward as a function of the intervention, and possibly the consumer state / features. The difference is that instead of estimating the model, they learn it. In the language of reinforcement learning (Sutton and Barto 2018), these approaches are similar to a *multi-armed bandit*, where the different arms represent the various interventions available to the firm. Works in this category include Gooley and Lattin 2000; Bertsimas and Mersereau 2007; Hauser et al. 2009; Urban et al. 2014; Hauser et al. 2014, Schwartz et al. 2017, and Cao et al. 2019. Although such learning-based approaches usually perform better than the estimation-based versions, both the approaches share the same key limitation: the decision is myopic (since it maximizes single-period reward) and hence, can result in suboptimal reward. We will highlight this in our numerical study that uses a real-world dataset (Section 3.6).

**Learning-based non-myopic approaches.** The focus of our work is learning-based non-myopic approaches. As in estimation-based non-myopic approaches, these approaches propose a model that predicts long-run reward as a function of the intervention and possibly the consumer state / features. The difference is that instead of estimating the model, they learn it. Though such

approaches have been around for almost two decades in the marketing literature (Abe et al. 2002; Pednault et al. 2002), there has been little progress in this space. Some other examples include Tkachenko 2015; Theocharous et al. 2015; Yao and Lu 2019, and AboElHamd et al. 2020. The existing works in this space have leveraged off-the-shelf reinforcement learning algorithms and there seems to be little (if any) rigorous methodological advancements that are tailored to the context of marketing, the focus of our work. One exception is the recent work of Moazeni et al. 2019, which focuses on designing marketing campaigns for market entry using a sequential learning approach. A key difference between Moazeni et al. 2019 and our work is that Moazeni et al. 2019 focus on aggregate-level dynamics whereas we focus on consumer-level dynamics.

We conclude our literature review with the following quote from Bertsimas and Mersereau 2007, that, although a decade old, is still relevant today:

“ . . . it remains a challenge to develop rigorous adaptive learning technologies for *more complicated and realistic models of customer behavior*, for example, allowing dependence among messages, dependence among customer segments, and customer heterogeneity. In short, we believe that there is much room for interesting future research in this area. . . . it remains a challenge to interface such techniques tractably with dynamic optimization formulations. We believe that approximations, either of the customer behavior model or of optimality or both, are necessary. . . . we also believe that *functional and distributional approximations of the Bayes updates warrant investigation*. Possible alternatives to the Bayes solution include non-Bayesian approaches or approaches that seek a *weaker form of optimality (e.g., asymptotic optimality)*.”

### 3.1.2 Our Approach and Contributions

Our approach is motivated by the above quote of Bertsimas and Mersereau 2007. We consider a fairly general model of consumer behavior and develop an asymptotically optimal decision-making algorithm that approximates the Bayes update. In terms of the model, at each point of time, we posit that the consumer is in some underlying state, which summarizes her interaction



history with the firm (and possibly includes segment information such as age, sex, location). The firm observes the state of the consumer and decides on which intervention to perform (if any). As a result, the consumer transitions to another state as a stochastic function of her current state and the firm’s intervention, i.e., we allow for interventions to have state-specific effects. The firm then performs another intervention and the process repeats until the consumer makes a decision to purchase or not. Due to the sequential nature of our model, it captures dependence among various interventions. Furthermore, our model’s sequential nature allows it to accommodate both the myopic and the long-run value of the interventions, which is in contrast with the existing approaches that only model myopic rewards. Our model is at a consumer level and hence, allows for personalized interventions (as a function of the consumer state). We emphasize that our model is general enough to capture a wide array of consumer behaviors.

The main contribution of our work is to propose a decision-making algorithm, which guides the firm in terms of what intervention to perform given the state of a consumer. Apriori, the firm does not know the state-specific effects of the interventions and needs to learn / estimate them in order to make optimal intervention decisions. To help the firm make decisions in such an unknown environment, we propose the *model-free<sup>2</sup> approximate Bayesian learning* (MFABL) algorithm. The high-level idea of MFABL is simple. It maintains a belief on the state-specific value (myopic plus long-run) of each intervention. As it interacts with the consumers and gathers data, it updates such beliefs in an approximate manner. Given the sequential nature of our model, maintaining the exact model-free belief is challenging and hence, MFABL approximates the Bayes update. MFABL’s belief structure, decision-making policy, and the update rule inherit the simplicity of Thompson sampling for a multi-armed bandit. In particular, MFABL assigns a Beta belief to the value of each state-specific intervention. When a consumer is in a certain state, MFABL uses the belief to sample the value of all interventions available at that state and performs the intervention

---

<sup>2</sup>We use the terminology “model-free” to be consistent with the existing literature (Sutton and Barto 2018) and note that “model-free” does not mean that there does not exist a true underlying model. Instead, it means that there exists a true underlying model but our algorithm does not learn / estimate the model. Approaches that estimate / learn the model are called “model-based”. We revisit this discussion in Section 3.3 (Footnote 4), where we formally define our model-free algorithm.

with the highest sample value. If doing so transitions the consumer to a state from where she is “likely” / “unlikely” to convert, MFABL positively / negatively reinforces the corresponding intervention using a Beta-Bernoulli style update. Despite being an approximation to the Bayes update, we prove that MFABL is asymptotically optimal.

We supplement our theoretical developments with numerical experiments using a real-world large-scale dataset corresponding to email interventions for a software product of a Fortune 500 firm, targeted at millions of consumers. We demonstrate the dominance of our algorithm over the traditional algorithms that either optimize for the myopic reward or are estimation-based. Furthermore, in contrast to the estimation-based approaches, our algorithm is able to adapt automatically to the underlying changes in consumer behavior (concept shift) and maintains a high level of uncertainty on the value of less explored consumer segments (covariate shift).

**Outline.** The rest of this chapter is organized as follows. In Section 3.2, we define our model for consumer behavior, i.e., the behavior of consumers as a function of the firm’s personalized sequential interventions. We call this model the conversion funnel and use it to define the underlying optimization problem the firm wishes to solve (conversion funnel optimization). In Section 3.3, we present the algorithm (model-free approximate Bayesian learning) that we propose in order to tackle the conversion funnel optimization problem. We discuss the properties of our proposed algorithm in Section 3.4, followed by some model extensions in Section 3.5. In Section 3.6, we apply our algorithm to a real-world email marketing dataset composed of millions of consumers corresponding to a Fortune 500 firm and benchmark it with some existing algorithms. In Section 3.7, we present further numerics showcasing the performance of our approach under practical issues such as concept shift and covariate shift. We conclude in Section 3.8 with some directions for ongoing and future research.

## 3.2 Consumer Model and Conversion Funnel Optimization

In this section, we formally define the underlying problem. To do so, we first discuss our model for consumer behavior (Section 3.2.1) and then, the conversion funnel optimization problem (Section 3.2.2). Finally, in Section 3.2.3, we map our model to a real-life example to provide intuition.

### 3.2.1 Model for Consumer Behavior: The Conversion Funnel

We now define a model that explains how a consumer behaves as a function of the firm's interventions. For ease of exposition, we start with a single firm promoting a single product and extend the model to multiple products in Section 3.5. Furthermore, as discussed in Section 3.1, we focus on a limited consumer features regime where we do not assume the firm has access to the features of the consumers (age, sex, location, for example). Nonetheless, we discuss an extension to a setting in which the firm has access to such features in Section 3.5. We note that we leverage the model from Chapter 2.

Motivated by the existing marketing literature on conversion funnel (Strong 1925; Howard and Sheth 1969; Barry 1987; Bettman et al. 1998; Court 2009; Elzinga et al. 2009; Kotler and Armstrong 2010; Mulpuru 2011; Jansen and Schuster 2011; Bruce et al. 2012), we model the consumer behavior using a state-based model, i.e., at each point of time, the consumer is in some state (possibly a function of her history). The firm observes<sup>3</sup> the state of the consumer and performs an intervention. As a result, the consumer transitions to some other state as a stochastic function of the firm's interventions and her current state. The process ends when the consumer converts or quits. If the consumer converts, the firm earns a corresponding reward. To showcase the flexibility of our model, we keep most of its elements in this section general and connect it to a specific application via an example in Section 3.2.3. We emphasize that our model is general enough to capture an arbitrary consumer behavior. Our model for consumer behavior can be casted as a

---

<sup>3</sup>In the traditional conversion funnel models, the state is usually assumed to be hidden. However, given the granularity of consumer-level data available these days, we assume the state to be observable. We revisit this discussion when we discuss our real-world dataset in Section 3.6.

Markov decision process (MDP) (Bertsekas 1995; Puterman 2014; Sutton and Barto 2018) and we define the five underlying components of the MDP next: (1) state space  $\mathbb{S}$ , (2) action space  $\mathbb{A}$ , (3) transition probabilities  $\mathcal{P}$ , (4) initial state probabilities  $\boldsymbol{\lambda}$ , and (5) reward  $r$ . We refer to this MDP as the *conversion funnel* and denote it by  $\mathcal{M} \equiv (\mathbb{S}, \mathbb{A}, \mathcal{P}, \boldsymbol{\lambda}, r)$ .

**State space.** We define  $\mathbb{S} := \{1, \dots, S\}$  as the set of states an “active” consumer can be in, i.e., a consumer who has not converted or quit. In addition, there are two absorbing states  $\{q, c\}$  with  $\mathbb{S}^+ := \mathbb{S} \cup \{q, c\}$ . State  $c$  refers to conversion (consumer buys the product) and  $q$  refers to the quit state (consumer leaves the system). At each point of time, a consumer is in one of the states in the set  $\mathbb{S}^+$  and the firm observes this information. We allow for any arbitrary definition of a state space and do not yet give it a physical meaning to highlight the flexibility of our framework.

**Action space.** The set of interventions available to the firm is defined as  $\mathbb{A} := \{0, 1, \dots, A\}$  with 0 denoting no intervention. All actions are assumed to be zero-cost (extension discussed in Section 3.5). Furthermore, we allow the action space to depend on the state, i.e.,  $\mathbb{A}_s$  for all  $s \in \mathbb{S}$ , but to keep the notation simple, we will use  $\mathbb{A}$  and note that all results hold for state-specific action space too. We allow for any arbitrary definition of an action space and do not yet give it a physical meaning to highlight the flexibility of our framework. We assume that the firm takes actions at arbitrary but discrete time points  $[t_k]_{k=1}^{\infty}$ . Note that we allow the gap between various time points to be unequal, i.e.,  $\delta_k := t_{k+1} - t_k$  can be such that  $\delta_k \neq \delta_{k'}$  for any  $k \neq k'$ . For ease of notation, we will use  $t_k = k$  for all  $k$  but we note that none of the results depend on this simplification.

**Transition probabilities.** If a consumer is in state  $s \in \mathbb{S}$  and the firm takes action  $a \in \mathbb{A}$ , the consumer transitions to state  $s' \in \mathbb{S}^+$  with probability  $p_{sas'} \in [0, 1]$ . Note that  $s'$  can equal  $s$  (self-loop). We allow for an arbitrary transition structure (as long as Assumption 3.1 stated below holds). Trivially,  $\sum_{s' \in \mathbb{S}^+} p_{sas'} = 1 \forall (s, a) \in \mathbb{S} \times \mathbb{A}$  and  $p_{cac} = p_{qaq} = 1 \forall a \in \mathbb{A}$  (absorbing states). Furthermore,  $p_{sac}$  denotes the *one-step conversion probability* corresponding to  $(s, a) \in \mathbb{S} \times \mathbb{A}$ . We denote by  $\mathcal{P}$  the collection of all transition probabilities. The only restriction we impose on the

transition behavior of a consumer is that she *eventually* absorbs, i.e., either converts or quits.

**Assumption 3.1 (Absorption).** *Under any policy, every consumer eventually converts or quits.*

A sufficient condition for Assumption 3.1 is that the sum of one-step conversion and one-step quit probabilities corresponding to each state-action pair is positive. We will discuss this condition in the context of our numerics in Section 6.

**Initial state probabilities.** We denote by  $\lambda_s$  the initial state probability corresponding to state  $s \in \mathbb{S}$ , i.e., the probability a consumer starts her journey in state  $s$ . Trivially,  $\sum_{s \in \mathbb{S}} \lambda_s = 1$ . We define  $\boldsymbol{\lambda} := [\lambda_s]_{s \in \mathbb{S}}$  and we allow the consumers to start in different initial states. Given  $\boldsymbol{\lambda}$ , consider the set of initial states  $\mathbb{S}_\lambda := \{s \in \mathbb{S} : \lambda_s > 0\}$ . We assume wlog that each state in  $\mathbb{S}$  is reachable via *some* state in  $\mathbb{S}_\lambda$  under *some* policy. This assumption is wlog since if there exists a state  $s \in \mathbb{S}$  that violates this assumption, then we can discard that state and re-define  $\mathbb{S} \leftarrow \mathbb{S} \setminus \{s\}$  (as no consumer will ever visit state  $s$ ). We will refer to this assumption as “connectedness”. (Since this assumption is wlog, we do not classify it as a formal assumption.)

**Reward structure.** The firm earns an immediate reward of  $r \in [0, \infty)$  when a consumer converts, i.e., transitions from a state in  $\mathbb{S}$  to the conversion state  $c$ . All other transitions result in an immediate reward of 0. Note that since  $r$  is a constant, we can assume it to equal 1 without loss of generality and we will do so. We emphasize that such a *terminal nature* of the reward is specific to the context of marketing and hence, our conversion funnel model belongs to a special class of MDPs. We will exploit this application-specific structure of reward when developing our algorithm in Section 3.3.

Before stating the conversion funnel optimization problem, we define some preliminaries of our MDP model, which will be useful later on. All these preliminaries are part of a standard MDP setup (Bertsekas 1995; Puterman 2014; Sutton and Barto 2018).

**Policy.** A policy  $\pi := [\pi_{sa}]_{(s,a) \in \mathbb{S} \times \mathbb{A}}$  is a mapping from states to action probabilities, i.e.,  $\pi_{sa}$  denotes the probability firm takes action  $a \in \mathbb{A}$  when a consumer is at state  $s \in \mathbb{S}$ .

**Value function of a policy.** Given that the firm employs policy  $\pi$ , state-value function  $V^\pi(s)$  denotes the expected reward the firm reaps from a consumer who is in state  $s \in \mathbb{S}$ . With  $r = 1$  wlog,  $V^\pi(s)$  equals the *eventual conversion probability* from state  $s$  under policy  $\pi$ . Using the Bellman equation (Bellman 1954) and the terminal reward structure of the conversion funnel, we get

$$V^\pi(s) = \sum_{a \in \mathbb{A}} \pi_{sa} \left\{ \sum_{s' \in \mathbb{S}} p_{sas'} V^\pi(s') + p_{sac} \right\} \quad \forall s \in \mathbb{S}.$$

Given  $\pi$ , define the action-value function

$$Q^\pi(s, a) := \underbrace{\sum_{s' \in \mathbb{S}} p_{sas'} V^\pi(s')}_{\text{long-run value}} + \underbrace{p_{sac}}_{\text{myopic value}} \quad \forall (s, a) \in \mathbb{S} \times \mathbb{A}, \quad (3.1)$$

which denotes the eventual conversion probability of a consumer who is in state  $s \in \mathbb{S}$  and firm takes action  $a \in \mathbb{A}$  at state  $s$  but follows policy  $\pi$  from then on. (If state  $s$  is visited in the future, then policy  $\pi$  is used.) Note that we have decoupled the eventual conversion probability into two components: (a) long-run value (probability of converting in more than one step) and (b) myopic value (one-step conversion probability).

**Optimal policy and optimal value function.** A policy  $\pi$  is optimal if and only if

$$V^\pi(s) \geq V^{\pi'}(s) \quad \forall s \in \mathbb{S}, \quad \forall \pi'.$$

We use the notation  $\pi^*$  to denote an optimal policy and define the optimal state-value function  $V^*(s) := V^{\pi^*}(s) \quad \forall s \in \mathbb{S}$  and the optimal action-value function  $Q^*(s, a) := Q^{\pi^*}(s, a) \quad \forall (s, a) \in \mathbb{S} \times \mathbb{A}$ . Trivially,

$$V^*(s) = \max_{a \in \mathbb{A}} Q^*(s, a) \quad \forall s \in \mathbb{S}.$$

We will use the notation  $\mathbf{Q}^* := [Q^*(s, a)]_{(s,a) \in \mathbb{S} \times \mathbb{A}}$ .

### 3.2.2 The Conversion Funnel Optimization Problem

We now state the optimization problem the firm wishes to solve, i.e., the *conversion funnel optimization*. Suppose there are  $N$  consumers, possibly interacting with the firm in parallel. Denote by  $y_n \in \{0, 1\}$  whether consumer  $n \in \{1, \dots, N\}$  converts or not. The objective of the firm is to maximize the average expected reward over all consumers, i.e.,  $\frac{r}{N} \sum_{n=1}^N \mathbb{E}[y_n]$ , which is equivalent to maximizing the average conversion probability

$$\frac{1}{N} \sum_{n=1}^N \mathbb{E}[y_n]$$

since  $r$  is a constant. (Hence, assuming  $r = 1$  is wlog.) Had the firm known the true conversion funnel MDP  $\mathcal{M}$ , then it could have computed an optimal policy using standard MDP solution techniques such as value / policy iteration (Bertsekas 1995; Puterman 2014; Sutton and Barto 2018) and hence, maximized the probability of converting each consumer. However, the main challenge in the conversion funnel optimization is that the firm does not know the true model  $\mathcal{M}$ . In particular, we assume that the firm knows  $\mathbb{S}$ ,  $\mathbb{A}$ , and  $r$  but does not know the consumers' transition behavior  $\mathcal{P}$  and  $\lambda$ .

In order to make intervention decisions, the firm can use information it collects by interacting with the consumers. Consider an arbitrary time  $t$  and an arbitrary consumer  $n$ . Denote by  $s_{nt}$  the state of consumer  $n$  at time  $t$  and by  $a_{nt}$  the intervention directed by the firm at consumer  $n$  at time  $t$ . As a result, consumer  $n$  transitions to state  $s_{n,t+1}$ . Define the interaction history till an arbitrary time  $t$  as

$$\mathcal{H}_t := \{(s_{n1}, a_{n1}, s_{n2}, a_{n2}, s_{n3}, \dots, s_{n,t-1}, a_{n,t-1}, s_{nt}) : n \in \{1, \dots, N\}\},$$

i.e., the set of all interactions the firm has had with all the consumers till time  $t$ . To make an intervention decision at time  $t$ , the firm is allowed to use the information in the set  $\mathcal{H}_t$ .

### 3.2.3 The Netflix Example

We now shed light on our general conversion funnel model by connecting it to a real-life setting. In particular, we revisit the Netflix example we introduced in Section 3.1 (recall Figure 3.1) and discuss how the components of our model map to such a setting. In that direction, we first elaborate on the sequence of events that we observed in real-life and then, show the mapping between real-life and our model. (Note that our numerics in Section 3.6 provide another such example.)

One of the authors (B from here on) provided his email address to Netflix on May 3, 2020 but did not subscribe to the paid membership. Netflix sent him a generic welcome email on May 3 (email #1 in Figure 3.1). 15 days later, on May 18, B received email #2 (see Figure 3.1), with a few “trending now” titles on Netflix. 20 days later, on June 8, B received email #3 (see Figure 3.1), displaying a diverse mix of shows across various genres (drama, comedies, documentaries, etc.).

To show the mapping between real-life and our model, it suffices to discuss the state space  $\mathbb{S}$  and the action space  $\mathbb{A}$ . In terms of the state space, the conversion state  $c$  refers to B subscribing to the paid membership of Netflix and the quit state  $q$  refers to B opting to unsubscribe from Netflix’s email list. On day 1 (May 3), B “arrives” with an initial state of 1, which corresponds to Netflix having no information on B except his email address. As B interacts with Netflix, the state of B evolves. For example, the state can capture information such as time since last email (to capture temporal effects) and the type of emails B interacted with. It can also include more granular information such as the titles B clicked on out of the ones displayed in emails (“Money Heist” in email #2 for example). In terms of the action space, the discretization of time at which Netflix sends emails can be at a weekly level. In addition to the zero action (do not send an email), the action space can include various types of emails, such as “trending now” (email #2), “diverse” (email #3), and “top 10 shows being watched on Netflix”. Another type of email can be to offer a one-month free trial. Intuitively, such an email will have a lower myopic value but a higher long-run value since it might not lead to an immediate conversion (subscription to *paid* membership) but push a consumer to a state from where she is more likely to convert.

Building further on this real-life example, after the first three emails, B opted in for a one-month



free trial, during which he received personalized emails specific to the shows he watched. After the trial expired, B did not become a paying member and received emails with a pricing theme. For example, the subject of one of the emails was “B, come back for just \$8.99”, illustrating an instance of state-specific email policy.

More broadly, we note that our conversion funnel model is not specific to email campaigns since one can model other applications using it too, one example being the e-commerce purchase funnel (a retailer selling clothes online and performing interventions such as changing the information displayed on its website as a function of the state of the consumer). For example, a consumer who has been reading product reviews might be persuaded by a message such as “1000 people like this product” whereas a consumer who has added the product to the cart might be persuaded by “only 1 unit left in the inventory”.

### 3.3 Model-Free Approximate Bayesian Learning

In this section, we propose a algorithm to tackle the conversion funnel optimization problem as defined in Section 3.2. To build intuition behind our algorithm, we first present the Thompson sampling algorithm for multi-armed bandit (Section 3.3.1) and use it to motivate the construction of our algorithm (Section 3.3.2).

#### 3.3.1 Motivation: Thompson Sampling for Multi-Armed Bandit

To discuss the Thompson sampling algorithm, we briefly define the multi-armed bandit model (Sutton and Barto 2018). The multi-armed bandit can be seen as a special case of our conversion funnel model. In particular, if we restrict the number of states in  $\mathbb{S}$  to 1 and only allow the firm to interact once with the consumer (as opposed to sequential interactions), our conversion funnel becomes a multi-armed bandit. The multiple actions available to the firm in the action space  $\mathbb{A}$  represent the “multiple arms” of the bandit. The consumer dynamics in a multi-armed model are as follows:

- Consumer 1 arrives and firm takes action  $a \in \mathbb{A}$ .

- Consumer converts with probability  $p_a$  and quits with probability  $1 - p_a$ .
- If consumer converts, firm earns a reward equal to  $r$  (equals 1 wlog).
- Then, consumer 2 arrives. And so on.

The objective of the firm is to maximize its expected reward over all consumers and the challenge is that the firm does not know the underlying parameters  $[p_a]_{a \in \mathbb{A}}$ . Thompson sampling (Thompson 1933) is a well-known algorithm to tackle the multi-armed bandit problem. The idea behind Thompson sampling (Algorithm 5) is simple: maintain a belief on the “value” of each action and when a consumer arrives, take action  $a \in \mathbb{A}$  with probability it is optimal. Once an action is taken and a corresponding outcome is observed (consumer converts or quits), update the belief corresponding to that action. For  $a \in \mathbb{A}$ , since the belief is over the conversion probability  $p_a \in [0, 1]$ , it seems natural to use a Beta distribution and exploit the Beta-Bernoulli conjugacy to update the belief. So, if the consumer converts, we increase the  $\alpha$  of the corresponding action by 1 (and hence, increase the expected value of our belief over the value of that action) and if the consumer quits, we increase the  $\beta$  of the corresponding action by 1 (and hence, decrease the expected value of our belief over the value of that action). Despite its simplicity, Thompson sampling for the multi-armed bandit problem is known to exhibit strong theoretical guarantees (Agrawal and Goyal 2012) and work well in marketing applications (Chapelle and Li 2011; Schwartz et al. 2017).

---

**Algorithm 5** Thompson Sampling for Multi-Armed Bandit with  $N$  Consumers

---

**Require:** Prior counts  $(\alpha_a, \beta_a) \forall a \in \mathbb{A}$

```

1: for  $n = 1$  to  $N$ 
2:    $q_a \sim \text{Beta}(\alpha_a, \beta_a) \forall a \in \mathbb{A}$                                      % generate samples
3:    $a^* = \arg \max_a q_a$                                                     % play action with the highest sample value
4:   if consumer  $n$  converts
5:      $\alpha_{a^*} \leftarrow \alpha_{a^*} + 1$                                    % update belief using Beta-Bernoulli conjugacy
6:   else
7:      $\beta_{a^*} \leftarrow \beta_{a^*} + 1$                                    % update belief using Beta-Bernoulli conjugacy
8:   end if
9: end for

```

---

### 3.3.2 Our Algorithm: Model-Free Approximate Bayesian Learning (MFABL)

Our algorithm, *model-free*<sup>4</sup> *approximate Bayesian learning* (MFABL), extends the simple Beta-Bernoulli structure of Thompson sampling to the general conversion funnel model and is presented as Algorithm 6. Due to the terminal reward structure in our conversion funnel, the  $Q$ -value of an action at a given state represents the eventual conversion probability (as discussed in Section 3.2.1). Hence, motivated by Thompson sampling, we give a  $\text{Beta}(\alpha_{sa}, \beta_{sa})$  belief to the “value” of taking action  $a \in \mathbb{A}$  at state  $s \in \mathbb{S}$ . We represent this belief by  $Q(s, a)$  for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ , i.e.,  $Q(s, a) \stackrel{d}{=} \text{Beta}(\alpha_{sa}, \beta_{sa})$  where  $\stackrel{d}{=}$  denotes “equal in distribution”.

---

#### Algorithm 6 Model-Free Approximate Bayesian Learning with $N$ Consumers

---

**Require:** Prior counts  $(\alpha_{sa}, \beta_{sa}) \forall (s, a) \in \mathbb{S} \times \mathbb{A}$ ,  $\epsilon$

- 1:  $(\alpha_{ca}, \beta_{ca}) = (\infty, 1)$  and  $(\alpha_{qa}, \beta_{qa}) = (1, \infty)$  for all  $a \in \mathbb{A}$      % prior counts for states  $c$  and  $q$
- 2: **for**  $t = 1, 2, \dots$      % until there exists an “active” consumer (has not converted or quit)
- 3:     **for**  $n = 1$  to  $N$
- 4:         Observe state  $s = s_{nt}$  of consumer  $n$  at time  $t$
- 5:         **if**  $s \in \mathbb{S}$      % filter for “active” consumers
- 6:              $q_{sa} \sim \text{Beta}(\alpha_{sa}, \beta_{sa}) \forall a \in \mathbb{A}$      % generate samples
- 7:              $a^* = \arg \max_a q_{sa}$  with  $\epsilon$ -greedy     % highest sample value with  $\epsilon$ -greedy
- 8:             Consumer transitions to state  $s' = s_{n,t+1} \in \mathbb{S}^+$
- 9:              $f_{s'} \sim \text{Bernoulli} \left( \max_{a' \in \mathbb{A}} \frac{\alpha_{s'a'}}{\alpha_{s'a'} + \beta_{s'a'}} \right)$      % generate feedback
- 10:            **if**  $f_{s'} = 1$
- 11:                 $\alpha_{sa^*} \leftarrow \alpha_{sa^*} + 1$      % approximate posterior update mimicking Beta-Bernoulli
- 12:                **else**
- 13:                 $\beta_{sa^*} \leftarrow \beta_{sa^*} + 1$      % approximate posterior update mimicking Beta-Bernoulli
- 14:                **end if**
- 15:            **end if**
- 16:         **end for**
- 17:     **end for**
- 18: **return**  $\mathbf{Q} := [Q(s, a)]_{(s,a) \in \mathbb{S} \times \mathbb{A}}$  where  $Q(s, a) \stackrel{d}{=} \text{Beta}(\alpha_{sa}, \beta_{sa})$

---

In terms of picking an action, MFABL mimics Thompson sampling (lines 6 and 7 in Algorithm 6). When a consumer is in state  $s \in \mathbb{S}$ , MFABL samples  $q_{sa} \sim \text{Beta}(\alpha_{sa}, \beta_{sa})$  for  $a \in \mathbb{A}$  and

<sup>4</sup>As noted in Section 3.1 (Footnote 2), “model-free” does not mean that there does not exist a true underlying model. Instead, it means that there exists a true underlying model  $\mathcal{M}$  (the conversion funnel MDP) but our algorithm does not learn / estimate the model  $\mathcal{M}$ . In particular, our algorithm (MFABL) never attempts to learn / estimate the transition probabilities  $\mathcal{P}$  corresponding to the MDP  $\mathcal{M}$  but only attempts at learning  $\mathbf{Q}^*$ . Approaches that estimate / learn the transition probabilities  $\mathcal{P}$  are called “model-based”.

plays the action with the highest sample value (with  $\epsilon$ -greedy), i.e.,

$$a^* = \begin{cases} \arg \max_{a \in \mathbb{A}} q_{sa} & \text{w.p. } 1 - \epsilon \\ \text{UniformAtRandom}(\mathbb{A}) & \text{w.p. } \epsilon. \end{cases}$$

In words, MFABL plays the action with the highest sample value with probability  $1 - \epsilon$  and plays an action sampled uniformly at random from  $\mathbb{A}$  with probability  $\epsilon$ . (Note that  $\epsilon$ -greedy is used to ensure convergence of MFABL to  $\mathbf{Q}^*$ , as it will become clear in Section 3.4. In theory, the parameter  $\epsilon$  can be arbitrarily small but strictly positive.)

However, unlike in a bandit, updating the belief in a conversion funnel is non-trivial since a reward is observed after a consumer visits a sequence of states. For example, suppose the consumer path is as follows:

$$(s, a) \rightarrow (s', a') \rightarrow c.$$

That is, consumer starts in state  $s \in \mathbb{S}$ . The firm takes action  $a \in \mathbb{A}$ . This makes the consumer transition to state  $s' \in \mathbb{S}$ , where the firm takes action  $a' \in \mathbb{A}$ , which leads to conversion. Given this path, how does one update the parameters  $(\alpha_{sa}, \beta_{sa})$  and  $(\alpha_{s'a'}, \beta_{s'a'})$ ? Does one increase both  $\alpha_{sa}$  and  $\alpha_{s'a'}$  by 1? As another example, suppose the consumer path is as follows:

$$(s, a) \rightarrow (s', a') \rightarrow q.$$

Given the consumer quit after visiting  $(s, a)$  and  $(s', a')$ , does one increase both  $\beta_{sa}$  and  $\beta_{s'a'}$  by 1? But what if taking action  $a$  at state  $s$  was optimal and it was the action  $a'$  at state  $s'$  that was the “culprit”? Should  $(s, a)$  be still “penalized”? The challenge here is to identify the “contributions” of various state-specific actions to the final outcome (consumer converting or quitting).

To overcome this challenge, MFABL adopts the following strategy (lines 9 to 13 in Algorithm 6). Suppose taking action  $a \in \mathbb{A}$  at state  $s \in \mathbb{S}$  transitions the consumer to state  $s' \in \mathbb{S}^+$ . Intu-

itively speaking, if the eventual conversion probability from state  $s'$  is “high”, the state-action pair  $(s, a)$  should be “positively reinforced”. For example, in the extreme case when  $s' = c$ , we should increase  $\alpha_{sa}$  by 1. On the other hand, if the eventual conversion probability from state  $s'$  is “low”, the state-action pair  $(s, a)$  should be “negatively reinforced”. For example, in the extreme case when  $s' = q$ , we should increase  $\beta_{sa}$  by 1. However, the eventual conversion probability from state  $s'$  is not necessarily known in advance and MFABL maintains a belief  $Q(s', a') \stackrel{d}{=} \text{Beta}(\alpha_{s'a'}, \beta_{s'a'})$  for each  $a' \in \mathbb{A}$ . To operationalize such positive / negative reinforcement, MFABL generates a binary “feedback”  $f_{s'}$  from state  $s'$  as follows:

$$f_{s'} \sim \text{Bernoulli} \left( \max_{a' \in \mathbb{A}} \frac{\alpha_{s'a'}}{\alpha_{s'a'} + \beta_{s'a'}} \right). \quad (3.2)$$

If the feedback equals 1, MFABL positively reinforces  $(s, a)$  by increasing  $\alpha_{sa}$  by 1. On the other hand, if the feedback equals 0, MFABL negatively reinforces  $(s, a)$  by increasing  $\beta_{sa}$  by 1. Note that the term  $\frac{\alpha_{s'a'}}{\alpha_{s'a'} + \beta_{s'a'}}$  in (3.2) equals the expected value of the belief  $Q(s', a')$ . Hence, the feedback in MFABL is generated using the action with the highest expected value at state  $s'$ . We note that MFABL recovers Thompson sampling when the conversion funnel is simplified to a bandit.

Despite aesthetical similarities, it is important to note that there are fundamental differences between Thompson sampling for bandit (Algorithm 5) and MFABL for conversion funnel (Algorithm 6). In particular, the Beta belief maintained in Thompson sampling is exact whereas the Beta belief maintained in MFABL is approximate. This is because the parameter update in Thompson sampling obeys the Bayes’ theorem (via Beta-Bernoulli conjugacy) whereas the update in MFABL does not necessarily do so. In fact, it is not clear if one can maintain an exact *model-free* belief in the conversion funnel model in a tractable manner. Accordingly, the belief we maintain in MFABL can be seen as an approximation to the true posterior. A natural question to ask is whether our approximate belief converges to  $\mathbf{Q}^*$ , which we will explore in Section 3.4 (and prove that it does).

We note that there do exist ideas such as Dearden et al. 1998, Osband et al. 2019, and Ryzhov et al. 2019 that maintain a *model-free* belief over the value of each state-action pair in an MDP.

However, none of these ideas exploit the terminal reward structure of the conversion funnel and hence, do not maintain a Beta belief. Instead, the belief structures they maintain are rather complicated and their update policy is significantly different than the simple “Beta-Bernoulli” type update in MFABL. Furthermore, their action selection can be quite different than MFABL’s. For example, Ryzhov et al. 2019 use a knowledge-gradient technique for action selection, which in fact, requires one to estimate the transition probabilities  $\mathcal{P}$ , making the overall algorithm model-based. It is also worth mentioning the recent work of Jin et al. 2018 and Dong et al. 2019, which attempt at solving similar problems in a model-free manner, but by using upper confidence bound (UCB) algorithms. We also briefly mention an alternative line of work (Strens 2000; Osband et al. 2013; Agrawal and Jia 2017) that attempts at solving similar problems but by maintaining a *model-based* belief over the underlying MDP. However, such model-based algorithms requires one to optimize an entire MDP in each “iteration”, which poses a higher computational burden than the model-free approaches, especially when the number of states  $S$  is large. Furthermore, since model-based approaches store belief over the transition probabilities  $\mathcal{P}$ , their storage requirement is  $O(S^2A)$ , compared with  $O(SA)$  of MFABL. In our numerical study (Section 3.6), we will benchmark MFABL with such state-of-the-art algorithms.

### 3.4 Properties of MFABL

We now discuss the theoretical properties (Section 3.4.1) and the practical appeal (Section 3.4.2) of MFABL. In this section, we hope to convince the reader that in addition to exhibiting desirable theoretical properties, MFABL is a practical algorithm, mainly due to the simplicity it inherits from Thompson sampling.

#### 3.4.1 Theoretical Properties

As we noted earlier, the belief we maintain in MFABL is an approximation to the true posterior and a natural question to ask is whether our approximate belief converges to  $\mathbf{Q}^*$ . The following theorem provides a positive answer to this question. A formal proof can be found in Appendix C.1

and we provide the high-level intuition in the proof sketch.

**Theorem 3.1** (Asymptotic convergence). *Denote by  $\mathbf{Q}^N$  the output of Algorithm 6 with  $N$  consumers. Then,*

$$\mathbf{Q}^N \rightarrow \mathbf{Q}^* \text{ with probability 1 as } N \rightarrow \infty.$$

**Proof sketch.** The key idea here is to analyze how the belief in MFABL evolves *in expectation*. Denote by  $\mathbf{Q}_i := [Q_i(s, a)]_{(s,a) \in \mathbb{S} \times \mathbb{A}}$  the belief in MFABL at the start of iteration  $i \in \{1, 2, \dots\}$  where  $Q_i(s, a) \stackrel{d}{=} \text{Beta}(\alpha_i(s, a), \beta_i(s, a)) \forall (s, a) \in \mathbb{S} \times \mathbb{A}$ . (By an “iteration”, we refer to a  $(t, n)$  pair in Algorithm 6.) We use  $\bar{\mathbf{Q}}_i := [\bar{Q}_i(s, a)]_{(s,a) \in \mathbb{S} \times \mathbb{A}}$  to denote the expected value of  $\mathbf{Q}_i$ . We first show how the MFABL update in the  $\mathbf{Q}$ -space translates to an update in the  $\bar{\mathbf{Q}}$ -space. Second, we establish that the update process in the  $\bar{\mathbf{Q}}$ -space is an asynchronous stochastic approximation scheme to the Bellman optimality equations corresponding to  $\mathbf{Q}^*$ . Third, we leverage the stochastic approximation theory to prove that  $\bar{\mathbf{Q}}_i$  converges to  $\mathbf{Q}^*$ . Finally, we show that the variance of the Beta belief  $\mathbf{Q}_i$  goes to zero and hence,  $\mathbf{Q}_i$  converges to  $\mathbf{Q}^*$ . ■

Theorem 3.1 establishes that our belief on the “value” of each state-action pair converges to the optimal value. Combined with how MFABL picks an action (lines 6 and 7 of Algorithm 6), an immediate consequence is that, asymptotically, MFABL picks an optimal action with probability at least  $1 - \epsilon$ . This establishes the asymptotic optimality of MFABL with high probability. In other words, as the firm interacts more and more with the consumers, MFABL learns the optimal personalized interventions. Note that this result holds irrespective of the prior counts supplied to Algorithm 6 as an input.

### 3.4.2 Practical Considerations

Despite the generality of our conversion funnel model, we wish to propose a solution that is both supported by theory and is practically appealing. Having stated the theoretical properties of

MFABL, we now discuss some properties of MFABL that highlight its desirability from a practitioner’s point-of-view. In this direction, we first touch upon the scalability of MFABL. We then mention the ability of MFABL to enable a practitioner to encode prior information. Finally, given the real-life importance of issues such as concept shift and covariate shift (Simester et al. 2020), we discuss how MFABL handles them.

## Scalability

Here, we discuss the tractability of MFABL (storage and computational requirements). Of course, tractability affects the scalability of a solution, which is critical when interacting with a large number of consumers on a daily basis. It is easy to see that MFABL only stores approximately  $2SA$  parameters<sup>5</sup>  $\{\alpha_{sa}, \beta_{sa}\}_{(s,a) \in \mathbb{S} \times \mathbb{A}}$ . In terms of computational cost, taking an intervention decision for a consumer at a given time requires MFABL to generate a sample from  $A + 1$  Beta distributions and pick the maximum among them (lines 6 and 7 of Algorithm 6), which can be done in real-time. We contrast such tractability with model-based approaches (Strens 2000; Osband et al. 2013; Agrawal and Jia 2017), which store  $\mathcal{O}(S^2A)$  parameters and require solving an MDP in each “iteration”, which can be computationally demanding, especially with large state and action spaces.

## Prior Information

Since a firm might have some prior information regarding the consumer behavior, it seems desirable to allow the firm to use such information when sending personalized interventions to consumers. For instance, using third-party data or its experience with an earlier version of the product, the firm might know that the consumers who are in a certain state are highly likely to convert if they are shown a specific intervention. MFABL allows the firm to encode such state-action specific prior knowledge via the prior counts on the “value” of each state-action pair. We will explore the value of such prior information in Section 3.7. It is worth mentioning that even if

---

<sup>5</sup>To be exact, MFABL stores  $2(S + 2)(A + 1)$  parameters (accounting for the two absorption states and the action of doing nothing).



the firm provides an incorrect prior, the convergence of MFABL to optimality (Theorem 3.1) still holds.

## Concept Shift

We now shift our attention to one of the key data challenges a practitioner faces, which has been highlighted in the recent work of Simester et al. 2020: *concept shift*. In the words of Simester et al. 2020,

“If the underlying response function is different in the training data than in the implementation setting in ways that are not captured by the predictor variables, this will generally result in suboptimal targeting policies. The risk of this is high if the targeting policy is implemented several months after the training data is collected. In the intervening period, changes in the environment through shifts in macroeconomic conditions, competitors’ actions, the firm’s other marketing activities, or just seasonality can all contribute to changes in how customers respond to the firm’s actions.”

Mapping to our conversion funnel model, concept shift refers to the possibility of the consumer behavior (transition probabilities) changing over time. Consider a two-phase setting such that the transition probabilities equal  $\mathcal{P}_1$  in phase 1 and change to  $\mathcal{P}_2$  in phase 2, with the underlying state space  $\mathbb{S}$  and action space  $\mathbb{A}$  held constant. The initial state probabilities can change too. The firm does not know apriori either  $\mathcal{P}_1$  or  $\mathcal{P}_2$ . Nor does the firm know when the phase will shift. As discussed in Simester et al. 2020, an algorithm that *estimates* the model using data from phase 1 but uses the estimated model to make decisions in phase 2 will generally result in suboptimal policies. Fortunately, given its *learning* nature, MFABL does not suffer from such a limitation. In particular, under such a two-phase setting, MFABL will initially make progress towards learning the  $\mathbf{Q}^*$  corresponding to  $\mathcal{P}_1$ , say  $\mathbf{Q}_1^*$ . As soon as there is a phase shift, the exploratory nature of MFABL will force it to learn the  $\mathbf{Q}^*$  corresponding to  $\mathcal{P}_2$ , say  $\mathbf{Q}_2^*$ . One way to see this is that when phase 2 begins, the status-quo of MFABL (the belief at the end of phase 1) can be seen as

the prior before phase 2 begins. Since Theorem 3.1 holds for all values of prior counts, it follows that MFABL will automatically detect the phase shift and start learning  $\mathbf{Q}_2^*$ . Of course, the speed at which MFABL moves towards  $\mathbf{Q}_2^*$  will depend on the strength of the belief at the end of phase 1 and the difference between the two phases. For instance, if the two phases are very different and phase 1 induces a strong belief, then MFABL can take longer to learn phase 2 dynamics as compared to a setting in which MFABL starts “fresh” (weak prior) in phase 2. Note that the same argument holds for a multi-phase setting and we will explore concept shift in more depth in Section 3.7.

### Covariate Shift

As discussed in Simester et al. 2020, another critical data challenge in real-life is that of *covariate shift*. In the words of Simester et al. 2020,

“Most methods assume that the distribution of the targeting variables in the training data will be representative of the implementation data, but this is not always the case. For example, if the targeting method is implemented in different geographic regions than the regions in which the training data was gathered, the characteristics of the targeting pool may differ from the implementation pool.”

Mapping to our conversion funnel model, covariate shift can refer to two possibilities: (1) encountering a state  $s \in \mathbb{S}$  that the firm has not encountered before and (2) encountering a consumer with features that the firm has not encountered before. We will defer the discussion of possibility 2 to Section 3.5.2 (Remark 3.2 in particular) since we introduce consumer features then. In terms of the first possibility, under MFABL, if a state-action pair  $(s, a) \in \mathbb{S} \times \mathbb{A}$  is unexplored, then the variance in the belief  $Q(s, a)$  will be as it was at initialization. Naturally, such variance affects the action selection in MFABL (since MFABL picks the action via sampling). Accordingly, MFABL accommodates covariate shift by implicitly accounting for the high variance in the parts of the state space that are unexplored. We will illustrate this in Section 3.7.

### 3.5 Model Extensions

In this section, we show how the conversion funnel model presented in Section 3.2 can be extended to capture a broader class of real-life settings. We discuss three extensions. In extension 1 (Section 3.5.1), we show how the conversion funnel for a single product can be generalized to accommodate multiple products. In extension 2 (Section 3.5.2), we allow for the actions to incur a non-negative cost (as opposed to zero-cost actions). In extension 3 (Section 3.5.3), we show how the conversion funnel can be used to model consumer behavior if the firm has access to exogenous consumer features such as age, sex, and location. For each extension, we also discuss the corresponding changes needed in the MFABL algorithm to ensure the validity of the properties discussed in Section 3.4. Note that we always start with the “base” conversion funnel (the one from Section 3.2) and make one change at a time (either multiple products or actions with costs or consumer features). However, we do so only to keep the presentation simple and note that one can come up with a “hybrid” model and a corresponding MFABL algorithm with the same properties as in Section 3.4.

#### 3.5.1 Multiple Products

We now show how the conversion funnel for a single product can be generalized to accommodate multiple products. Suppose there are  $M \in \mathbb{N}$  underlying products. The partial state space  $\mathbb{S}$  remains the same as in the base conversion funnel but now, there are  $M + 1$  absorbing states  $\{c_1, \dots, c_M, q\}$  with  $\mathbb{S}^+ := \mathbb{S} \cup \{c_1, \dots, c_M, q\}$ . State  $c_m$  refers to conversion to product  $m \in \{1, \dots, M\}$ . The action space and the timeline remains the same as before and so do the transition probabilities along with the absorption assumption (consumer eventually converts to some product or quits). Initial state probabilities remain unchanged. The firm earns an immediate reward of  $r_m \in [0, \infty)$  when a consumer buys product  $m \in \{1, \dots, M\}$ , i.e., transition from a state in  $\mathbb{S}$  to the state  $c_m$ . All other transitions result in an immediate reward of 0. As before, we normalize the

rewards wlog so that the maximum reward equals 1, i.e.,

$$r_m \leftarrow \frac{r_m}{\max_{m' \in \{1, \dots, M\}} r_{m'}} \quad \forall m \in \{1, \dots, M\}.$$

This is wlog since dividing by the constant  $\max_{m' \in \{1, \dots, M\}} r_{m'}$  does not change the objective of maximizing expected reward. As before, the reward exhibits a terminal nature. The corresponding preliminaries (value function, optimal policy, etc.) can be defined as before and we note that the  $Q$ -value of each state-action still lies in  $[0, 1]$  due to the normalization and the terminal nature of reward. Furthermore, the conversion funnel optimization problem can be defined as before with the objective of maximizing  $\frac{1}{N} \sum_{n=1}^N \mathbb{E}[y_n]$ , where for all  $n = 1, \dots, N$ ,

$$y_n = \begin{cases} 0 & \text{if consumer } n \text{ does not convert} \\ r_m & \text{if consumer } n \text{ converts to product } m \in \{1, \dots, M\}. \end{cases}$$

To be concise, the corresponding algorithm, which generalizes MFABL to a multiple products setting, is presented in Appendix C.2.1. We note that all the properties as discussed in Section 3.4 apply to this generalization of MFABL.

### 3.5.2 Actions with Costs

We now allow for the actions to incur a cost to the firm. Everything except the reward structure remains the same as in the base conversion funnel (Section 3.2). The terminal reward firm reaps when a consumer converts remains the same, i.e.,  $r \in [0, \infty)$ . However, we now introduce immediate costs. In particular, the firm incurs a (possibly random) cost  $c_{sa} \in (-\infty, 0]$  for taking action  $a \in \mathbb{A}$  when the consumer is in state  $s \in \mathbb{S}$ . (If the cost  $c_{sa}$  is random, we assume its variance to be finite as in Tsitsiklis 1994.)

Given that the firm employs policy  $\pi$ , state-value function  $V^\pi(s)$  no longer equals the eventual

conversion probability (due to immediate costs) from state  $s$  under policy  $\pi$ . It satisfies

$$V^\pi(s) = \sum_{a \in \mathbb{A}} \pi_{sa} \mathbb{E}[c_{sa}] + \sum_{a \in \mathbb{A}} \pi_{sa} \left\{ \sum_{s' \in \mathbb{S}} p_{sas'} V^\pi(s') + p_{sac} r \right\} \quad \forall s \in \mathbb{S}.$$

The action-value function corresponding to policy  $\pi$  is defined as

$$Q^\pi(s, a) := \mathbb{E}[c_{sa}] + \sum_{s' \in \mathbb{S}} p_{sas'} V^\pi(s') + p_{sac} r \quad \forall (s, a) \in \mathbb{S} \times \mathbb{A}.$$

Optimal policy and the corresponding value functions can be defined in a similar fashion to before. They key observation here is that the  $Q$ -value of a state-action pair no longer represents a probability and can be outside  $[0,1]$  even if we normalize the costs and reward (due to the non-terminal nature of reward). Hence, assigning a Beta belief to the value of a state-action pair seems inappropriate. In this setting, we modify the MFABL algorithm such that it maintains Gaussian beliefs on the values of various state-action pairs. The modification is presented in Appendix C.2.2. We note that all the properties as discussed in Section 3.4 apply to this modification of MFABL.

### 3.5.3 Consumer Features

We now show how the conversion funnel can be used to model consumer behavior if the firm has access to *exogenous* consumer features such as age, sex, and location. By “exogenous”, we mean that such features do not change as a function of the action the firm takes. We denote the features of a consumer by  $x \in \mathbb{X}$ , where  $\mathbb{X}$  denotes the features space. For example,  $x = (\text{young}, \text{female})$  denotes a consumer whose age category is “young” and sex is “female”. The underlying feature space might have  $\text{age} \in \{\text{young}, \text{middle}, \text{old}\}$  and  $\text{sex} \in \{\text{male}, \text{female}\}$  with  $\mathbb{X}$  containing all 6 possible combinations of the (age, sex) pair. To accommodate such consumer features, we can simply define a new state space  $\mathbb{S} \times \mathbb{X}$  where the new state corresponds to  $(s, x)$ . Doing so allows for the possibility of the consumer behavior to be a function of both  $s \in \mathbb{S}$  and  $x \in \mathbb{X}$ . In particular, the transition probabilities are denoted by  $p_{(s,x),a,(s',x)}$  for all  $(s, a, s') \in \mathbb{S} \times \mathbb{A} \times \mathbb{S}^+$  and  $x \in \mathbb{X}$ . Everything else (including the MFABL algorithm) remains the same as before with

the understanding that the state captures both the endogenous component  $s \in \mathbb{S}$  and the exogenous component  $x \in \mathbb{X}$ , i.e.,  $s \leftarrow (s, x)$  with  $\mathbb{S} \leftarrow \mathbb{S} \times \mathbb{X}$ . All properties as discussed in Section 3.4 apply trivially.

**Remark 3.1** (High-dimensional features). *The above definition of state space ( $\mathbb{S} \leftarrow \mathbb{S} \times \mathbb{X}$ ) can potentially result in the number of states to be extremely large, especially if the firm has “fine-grained” data on consumer features. However, as discussed in Section 3.1, when a consumer journey begins (for instance, consumer providing her email to Netflix), the firm usually has very limited data on the consumer. Accordingly, such settings can be modeled tractably. Nonetheless, in applications other than email campaign management, the firm might have access to lots of data on consumer features. Even though our model-free approach is quite scalable (as discussed in Section 3.4.2), we reckon a high-dimensional feature space can lead to tractability concerns. Accordingly, it is of interest to extend our ideas to such high-dimensional settings. We mention in passing that there exists work for such “contextual” settings but when the underlying model is myopic (Li et al. 2010; Agrawal and Goyal 2013; Tang et al. 2013). Extending such ideas to non-myopic models (such as our conversion funnel) in a model-free manner remains an open question. We refer the reader to the relatively recent work of Hallak et al. 2015 for a model that accounts for contextual information in a non-myopic manner. To the best of our knowledge, there does not exist any model-free approaches that solve the optimization problem in Hallak et al. 2015.*

**Remark 3.2** (Covariate shift). *As mentioned in Section 3.4.2, we discuss here covariate shift in the presence of consumer features, which refers to the firm encountering a consumer with features previously not encountered. Similar to our discussion in Section 3.4.2, if  $(s, x, a) \in \mathbb{S} \times \mathbb{X} \times \mathbb{A}$  in unexplored, then the variance in the corresponding belief will be as it was at initialization and MFABL will account for it in its action selection.*

### 3.6 Numerical Experiments

We now shift our focus to numerically evaluating the proposed algorithm (MFABL). To do so, we benchmark its performance on a large-scale real-world dataset against some existing algo-

rithms. In Section 3.6.1, we provide details on the real-world dataset we use. In Section 3.6.2, we define a possible state space and an action space of the conversion funnel specific to our data and discuss its estimation. Note that we estimate the conversion funnel (in particular, its transition probabilities) to enable us to simulate consumer behavior from a “ground truth” model. None of the decision-making algorithms (including MFABL) will have access to the “ground truth” but only observe data sampled from it. We briefly discuss the benchmark algorithms in Section 3.6.3. In Section 3.6.4, we compare the performance of the various algorithms using the estimated “ground truth” model to simulate consumer behavior, followed by a sensitivity analysis in Section 3.6.5. We note that our numerics are for illustrative purposes and in particular, to see how various decision-making algorithms differ in terms of the number of conversions (given an underlying conversion funnel model). We admit that the ultimate test would be to perform an experiment in real-life; however, obtaining such decision-making control is challenging, as noted by Bertsimas and Mersereau 2007:

“A second challenge in studying adaptive experimentation in marketing contexts is in testing the methodologies. Testing an adaptive learning model requires not just data but also decision-making control in a real-world system. While examples of successful field testing exist (e.g., Simester et al. 2006), it is understandable why marketers have been reluctant to cede this control to academic researchers.”

### 3.6.1 Dataset

Our real-world data is for a software product promoted and sold on the Internet by a Fortune 500 firm. Firm details and microscopic data statistics are not disclosed for anonymity reasons. The dataset consists of a couple million consumer paths, out of which a few ten thousand consumers made a purchase. Each path corresponds to a unique consumer and starts with the consumer creating an account on the firm’s website (*sign-up*). Once a consumer signs-up, the firm sends her a sequence of emails. The firm classifies these emails into four categories: (1) awareness email, (2)

promotional email, (3) ROI email, and (4) type 4 email<sup>6</sup>. For each type of email, the firm knows the time it was sent to a specific consumer and the time at which the consumer opened the email (if at all). In addition, for type 1 and 3 emails, the firm can track the time at which the consumer clicked on an embedded link in the email (if at all). From the date of the sign-up, the data tracks a consumer for 8 months to give her enough time to make a decision<sup>7</sup>. After discussing with the firm, certain consumer paths were removed. These include paths with over 100 “touchpoints” (every firm and consumer activity is called a touchpoint) with the suspicion of bots. Illogical paths (for example, consumer opening the email before receiving it) were also discarded. The frequency of such paths was low (less than 1%).

Some high-level statistics of the dataset are as follows. A total of over 20 million emails were sent, with an average of around 5 emails per consumer. Note that a consumer can receive same type of email multiple number of times. Type 3 email (“ROI”) was the most prevalent in the dataset and each type of email was sent at least a few hundred thousand times. In terms of the frequency of emails, average time between two successive emails to a consumer was around 7 days. The average length of paths (in days) that ended up in a purchase (conversion) was approximately 75 days but most conversions happened within the first couple of weeks after sign-up. The median was slightly below 50 days. In Figure 3.2, we illustrate the consumer journey as a function of her interactions with the firm. Out of all the consumers who sign-up, around 65% *received* at least one email, around 30% *opened* at least one email, around 2% *clicked* on at least one link, and around 1.5% *converted*. Such a progression is not unexpected and it resembles the well-known “funnel” shape in marketing. The eventual conversion probability of a consumer who received at least one email is around 2%. It increases to around 3% for a consumer who opens at least one email and to around 7% for a consumer who clicks on at least one link. Hence, as a consumer goes “deeper” in the funnel, she is more likely to convert.

---

<sup>6</sup>Actual name of “type 4” email not disclosed for anonymity reasons.

<sup>7</sup>Our data suffers from common issues such as we lose track of a consumer if she clears cookies from the web browser.



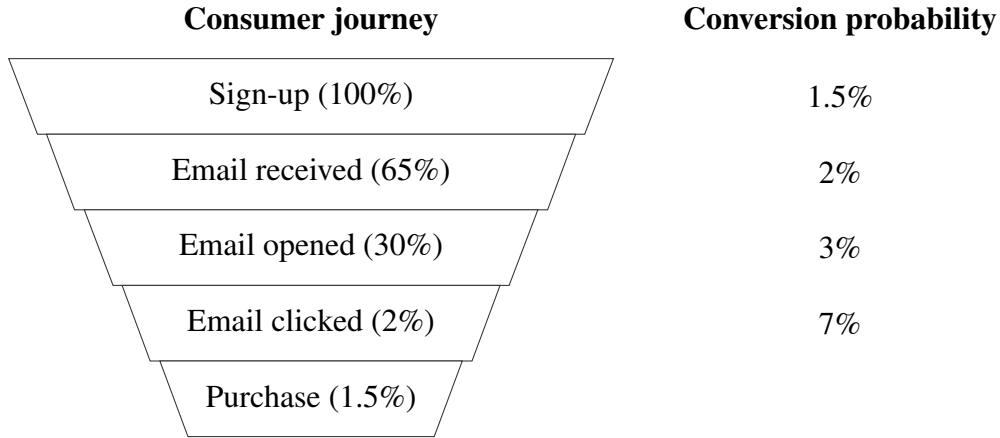


Figure 3.2: Consumer journey as a function of her interactions with the firm. On average, out of 100 consumers who sign-up, around 65, 30, and 2 receive, open, and click an email, respectively. Around 1.5 convert. As a consumer interacts more with the firm, her eventual conversion probability increases from 1.5% to 7%.

### 3.6.2 Conversion Funnel and Estimation

Motivated by the dataset, we now postulate a conversion funnel  $\mathcal{M}$ . In particular, we define the underlying components of  $\mathcal{M}$ , i.e., action space  $\mathbb{A}$ , state space  $\mathbb{S}$ , initial state probabilities  $\lambda$ , reward  $r$ , and transition probabilities  $\mathcal{P}$ . We note that the funnel we propose below is one possibility out of many and in practice, the construction of the funnel will very much be context-specific and the granularity of data the firm can observe. Constructing context-specific state space is an active area of research and we refer the reader to Hallak et al. 2013 and references therein. We emphasize that our framework (both model and algorithm) is general enough to capture an arbitrary (finite) state space. Our goal is to work with a relatively straightforward yet a practical state space and compare various decision-making algorithms using it. As we will see in our results (Section 3.6.4), even a simple setup can be useful in understanding shortcomings of various algorithms, especially the ones that are either myopic or estimation-based.

The action space consists of the 4 types of emails ( $a = 1, 2, 3, 4$ ) along with the action of sending no email ( $a = 0$ ), i.e.,  $\mathbb{A} = \{0, 1, 2, 3, 4\}$ . The state tracks the *stage* of the consumer in the funnel, i.e., her interaction history with the firm as illustrated in Figure 3.2. In particular, the interaction history can be classified into the following 5 stages:

1. **Unaware:** Consumer has signed-up but not received any email yet.
2. **Aware:** Consumer has received at least one email but not opened / clicked on any email.
3. **Interest:** Consumer has opened at least one email but not clicked.
4. **Desire:** Consumer has clicked on a link in at least one email but not made a purchase.
5. **Conversion:** Consumer has made a purchase.

Accordingly,  $\mathbb{S}^+ = \{1, 2, 3, 4\} \cup \{q, c\}$  where “1” denotes stage 1 (unaware) and so on. Such a state space is motivated by the widespread literature on conversion funnel models (Strong 1925; Howard and Sheth 1969; Barry 1987; Bettman et al. 1998; Court 2009; Elzinga et al. 2009; Kotler and Armstrong 2010; Mulpuru 2011; Jansen and Schuster 2011; Bruce et al. 2012), with the key difference being we assume the state to be observable (given our path-level data on each consumer). Our state space allows us to capture consumer behavior in a dynamic fashion and perform interventions at a personalized level. For example, a consumer who is in the initial stages of the funnel might engage with more “broader” messages to help her learn about the firm / product whereas a consumer who has already shown interest (by clicking for instance) might be persuaded by “call to action” type interventions. Having the ability to encode such information in the consumer state is useful as it allows the decision-making algorithm to personalize appropriately.

For simplicity, we do not explicitly model time with the understanding that the frequency of decision-making is fixed, once every week for example. Note that since  $\mathbb{A}$  includes no email ( $a = 0$ ), this does not mean an email will be sent every week. Furthermore, we note that it is possible to capture the time dimension as part of the state space. For example, the state can track “number of days since last email”.

In terms of the initial state probability vector  $\boldsymbol{\lambda} = [\lambda_s]_{s \in \mathbb{S}}$ , since each consumer path starts with a sign-up, the initial state is always  $s = 1$ . Hence,  $\lambda_1 = 1$  and  $\lambda_s = 0$  for all  $s \in \mathbb{S} \setminus \{1\}$ . As discussed in Section 3.2, we normalize the terminal reward  $r$  to 1 wlog.

The only component of the conversion funnel  $\mathcal{M}$  that remains to be discussed is the consumer transition behavior, which is governed by the transition probabilities  $\mathcal{P} = [p_{sas'}]_{(s,a,s') \in \mathbb{S} \times \mathbb{A} \times \mathbb{S}^+}$ , and

we discuss it now. Recall that  $p_{sas'} \in [0, 1]$  denotes the probability consumer transitions from state  $s \in \mathbb{S}$  to state  $s' \in \mathbb{S}^+$  given the firm takes action  $a \in \mathbb{A}$ . Given our definitions of action space and state space as above, the physical meaning of a transition is straightforward. By construction of our state space, the only transitions allowed are in the “forward” direction, i.e., a consumer can either remain at her state and go “deeper” in the funnel, in addition to either quitting or converting. Formally, from  $s \in \mathbb{S}$ , a consumer can transition to  $s' \geq s$  or  $s' \in \{q, c\}$ . This is an implication of the way we constructed the state space in this section and not a necessity of our framework, for which Assumption 3.1 is sufficient.

Given the dataset we described in Section 3.6.1, it should be apparent that the state space defined above is observable to the firm at a consumer level. All that remains before benchmarking MFABL with existing algorithms is to estimate the transition probabilities  $\mathcal{P}$ . As discussed at the beginning of Section 3.6, this will give us a handle on a “ground truth” model, allowing us to simulate consumer behavior and do an “in vitro” comparison of the various decision-making algorithms. We emphasize that none of the decision-making algorithms will have access to the “ground truth” and will only observe paths sampled from it. Furthermore, we note that for model-free algorithms such as MFABL, this step of estimating  $\mathcal{P}$  will not be required in practice since the firm will have access to the real-world paths generated as a function of MFABL. We follow Simester et al. 2006 and estimate the transition probabilities by taking the ratio of counts in the data, i.e., we estimate  $p_{sas'}$  for  $(s, a, s') \in \mathbb{S} \times \mathbb{A} \times \mathbb{S}^+$  as the ratio of the number of times taking action  $a$  at state  $s$  resulted in a consumer transitioning to state  $s'$  and the number of times action  $a$  was taken at state  $s$ . We refer the reader to Simester et al. 2006, which discusses multiple advantages of using such a non-parametric approach. Note that we do not explicitly observe the consumer quitting in the dataset and we use the following heuristic to classify a transition to the quit state  $q$ : if there is no consumer activity (opening / clicking / purchasing) in the future, we mark the next state as  $q$ . (Receiving an email is not a consumer activity but an intervention by the firm.) Furthermore, since we do not observe the action of not sending an email ( $a = 0$ ), we simply implant no emails whenever there is a gap of 24 hours since a previous email. We acknowledge

that our simple estimation procedure could potentially be improved. However, our primary focus is on decision-making algorithms and we emphasize that model-free algorithms such as MFABL do not require one to estimate  $\mathcal{P}$  in practice. If anything, this highlights an advantage of model-free approaches. Irrespective, in Section 3.6.5, we show that our results are robust to changes in the estimate of  $\mathcal{P}$ .

We found the estimates of the funnel to be in alignment with our intuition. For example, averaged over  $s \in \mathbb{S}$ , the one-step conversion probability corresponding to the no email action was the lowest among all actions whereas the one-step quit probability was the highest. The one-step conversion probabilities were generally very low (less than 0.1%) and the self-loop probabilities (i.e., consumer not changing her state) were high (over 90%). Given a consumer in state 1, the average email open rate was around 20% and the click rate was around 2%. Such a “slow-moving” traffic seems consistent with the data in the email marketing industry (CampaignMonitor 2020). Finally, we note that our estimated funnel satisfies Assumption 3.1 as we found the sum of one-step conversion and one-step quit probabilities to be strictly positive for each state-action pair.

### 3.6.3 Benchmark Algorithms

Before presenting our results, we discuss the benchmark algorithms we use. These cover all four possibilities presented in Table 3.1 and we show such categorization in Table 3.2. For each of the algorithms, we assume  $N$  sequential consumers (as not all benchmarks can be defined over parallel consumers). To the best of our knowledge, this work is the first to evaluate state-of-the-art non-myopic learning algorithms (e.g. QL-UCB and PSRL) for a marketing application.

**Random policy.** Given a state  $s \in \mathbb{S}$ , the random policy samples an action uniformly at random from the action set  $\mathbb{A}$ . We use it to establish a lower bound on the performance.

**Myopic estimate-then-optimize (mETO).** mETO works in two phases: exploration and exploitation. For the first  $N_1 \leq N$  consumers, it employs the random policy (“exploration”). At the end of phase 1, mETO uses the collected data to estimate one-step conversion probability  $p_{sac}$  for

each  $(s, a) \in \mathbb{S} \times \mathbb{A}$ . To do so, it simply takes the ratio of the counts. In phase 2 (consumers  $N_1 + 1$  to  $N$ ), it uses the estimates to play a myopic policy, i.e., at state  $s \in \mathbb{S}$ , it plays action  $\arg \max_{a \in \mathbb{A}} \hat{p}_{sac}$ , where  $\hat{p}_{sac}$  denotes the estimate of  $p_{sac}$  (“exploitation”).  $N_1$  is a parameter that needs to be tuned.

**(Myopic) Thompson sampling (TS).** TS is a Bayesian version of mETO. Instead of explicitly dividing into two phases, it maintains a Beta belief over the one-step conversion probability of each state-action pair, i.e.,  $\text{Beta}(\alpha_{sa}, \beta_{sa})$ . When a consumer is in state  $s \in \mathbb{S}$ , TS generates a sample from  $\text{Beta}(\alpha_{sa}, \beta_{sa})$  for all  $a \in \mathbb{A}$  and plays the action with the highest sample value. If the consumer transitions to the conversion state (in one-step), TS increases  $\alpha_{sa}$  by 1 of the corresponding action. Else, it increases  $\beta_{sa}$  by 1 of the corresponding action. At initialization,  $\alpha_{sa}$  and  $\beta_{sa}$  are set to 1 for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ . There are no additional tuning parameters.

**Q-learning with  $\epsilon$ -greedy (QL).** QL (Watkins and Dayan 1992) is perhaps the most widely used non-myopic learning algorithm. It maintains an estimate on the value of each state-action pair. Denote the estimate by  $Q(s, a)$  for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ . Given a consumer at state  $s \in \mathbb{S}$ , it plays an action with highest  $Q(s, a)$  over  $a \in \mathbb{A}$  (with  $\epsilon$ -greedy). Then, if the consumer transitions to state  $s' \in \mathbb{S}^+$ , it updates  $Q(s, a)$  as follows:

$$Q(s, a) \leftarrow (1 - \kappa_{sa}) \times Q(s, a) + \kappa_{sa} \times \max_{a' \in \mathbb{A}} Q(s', a').$$

The parameter  $\kappa_{sa}$  denotes the stepsize and we set it to  $\frac{1}{n_{sa}}$ , where  $n_{sa}$  denotes the number of visits to  $(s, a)$  so far. We initialize  $Q(s, a) = 0$  for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ ,  $Q(q, a) = 0$ , and  $Q(c, a) = 1$  for all  $a \in \mathbb{A}$ . The only remaining tuning parameter is  $\epsilon$ .

**Q-learning with upper confidence bounds (QL-UCB).** QL-UCB (Jin et al. 2018; Dong et al. 2019) is a variant of QL that leverages UCB ideas and is known to exhibit strong theoretical properties. We refer the reader to Jin et al. 2018 and Dong et al. 2019 for its formal description. We use the version presented in Dong et al. 2019 since it allows for a variable horizon (i.e., length

of a consumer path) whereas the version in Jin et al. 2018 assumes a deterministic horizon. We note that both Jin et al. 2018 and Dong et al. 2019 assume the presence of a discount factor  $\gamma \in [0, 1)$ . In our MDP, we do not discount the terminal reward and hence, MFABL implicitly sets  $\gamma = 1$ . For comparison purposes, in the algorithm of Dong et al. 2019, we set  $\gamma$  to be close to 1 (as setting it to 1 results in some of their parameters being undefined). There are 2 other parameters:  $\epsilon$  and  $\delta$ . The physical meaning of  $\delta$  is that the theoretical guarantee of Dong et al. 2019 holds with probability  $1 - \delta$  (hence,  $\delta$  should be close to 0) and  $\epsilon$  quantifies the degree of suboptimality (hence,  $\epsilon$  should be close to 0 as well). We initialize  $Q(s, a)$  for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$  as suggested in Dong et al. 2019,  $Q(q, a) = 0$ , and  $Q(c, a) = 1$  for all  $a \in \mathbb{A}$ .

**Estimate-then-optimize (ETO).** ETO is a non-myopic extension of mETO. Similar to mETO, it works in two phases. For the first  $N_1 \leq N$  consumers, it employs the random policy. At the end of phase 1, ETO uses the collected data to estimate the *entire* transition structure  $\mathcal{P}$ , i.e.,  $p_{sas'}$  for all  $(s, a, s') \in \mathbb{S} \times \mathbb{A} \times \mathbb{S}^+$ . To do so, it simply takes the ratio of the counts. At the end of phase 1, it computes an optimal policy to the estimated MDP  $\widehat{\mathcal{M}} \equiv (\mathbb{S}, \mathbb{A}, \widehat{\mathcal{P}}, \boldsymbol{\lambda}, r)$ , where  $\widehat{\mathcal{P}}$  denotes the estimate of  $\mathcal{P}$ . This computed policy is used to interact with consumers in phase 2, i.e., consumers  $N_1 + 1$  to  $N$ . As in mETO,  $N_1$  is a parameter. ETO requires solving an MDP only once.

**Posterior sampling for reinforcement learning (PSRL).** PSRL (Strens 2000; Osband et al. 2013; Agrawal and Jia 2017) generalizes Thompson sampling in a model-based manner. It maintains a belief over the entire transition probabilities  $\mathcal{P}$ . In particular, given  $(s, a) \in \mathbb{S} \times \mathbb{A}$ , it maintains a Dirichlet( $\boldsymbol{\alpha}_{sa}$ ) belief over the one-step transition probability vector  $\mathbf{p}_{sa} := [p_{sas'}]_{s' \in \mathbb{S}^+}$  where  $\boldsymbol{\alpha}_{sa} := [\alpha_{sas'}]_{s' \in \mathbb{S}^+}$ . Before each consumer arrives, PSRL generates a sample from Dirichlet( $\boldsymbol{\alpha}_{sa}$ ) for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ . Denoting the corresponding sampled transition probabilities by  $\widehat{\mathcal{P}}$ , it computes an optimal policy of the MDP  $\widehat{\mathcal{M}} \equiv (\mathbb{S}, \mathbb{A}, \widehat{\mathcal{P}}, \boldsymbol{\lambda}, r)$  and plays the computed policy. Using the transition data of the form “ $(s, a, s')$ ” observed in the realized consumer path, it updates the belief over  $\mathcal{P}$  using Dirichlet-multinomial conjugacy. For example, if taking action  $a \in \mathbb{A}$  at state  $s \in \mathbb{S}$  transitioned the consumer to state  $s' \in \mathbb{S}^+$ , PSRL increases  $\alpha_{sas'}$  by 1. This updated belief

	<b>myopic</b>	<b>non-myopic</b>
<b>estimation</b>	mETO	ETO
<b>learning</b>	TS	QL, QL-UCB, PSRL

Table 3.2: Classification of the benchmark algorithms into four sets we introduced in our literature review.

is used to generate a sample of  $\mathcal{P}$  and re-compute an optimal policy for the next consumer and so on. Hence, given  $N$  consumers, PSRL solves  $N$  MDPs. At initialization,  $\alpha_{sas'}$  is set to 1 for all  $(s, a, s') \in \mathbb{S} \times \mathbb{A} \times \mathbb{S}^+$ . There is no additional parameter.

Table 3.2 categorizes the benchmarks along the two dimensions of interest, namely, myopic / non-myopic and estimate / learn. MFABL, though not shown in Table 3.2, lies in the fourth category: non-myopic and learning-based. In addition to these two dimensions, we highlight a third dimension of model-free versus model-based since that affects the scalability of an algorithm. In particular, all algorithms except ETO and PSRL are model-free since they do not estimate / learn the entire  $\mathcal{P}$ . ETO and PSRL, on the other hand, are model-based as they attempt to estimate / learn the entire  $\mathcal{P}$  and compute an optimal policy using such information. Note that benchmarking a model-free approach such as MFABL with model-based approaches such as ETO and PSRL can be seen as “unfair” since model-free approaches are constrained in the type of information they can store. In particular, they do not store information on the transition dynamics of the funnel, which leads to better tractability, but possibly at a loss in performance. To state it differently, the information stored by model-based approaches is a superset of the information stored by model-free approaches since the  $Q$ -values can be computed using the transition probabilities  $\mathcal{P}$  but not vice-versa. In fact, empirical work has suggested that model-based approaches may require less samples to learn as compared to model-free (Deisenroth and Rasmussen 2011; Schulman et al. 2015). Hence, model-based benchmarks seem too optimistic to begin with but nonetheless, we include them to understand how much performance loss we incur by being model-free.

Finally, we introduce a variant of MFABL that we refer to as MFABL2. There is only one

change. In particular, if taking action  $a \in \mathbb{A}$  at state  $s \in \mathbb{S}$  results in a self-loop, i.e., a transition to  $s$  itself, then MFABL2 does not do any update. Intuitively, this should increase the learning rate of MFABL by discarding redundant transitions.

### 3.6.4 Results and Discussion

We now present our results. We first discuss the setup we used followed by the parameter values for various algorithms. Then, we define the two metrics used to compare the algorithms and finally, we showcase and discuss the results.

In terms of the setup, we used the estimated MDP from Section 3.6.2 as the “ground truth” funnel. For each algorithm, we let it “interact” with a sequence of  $N = 100,000$  consumers sampled from the ground truth and repeated the entire process for  $R = 100$  runs (each with a different seed) to account for the random evolution. Given algorithm  $\text{ALG}$ , denote by  $y_{nr}^{\text{ALG}} \in \{0, 1\}$  the indicator whether consumer  $n \in \{1, \dots, N\}$  converted in run  $r \in \{1, \dots, R\}$ . Given the extensive computational load required, we used over 100 cores on a high-performance computing cluster and parallelized over  $r \in \{1, \dots, R\}$ .

For estimation-based approaches (mETO and ETO), we set  $N_1 = 2,500$  (number of consumers used to estimate). Given that we are estimating around 50 parameters, this number seems high enough to estimate the transition probabilities of our relatively simple funnel and not high enough (contrast with  $N = 100,000$ ) to “sacrifice” too many consumers. (In fact, as we will see in our sensitivity analysis (Section 3.6.5),  $N_1 = 2500$  is near-optimal.) For  $\epsilon$ -greedy, we used  $\epsilon = 0.01$  for MFABL / MFABL2 and  $\epsilon = 0.02$  for QL. We discuss the sensitivity of these four algorithms with respect to the underlying parameters in Section 3.6.5. For QL-UCB, we tried 8 possible combinations in the set  $\{(\epsilon, \gamma, \delta) : \epsilon \in \{0.01, 0.05\}, \gamma \in \{0.95, 0.99\}, \delta \in \{0.01, 0.05\}\}$  and obtained very similar results for each combination. So, we arbitrarily picked  $(0.01, 0.95, 0.01)$ . None of the other algorithms require any parameter to be set.

We use two performance metrics. First, for algorithm  $\text{ALG}$ , we define the *performance ratio*



as

$$\text{PR}^{\text{ALG}} = \frac{\text{Expected number of conversions under ALG}}{\text{Expected number of conversions under OPT}},$$

where OPT denotes an optimal policy corresponding to the ground truth funnel. The denominator equals  $Nv^*$ , where  $v^*$  denotes the optimal conversion probability. Since  $\lambda_1 = 1$  in our funnel,  $v^* = V^*(1)$ , where  $V^*(s)$  denotes the optimal value function corresponding to state  $s \in \mathbb{S}$  (recall from Section 3.2). We estimate the performance ratio using the following unbiased Monte-Carlo estimator:

$$\frac{1}{R} \sum_{r=1}^R \frac{\sum_{n=1}^N y_{nr}^{\text{ALG}}}{Nv^*}.$$

Trivially,  $\text{PR}^{\text{ALG}} \in [0, 1]$  and a value closer to 1 denotes a better performance. Moreover, we note that the denominator here is “too optimistic” since it assumes knowledge of the ground truth from the beginning (consumer 1). Second, we evaluate algorithms using their *expected total regret* as a function of  $N$ , an unbiased estimator for which is

$$Nv^* - \frac{1}{R} \sum_{r=1}^R \sum_{n=1}^N y_{nr}^{\text{ALG}}.$$

Naturally, a smaller regret is better. Note that our first metric provides a static snapshot of the performance after  $N$  consumers whereas the second metric provides a fluid story from consumer 1 to  $N$ . We split our presentation of results in two steps. First, we benchmark MFABL with respect to model-free approaches and second, with respect to model-based approaches.

Figure 3.3(a) shows the performance ratio of MFABL benchmarked with various model-free algorithms. For the performance ratio, we have shown the unbiased estimate discussed above (blue dots in the figure) along with the standard deviation over  $r \in \{1, \dots, R\}$  (blue lines in the figure denote  $\pm 1$  standard deviation). MFABL2 achieves a performance ratio of around 93%, outperforming all other model-free algorithms. MFABL is the second-best (73%), followed by

QL, TS, and mETO (all three around 66%). Surprisingly, QL-UCB is as good as the random policy (45%). In Figure 3.3(b), we plot the corresponding regret as the number of consumers increases from 1 to 100,000. MFABL2 clearly dominates all others. Notice that MFABL2 seems to have “converged” for  $N$  as low as 10,000. (Since MFABL2 employs  $\epsilon$ -greedy, its regret curve will always have a positive slope.)

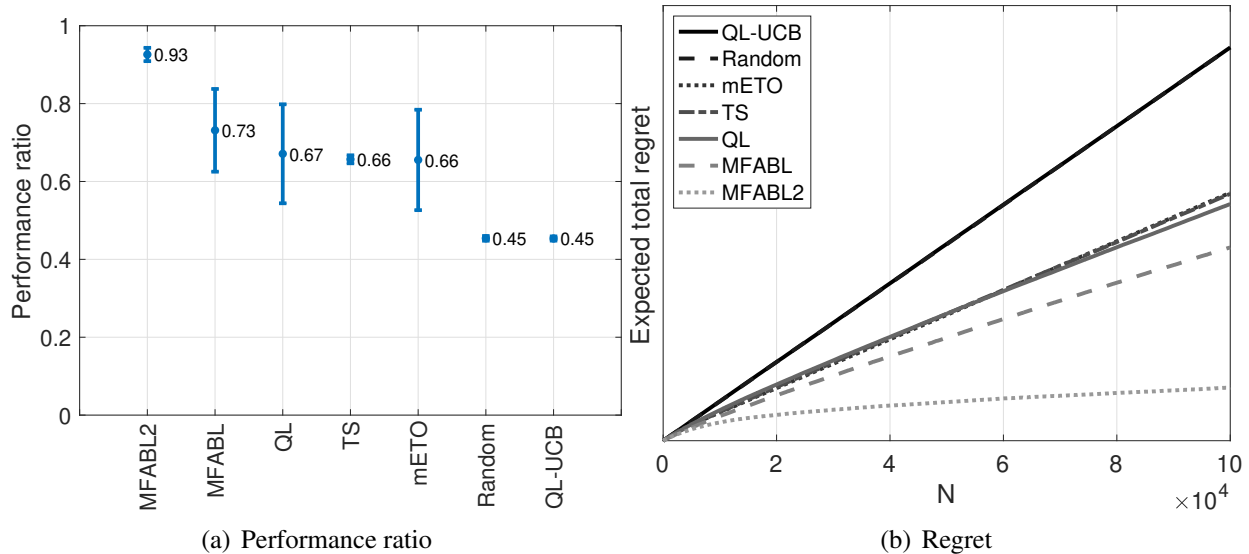


Figure 3.3: Benchmarking MFABL / MFABL2 with model-free algorithms.

Before discussing model-based algorithms, we discuss the key shortcoming of myopic approaches such as TS and mETO. Intuitively, myopic approaches “fail” when the long-run value of an action dominates its myopic value (see (3.1)). In our MDP, we expect such instances to occur in earlier stages of the funnel since there might be an intervention that does a good job of pushing the consumer deeper into the funnel but has a lower one-step conversion probability. In fact, this is what we found in our data. In particular, at stage 1, the myopic value (one-step conversion probability) was highest for email type 1 whereas the optimal policy was to use email type 2, which had the third highest myopic value. We observed a similar finding at stage 2. At stage 3, the myopic value of the non-myopic optimal action was close to that of the myopic optimal action and at stage 4, the myopic optimal was the non-myopic optimal. Accordingly, as we move away from the end of the funnel, a myopic policy degrades as it fails to account for the long-run value. This explains

the significant difference observed between the performance of myopic approaches and MFABL2 and highlights the importance of capturing long-run value in such settings.

We benchmark MFABL2 with model-based approaches (ETO and PSRL) in Figure 3.4. The performance ratio is shown in Figure 3.4(a) and the regret in Figure 3.4(b). All three approaches achieve a performance ratio of over 90%, with PSRL almost close to 1 (98%), followed by MFABL2 (93%) and ETO (92%). Given our earlier discussion surrounding the appropriateness of benchmarking a model-free approach with model-based, it is interesting to see that MFABL2 can be comparable to ETO. In fact, as seen in Figure 3.4(b), the performance of ETO seems to degrade (with respect to that of MFABL2) as the number of consumers increases. This observation is supported by the fact that ETO, being an estimation-based approach, stops learning after the initial phase of  $N_1$  consumers. It employs the policy computed using the estimate of  $\mathcal{P}$  from the first  $N_1$  consumers, which might be suboptimal (and it is in our numerics as seen in Figure 3.4(b)!) due to the noise in the estimate of  $\mathcal{P}$ . In general, this is a key limitation of estimation-based approaches and this is amplified under the presence of concept shift, as we will see in Section 3.7.

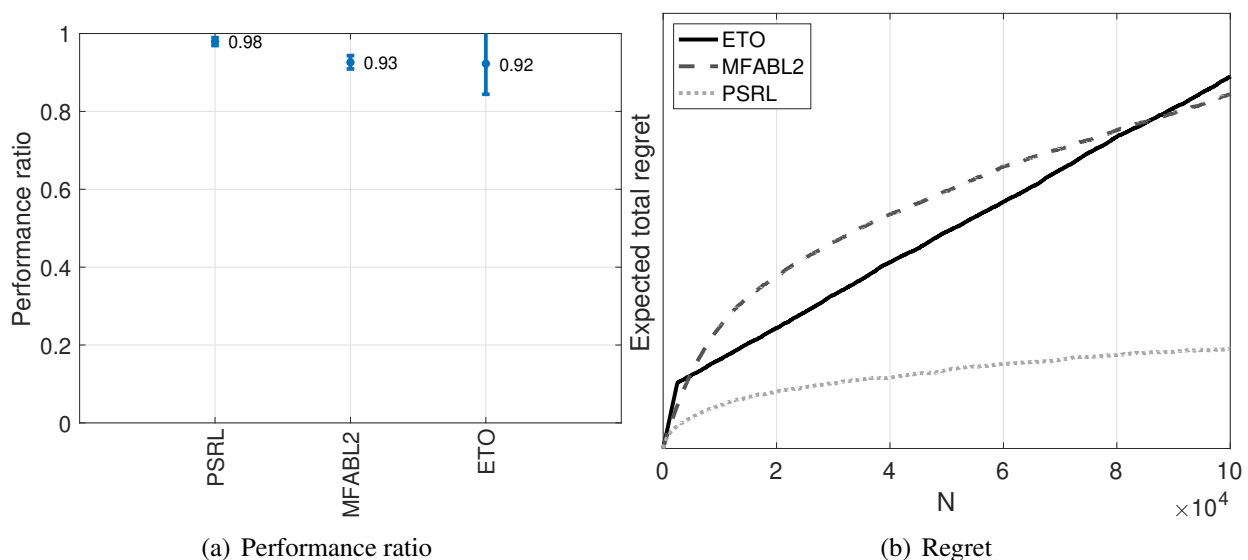


Figure 3.4: Benchmarking MFABL2 with model-based algorithms.

PSRL achieves a better performance than MFABL2. Given the near-optimality of PSRL with respect to any learning algorithm (Osband and Van Roy 2017), this is not too surprising. Intuitively,

since PSRL is model-based and learns on a continuing basis, it is able to use data much more efficiently. However, this comes at an increased computational cost of solving an MDP in each iteration and storing  $O(S^2A)$  parameters.

We summarize our findings in Figure 3.5, where we plot the various algorithms along two dimensions: (1) performance and (2) scalability. By scalability, as discussed in Section 3.4.2, we refer to both the storage and computational requirements of an algorithm. Among the model-free approaches, MFABL2 dominates since it has a better performance and similar storage and computational requirements. ETO achieves similar performance as MFABL2 but has a lower scalability (since it stores the entire  $\mathcal{P}$  and requires solving the MDP once). PSRL has the best performance but the least scalability. It should be clear that myopic approaches are dominated by non-myopic (MFABL2 for instance delivers a better performance than all myopic approaches at a similar computation cost) and that estimation-based approaches are dominated by learning-based ones (consider MFABL2 versus mETO / ETO).

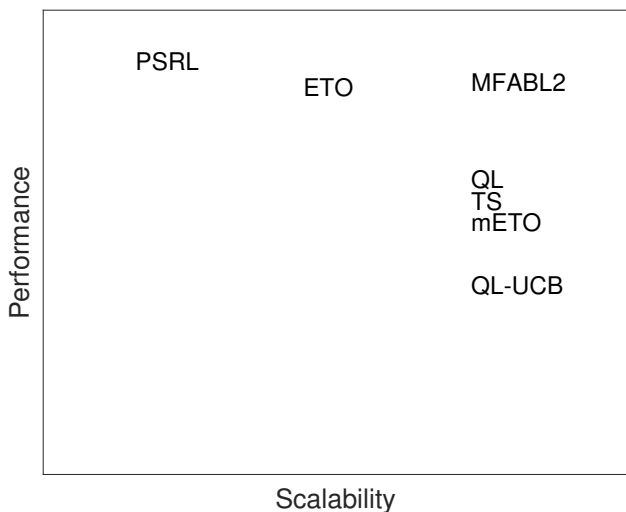


Figure 3.5: Summarizing the various algorithms across two dimensions: (1) performance and (2) scalability. The scale of the  $y$ -axis (performance) is consistent with the performance ratios in Figures 3.3(a) and 3.4(a). The  $x$ -axis (scalability) is partitioned into 3 levels as follows. PSRL is least scalable since it stores the entire transition structure and solves an MDP per consumer (i.e.,  $N$  MDPs across all consumers) whereas ETO stores the entire transition structure but only solves one MDP (across all consumers). All other algorithms are model-free and exhibit similar scalability.

### 3.6.5 Sensitivity Analysis

We conclude this section with some sensitivity analysis. First, we evaluate the performance ratio of various algorithms by varying the underlying parameters. For  $\epsilon$ -greedy approaches (MFABL2 and QL), we vary  $\epsilon \in \{0, 0.01, \dots, 0.10\}$ . For estimation-based approaches (mETO and ETO), we vary  $N_1 \in \{100, 1000, 2500, 5000, 10000, 15000, 20000\}$ . For each parameter setting, we use  $N = 100,000$  (number of consumers) and  $R = 100$  (number of runs) as before. Figure 3.6 shows the results corresponding to  $\epsilon$ -greedy approaches. The performance ratio of MFABL2 seems to degrade as  $\epsilon$  decreases. This is not unexpected (especially for large  $N$ ) as MFABL2 sacrifices approximately  $\epsilon$ -optimality by definition. Despite this decreasing pattern, MFABL2 dominates among the model-free approaches. The performance ratio of QL seems to be around 0.66 for all values of  $\epsilon$  we experimented with. In Figure 3.7, we plot the sensitivity results corresponding to estimation-based approaches. As can be seen, our choice of  $N_1 = 2,500$  reflected almost the best-case scenario for estimation-based approaches since the performance ratio seems to exhibit a unimodal pattern with respect to  $N_1$  and 2,500 is near-optimal.

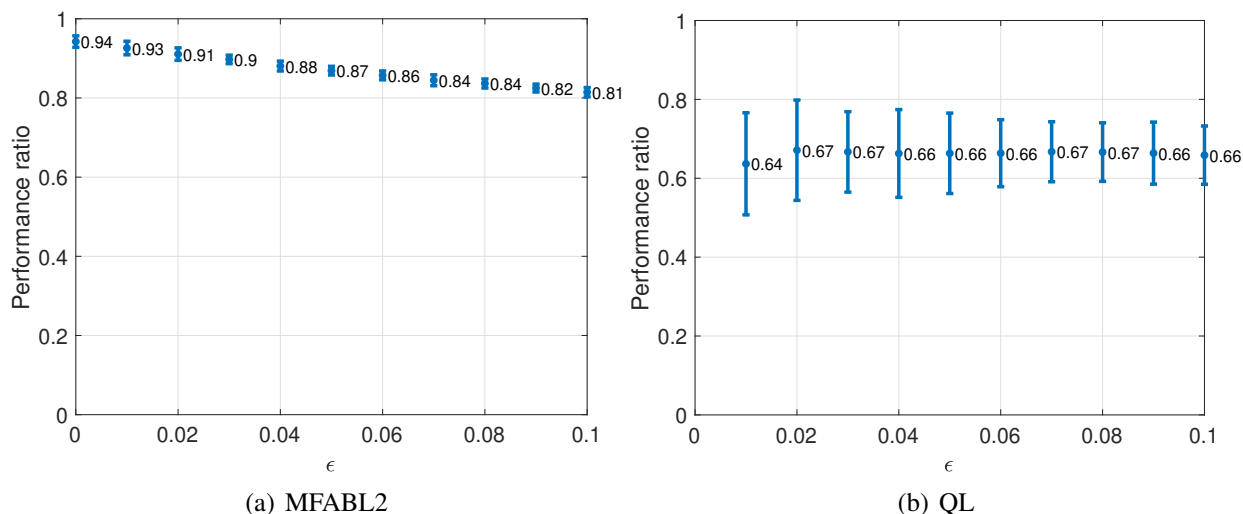


Figure 3.6: Sensitivity of MFABL2 and QL with respect to  $\epsilon$ . (Note that QL with  $\epsilon = 0$  is equivalent to the random policy.)

Second, to check sensitivity of our results to changes in the estimate of  $\mathcal{P}$ , we do the following. We repeat our computations from Section 3.6.4 but for each run  $r \in \{1, \dots, R\}$  ( $R = 100$ ), we

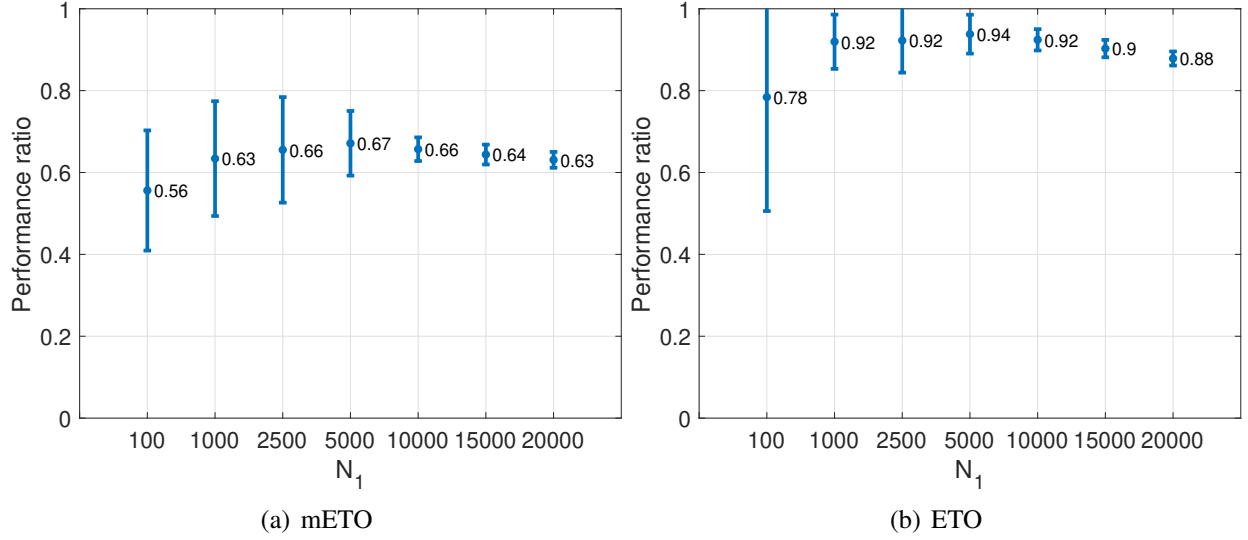


Figure 3.7: Sensitivity of mETO and ETO with respect to  $N_1$ .

perturb our estimate of  $\mathcal{P}$ . In particular, for  $(s, a, s') \in \mathbb{S} \times \mathbb{A} \times \mathbb{S}^+$ , we add noise to  $p_{sas'}$  as follows:

$$p_{sas'} \leftarrow p_{sas'} \times (1 + \eta_{sas'}^r),$$

where  $\eta_{sas'}^r \sim \text{uniform}[-0.25, 0.25]$ . After adding the noise, we ensure  $[p_{sas'}]_{s' \in \mathbb{S}^+}$  sums to 1 by appropriate normalization. We report the corresponding performance ratios in Figure 3.8 and observe they are almost identical to the ones in Figures 3.3(a) and 3.4(a). This highlights the robustness of our results to changes in  $\mathcal{P}$ .

### 3.7 Value of Prior Information, Concept Shift, and Covariate Shift

In this section, using the estimated funnel from Section 3.6.2, we shed light on some of the practical considerations we discussed in Section 3.4.2. In particular, we first showcase the ability of MFABL to let the firm encode prior information and the value of such information (Section 3.7.1). Second, we demonstrate the robustness of MFABL to concept shift and benchmark it with ETO (Section 3.7.2). Third, we discuss the working of MFABL under covariate shift (Section 3.7.3).

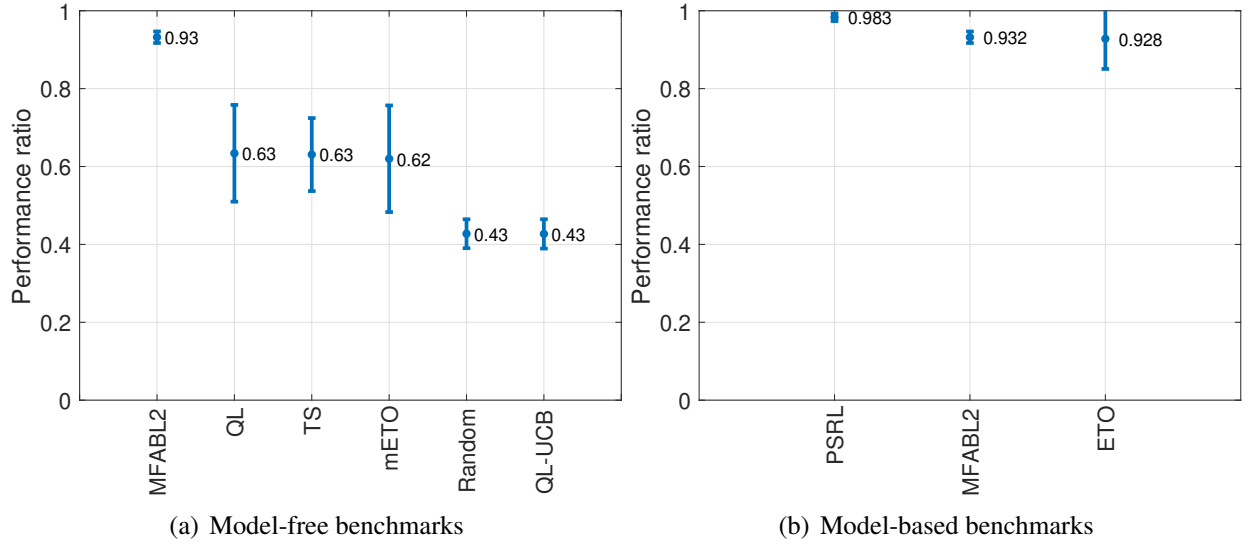


Figure 3.8: Performance ratios corresponding to perturbations in the transition probabilities.

### 3.7.1 Value of Prior Information

As mentioned in Section 3.4.2, if a firm might have some prior information regarding the consumer behavior, it seems desirable to allow the firm to leverage such information. Our algorithm allows the firm to encode state-action specific prior knowledge via the prior counts on the “value” of each state-action pair (recall the inputs to Algorithm 6). In this subsection, under a somewhat stylized setup, we illustrate the ease with which prior information can be encoded and how valuable such information can be. In particular, we perform the following experiments using the estimated funnel from Section 3.6.2. In the zeroth experiment, we assume the firm has access to no prior information whereas in experiment  $x \in \{1, 2, 3, 4\}$ , we assume the firm has prior information on stage  $x$  and beyond. For example, if  $x = 3$ , we assume the firm knows the true values of  $Q^*(s, a)$  for all  $s \geq 3$  and  $a \in \mathbb{A}$  and sets the prior counts  $(\alpha_{sa}, \beta_{sa})$  such that the expected value equals  $Q^*(s, a)$  and the variance is essentially zero. We reduce the value of  $N$  from 100,000 to 10,000 to leave more room for improvement (recall performance ratio with  $N = 100,000$  was already 0.93).

The results for MFABL2 (with  $\epsilon = 0.01$  as before) are shown in Figure 3.9. In Figure 3.9(a), we display the performance ratios corresponding to the five experiments. Note that under the “no prior information” setting, the performance ratio is now 0.75. This decrease from 0.93 (when  $N$

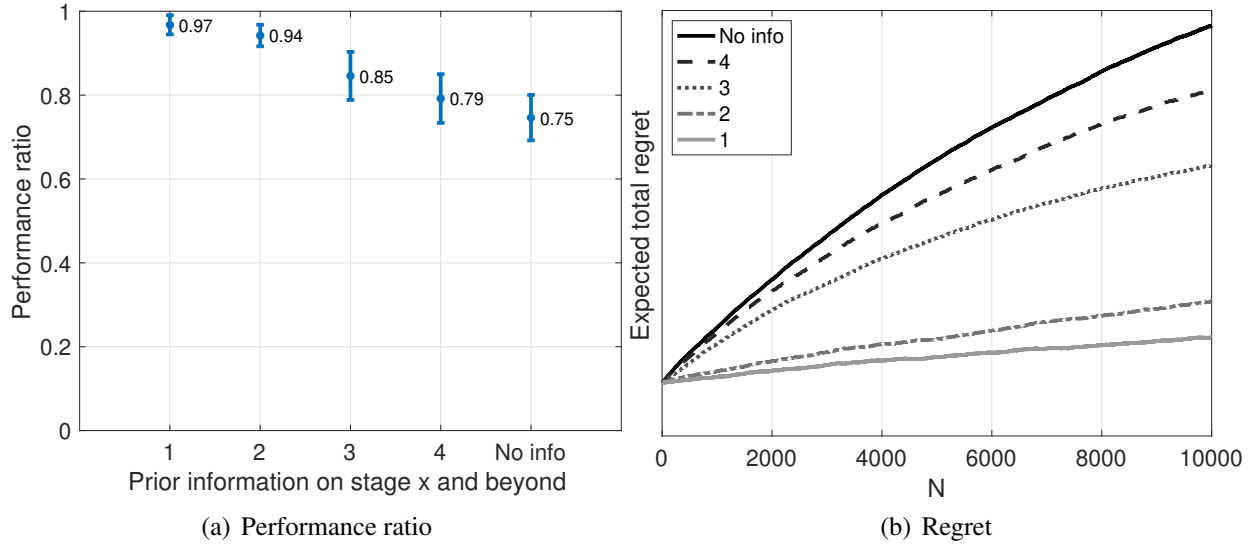


Figure 3.9: Value of prior information corresponding to MFABL2.

was set to 100,000) seems intuitive since we are giving the firm less number of consumers to learn from. As the  $x$ -axis decreases, the firm has more prior information and hence, the performance gets better. In particular, having information on just the last stage improves the performance ratio from 0.75 to 0.79 whereas having information on last 2 and 3 stages increases it to 0.85 and 0.94, respectively. Figure 3.9(b) shows the corresponding regret curves.

### 3.7.2 Concept Shift

We now discuss the topic of concept shift. As discussed in Section 3.4.2, in our conversion funnel model, concept shift refers to the possibility of the consumer behavior (transition probabilities) changing over time. To understand how our algorithm operates under concept shift, we do the following simple experiment. We set  $N = 10,000$  and split the consumer behavior in two phases. In phase 1, transition probabilities are  $\mathcal{P}_1$  (the ones estimated in Section 3.6.2) and in phase 2, they change to  $\mathcal{P}_2$ . To generate  $\mathcal{P}_2$ , we randomly permute the actions so that the optimal conversion probability remains the same. Phase 1 corresponds to the first  $N_1 \leq N$  consumers and phase 2 corresponds to consumers  $N_1 + 1$  to  $N$ . We experiment with  $N_1 \in \{1000, 2500, 5000\}$  and evaluate MFABL2 (with  $\epsilon = 0.01$  as before) and ETO, where ETO uses the first  $N_1$  consumers to estimate



(similar to the real-life setup of Simester et al. 2020). As before, we perform  $R = 100$  runs ( $\mathcal{P}_2$  in run  $r$  can be different from  $\mathcal{P}_2$  in run  $r'$  due to the random permutation). The decision-making algorithms do not know that a switch from  $\mathcal{P}_1$  to  $\mathcal{P}_2$  will happen.

The performance ratios for both MFABL2 and ETO are shown in Figure 3.10. MFABL2 performs better for all values of  $N_1$ . In fact, on further digging, we found that MFABL2 exhibits a “win-win” behavior in the sense that it has a higher performance ratio in both phases. This makes sense because in phase 1, ETO just plays a random policy whereas MFABL2 makes an attempt to learn. In phase 2, ETO plays the policy computed using phase 1 data (which is likely to be suboptimal) whereas MFABL2 eventually adapts to phase 2 consumers (due to its learning nature).

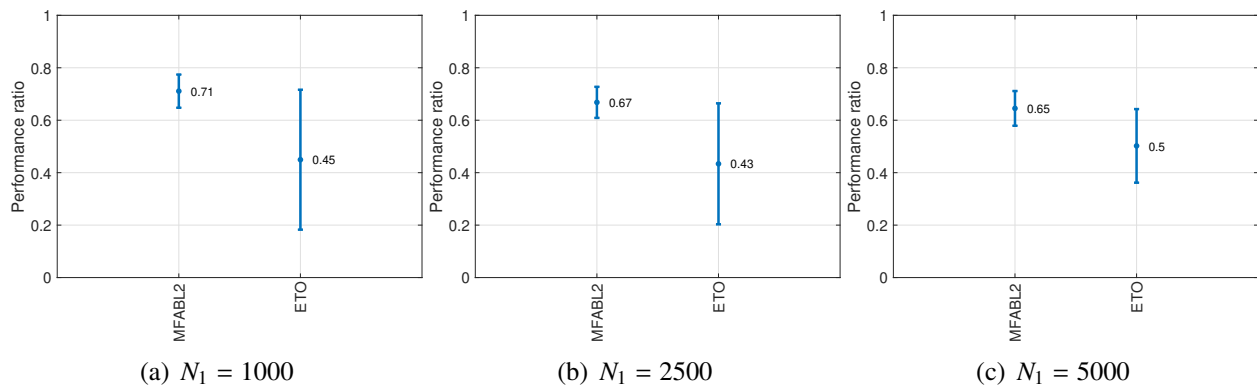


Figure 3.10: Performance ratio for concept shift corresponding to various values of  $N_1$  with  $N = 10,000$ .

In Figure 3.11, we display the corresponding regret curves. It is clear that MFABL2 incurs a lower regret. Furthermore, on closer examination, one should be able to see kinks corresponding to the concept shift in these curves (especially for MFABL2). Initially, MFABL2 learns the behavior of phase 1 consumers and hence, the regret starts to “decay”. However, as soon as the phase shifts, the regret jumps up but eventually, MFABL2 adapts. We note that MFABL2 detects the phase shift *automatically* due to its learning nature.

As noted in Section 3.4.2, the speed at which our learning algorithm adapts to the phase shift will depend on the strength of the belief at the end of phase 1. To induce a strong belief, we increased  $N_1$  to 50,000 with  $N = 100,000$ . The corresponding results are shown in Figure 3.12.

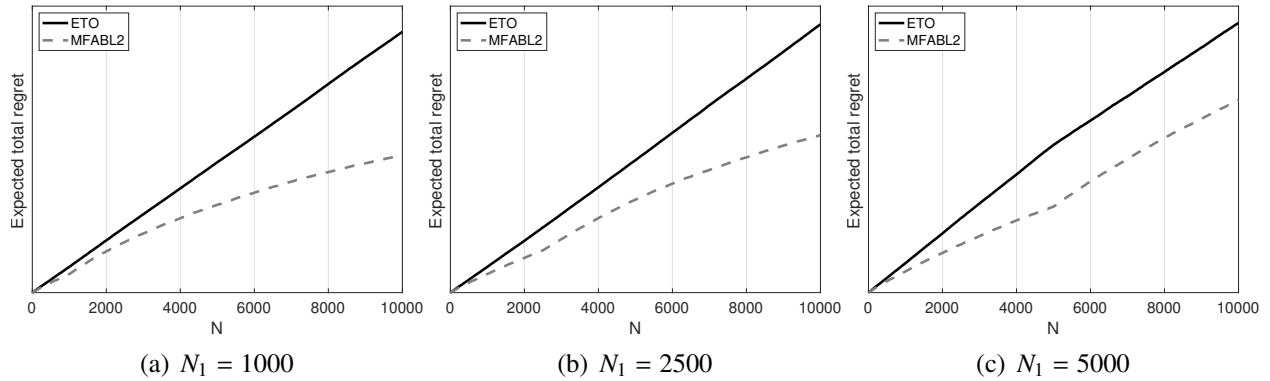


Figure 3.11: Regret for concept shift corresponding to various values of  $N_1$  with  $N = 10,000$ .

(We did not change  $\epsilon$  for MFABL2, i.e.,  $\epsilon = 0.01$ .) Observations from above still hold, showing the robustness of our approach. In fact, we highlight an important trade-off in a setting with large  $N_1$ . Due to large  $N_1$ , the belief at the end of phase 1 is strong, making the adaptation to phase 2 slower. However, at the same time, our algorithm enjoys a higher performance ratio in phase 1 since it can exploit its learning on more consumers (compared to a setting with a smaller  $N_1$ ). In fact, the average performance ratio of MFABL2 in Figure 3.12(a) (78%) is higher than in Figure 3.10 (65% to 71%).

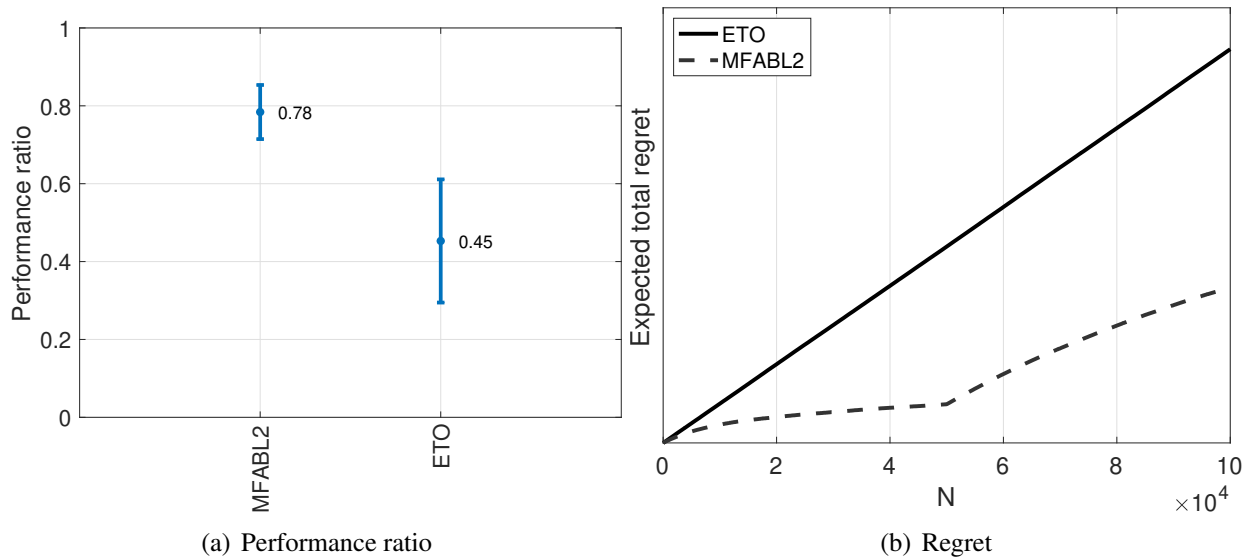


Figure 3.12: Illustration of concept shift for  $N_1 = 50,000$  with  $N = 100,000$ .

### 3.7.3 Covariate Shift

To understand covariate shift, we perform the following experiment. We use the estimated funnel from Section 3.6.2 and augment the state space to include a second dimension  $x \in \{1, 2\}$ , which one can think of capturing a consumer feature such as age (young / old), sex (male / female), or location (city / rural). So, the new state is represented as  $(s, x)$  where  $s$  denotes the stage and  $x$  denotes the type. For  $x = 1$ , we let the transition behavior be the same as in the estimated funnel and for  $x = 2$ , we randomly permuted the actions (not important). We sampled  $N = 10,000$  consumers under the following split: a consumer is of type 1 with probability  $\omega_1 = 95\%$  and type 2 with probability  $\omega_2 = 5\%$ . Hence, we expect to see 9500 type 1 and 500 type 2 consumers. In other words, this dataset is not very representative of type 2 consumers and if in the future, the firm encounters a type 2 consumer, the information gathered from this dataset might not be enough to make a good decision. Note that the absolute value of  $N$  is not important here but the number of type 2 consumers the firm encounters is, i.e., the scale of  $\omega_2$  with respect to  $N$  is important. As noted in Simester et al. 2020, estimation-based approaches struggle in such settings. To illustrate the mechanics of MFABL2, we show in Figure 3.13 the Beta belief over the state-action pairs  $((s, x), a)$  for  $s = 1$ ,  $x \in \{1, 2\}$ , and  $a \in \mathbb{A}$  after MFABL2 has interacted with 10,000 consumers (corresponding to just one run). Given the 0.95-0.05 split between type 1 and 2 consumers, the posterior variance in the beliefs corresponding to  $x = 1$  is quite low compared to that of  $x = 2$ . By construction of how MFABL2 picks an action (mimicking Thompson sampling with  $\epsilon$ -greedy), such higher posterior variance for  $x = 2$  forces MFABL2 to explore the unexplored actions. So, if in the future, the firm encounters a consumer of type 2, MFABL2 implicitly accounts for the uncertainty present due to lack of encounters with type 2 consumers in the initial dataset.

## 3.8 Conclusions and Further Research

In this work, we study the problem of optimal sequential personalized interventions from the point-of-view of a firm promoting a product under a fairly general model of consumer behavior,

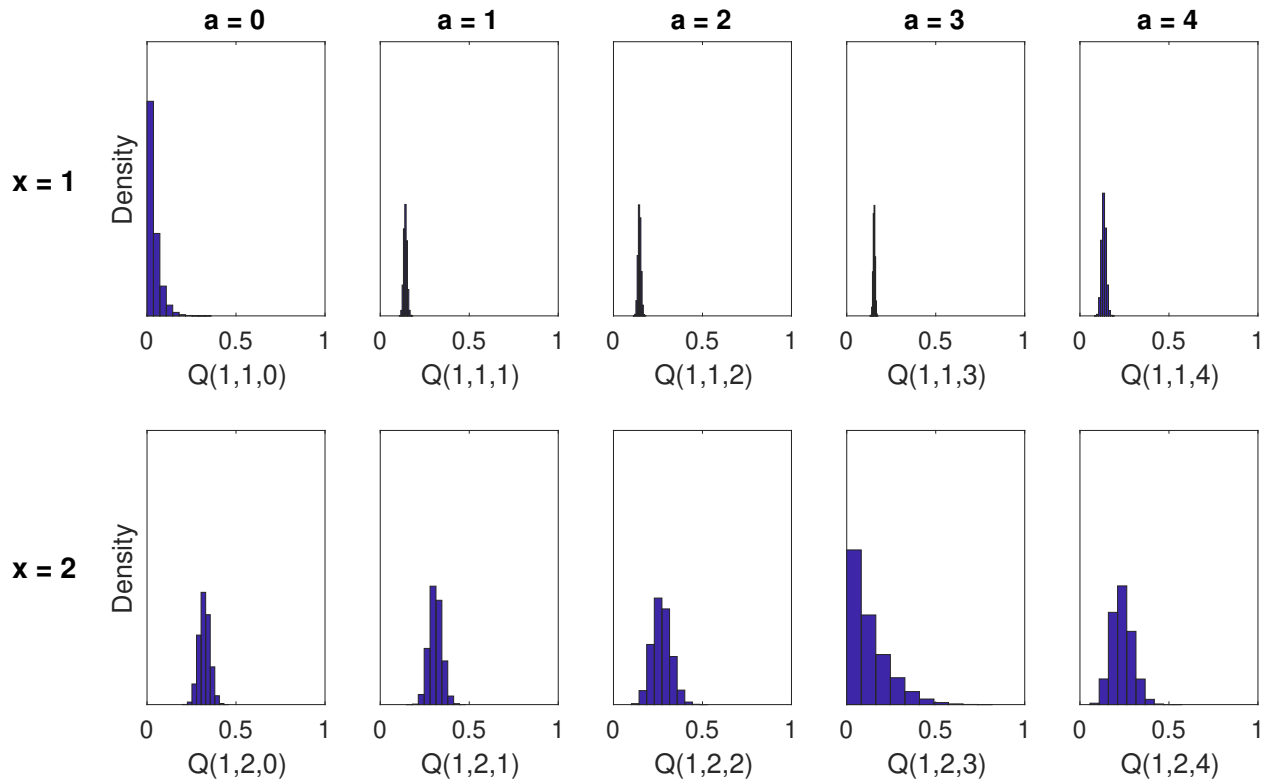


Figure 3.13: Covariate shift illustration.

which we call the conversion funnel. Our state-based conversion funnel model explains the journey of a consumer (at an individual level) from her initial interactions with the firm till the time she makes a decision to purchase or not. The behavior of the consumer is driven by the sequential intervention decisions the firm makes as a function of the consumer state at various points of time. The effect of each intervention on the consumer behavior is allowed to be state-specific. Due to its sequential nature, our model captures both the myopic value and the long-run value of each intervention. Apriori, the firm does not know the state-specific effects of the interventions and needs to learn / estimate them in order to make optimal intervention decisions. In that direction, we propose a decision-making algorithm, which we call model-free approximate Bayesian learning. Our algorithm inherits the simplicity of Thompson sampling for a multi-armed bandit in the sense that it maintains an approximate belief on the value of each state-specific intervention and updates this belief (mimicking the Beta-Bernoulli rule in Thompson sampling) as it interacts with

the consumers. We establish the asymptotic optimality of our algorithm and benchmark it with various algorithms on a real-world large-scale dataset. In the numerics, our algorithm dominates the traditional algorithms that either optimize for the myopic reward or are estimation-based. It also outperforms state-of-the-art model-free learning algorithms that are non-myopic. Furthermore, in contrast to the estimation-based approaches, our algorithm is able to adapt automatically to the underlying changes in consumer behavior (concept shift) and maintains a high level of uncertainty on the value of less explored consumer segments (covariate shift).

We believe there is immense potential for non-myopic learning-based approaches to personalized marketing and we highlight some interesting research directions now. In our conversion funnel model, consumer behavior is captured using the notion of a state and the firm makes decisions as it observes the consumer state. It should be clear that the out-of-sample performance of any decision-making algorithm that optimizes conditioned on a state space would depend on how well the state space represents reality. Though our framework (both model and algorithm) is general enough to capture an arbitrary (finite) state space, constructing an appropriate state space is an interesting research avenue. We note that the “appropriateness” of a state space would very much vary between firms and industries and constructing such context-specific state space is an active area of research (Hallak et al. 2013).

Our work focuses on a “limited consumer features regime” and it is of interest to extend our framework to a setting where consumers can have high-dimensional features (and the firm can observe such information). As noted in Remark 3.1, works such as Li et al. 2010; Agrawal and Goyal 2013, and Tang et al. 2013 handle high-dimensional consumer features but they optimize for a myopic reward function. There appears to be some progress in modeling high-dimensional contextual information along with long-run rewards (see Hallak et al. 2015). We believe the idea of “value function approximation” from the reinforcement learning literature (Sutton et al. 2000; Sutton and Barto 2018) can be leveraged here, especially if the value function is parameterized using domain expertise from marketing. Such approximations can allow efficient learning across consumers.

From a theoretical perspective, it is important to understand the properties such as regret and convergence rate of our algorithm (or its variants). Given the connection of our algorithm to Q-learning (which we used while proving its asymptotic convergence), recent works such as Wainwright 2019a; Wainwright 2019b; Qu and Wierman 2020, and Li et al. 2020 that establish theoretical guarantees for Q-learning can possibly be leveraged. Furthermore, as touched upon at the beginning of Section 3.6, the ultimate test to evaluate any proposal is to perform a real-life experiment. Though we do not have such decision-making control yet, we hope to follow-up with an empirical study.

We conclude this chapter with some thoughts on the connection between attribution (Chapter 2) and optimizing actions. The goal in attribution is to compute the return-on-investment of various advertising actions. Therefore, it is natural to ask whether attribution can be utilized to construct an improved advertising policy: an advertising action that is attributed “high” value should be used “more” often. In fact, from our interaction with the industry, such a policy iteration is one of the desired property for any attribution scheme. However, from a theoretical perspective, the mapping from attribution to prescription is not as straightforward. In the language of learning theory, attribution is a form of “policy evaluation” (given an existing advertising policy) and using attribution to increase the conversion probability corresponds to “policy improvement”. For attribution to serve as an input to a policy improvement algorithm, it needs to point in the direction of the gradient corresponding to the underlying objective function (maximize conversion probability). However, to the best of our knowledge, we are not aware of such a connection. We leave this as an open question and hope to pursue it in the future.

## Concluding Remarks

We leveraged the granularity of data available in the new economy to develop application-driven models, leading to better understanding and decision-making in complex systems. In daily fantasy sports, we used data on athletes and opponents to construct strategic portfolios and in online advertising, we used user-level data to understand the value of a targeted ad and propose a decision-making algorithm for sequential personalized marketing. In addition to our thoughts at the end of each chapter, we briefly highlight some broader avenues of research that we are pursuing.

**Fairness and pricing dynamics in ridesharing.** We are developing models for the ridesharing industry with the goal of improving driver welfare. Though the rise of the gig economy has given drivers flexibility, “fairness” to drivers has been debated. With platforms being able to track local supply and demand conditions, dynamic policies such as turning a driver off under demand surplus has become the norm. Such profit-maximizing policies often result in a high uncertainty for drivers, forcing them to drive without passengers for extended periods and earn low wages. Using data on temporal preferences of drivers, we are developing models that increase driver welfare without hurting platform profit. In addition, motivated by a real-world dataset on trip price quotes of major platforms, we are developing models to understand the temporal pricing dynamics under competition.

**Experimentation in two-sided markets.** With access to a user-level dataset corresponding to a music streaming app, we are exploring optimal ways to experiment in two-sided markets. To maximize revenue in such settings, the platform needs to satisfy the two sides: content creators and viewers. Given a newly created content, a key operational decision is how to experiment in order to learn the quality of the content. In order to retain viewers, the platform wants to show “high” quality content to more viewers and discard “low” quality content without exposing it to too many viewers. Using the granular dataset, we are developing models to understand optimal experimentation policies as a function of the “state” of the viewer.

**Hot-hand in sports.** Finally, we are developing a Bayesian statistical framework to test for hot-hand with a rigorous characterization of its statistical power. We plan to test our framework on a point-level tennis dataset, consisting of over 25,000 matches between 2011 and 2018.

Given the rise of the new economy, we believe this is an exciting time to pursue data-driven research in operations, capturing aspects of reality that were previously not possible. In addition to providing academic insights, we hope our models ultimately lead to a better understanding of complex systems and equip practitioners with improved decision-making capabilities.



## References

- Abe, Naoki et al. (2002). “Empirical comparison of various reinforcement learning strategies for sequential targeted marketing”. In: *2002 IEEE International Conference on Data Mining, 2002. Proceedings*. IEEE, pp. 3–10.
- Abebe, Rediet, Lada A Adamic, and Jon Kleinberg (2018). “Mitigating overexposure in viral marketing”. In: *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Abhishek, Vibhanshu, Peter Fader, and Kartik Hosanagar (2012). “Media exposure through the funnel: A model of multi-stage attribution”. In: Working paper, available at: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2158421](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2158421).
- Abhishek, Vibhanshu, Stylianos Despotakis, and R Ravi (2017). “Multi-Channel Attribution: The Blind Spot of Online Advertising”. In: Working paper, available at: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2959778](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2959778).
- AboElHamd, Eman, Hamed M Shamma, and Mohamed Saleh (2020). “Maximizing Customer Lifetime Value Using Dynamic Programming: Theoretical and Practical Implications”. In: *Academy of Marketing Studies Journal* 24.1.
- Agrawal, Shipra and Navin Goyal (2012). “Analysis of thompson sampling for the multi-armed bandit problem”. In: *Conference on learning theory*, pp. 39–1.
- (2013). “Thompson sampling for contextual bandits with linear payoffs”. In: *International Conference on Machine Learning*, pp. 127–135.
- Agrawal, Shipra and Randy Jia (2017). “Optimistic posterior sampling for reinforcement learning: worst-case regret bounds”. In: *Advances in Neural Information Processing Systems*, pp. 1184–1194.
- Anderl, Eva et al. (2016). “Mapping the customer journey: Lessons learned from graph-based online attribution modeling”. In: *International Journal of Research in Marketing* 33.3, pp. 457–474.
- Anderton, Kevin (2016). *FanDuel And DraftKings Are Dominating The Daily Fantasy Sports Market*. URL: <https://www.forbes.com/sites/kevinanderton/2016/11/30/fanduel-and-draftkings-are-dominating-the-daily-fantasy-sports-market-infographic/#2979acb87c4f>.
- Ansari, Asim and Carl F Mela (2003). “E-customization”. In: *Journal of marketing research* 40.2, pp. 131–145.

- Arava, Sai Kumar et al. (2018). “Deep Neural Net with Attention for Multi-channel Multi-touch Attribution”. In: Working paper, available at: <https://arxiv.org/abs/1809.02230>.
- Archak, Nikolay, Vahab S Mirrokni, and S Muthukrishnan (2010). “Mining advertiser-specific user behavior using adfactors”. In: *Proceedings of the 19th International Conference on World Wide Web*. ACM, pp. 31–40.
- Arensman, Joan and Wilfred Yeung (2016). *Move beyond last click attribution in AdWords*. URL: <https://adwords.googleblog.com/2016/05/move-beyond-last-click-attribution.html>.
- Ariely, Dan, Gabriel Bitran, et al. (2013). “Design to learn: Customizing services when the future matters”. In: *Pesquisa Operacional* 33.1, pp. 37–61.
- Asmussen, Søren and Peter W Glynn (2007). *Stochastic simulation: algorithms and analysis*. Vol. 57. Springer Science & Business Media.
- Avrachenkov, Konstantin, Laura Cottatellucci, and Lorenzo Maggi (2012). “Confidence intervals for Shapley value in Markovian dynamic games”. In: Technical report (Eurecom), available at: <http://www.eurecom.fr/publication/3602>.
- Baardman, Lennart et al. (2019). “Learning Optimal Online Advertising Portfolios with Periodic Budgets”. In: Working paper, available at: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3346642](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3346642).
- Baehrens, David et al. (2010). “How to explain individual classification decisions”. In: *Journal of Machine Learning Research* 11.Jun, pp. 1803–1831.
- Bales, Jon (2016). *Pairing a QB with his Receiver(s)*. URL: <https://rotogrinders.com/articles/pairing-a-qb-with-his-receiver-s-481544>.
- Balseiro, Santiago and Yonatan Gur (2017). “Learning in repeated auctions with budgets: Regret minimization and equilibrium”. In: Working paper, available at: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2921446](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2921446).
- Balseiro, Santiago R et al. (2014). “Yield optimization of display advertising with ad exchange”. In: *Management Science* 60.12, pp. 2886–2907.
- Balseiro, Santiago R, Omar Besbes, and Gabriel Y Weintraub (2015). “Repeated auctions with budgets in ad exchanges: Approximations and design”. In: *Management Science* 61.4, pp. 864–884.
- Barry, Thomas E (1987). “The development of the hierarchy of effects: An historical perspective”. In: *Current issues and Research in Advertising* 10.1-2, pp. 251–295.

- Bayraktar, Erhan and Alexander Munk (2017). “High-roller impact: a large generalized game model of parimutuel wagering”. In: *Market Microstructure and Liquidity* 3.01, p. 1750006.
- Becker, Adrian and Xu Andy Sun (2016). “An analytical approach for fantasy football draft and lineup management”. In: *Journal of Quantitative Analysis in Sports* 12.1, pp. 17–30.
- Bellman, Richard (1954). *The theory of dynamic programming*. Tech. rep. Rand corp santa monica ca.
- Bergman, David and Jason Imbrogno (2017). “Surviving a national football league survivor pool”. In: *Operations Research* 65.5, pp. 1343–1354.
- Berman, Ron (2018). “Beyond the last touch: Attribution in online advertising”. In: *Marketing Science* 37.5, pp. 771–792.
- Bertsekas, Dimitri P (1995). *Dynamic programming and optimal control*. Vol. 1. 2. Athena scientific Belmont, MA.
- Bertsimas, Dimitris and Adam J Mersereau (2007). “A learning approach for interactive marketing to a customer segment”. In: *Operations Research* 55.6, pp. 1120–1135.
- Bettman, James R, Mary Frances Luce, and John W Payne (1998). “Constructive consumer choice processes”. In: *Journal of Consumer Research* 25.3, pp. 187–217.
- Bhandari, Jalaj, Daniel Russo, and Raghav Singal (2018). “A Finite Time Analysis of Temporal Difference Learning With Linear Function Approximation”. In: *Conference On Learning Theory*, pp. 1691–1692.
- Bitran, Gabriel R and Susana V Mondschein (1996). “Mailing decisions in the catalog sales industry”. In: *Management science* 42.9, pp. 1364–1381.
- Blake, Thomas, Chris Nosko, and Steven Tadelis (2015). “Consumer heterogeneity and paid search effectiveness: A large-scale field experiment”. In: *Econometrica* 83.1, pp. 155–174.
- Bleier, Alexander and Maik Eisenbeiss (2015). “Personalized online advertising effectiveness: The interplay of what, when, and where”. In: *Marketing Science* 34.5, pp. 669–688.
- Bottou, Léon et al. (2013). “Counterfactual reasoning and learning systems: The example of computational advertising”. In: *The Journal of Machine Learning Research* 14.1, pp. 3207–3260.
- Braun, Michael and Wendy W Moe (2013). “Online display advertising: Modeling the effects of multiple creatives and individual impression histories”. In: *Marketing Science* 32.5, pp. 753–767.

- Brown, Larry (2016). *DraftKings investigating potential collusion in millionaire contest*. URL: <http://larrybrownsports.com/sports-business/draftkings-investigating-collusion-millionaire-contest/325182>.
- Bruce, Norris I, Kay Peters, and Prasad A Naik (2012). “Discovering how advertising grows sales and builds brands”. In: *Journal of Marketing Research* 49.6, pp. 793–806.
- Bruce, Norris I, BPS Murthi, and Ram C Rao (2017). “A dynamic model for digital advertising: The effects of creative format, message content, and targeting on engagement”. In: *Journal of marketing research* 54.2, pp. 202–218.
- Bult, Jan Roelf and Tom Wansbeek (1995). “Optimal selection for direct mail”. In: *Marketing Science* 14.4, pp. 378–394.
- Byers, John W, Michael Mitzenmacher, and Georgios Zervas (2012). “The groupon effect on yelp ratings: a root cause analysis”. In: *Proceedings of the 13th ACM conference on electronic commerce*, pp. 248–265.
- CampaignMonitor (2020). *Ultimate Email Marketing Benchmarks for 2020: By Industry and Day*. URL: <https://www.campaignmonitor.com/resources/guides/email-marketing-benchmarks/>.
- Cao, Junyu, Wei Sun, and Zuo-Jun Max Shen (2019). “Sequential choice bandits: Learning with marketing fatigue”. In: *Available at SSRN 3355211*.
- Castro, Javier, Daniel Gómez, and Juan Tejada (2009). “Polynomial calculation of the Shapley value based on sampling”. In: *Computers & Operations Research* 36.5, pp. 1726–1730.
- Chapelle, Olivier and Lihong Li (2011). “An empirical evaluation of thompson sampling”. In: *Advances in neural information processing systems*, pp. 2249–2257.
- Chatterjee, Patrali, Donna L Hoffman, and Thomas P Novak (2003). “Modeling the clickstream: Implications for web-based advertising efforts”. In: *Marketing Science* 22.4, pp. 520–541.
- Cheng, Jiesi, Aaron Sun, and Daniel Zeng (2010). “Information overload and viral marketing: countermeasures and strategies”. In: *International Conference on Social Computing, Behavioral Modeling, and Prediction*. Springer, pp. 108–117.
- Chockler, Hana and Joseph Y Halpern (2004). “Responsibility and blame: A structural-model approach”. In: *Journal of Artificial Intelligence Research* 22, pp. 93–115.
- Choi, Hana et al. (2017). “Online Display Advertising Markets: A Literature Review and Future Directions”. In: Working paper, available at: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3070706](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3070706).

- Clair, Bryan and David Letscher (2007). “Optimal strategies for sports betting pools”. In: *Operations Research* 55.6, pp. 1163–1177.
- Collins, John David et al. (2004). *Causation and counterfactuals*. MIT Press.
- Cook, John D (2010). *Determining distribution parameters from quantiles*. URL: [https://www.johndcook.com/quantiles\\_parameters.pdf](https://www.johndcook.com/quantiles_parameters.pdf).
- Court, David (2009). “The consumer decision journey”. In: *McKinsey Quarterly*.
- Dalessandro, Brian et al. (2012). “Causally motivated attribution for online advertising”. In: *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy*. ACM, p. 7.
- Danaher, Peter J and Harald J van Heerde (2018). “Delusion in Attribution: Caveats in Using Attribution for Multimedia Budget Allocation”. In: *Journal of Marketing Research* 55.5, pp. 667–685.
- David, Herbert Aron and Haikady Navada Nagaraja (2004). “Order statistics”. In: *Encyclopedia of Statistical Sciences*.
- Davis, Alexander (2017). *Data-Driven Portfolios Power ‘Home-Run’ Exits in MIT Study*. URL: <https://www.wsj.com/articles/data-driven-portfolios-power-home-run-exits-in-mit-study-1502105402>.
- Dearden, Richard, Nir Friedman, and Stuart Russell (1998). “Bayesian Q-learning”. In: *Aaai/iaai*, pp. 761–768.
- Deisenroth, Marc and Carl E Rasmussen (2011). “PILCO: A model-based and data-efficient approach to policy search”. In: *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pp. 465–472.
- Dhamdhere, Kedar, Mukund Sundararajan, and Qiqi Yan (2018). “How important is a neuron?” In: Working paper, available at: <https://arxiv.org/abs/1805.12233>.
- Dong, Kefan et al. (2019). “Q-learning with ucb exploration is sample efficient for infinite-horizon mdp”. In: *arXiv preprint arXiv:1901.09311*.
- Drape, Joe and Jacqueline Williams (2015a). *In Fantasy Sports, Signs of Insiders’ Edge*. URL: <https://www.nytimes.com/2015/10/12/sports/fantasy-sports-draftkings-fanduel-insiders-edge-football.html>.
- (2015b). *Scandal Erupts in Unregulated World of Fantasy Sports*. URL: <https://www.nytimes.com/2015/10/06/sports/fanduel-draftkings-fantasy-employees-bet-rivals.html>.

- Eells, Ellery (1991). *Probabilistic causality*. Vol. 1. Cambridge University Press.
- Elzinga, Dave, Susan Mulder, Ole Jorgen Vetvik, et al. (2009). “The consumer decision journey”. In: *McKinsey Quarterly* 3, pp. 96–107.
- FanDuel (2016). *Rules & Scoring*. URL: <https://www.fanduel.com/rules>.
- FantasyPros (2017). URL: <https://www.fantasypros.com>.
- Fatima, Shaheen S, Michael Wooldridge, and Nicholas R Jennings (2008). “A linear approximation method for the Shapley value”. In: *Artificial Intelligence* 172.14, pp. 1673–1699.
- Feit, Elea McDonnell and Ron Berman (2019). “Test & Roll: Profit-Maximizing A/B Tests”. In: *Marketing Science* 38.6, pp. 1038–1058.
- Fry, Michael J, Andrew W Lundberg, and Jeffrey W Ohlmann (2007). “A player selection heuristic for a sports league draft”. In: *Journal of Quantitative Analysis in Sports* 3.2.
- FSTA (2015). *Industry demographics*. URL: <http://fsta.org/research/industry-demographics/>.
- Gelman, Andrew et al. (2013). *Bayesian data analysis*. CRC press.
- Ghose, Anindya and Vilma Todri (2015). “Towards a digital attribution model: Measuring the impact of display advertising on online consumer behavior. Forthcoming”. In: *MIS quarterly*.
- Gibbs, Jacob (2017). *Week 1 FanDuel NFL Tournament Pivots*. URL: <https://www.numberfire.com/nfl/lists/16195/week-1-fanduel-nfl-tournament-pivots>.
- Gönül, Füsün and Meng Ze Shi (1998). “Optimal mailing of catalogs: A new methodology using estimable structural dynamic programming models”. In: *Management Science* 44.9, pp. 1249–1262.
- Google (2019). *MCF Data-Driven Attribution methodology*. URL: <https://support.google.com/analytics/answer/3191594?hl=en>.
- Gooley, Christopher G and James M Lattin (2000). *Dynamic customization of marketing messages in interactive media*. Graduate School of Business, Stanford University.
- Grinstead, Charles Miller and James Laurie Snell (2012). *Introduction to probability*. American Mathematical Society.
- Gurobi Optimization, LLC (2016). *Gurobi Optimizer Reference Manual*. URL: <http://www.gurobi.com>.

- Hagan, Shelly (2018). *Digital economy has been growing at triple the pace of U.S. GDP*. URL: <https://www.bloomberg.com/news/articles/2018-03-15/digital-economy-has-been-growing-at-triple-the-pace-of-u-s-gdp>.
- Hallak, Assaf, Dotan Di-Castro, and Shie Mannor (2013). “Model selection in markovian processes”. In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 374–382.
- (2015). “Contextual markov decision processes”. In: *arXiv preprint arXiv:1502.02259*.
- Halpern, Joseph Y and Judea Pearl (2005). “Causes and explanations: A structural-model approach. Part I: Causes”. In: *The British Journal for the Philosophy of Science* 56.4, pp. 843–887.
- Harwell, Drew (2015). *Why you (probably) won't win money playing DraftKings, FanDuel*. URL: <http://www.dailyherald.com/article/20151012/business/151019683/>.
- Haugh, Martin B and Raghav Singal (2020). “How to play fantasy sports strategically (and win)”. In: *Management Science*.
- Hauser, John R et al. (2009). “Website morphing”. In: *Marketing Science* 28.2, pp. 202–223.
- Hauser, John R, Guilherme Liberali, and Glen L Urban (2014). “Website morphing 2.0: Switching costs, partial exposure, random exit, and when to morph”. In: *Management science* 60.6, pp. 1594–1616.
- Hitchcock, Christopher (1997). “Probabilistic causation.” In: URL: <https://plato.stanford.edu/entries/causation-probabilistic/>.
- Hojjat, Ali et al. (2017). “A unified framework for the scheduling of guaranteed targeted display advertising under reach and frequency requirements”. In: *Operations Research* 65.2, pp. 289–313.
- Howard, John A and Jagdish N Sheth (1969). *The theory of buyer behavior*. John Wiley & Sons.
- Howard, Ronald A (2002). “Comments on the origin and application of Markov decision processes”. In: *Operations Research* 50.1, pp. 100–102.
- Hume, David (2003). *A treatise of human nature*. Courier Corporation.
- Hunter, David Scott, Juan Pablo Vielma, and Tauhid Zaman (2016). “Picking winners using integer programming”. In: Working paper, available at: <https://arxiv.org/abs/1604.01455>.

- IBM (2020). *The Four V's of Big Data*. URL: <https://www.ibmbigdatahub.com/infographic/four-vs-big-data>.
- Institute, Marketing Science (2016). *Research Priorities 2016-2018*. URL: [https://www.msi.org/uploads/articles/MSI\\_RP16-18.pdf](https://www.msi.org/uploads/articles/MSI_RP16-18.pdf).
- (2018). *Research Priorities 2018-2020*. URL: [https://www.msi.org/wp-content/uploads/2020/06/MSI\\_RP18-20.pdf](https://www.msi.org/wp-content/uploads/2020/06/MSI_RP18-20.pdf).
- (2020). *Research Priorities 2020-2022*. URL: [https://www.msi.org/wp-content/uploads/2020/06/MSI\\_RP20-22.pdf](https://www.msi.org/wp-content/uploads/2020/06/MSI_RP20-22.pdf).
- Iyer, Krishnamurthy, Ramesh Johari, and Mukund Sundararajan (2014). “Mean field equilibria of dynamic auctions with learning”. In: *Management Science* 60.12, pp. 2949–2970.
- Jansen, Bernard J and Simone Schuster (2011). “Bidding on the buying funnel for sponsored search and keyword advertising”. In: *Journal of Electronic Commerce Research* 12.1, p. 1.
- Ji, Wendi and Xiaoling Wang (2017). “Additional Multi-Touch Attribution for Online Advertising”. In: *AAAI*, pp. 1360–1366.
- Ji, Wendi, Xiaoling Wang, and Dell Zhang (2016). “A probabilistic multi-touch attribution model for online advertising”. In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, pp. 1373–1382.
- Jin, Chi et al. (2018). “Is Q-learning provably efficient?” In: *Advances in Neural Information Processing Systems*, pp. 4863–4873.
- Johnson, Alex (2016). *After Ferocious Battle, New York Legalizes DraftKings, FanDuel*. URL: <https://www.nbcnews.com/news/us-news/after-ferocious-battle-new-york-legalizes-draftkings-fanduel-n622606>.
- Jordan, Patrick et al. (2011). “The multiple attribution problem in pay-per-conversion advertising”. In: *International Symposium on Algorithmic Game Theory*. Springer, pp. 31–43.
- Kakalejčik, Lukáš et al. (2018). “Multichannel Marketing Attribution Using Markov Chains”. In: *Journal of Applied Management and Investments* 7.1, pp. 49–60.
- Kannan, PK, Werner Reinartz, and Peter C Verhoef (2016). “The path to purchase and attribution modeling: Introduction to special section”. In: *International Journal of Research in Marketing* 33.3, pp. 449–456.
- Kaplan, Edward H and Stanley J Garstka (2001). “March madness and the office pool”. In: *Management Science* 47.3, pp. 369–382.



- Kireyev, Pavel, Koen Pauwels, and Sunil Gupta (2016). “Do display ads influence search? Attribution and dynamics in online advertising”. In: *International Journal of Research in Marketing* 33.3, pp. 475–490.
- Kolodny, Lora (2015). *Fantasy Sports Create Billion-Dollar Startups*. URL: <https://www.wsj.com/articles/fantasy-sports-create-billion-dollar-startups-1436846402>.
- Kotler, Philip and Gary Armstrong (2010). *Principles of marketing*. Pearson Education.
- Leino, Klas et al. (2018). “Influence-directed explanations for deep convolutional networks”. In: *2018 IEEE International Test Conference (ITC)*. IEEE, pp. 1–8.
- Lejeune, Miguel and John Turner (2019). “Planning online advertising using Lorenz curves”. In: *Operations Research (In press)*.
- Li, Gen et al. (2020). “Sample Complexity of Asynchronous Q-Learning: Sharper Analysis and Variance Reduction”. In: *arXiv preprint arXiv:2006.03041*.
- Li, Hongshuang and PK Kannan (2014). “Attributing conversions in a multichannel online marketing environment: An empirical model and a field experiment”. In: *Journal of Marketing Research* 51.1, pp. 40–56.
- Li, Lihong et al. (2010). “A contextual-bandit approach to personalized news article recommendation”. In: *Proceedings of the 19th international conference on World wide web*, pp. 661–670.
- Li, Yiyi, Ying Xie, and Eric Zheng (2017). “Modeling Multi-Channel Advertising Attribution Across Competitors”. In: Working paper, available at: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3047981](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3047981).
- Liben-Nowell, David et al. (2012). “Computing Shapley value in supermodular coalitional games”. In: *International Computing and Combinatorics Conference*. Springer, pp. 568–579.
- Liberali, Gui and Alina Ferecatu (2019). “Morphing Consumer Dynamics: Bandits Meet HMM”. In: *Available at SSRN 3495518*.
- Littlechild, Stephen C and Guillermo Owen (1973). “A simple expression for the Shapley value in a special case”. In: *Management Science* 20.3, pp. 370–372.
- Lundberg, Scott M and Su-In Lee (2017). “A unified approach to interpreting model predictions”. In: *Advances in Neural Information Processing Systems*, pp. 4765–4774.

- Ma, Shaohui et al. (2016). “A nonhomogeneous hidden Markov model of response dynamics and mailing optimization in direct marketing”. In: *European Journal of Operational Research* 253.2, pp. 514–523.
- Maleki, Sasan et al. (2013). “Bounding the estimation error of sampling-based Shapley value approximation”. In: Working paper, available at: <https://arxiv.org/abs/1306.4265>.
- Manchanda, Puneet et al. (2006). “The effect of banner advertising on internet purchasing”. In: *Journal of Marketing Research* 43.1, pp. 98–108.
- Michalak, Tomasz P et al. (2013). “Efficient computation of the Shapley value for game-theoretic network centrality”. In: *Journal of Artificial Intelligence Research* 46, pp. 607–650.
- Miller, Ed and Daniel Singer (2015). *For daily fantasy-sports operators, the curse of too much skill*. URL: <https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/for-daily-fantasy-sports-operators-the-curse-of-too-much-skill#>.
- Mišić, Velibor V and Georgia Perakis (2020). “Data analytics in operations management: A review”. In: *Manufacturing & Service Operations Management* 22.1, pp. 158–169.
- Moazeni, Somayeh, Boris Defourny, and Monika J Wilczak (2019). “Sequential Learning in Designing Marketing Campaigns for Market Entry”. In: *Management Science*.
- Montoya, Ricardo, Oded Netzer, and Kamel Jedidi (2010). “Dynamic allocation of pharmaceutical detailing and sampling for long-term profitability”. In: *Marketing Science* 29.5, pp. 909–924.
- Morgan, Stephen L and Christopher Winship (2014). *Counterfactuals and causal inference*. Cambridge University Press.
- Morton, David P et al. (2003). “Optimizing benchmark-based utility functions”. In: *Bulletin of the Czech Econometric Society* 10.18.
- Mulpuru, Sucharita (2011). “The purchase path of online buyers”. In: *Forrester Report* 5, p. 12.
- Mulshine, Molly (2015). *How one man made hundreds of thousands of dollars playing daily fantasy sports*. URL: <http://www.businessinsider.com/draft-kings-jonathan-bales-a-day-in-the-life-2015-10>.
- Nelsen, Roger B (2007). *An introduction to copulas*. Springer Science & Business Media.
- Nemhauser, George L, Laurence A Wolsey, and Marshall L Fisher (1978). “An analysis of approximations for maximizing submodular set functions—I”. In: *Mathematical programming* 14.1, pp. 265–294.

- Nickish, Curt (2015). *Meet A Bostonian Who's Made \$3 Million This Year Playing Daily Fantasy Sports*. URL: <http://www.wbur.org/news/2015/11/23/dfs-power-player-profile>.
- O’Keeffe, Kate (2015). *Daily Fantasy-Sports Operators Await Reality Check*. URL: <https://www.wsj.com/articles/daily-fantasy-sports-operators-await-reality-check-1441835630>.
- Osband, Ian and Benjamin Van Roy (2017). “Why is posterior sampling better than optimism for reinforcement learning?” In: *International Conference on Machine Learning*, pp. 2701–2710.
- Osband, Ian, Daniel Russo, and Benjamin Van Roy (2013). “(More) efficient reinforcement learning via posterior sampling”. In: *Advances in Neural Information Processing Systems*, pp. 3003–3011.
- Osband, Ian et al. (2019). “Deep Exploration via Randomized Value Functions.” In: *Journal of Machine Learning Research* 20.124, pp. 1–62.
- Owen, Guillermo (1972). “Multilinear extensions of games”. In: *Management Science* 18.5-part-2, pp. 64–79.
- Pearl, Judea (2009). *Causality*. Cambridge University Press.
- Pednault, Edwin, Naoki Abe, and Bianca Zadrozny (2002). “Sequential cost-sensitive decision making with reinforcement learning”. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 259–268.
- Plott, Charles R, Jorgen Wit, and Winston C Yang (2003). “Parimutuel betting markets as information aggregation devices: experimental results”. In: *Economic Theory* 22.2, pp. 311–351.
- Pramuk, Jacob (2015). *Former pro poker player makes a living on fantasy sports*. URL: <http://www.cnbc.com/2015/10/06/former-pro-poker-player-makes-a-living-on-fantasy-sports.html>.
- Priest, Colin (2017). “Multichannel Marketing Attribution with DataRobot.” In: URL: <https://www.datarobot.com/resource/mktattribution/>.
- Puterman, Martin L (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.
- Qu, Guannan and Adam Wierman (2020). “Finite-Time Analysis of Asynchronous Stochastic Approximation and  $Q$ -Learning”. In: *arXiv preprint arXiv:2002.00260*.

- Quantcast (2013). “Beyond Last Touch: Understanding Campaign Effectiveness.” In: URL: [http://info.quantcast.com/rs/quantcast/images/Quantcast%20White%20Paper\\_Beyond%20Last%20Touch.pdf](http://info.quantcast.com/rs/quantcast/images/Quantcast%20White%20Paper_Beyond%20Last%20Touch.pdf).
- (2016). “Guide to marketing attribution.” In: URL: <https://www.iabuk.com/sites/default/files/case-study-docs/Quantcast%20-%20Guide%20to%20Marketing%20Attribution%20-%20EN%20-%202016%20-%20print.pdf>.
- Reagan, Brad (2016). *DraftKings Investigating Potential Collusion in \$1 Million Contest*. URL: <https://www.wsj.com/articles/draftkings-investigating-potential-collusion-in-1-million-contest-1475091553>.
- ReportLinker (2020). *Global Email Marketing Industry*. URL: [https://www.reportlinker.com/p05900877/Global-Email-Marketing-Industry.html?utm\\_source=GNW](https://www.reportlinker.com/p05900877/Global-Email-Marketing-Industry.html?utm_source=GNW).
- Rossi, Peter E, Robert E McCulloch, and Greg M Allenby (1996). “The value of purchase history data in target marketing”. In: *Marketing Science* 15.4, pp. 321–340.
- RotoViz (2017). URL: <http://rotoviz.com>.
- Rubin, Donald B (1974). “Estimating causal effects of treatments in randomized and nonrandomized studies.” In: *Journal of Educational Psychology* 66.5, p. 688.
- Ryzhov, Ilya O et al. (2019). “Bayesian exploration for approximate dynamic programming”. In: *Operations research* 67.1, pp. 198–214.
- Sahni, Navdeep S (2015). “Effect of temporal spacing between advertising exposures: Evidence from online field experiments”. In: *Quantitative Marketing and Economics* 13.3, pp. 203–247.
- Sahni, Navdeep S, Sridhar Narayanan, and Kirthi Kalyanam (2019). “An experimental investigation of the effects of retargeted advertising: The role of frequency and timing”. In: *Journal of Marketing Research* 56.3, pp. 401–418.
- Schrijver, Alexander (2003). *Combinatorial optimization: polyhedra and efficiency*. Vol. 24. Springer Science & Business Media.
- Schulman, John et al. (2015). “Trust region policy optimization”. In: *International conference on machine learning*, pp. 1889–1897.
- Schwartz, Eric M, Eric T Bradlow, and Peter S Fader (2017). “Customer acquisition via display advertising using multi-armed bandit experiments”. In: *Marketing Science* 36.4, pp. 500–522.
- Schweidel, David A, Eric T Bradlow, and Peter S Fader (2011). “Portfolio dynamics for customers of a multiservice provider”. In: *Management Science* 57.3, pp. 471–486.

- Shao, Xuhui and Lexin Li (2011). “Data-driven multi-touch attribution models”. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 258–264.
- Shapley, Lloyd S (1953). “A value for n-person games”. In: *Contributions to the Theory of Games* 2.28, pp. 307–317.
- Simester, Duncan, Artem Timoshenko, and Spyros I Zoumpoulis (2020). “Targeting prospective customers: Robustness of machine-learning methods to typical data challenges”. In: *Management Science* 66.6, pp. 2495–2522.
- Simester, Duncan I, Peng Sun, and John N Tsitsiklis (2006). “Dynamic catalog mailing policies”. In: *Management science* 52.5, pp. 683–696.
- Singal, Raghav et al. (2019). “Shapley Meets Uniform: An Axiomatic Framework for Attribution in Online Advertising”. In: *The World Wide Web Conference*, pp. 1713–1723.
- Sinha, Indrajit and Thomas Foscht (2007). “Over-marketing and brand suicide”. In: *Reverse Psychology Marketing*. Springer, pp. 23–50.
- Sklar, M (1959). “Fonctions de repartition an dimensions et leurs marges”. In: *Publ. inst. statist. univ. Paris* 8, pp. 229–231.
- Sluis, Sarah (2018). *Digital ad market soars to \$88 billion, Facebook and Google contribute 90% of growth*. URL: <https://adexchanger.com/online-advertising/digital-ad-market-soars-to-88-billion-facebook-and-google-contribute-90-of-growth/>.
- Strens, Malcolm (2000). “A Bayesian framework for reinforcement learning”. In: *ICML*. Vol. 2000, pp. 943–950.
- Strong, Edward Kellogg (1925). *The psychology of selling and advertising*. McGraw-Hill Book Company, Incorporated.
- Sundararajan, Mukund, Ankur Taly, and Qiqi Yan (2017). “Axiomatic attribution for deep networks”. In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70, pp. 3319–3328.
- Sutton, Richard S and Andrew G Barto (2018). *Reinforcement learning: An introduction*. MIT press.
- Sutton, Richard S et al. (2000). “Policy gradient methods for reinforcement learning with function approximation”. In: *Advances in neural information processing systems*, pp. 1057–1063.

- Tang, Liang et al. (2013). “Automatic ad format selection via contextual bandits”. In: *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pp. 1587–1594.
- Team, Stan Development (2017). *RStan: the R interface to Stan*. URL: <http://mc-stan.org>.
- Terrell, Dek and Amy Farmer (1996). “Optimal betting and efficiency in parimutuel betting markets with information costs”. In: *The Economic Journal* 106.437, pp. 846–868.
- Thaler, Richard H and William T Ziemba (1988). “Anomalies: Parimutuel betting markets: Race-tracks and lotteries”. In: *Journal of Economic perspectives* 2.2, pp. 161–174.
- Theocharous, Georgios, Philip S Thomas, and Mohammad Ghavamzadeh (2015). “Personalized ad recommendation systems for life-time value optimization with guarantees”. In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*.
- Thompson, William R (1933). “On the likelihood that one unknown probability exceeds another in view of the evidence of two samples”. In: *Biometrika* 25.3/4, pp. 285–294.
- Tkachenko, Yegor (2015). “Autonomous CRM control via CLV approximation with deep reinforcement learning in discrete and continuous action space”. In: *arXiv preprint arXiv:1504.01840*.
- Tsitsiklis, John N (1994). “Asynchronous stochastic approximation and Q-learning”. In: *Machine learning* 16.3, pp. 185–202.
- UNCTAD (2019). *Digital economy report*. URL: [https://unctad.org/en/PublicationsLibrary/der2019\\_overview\\_en.pdf](https://unctad.org/en/PublicationsLibrary/der2019_overview_en.pdf).
- United, Econsultancy Digital Marketers (2012). “Quarterly Digital Intelligence Briefing: Making Sense of Marketing Attribution (in association with Adobe).” In: URL: <http://success.adobe.com/assets/en/downloads/whitepaper/Adobe-Quarterly-Digital-Intelligence-Briefing-Digital-Trends-for-2013.pdf>.
- Urban, Glen L et al. (2014). “Morphing banner advertising”. In: *Marketing Science* 33.1, pp. 27–46.
- Wainwright, Martin J (2019a). “Stochastic approximation with cone-contractive operators: Sharp  $\ell_\infty$ -bounds for Q-learning”. In: *arXiv preprint arXiv:1905.06265*.
- (2019b). “Variance-reduced Q-learning is minimax optimal”. In: *arXiv preprint arXiv:1906.04697*.
- Watkins, Christopher JCH and Peter Dayan (1992). “Q-learning”. In: *Machine learning* 8.3-4, pp. 279–292.

- Wong, Kristin (2015). *The Fantasy Sports Industry, by the Numbers*. URL: <http://www.nbcnews.com/business/business-news/fantasy-sports-industry-numbers-n439536>.
- Woodward, Curt (2015). *Top fantasy sports player uses software, analytics to reap millions*. URL: <https://www.bostonglobe.com/business/2015/12/23/from-boston-penthouse-world-best-fantasy-player-plunges-into-startup-world/QHNpLh003QMyUDqTd4t27N/story.html>.
- (2016). *DraftKings, FanDuel collected \$3b in entry fees last year, analyst says*. URL: <https://www.bostonglobe.com/business/2016/02/29/fantasy-sports-industry-hits-amid-legal-questions-analyst-says/NKw364kiLjv8XcD54vRr4H/story.html>.
- Xu, Lizhen, Jason A Duan, and Andrew Whinston (2014). “Path to purchase: A mutually exciting point process model for online advertising and conversion”. In: *Management Science* 60.6, pp. 1392–1412.
- Yao, Xinlin and Xianghua Lu (2019). “Optimizing Digital Coupon Assignment Using Constrained Reinforcement Learning”. In: *Proceedings of the 3rd International Conference on Machine Learning and Soft Computing*, pp. 143–147.
- Zantedeschi, Daniel, Eleanor McDonnell Feit, and Eric T Bradlow (2017). “Measuring multichannel advertising response”. In: *Management Science* 63.8, pp. 2706–2728.
- Zhang, Jonathan Z et al. (2016). “Dynamic relationship marketing”. In: *Journal of Marketing* 80.5, pp. 53–75.
- Zhang, Xi, V Kumar, and Koray Cosguner (2017). “Dynamically managing a profitable email marketing program”. In: *Journal of marketing research* 54.6, pp. 851–866.
- Zhang, Ya, Yi Wei, and Jianbiao Ren (2014). “Multi-touch attribution in online advertising with survival theory”. In: *2014 IEEE International Conference on Data Mining (ICDM)*. IEEE, pp. 687–696.
- Zhao, Kaifeng, Seyed Hanif Mahboobi, and Saeed R Bagheri (2018). “Revenue-based attribution modeling for online advertising”. In: *International Journal of Market Research* 61.2, pp. 195–209.

# Appendix A: Additional Details for Chapter 1

## A.1 Further Technical Details

### A.1.1 Details for Modeling Opponents' Team Selections

#### *The Copula*

Determining which copula  $C$  to use when modeling opponents via (1.3) is a difficult issue as we will generally have very limited data available to estimate  $C$ . To be clear, the data we do have available is typically data on the positional marginals. In order to obtain data that is useful for estimating  $C$ , we would need to have access to the teams selected by contestants in historical DFS contests. Such data is hard to come by although it is often possible to obtain a small sample of selected teams via manual<sup>1</sup> inspection on the contest web-sites. As a result, we restrict ourselves to three possible choices of  $C$ :

1. The *independence* copula  $C_{\text{ind}}$  satisfies  $C_{\text{ind}}(u_1, \dots, u_m) = \prod_{i=1}^m u_i$  where  $m$  is the number of positional marginals. As the name suggests, the independence copula models independence among the selected positions so that when a contestant is choosing her TE position, for example, it is done so independently of her selections for the other positions. The independence copula can therefore be interpreted as the copula of a non-strategic contestant.
2. The *stacking* copula  $C_{\text{stack}}$  is intended to model the well known (Bales 2016) stacking behavior of some strategic contestants. In the NFL setting, for example, it is well known that the points scored by a given team's QB and main WR<sup>2</sup> are often strongly positively correlated. Selecting both players then becomes attractive to contestants who understand that positive correlation will serve to increase the overall variance of their entry, which is generally a desirable feature in top-heavy contests as we will argue in Section 1.5. Rather than explicitly

---

<sup>1</sup>These web-sites are generally not easy to "scrape" nor do the owners look favorably on web-scrapers.

<sup>2</sup>By "main" WR of a team, we refer to the WR with the highest expected points among all the WRs in the same team.



defining the stacking copula (which would require further notation), we simply note that it is straightforward to simulate a value of  $\mathbf{w}_o$  when  $C$  is the stacking copula. We first generate the QB position, i.e.,  $\mathbf{w}_o^{\text{QB}}$ . We then select the first WR to be the main WR from the generated QB's team. The remaining 2 WRs are generated from the Multinomial( $2, \mathbf{p}_{\text{WR}}$ ) distribution. It is easy to ensure that all 3 WRs are different; see Footnote 8 (main text).

3. The *mixture* copula sets  $C(\cdot) := (1 - q)C_{\text{ind}}(\cdot) + qC_{\text{stack}}(\cdot)$  for some  $q \in (0, 1)$ . Note that a finite mixture of copulas remains a copula. While very little data on complete team selections is available, as mentioned above, it is possible to observe a small sample of teams and such a sample could be used to estimate  $q$ . We can then interpret  $q$  as being the probability that a random contestant will be a “stacker” and therefore be represented by the stacking copula.

We note that different contest structures tend to result in more or less strategic behavior. Top-heavy contests, for example, encourage high variance teams, which suggests that stacking might be more common in those contests. Indeed that is what we observe and so the estimated  $q$  is typically higher for top-heavy contests. Finally, we note that the main fantasy sports companies will themselves have access to the team selections of all players and these companies could easily fit more sophisticated copulas to the data. This might be of general interest to these companies but it might also be useful to help them understand the skill-luck tradeoff in playing fantasy sports.

#### *Algorithm for Sampling Opponent Portfolios*

See Algorithm 7. We note that Algorithm 7 is guaranteed to terminate in a finite time since each accepted  $\mathbf{w}_o$  may be viewed as a draw from a geometric distribution.

---

**Algorithm 7** Sampling  $O$  Opponent Portfolios

---

**Require:**  $(\boldsymbol{\beta}_{\text{QB}}, \dots, \boldsymbol{\beta}_{\text{D}})$ ,  $(\mathbf{X}_{\text{QB}}, \dots, \mathbf{X}_{\text{D}})$ ,  $\mathbf{c}$ ,  $B_{\text{lb}}$ ,  $q$

```
1:  $(\boldsymbol{\alpha}_{\text{QB}}, \dots, \boldsymbol{\alpha}_{\text{D}}) = (\exp(\mathbf{X}_{\text{QB}}\boldsymbol{\beta}_{\text{QB}}), \dots, \exp(\mathbf{X}_{\text{D}}\boldsymbol{\beta}_{\text{D}}))$ 
2:  $(\mathbf{p}_{\text{QB}}, \dots, \mathbf{p}_{\text{D}}) \sim (\text{Dir}(\boldsymbol{\alpha}_{\text{QB}}), \dots, \text{Dir}(\boldsymbol{\alpha}_{\text{D}}))$ 
3: for  $o = 1 : O$ 
4:   Stack  $\sim \text{Bernoulli}(q)$ 
5:   Reject = True
6:   while Reject
7:      $(k_{\text{QB}}, k_{\text{RB}}, \dots, k_{\text{D}}) \sim (\text{Mult}(1, \mathbf{p}_{\text{QB}}), \text{Mult}(2, \mathbf{p}_{\text{RB}}), \dots, \text{Mult}(1, \mathbf{p}_{\text{D}}))$ 
8:     % Mult(2,  $\mathbf{p}_{\text{RB}}$ ) etc. should be understood as being without replacement; see Footnote 8 (main text)
9:     if Stack = 1
10:       Replace  $k_{\text{WR}}(1)$  with main WR from team of  $k_{\text{QB}}$ 
11:     end if
12:     Let  $\mathbf{w}_o$  denote the portfolio corresponding to  $(k_{\text{QB}}, \dots, k_{\text{D}})$ 
13:     if  $\mathbf{w}_o \in \mathbb{W}$  and  $\mathbf{c}^\top \mathbf{w}_o \geq B_{\text{lb}}$ 
14:       Reject = False and
15:       Accept  $\mathbf{w}_o$ 
16:     end if
17:   end while
18: end for
19: return  $\mathbf{W}_{\text{op}} = \{\mathbf{w}_o\}_{o=1}^O$ 
```

---

### A.1.2 Efficient Sampling of Order Statistic Moments

Monte Carlo simulation is required to generate samples of  $(\boldsymbol{\delta}, G^{(r')})$ , which are required to:

1. Estimate the input parameters  $(\mu_{G^{(r')}})$ ,  $(\boldsymbol{\sigma}_{\boldsymbol{\delta}, G^{(r')}})$  that are required by the various algorithms in Sections 1.4, 1.5 and 1.7.

2. Estimate the expected payoff from a given entry in the various algorithms, e.g. line 4 in Algorithm 2.
3. Estimate the P&L distribution for a given portfolio of entries using samples of  $(\boldsymbol{\delta}, G^{(r')})$ .

Recalling  $G_o = \mathbf{w}_o^\top \boldsymbol{\delta}$  is the fantasy points score of the  $o^{th}$  opponent, we first note the  $G_o$ 's,  $o = 1, \dots, O$ , are IID given  $(\boldsymbol{\delta}, \mathbf{p})$  where  $\mathbf{p}$  denotes the multinomial probability vectors for the positional marginals as discussed in Section 1.3. This then suggests the following algorithm for obtaining independent samples of  $(\boldsymbol{\delta}, G^{(r')})$ :

1. Generate<sup>3</sup>  $\boldsymbol{\delta} \sim N(\boldsymbol{\mu}_\delta, \boldsymbol{\Sigma}_\delta)$  and  $(\mathbf{p}, \mathbf{W}_{op})$  using Algorithm 7 where  $\mathbf{p} := (\mathbf{p}_{QB}, \dots, \mathbf{p}_D)$  and  $\mathbf{W}_{op} = \{\mathbf{w}_o\}_{o=1}^O$ .
2. Compute  $G_o := \mathbf{w}_o^\top \boldsymbol{\delta}$  for  $o = 1, \dots, O$ .
3. Order the  $G_o$ 's.
4. Return  $(\boldsymbol{\delta}, G^{(r')})$ .

While all of the contests that we participated in had relatively large values of  $O$ , it is worth noting there are also some very interesting DFS contests with small values of  $O$  that may range<sup>4</sup> in value from  $O = 1$  to  $O = 1000$ . These small- $O$  contests often have very high entry fees with correspondingly high payoffs and there is therefore considerable interest in them. At this point we simply note that (based on unreported numerical experiments) the algorithm described above seems quite adequate for handling small- $O$  contests. Of course, if we planned to participate in small- $O$  contests and also be able to quickly respond to developing news in the hours and minutes before the games, then it may well be necessary to develop a more efficient Monte Carlo algorithm. This of course is also true for the large- $O$  algorithm we develop below.

---

<sup>3</sup>We note that the normal assumption for  $\boldsymbol{\delta}$  is not necessary and any multivariate distribution with mean vector  $\boldsymbol{\mu}_\delta$  and variance-covariance matrix  $\boldsymbol{\Sigma}_\delta$  could also be used.

<sup>4</sup>All of the contests that we participated in during the 2017 NFL season had values of  $O$  that exceeded 8,000. That said, the cutoff between small  $O$  and large  $O$  is entirely subjective and indeed we could also add a third category – namely moderate- $O$  contests. These contests might refer to contests with values of  $O$  ranging from  $O = 500$  to  $O = 5,000$ .

## Efficient Monte Carlo when $O$ is Large

When  $O$  is large, e.g. when  $O = 500,000$  which is often the case in practice, the algorithm above is too computationally expensive and so a more efficient algorithm is required. Recalling that the conditional random variables  $G_o \mid (\boldsymbol{\delta}, \mathbf{p})$  are IID for  $o = 1, \dots, O$ , it follows (David and Nagaraja 2004) from the theory of order statistics that  $G^{(r')} \mid (\boldsymbol{\delta}, \mathbf{p})$  satisfies

$$G^{(qO)} \mid (\boldsymbol{\delta}, \mathbf{p}) \xrightarrow{p} F_{G \mid (\boldsymbol{\delta}, \mathbf{p})}^{-1}(q) \quad \text{as } O \rightarrow \infty \quad (\text{A.1})$$

where  $q \in (0, 1)$  and “ $\xrightarrow{p}$ ” denotes convergence in probability. In large- $O$  contests, we can use the result in (A.1) by simply setting  $G^{(r')} = F_{G \mid (\boldsymbol{\delta}, \mathbf{p})}^{-1}\left(\frac{r'}{O}\right)$ . Of course in practice we do not know the CDF  $F_{G \mid (\boldsymbol{\delta}, \mathbf{p})}$  and so we will have to estimate it as part of our algorithm. The key observation now is that even if the DFS contest in question has say 500,000 contestants, we can estimate  $F_{G \mid (\boldsymbol{\delta}, \mathbf{p})}$  with potentially far fewer samples. Our algorithm for generating Monte Carlo samples of  $(\boldsymbol{\delta}, G^{(r')})$  therefore proceeds as follows:

1. Generate  $\boldsymbol{\delta} \sim N(\boldsymbol{\mu}_\delta, \boldsymbol{\Sigma}_\delta)$  and  $(\mathbf{p}, \mathbf{W}_{\text{op}})$  using Algorithm 7 (or the stacking variant of it) where  $\mathbf{p} := (\mathbf{p}_{\text{QB}}, \dots, \mathbf{p}_{\text{D}})$  and  $\mathbf{W}_{\text{op}} = \{\mathbf{w}_o\}_{o=1}^O$ .
2. Compute  $G_o := \mathbf{w}_o^\top \boldsymbol{\delta}$  for  $o = 1, \dots, O$ .
3. Use the  $G_o$ 's to construct  $\widehat{F}_{G \mid (\boldsymbol{\delta}, \mathbf{p})}(\cdot)$ .
4. Set  $G^{(r')} = \widehat{F}_{G \mid (\boldsymbol{\delta}, \mathbf{p})}^{-1}\left(\frac{r'}{O}\right)$ .
5. Return  $(\boldsymbol{\delta}, G^{(r')})$ .

Note in this algorithm  $O$  now represents the number of Monte Carlo samples we use to estimate  $F_{G \mid (\boldsymbol{\delta}, \mathbf{p})}$  rather than the number of contestants in a given DFS contest. One issue still remains with this new algorithm, however. Consider for example the case where  $r' = O + N - 1$  (corresponding to the # 1 ranked opponent) in a top-heavy contest with say 100,000 contestants. This corresponds to the quantile  $q = 1 - 10^{-5}$  and according to line 4 of the algorithm we can generate a sample of

$G^{(O+N-1)}$  by setting it equal to  $\widehat{F}_{G|(\delta, \mathbf{p})}^{-1}(1 - 10^{-5})$ . We cannot hope to estimate  $\widehat{F}_{G|(\delta, \mathbf{p})}^{-1}(1 - 10^{-5})$  with just a moderate number of samples from line 2 of the algorithm, however, and this of course also applies to the values of  $r'$  corresponding to the # 2 ranked opponents, the # 3 ranked opponent etc.

We overcome this challenge as follows. We set  $O$  to a moderate value, e.g.  $O = 10,000$ , and then estimate the conditional CDF  $\widehat{F}_{G|(\delta, \mathbf{p})}(\cdot)$  with the empirical CDF of those  $O$  samples from line 2 of the algorithm. For  $r'$  values that are not deep in the tail, we use  $\widehat{F}_{G|(\delta, \mathbf{p})}^{-1}(\cdot)$  to sample  $G^{(r')}$ . For  $r'$  values that are deep in the right tail (corresponding to the largest payoffs), however, we will use an approximation based on the normal distribution. Specifically, we choose the mean and variance of the normal distribution so that it has the same 99.0<sup>th</sup> and 99.5<sup>th</sup> percentiles as  $\widehat{F}_{G|(\delta, \mathbf{p})}(\cdot)$ ; see Cook 2010. We then use this normal distribution in place of  $\widehat{F}$  in line 4 of the algorithm for values of  $r'$  that correspond to extreme percentiles.

Further efficiencies were obtained through the use of *splitting*. The high-level idea behind splitting is as follows. If a system is dependent on two random variables and it takes more time to sample the second variable but the first variable influences the system more, then one should generate multiple samples of the first variable for each sample of the second variable. In our context,  $\mathbf{W}_{\text{op}}$  takes more time to sample but  $\delta$  appears to influence  $G^{(r')}$  more. Accordingly, in our experiments we implemented splitting<sup>5</sup> with a ratio of 50:1 so that for each sample of  $\mathbf{W}_{\text{op}}$  we generated 50 samples of  $\delta$ .

### A.1.3 Additional Details for the Double-Up Problem Formulation

#### *Accounting for a Skew in the Distribution of $Y_{\mathbf{w}}$ Double-Up Contests*

One difficulty that might arise with the mean-variance approach is if the distribution of the  $Y_{\mathbf{w}}$ 's display a significant skew. While we have seen no evidence<sup>6</sup> of this when  $\delta$  is assumed multivariate

---

<sup>5</sup>We empirically tested several split ratios and found a ratio of 50:1 to perform best. See Chapter V of Asmussen and Glynn 2007 for further details on splitting.

<sup>6</sup>In unreported experiments, we found  $Y_{\mathbf{w}}$  to be unimodal and very well approximated by a normal distribution for a test-set of  $\mathbf{w}$ 's when  $\delta$  was also normally distributed.

normally distributed, we might see such a skew if we assumed a distribution for  $\delta$  which also had a skew. A significant skew in the  $Y_{\mathbf{w}}$ 's could then result in us mistakenly seeking a portfolio with a small variance or vice versa. To see this, consider a double-up contest where the top 50% of contestants earn a reward. If the  $Y_{\mathbf{w}}$ 's display a significant right skew, then their medians will be less than their means. It's then possible there exists a  $\mathbf{w}$  such  $\mu_{Y_{\mathbf{w}}} \geq 0$  but that  $\text{median}(Y_{\mathbf{w}'}) < 0$  for all  $\mathbf{w}' \in \mathbb{W}$ . In that event, the condition of the if statement on line 1 of Algorithm 1 will be satisfied and so we end up seeking a team with a large mean and a small variance. It's possible, however, that we should be seeking a team with a large mean and a large variance since  $\text{median}(Y_{\mathbf{w}'}) < 0$  for all  $\mathbf{w}' \in \mathbb{W}$ . Note that such a mistake might occur because the reward cutoff of the contest is determined by the median and not the mean, which is what we use in Algorithm 1. Of course this issue doesn't arise if the  $Y_{\mathbf{w}}$ 's are normal since then the means and medians coincide. An easy solution to this problem is to simply ignore the if-else statements in Algorithm 1 and consider both possibilities. This results in Algorithm 8.

---

**Algorithm 8** Adjusted Optimization for the Double-Up Problem with a Single Entry

---

**Require:**  $\mathbb{W}, \Lambda, \boldsymbol{\mu}_{\delta}, \boldsymbol{\Sigma}_{\delta}, \mu_{G^{(r')}}, \sigma_{G^{(r')}}^2, \boldsymbol{\sigma}_{\delta, G^{(r')}}$  and Monte Carlo samples of  $(\delta, G^{(r')})$

- 1: **for all**  $\lambda \in \Lambda$
  - 2:    $\mathbf{w}_{\lambda^-} = \underset{\mathbf{w} \in \mathbb{W}, \mu_{Y_{\mathbf{w}}} \geq 0}{\text{argmax}} \left\{ \mu_{Y_{\mathbf{w}}} - \lambda \sigma_{Y_{\mathbf{w}}}^2 \right\}$
  - 3:    $\mathbf{w}_{\lambda^+} = \underset{\mathbf{w} \in \mathbb{W}}{\text{argmax}} \left\{ \mu_{Y_{\mathbf{w}}} + \lambda \sigma_{Y_{\mathbf{w}}}^2 \right\}$
  - 4: **end for**
  - 5: **return**  $\mathbf{w}^* = \underset{\mathbf{w} \in \{\mathbf{w}_{\lambda^-}, \mathbf{w}_{\lambda^+} : \lambda \in \Lambda\}}{\text{argmax}} \mathbb{P}\{Y_{\mathbf{w}} > 0\}$
- 

*The Double-Up Problem with Multiple Entries*

Here we consider the problem of submitting a fixed but finite number of  $N > 1$  entries to the double-up contest. In the case of a risk-neutral DFS player, it seems intuitively clear that if  $O \rightarrow \infty$  so that  $N/O \rightarrow 0$ , then a *replication* strategy is an optimal or near-optimal strategy. In particular, under such a replication strategy, the DFS player should submit  $N$  copies of  $\mathbf{w}^*$ , which is defined as

follows. Consider a double-up style contest in which we have purchased  $N$  entries<sup>7</sup> and let  $\mathcal{R}(\mathbf{W})$  denote the expected reward function when we submit the portfolio of entries  $\mathbf{W} := \{\mathbf{w}_i\}_{i=1}^{|\mathbf{W}|}$ . That is,

$$\mathcal{R}(\mathbf{W}) := \sum_{i=1}^{|\mathbf{W}|} \mathbb{P} \left\{ \mathbf{w}_i^\top \boldsymbol{\delta} > G_{-i}^{(r')}(\mathbf{W}_{-i}, \mathbf{W}_{\text{op}}, \boldsymbol{\delta}) \right\} \quad (\text{A.2})$$

where  $r' := O + |\mathbf{W}| - r$ ,  $G_{-i}^{(r')}$  is the  $r^{\text{th}}$  order statistic of  $\{G_o\}_{o=1}^O \cup \{F_j\}_{j=1}^{|\mathbf{W}|} \setminus F_i$  and  $\mathbf{W}_{-i} := \mathbf{W} \setminus \mathbf{w}_i$ . The entry that we replicate is defined as

$$\mathbf{w}^* := \operatorname{argmax}_{\mathbf{w} \in \mathbb{W}} \mathcal{R}(\mathbf{w}) \quad (\text{A.3})$$

where it follows from (A.2) that

$$\mathcal{R}(\mathbf{w}) = \mathbb{P} \left\{ \mathbf{w}^\top \boldsymbol{\delta} > G^{(r')}(\mathbf{W}_{\text{op}}, \boldsymbol{\delta}) \right\}$$

with  $r' = O + 1 - r$ .

To gain some intuition for our advocacy of the replication strategy where we submit  $N$  copies of  $\mathbf{w}^*$  to the double-up contest, consider such a contest where the top 50% of entries double their money and let  $\mathbf{w}^*$  be the optimal entry as defined in (A.3). Now consider the  $|\mathbf{W}| = 2$  case with  $O$  large and suppose we submit two copies of  $\mathbf{w}^*$ . Given the optimality of  $\mathbf{w}^*$ , submitting two copies of it can only be suboptimal to the extent that  $\mathbf{w}^*$  is at or near the boundary cutoff  $G^{(r')}$ . But this event will (in general) occur with vanishingly small probability in the limit as  $O \rightarrow \infty$ . Even when  $O$  is not large, we suspect the replication strategy will still be close to optimal. While we can derive conditions guaranteeing the optimality or near-optimality of replication for double-up contests, these conditions are not easily expressed in terms of the *observable* parameters of the

---

<sup>7</sup>We assume we have already purchased the  $N$  entries and hence the cost associated with the purchase of these  $N$  entries can be viewed as a sunk cost. However, we allow ourselves to submit less than (or equal to)  $N$  entries, i.e.,  $|\mathbf{W}| \leq N$  with the understanding that if  $|\mathbf{W}| < N$ , then we simply “waste”  $N - |\mathbf{W}|$  entries. As such the rank  $r$  that determines the cutoff between winning and not winning in double-up style contests (as discussed near the beginning of Section 1.2.1) is constant and does not depend on the number  $|\mathbf{W}|$  of entries that we actually submit. Note that we also need to allow for the possibility of submitting less than  $N$  entries when we discuss the submodularity of the top-heavy objective function in Appendix A.1.4.

contest. This is not surprising since we know the double-up payoff structure can be viewed as a special case of the top-heavy payoff structure and certainly replication in general is not optimal for top-heavy contests.

There is a simple test we can deploy, however, to check if replication is indeed optimal in any given double-up style contest. To see this, let  $\mathcal{R}(N \times \mathbf{w}^*)$  denote the expected reward when we replicate  $\mathbf{w}^*$   $N$  times. Using Monte-Carlo, we can easily check to see whether or not  $\mathcal{R}(N \times \mathbf{w}^*) \approx N \mathcal{R}(\mathbf{w}^*)$ . If this is the case, then we know that replication of  $\mathbf{w}^*$  is near-optimal because it must be the case that the expected reward of any portfolio of  $N$  entries is less than or equal to  $N \mathcal{R}(\mathbf{w}^*)$ . We formally state these observations regarding a “certificate-of-optimality” as a proposition.

**Proposition A.1.** *Consider  $\mathbf{w}^*$  as defined in (A.3) and denote by  $N \times \mathbf{w}^*$  the portfolio consisting of  $N$  replications of  $\mathbf{w}^*$ . Define  $\mathfrak{d} := N \mathcal{R}(\mathbf{w}^*) - \mathcal{R}(N \times \mathbf{w}^*)$  where  $\mathcal{R}(\cdot)$  is as defined in (A.2). Finally, denote by  $\mathbf{W}^\# = \{\mathbf{w}_i^\#\}_{i=1}^N$  an optimal portfolio of  $N$  entries for the double-up contest, i.e.,*

$$\mathbf{W}^\# := \underset{\mathbf{W} \in \mathbb{W}^N}{\operatorname{argmax}} \mathcal{R}(\mathbf{W}).$$

*Then, the following statements hold:*

(a) *[Suboptimality bound]. The suboptimality of  $N \times \mathbf{w}^*$  is bounded above by  $\mathfrak{d}$ , i.e.,*

$$\mathcal{R}(\mathbf{W}^\#) - \mathcal{R}(N \times \mathbf{w}^*) \leq \mathfrak{d}$$

(b) *[Certificate-of-optimality]. If  $\mathfrak{d}$  equals 0, then  $N \times \mathbf{w}^*$  is optimal, i.e., replication is optimal.*

**Proof.** To prove the suboptimality bound (a), it suffices to show that  $\mathcal{R}(\mathbf{W}^\#) \leq N \mathcal{R}(\mathbf{w}^*)$ , which holds since  $\mathcal{R}(\mathbf{W}^\#) \leq \mathcal{R}(\mathbf{w}_1^\#) + \dots + \mathcal{R}(\mathbf{w}_N^\#) \leq N \mathcal{R}(\mathbf{w}^*)$ . The second inequality follows from the optimality of  $\mathbf{w}^*$  (see (A.3)). The first inequality holds since for an arbitrary portfolio  $\mathbf{W} := \{\mathbf{w}_i\}_{i=1}^N$  of  $N$  entries, we have  $\mathcal{R}(\mathbf{W}) \leq \sum_{i=1}^N \mathcal{R}(\mathbf{w}_i)$ . To see this, recall that the expected reward  $\mathcal{R}(\cdot)$  is an expectation over  $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$  so consider an arbitrary  $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$  realization and denote by  $\mathcal{R}_{\boldsymbol{\delta}, \mathbf{W}_{\text{op}}}(\cdot)$  the



reward function conditional on the realization  $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$ . Using  $\mathbb{I}\{\cdot\}$  to denote the indicator function, it follows from the definition of  $\mathcal{R}(\cdot)$  in (A.2) that

$$\mathcal{R}_{\boldsymbol{\delta}, \mathbf{W}_{\text{op}}}(\mathbf{W}) = \sum_{i=1}^N \mathbb{I}\left\{\mathbf{w}_i^\top \boldsymbol{\delta} > G_{-i}^{(O+N-r)}(\mathbf{W}_{-i}, \mathbf{W}_{\text{op}}, \boldsymbol{\delta})\right\} \quad (\text{A.4})$$

$$\leq \sum_{i=1}^N \mathbb{I}\left\{\mathbf{w}_i^\top \boldsymbol{\delta} > G^{(O+1-r)}(\mathbf{W}_{\text{op}}, \boldsymbol{\delta})\right\} \quad (\text{A.5})$$

$$= \sum_{i=1}^N \mathcal{R}_{\boldsymbol{\delta}, \mathbf{W}_{\text{op}}}(\mathbf{w}_i)$$

where the inequality holds since  $G_{-i}^{(O+N-r)}(\mathbf{W}_{-i}, \mathbf{W}_{\text{op}}, \boldsymbol{\delta}) \geq G^{(O+1-r)}(\mathbf{W}_{\text{op}}, \boldsymbol{\delta})$  for all  $i \in \{1, \dots, N\}$ .

Taking an expectation over  $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$  allows us to conclude  $\mathcal{R}(\mathbf{W}) \leq \sum_{i=1}^N \mathcal{R}(\mathbf{w}_i)$ .

To prove the certificate-of-optimality (b), we note that it follows directly from the suboptimality bound since  $\mathfrak{d} = 0$  implies

$$0 \leq \mathcal{R}(\mathbf{W}^\#) - \mathcal{R}(N \times \mathbf{w}^*) \leq \mathfrak{d} = 0,$$

where the first inequality holds due to the optimality of  $\mathbf{W}^\#$ . ■

Part (b) of Proposition A.1 allows us to check if replication is indeed optimal in any given double-up contest. Indeed in the various numerical experiments of Section 1.6 we found (modulo the statistical noise arising from our Monte-Carlo simulations) that replication was optimal for all of our double-up contests.

#### A.1.4 Details for the Top-Heavy Problem Formulation

##### *Justification of Assumptions 1.1 and 1.2*

Assumption 1.1 can be interpreted as stating that, in expectation, the points total of our optimal portfolio will not be sufficient to achieve the minimum payout  $R_D$ . In option-pricing terminology, we are therefore assuming our optimal portfolio is “out-of-the-money”. This is a very reasonable

assumption to make for top-heavy contests where it is often the case that only the top 20% or so of entries earn a cash payout. In numerical experiments, our model often predicts that our optimal portfolio will (in expectation) be at or around the top 20<sup>th</sup> percentile. The assumption therefore may break down if payoffs extend beyond the top 20% of entries. Nonetheless, the payoff sizes around the 20<sup>th</sup> percentile are very small and almost negligible. Indeed within our model, most of the expected profit and loss (P&L) comes from the top few percentiles and  $\mu_{Y_w^d} < 0$  is certainly true for these values of  $d$ . Finally, we note the well-known general tendency of models to overestimate the performance of an optimally chosen quantity (in this case our portfolio). We therefore anticipate that our optimal portfolio will not quite achieve (in expectation) the top 20<sup>th</sup> percentile and may well be out of the money for all payoff percentiles as assumed in Assumption 1.1.

**Proof of Proposition 1.2.** First note that the number of feasible lineups is finite and so any  $w_o$  which is selected with strictly positive probability will be chosen infinitely often as  $O \rightarrow \infty$ . In particular, the top team will be chosen infinitely often and so it follows that the top  $D$  teams will be identical for any finite  $D$  and any realisation of  $(\delta, p)$ . It therefore follows that conditioned on  $(\delta, p)$ ,  $G^{(r'_d)} = G^{(r'_{d'})}$  w.p. 1 in the limit as  $O \rightarrow \infty$ . (1.16) will then follow from a simple interchange of limit and expectation, which can easily be justified assuming  $\delta$  is integrable. ■

In many of the contests we participated in, we saw values of  $O \approx 200,000$  which, while large, is actually quite small relative to the total number of feasible lineups. As such, we do not expect to see the top  $D$  teams being identical in practice or even to see much if any repetition among them. Nonetheless, we do expect to see sizeable overlaps in these top teams, especially for the very highest ranks, which are our ultimate target given the lop-sided reward structure of typical top-heavy contests. It was no surprise then that in our numerical experiments we observed a very weak dependence of  $\text{Cov}(\delta_p, G^{(r'_d)})$  on  $d$  as stated earlier.

### Submodularity of Top-Heavy Objective Function

We first discuss here the practicality of Assumption 1.3, which assumes  $\Delta_k \geq \Delta_{k+1}$  for all  $k = 1, \dots, K - 2$  where  $\Delta_k := V_k - V_{k+1}$  and  $V_k$  denotes the payoff corresponding to rank  $k$  entry. We first give an example of a simple top-heavy payoff structure that satisfies this assumption and then, discuss how the payoff structure of real-world top-heavy contests compares with our simple structure.

To gain intuition regarding what type of payoff structure satisfies Assumption 1.3, consider the payoff structure in which  $V_k = \tau \times V_{k+1}$  for all  $k = 1, \dots, K - 1$  with the parameter  $\tau > 1$ . Then, it follows trivially that  $\Delta_k \geq \Delta_{k+1}$  for all  $k = 1, \dots, K - 2$ . Furthermore, observe that one can make the parameter  $\tau$  specific to rank, i.e.,  $V_k = \tau_k \times V_{k+1}$ , as long as  $\tau_k > 1$  for all  $k = 1, \dots, K - 1$ . Connecting this simple payoff structure to the real-world top-heavy contests we participated in, we show in Figure A.1 the payoffs corresponding to the top few ranks and observe that the “convex” shape clearly demonstrates that the payoff structure satisfies  $\tau_k > 1$  for all  $k$  shown.

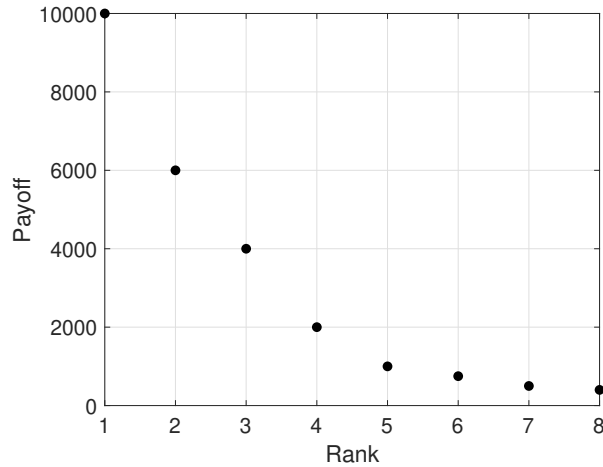


Figure A.1: Payoffs corresponding to ranks 1 to 8 of the week 10 top-heavy contest we participated in during the 2017 NFL season. We note the payoff structure of the top-heavy contests in other weeks were very similar and we do not show them here for the sake of brevity.

We note however that not all ranks satisfy Assumption 1.3. For example, in the contest corresponding to Figure A.1, both ranks 999 and 1000 had a payoff of \$10 each whereas rank 1001 had a payoff of \$8. Hence,  $\Delta_{999} = 0 < 2 = \Delta_{1000}$ , which violates Assumption 1.3. But it is

worth mentioning that Assumption 1.3 holds for the top few ranks where most of the action in top-heavy contests lies. Furthermore, as discussed in Section 1.5.1, though our proposed approach (Algorithm 3) is motivated by the submodularity considerations, it does not necessarily enjoy the classical “ $1 - 1/e$ ” theoretical guarantee since finding an entry with the highest value-add is non-trivial. Recognizing this, we provide a bound (Proposition 1.3) in Section 1.5.1 on the degree of suboptimality of our approach and used it to evaluate the performance of our approach on real-world data. This bound does not require Assumption 1.3.

We now recall the definition of a submodular function (Schrijver 2003).

**Definition A.1** (Submodular function). *Let  $\Omega$  be a finite set and suppose  $f : 2^\Omega \rightarrow \mathbb{R}$  is a function where  $2^\Omega$  denotes the power set of  $\Omega$ . Suppose  $f(X \cup \{x_1\}) + f(X \cup \{x_2\}) \geq f(X \cup \{x_1, x_2\}) + f(X)$  for every  $X \subseteq \Omega$  and every  $x_1, x_2 \in \Omega \setminus X$  such that  $x_1 \neq x_2$ . Then,  $f$  is submodular.*

We also need the following definition.

**Definition A.2** (Monotonic function). *A function  $f : 2^\Omega \rightarrow \mathbb{R}$  is monotonic if  $f(X) \leq f(Y)$  whenever  $X \subseteq Y$ .*

A function that is both monotonic and submodular is called a *monotone submodular function*. We now state the proof of Theorem 1.1, which asserts that the objective function of the top-heavy DFS problem with  $N$  entries, denoted by  $\mathcal{R}(\mathbf{W})$  is monotone submodular in  $\mathbf{W}$  when Assumption 1.3 holds.

**Proof of Theorem 1.1.** Consider a top-heavy style contest and let  $\mathcal{R}(\mathbf{W})$  denote the expected reward function when we submit the portfolio of entries  $\mathbf{W} := \{\mathbf{w}_i\}_{i=1}^n$ . Recall from (1.17) the definition

$$\mathcal{R}(\mathbf{W}) := \sum_{i=1}^n \sum_{d=1}^D (R_d - R_{d+1}) \mathbb{P} \left\{ \mathbf{w}_i^\top \boldsymbol{\delta} > G_{-i}^{(r'_d)}(\mathbf{W}_{-i}, \mathbf{W}_{\text{op}}, \boldsymbol{\delta}) \right\},$$

where  $r'_d := O + n - r_d$ ,  $G_{-i}^{(r)}$  is the  $r^{\text{th}}$  order statistic of  $\{G_o\}_{o=1}^O \cup \{F_j\}_{j=1}^n \setminus F_i$  and  $\mathbf{W}_{-i} := \mathbf{W} \setminus \mathbf{w}_i$ .

We first show that  $\mathcal{R}(\mathbf{W})$  is monotonic increasing in  $\mathbf{W}$ . Note that the expected reward  $\mathcal{R}(\cdot)$  is an expectation over  $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$  so consider an arbitrary  $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$  realization and two arbitrary and feasible portfolio choices  $\mathbf{W}_1$  and  $\mathbf{W}_2$  such that  $\mathbf{W}_1 \subseteq \mathbf{W}_2$ . It is easy to see that

$$\mathcal{R}_{\boldsymbol{\delta}, \mathbf{W}_{\text{op}}}(\mathbf{W}_1) \leq \mathcal{R}_{\boldsymbol{\delta}, \mathbf{W}_{\text{op}}}(\mathbf{W}_2)$$

where  $\mathcal{R}_{\boldsymbol{\delta}, \mathbf{W}_{\text{op}}}(\cdot)$  denotes the reward function conditional on the realization  $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$ . Taking an expectation over  $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$  allows us to conclude that  $\mathcal{R}(\mathbf{W})$  is monotonic increasing.

Second, we show the submodularity of  $\mathcal{R}(\mathbf{W})$  with respect to  $\mathbf{W}$ . Consider an arbitrary  $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$  realization, an arbitrary feasible portfolio choice  $\mathbf{W} := \{\mathbf{w}_i\}_{i=1}^n$ , and two arbitrary feasible entries  $\mathbf{x}_1$  and  $\mathbf{x}_2$ . For all  $i = 1, \dots, n$ , denote by  $F_i := \mathbf{w}_i^\top \boldsymbol{\delta}$  the fantasy points of entry  $\mathbf{w}_i$  and by  $\hat{F}_1 := \mathbf{x}_1^\top \boldsymbol{\delta}$  and  $\hat{F}_2 := \mathbf{x}_2^\top \boldsymbol{\delta}$  the fantasy points of entries  $\mathbf{x}_1$  and  $\mathbf{x}_2$ , respectively. As explained in Remark A.1 below, assume without loss of generality that all of the values in the collection  $\{F_1, \dots, F_n, \hat{F}_1, \hat{F}_2\}$  are unique. Without loss of generality we can assume the entries are ranked according to

$$\underbrace{F_1 > F_2 > \dots > F_{n_0}}_{\mathbf{W}_0 := \{\mathbf{w}_i\}_{i=1}^{n_0}} > \hat{F}_1 > \underbrace{F_{n_0+1} > \dots > F_{n_0+n_1}}_{\mathbf{W}_1 := \{\mathbf{w}_{n_0+i}\}_{i=1}^{n_1}} > \hat{F}_2 > \underbrace{F_{n_0+n_1+1} > \dots > F_{n_0+n_1+n_2}}_{\mathbf{W}_2 := \{\mathbf{w}_{n_0+n_1+i}\}_{i=1}^{n_2}} \quad (\text{A.6})$$

with  $\mathbf{W}$  partitioned into  $\{\mathbf{W}_0, \mathbf{W}_1, \mathbf{W}_2\}$  as shown in (A.6) with  $n_0 := |\mathbf{W}_0|$ ,  $n_1 := |\mathbf{W}_1|$ ,  $n_2 := |\mathbf{W}_2|$ , and  $n_0 + n_1 + n_2 = n$  trivially. Now, for any entry  $\mathbf{w} \in \mathbf{W} \cup \{\mathbf{x}_1, \mathbf{x}_2\}$ , define  $\rho(\mathbf{w})$  as the rank of  $\mathbf{w}$  in a contest in which the entries  $\{\mathbf{W}_{\text{op}}, \mathbf{W}, \mathbf{x}_1, \mathbf{x}_2\}$  are submitted. Recalling that  $V_k$  denotes the payoff

corresponding to rank  $k$ , the reward function conditioned on  $(\delta, \mathbf{W}_{\text{op}})$  can be expressed as follows:

$$\mathcal{R}_{\delta, \mathbf{W}_{\text{op}}}(\mathbf{W} \cup \{\mathbf{x}_1, \mathbf{x}_2\}) = \left\{ \sum_{\mathbf{w} \in \mathbf{W}_0} V_{\rho(\mathbf{w})} + \sum_{\mathbf{w} \in \mathbf{W}_1} V_{\rho(\mathbf{w})} + \sum_{\mathbf{w} \in \mathbf{W}_2} V_{\rho(\mathbf{w})} \right\} + V_{\rho(\mathbf{x}_1)} + V_{\rho(\mathbf{x}_2)} \quad (\text{A.7a})$$

$$\mathcal{R}_{\delta, \mathbf{W}_{\text{op}}}(\mathbf{W}) = \sum_{\mathbf{w} \in \mathbf{W}_0} V_{\rho(\mathbf{w})} + \sum_{\mathbf{w} \in \mathbf{W}_1} V_{\rho(\mathbf{w})-1} + \sum_{\mathbf{w} \in \mathbf{W}_2} V_{\rho(\mathbf{w})-2} \quad (\text{A.7b})$$

$$\mathcal{R}_{\delta, \mathbf{W}_{\text{op}}}(\mathbf{W} \cup \{\mathbf{x}_1\}) = \left\{ \sum_{\mathbf{w} \in \mathbf{W}_0} V_{\rho(\mathbf{w})} + \sum_{\mathbf{w} \in \mathbf{W}_1} V_{\rho(\mathbf{w})} + \sum_{\mathbf{w} \in \mathbf{W}_2} V_{\rho(\mathbf{w})-1} \right\} + V_{\rho(\mathbf{x}_1)} \quad (\text{A.7c})$$

$$\mathcal{R}_{\delta, \mathbf{W}_{\text{op}}}(\mathbf{W} \cup \{\mathbf{x}_2\}) = \left\{ \sum_{\mathbf{w} \in \mathbf{W}_0} V_{\rho(\mathbf{w})} + \sum_{\mathbf{w} \in \mathbf{W}_1} V_{\rho(\mathbf{w})-1} + \sum_{\mathbf{w} \in \mathbf{W}_2} V_{\rho(\mathbf{w})-1} \right\} + V_{\rho(\mathbf{x}_2)-1}. \quad (\text{A.7d})$$

Following Definition A.1 we must show

$$\mathcal{R}_{\delta, \mathbf{W}_{\text{op}}}(\mathbf{W} \cup \{\mathbf{x}_1\}) + \mathcal{R}_{\delta, \mathbf{W}_{\text{op}}}(\mathbf{W} \cup \{\mathbf{x}_2\}) \geq \mathcal{R}_{\delta, \mathbf{W}_{\text{op}}}(\mathbf{W} \cup \{\mathbf{x}_1, \mathbf{x}_2\}) + \mathcal{R}_{\delta, \mathbf{W}_{\text{op}}}(\mathbf{W}) \quad (\text{A.8})$$

in order to establish the submodularity of  $\mathcal{R}_{\delta, \mathbf{W}_{\text{op}}}(\cdot)$ . Using (A.7a) to (A.7d), we can see that (A.8) is equivalent to

$$\begin{aligned} & \left\{ \sum_{\mathbf{w} \in \mathbf{W}_2} V_{\rho(\mathbf{w})-1} + \sum_{\mathbf{w} \in \mathbf{W}_2} V_{\rho(\mathbf{w})-1} \right\} + V_{\rho(\mathbf{x}_2)-1} \geq \left\{ \sum_{\mathbf{w} \in \mathbf{W}_2} V_{\rho(\mathbf{w})-2} + \sum_{\mathbf{w} \in \mathbf{W}_2} V_{\rho(\mathbf{w})} \right\} + V_{\rho(\mathbf{x}_2)} \quad (\text{A.9}) \\ \Leftrightarrow & \underbrace{\left\{ V_{\rho(\mathbf{x}_2)-1} - V_{\rho(\mathbf{x}_2)} \right\}}_{=:\Delta_{\rho(\mathbf{x}_2)-1}} + \sum_{i=1}^{n_2} \underbrace{\left\{ V_{\rho(\mathbf{w}_{n_0+n_1+i})-1} - V_{\rho(\mathbf{w}_{n_0+n_1+i})} \right\}}_{=:\Delta_{\rho(\mathbf{w}_{n_0+n_1+i})-1}} \geq \sum_{i=1}^{n_2} \underbrace{\left\{ V_{\rho(\mathbf{w}_{n_0+n_1+i})-2} - V_{\rho(\mathbf{w}_{n_0+n_1+i})-1} \right\}}_{=:\Delta_{\rho(\mathbf{w}_{n_0+n_1+i})-2}} \\ \Leftrightarrow & \Delta_{\rho(\mathbf{x}_2)-1} + \left\{ \sum_{i=1}^{n_2-1} \Delta_{\rho(\mathbf{w}_{n_0+n_1+i})-1} \right\} + \Delta_{\rho(\mathbf{w}_n)-1} \geq \Delta_{\rho(\mathbf{w}_{n_0+n_1+1})-2} + \sum_{i=2}^{n_2} \Delta_{\rho(\mathbf{w}_{n_0+n_1+i})-2} \\ \Leftrightarrow & \underbrace{\left\{ \Delta_{\rho(\mathbf{x}_2)-1} - \Delta_{\rho(\mathbf{w}_{n_0+n_1+1})-2} \right\}}_{=:(\star)} + \sum_{i=1}^{n_2-1} \underbrace{\left\{ \Delta_{\rho(\mathbf{w}_{n_0+n_1+i})-1} - \Delta_{\rho(\mathbf{w}_{n_0+n_1+i+1})-2} \right\}}_{=:(\diamond_i)} + \underbrace{\Delta_{\rho(\mathbf{w}_n)-1}}_{=:(\square)} \geq 0. \quad (\text{A.10}) \end{aligned}$$

But (A.10) is true because:

- $(\star) \geq 0$  since  $\rho(\mathbf{x}_2) \leq \rho(\mathbf{w}_{n_0+n_1+1}) - 1$  and  $\Delta_k - \Delta_\ell \geq 0$  for  $k \leq \ell$  by Assumption 1.3.
- $(\diamond_i) \geq 0$  since  $\rho(\mathbf{w}_{n_0+n_1+i}) - 1 \leq \rho(\mathbf{w}_{n_0+n_1+i+1}) - 2$  for all  $i = 1, \dots, n_2 - 1$  and  $\Delta_k - \Delta_\ell \geq 0$

for  $k \leq \ell$  again by Assumption 1.3.

- $(\square) \geq 0$  since  $\Delta_k \geq 0$  for all  $k$  since by assumption the payoffs are non-increasing in  $k$ .

It therefore follows that  $\mathcal{R}_{\delta, \mathbf{W}_{\text{op}}}$  is submodular for each  $(\delta, \mathbf{W}_{\text{op}})$  realization. Taking expectation over  $(\delta, \mathbf{W}_{\text{op}})$  then yields the submodularity of  $\mathcal{R}(\mathbf{W})$ . ■

**Remark A.1.** *While establishing the submodularity in the proof of Theorem 1.1, we assumed that each of the values in  $\{F_1, \dots, F_n, \hat{F}_1, \hat{F}_2\}$  was unique. This assumption is without loss of generality. To see this, suppose  $F_1 = F_2 > F_3$  and hence the entries  $\mathbf{w}_1$  and  $\mathbf{w}_2$  have the same rank, say  $k$ . Then the decision-maker will earn a payoff of  $(V_k + V_{k+1})/2$  from  $F_1$  and  $F_2$  and hence, earn a total payoff of  $V_k + V_{k+1}$  from these two entries. However, the decision-maker also earns a total payoff of  $V_k + V_{k+1}$  from these two entries if we re-define  $F_2$  to equal  $F_1 - \epsilon$  for  $\epsilon$  sufficiently small so that the rank of  $\mathbf{w}_2$  equals  $k + 1$ , the rank of  $\mathbf{w}_1$  remains  $k$  and the ranks (and hence, rewards) of the other entries  $\{\mathbf{w}_3, \dots, \mathbf{w}_n, \mathbf{x}_1, \mathbf{x}_2\}$  remains unchanged. More generally, if  $\ell$  values are equal, say  $F_1 = F_2 = \dots = F_\ell$ , then we can re-define  $F_i \leftarrow F_1 - (i - 1) \times \epsilon$  for all  $i = 2, \dots, \ell$  where  $\epsilon$  is again sufficiently small. We can extend this reasoning to scenarios in which there are multiple “blocks” of equal values. In particular, we can adjust them so that there are no ties but the total payoff to the DFS portfolio remains unchanged.*

**Remark A.2.** *Theorem 1.1 states that the top-heavy objective  $\mathcal{R}(\cdot)$  is submodular under the “convexity” assumption (Assumption 1.3), i.e. if  $\Delta_k \geq \Delta_{k+1}$  for all  $k$ . In fact it is even easier (and perhaps surprising) to see that  $\mathcal{R}(\cdot)$  is also submodular under the “concavity” assumption, i.e.,  $\Delta_k \leq \Delta_{k+1}$  for all  $k$ . To see this, note that (A.9) is equivalent to*

$$\sum_{\mathbf{w} \in \mathbf{W}_2} \underbrace{\{V_{\rho(\mathbf{w})-1} - V_{\rho(\mathbf{w})}\}}_{=\Delta_{\rho(\mathbf{w})-1}} + \underbrace{\{V_{\rho(\mathbf{x}_2)-1} - V_{\rho(\mathbf{x}_2)}\}}_{\geq 0} \geq \sum_{\mathbf{w} \in \mathbf{W}_2} \underbrace{\{V_{\rho(\mathbf{w})-2} - V_{\rho(\mathbf{w})-1}\}}_{=\Delta_{\rho(\mathbf{w})-2}}.$$

Hence, submodularity follows even if  $\Delta_k \leq \Delta_{k+1}$  for all  $k$ . We chose to focus on Assumption 1.3, however, as the payoff structures in real-world top-heavy contests are more convex in nature as

discussed around Figure A.1.

### Proof of Proposition 1.3

**Proof of Proposition 1.3.** We have  $\mathcal{R}(\mathbf{W}) \leq \mathcal{R}(\mathbf{W}^\#) \leq \mathcal{R}(\mathbf{w}_1^\#) + \dots + \mathcal{R}(\mathbf{w}_N^\#) \leq N \mathcal{R}(\mathbf{w}^*)$  where the first inequality follows from the optimality of  $\mathbf{W}^\#$  for the  $N$ -entry problem, the second inequality holds due to reasons discussed below, and the third inequality follows from the optimality of  $\mathbf{w}^*$ . Dividing across by  $\mathcal{R}(\mathbf{W})$  yields  $1 \leq \mathcal{R}(\mathbf{W}^\#)/\mathcal{R}(\mathbf{W}) \leq N \mathcal{R}(\mathbf{w}^*)/\mathcal{R}(\mathbf{W}) = 1/\nu_{\mathbf{W}}$  from which the result follows.

The second inequality holds since for an arbitrary portfolio  $\mathbf{W} := \{\mathbf{w}_i\}_{i=1}^N$  of  $N$  entries, we have  $\mathcal{R}(\mathbf{W}) \leq \sum_{i=1}^N \mathcal{R}(\mathbf{w}_i)$ . To see this, we follow the same argument used for proving part (a) of Proposition A.1. In particular, recall that the expected reward  $\mathcal{R}(\cdot)$  is an expectation over  $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$  so consider an arbitrary  $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$  realization and denote by  $\mathcal{R}_{\boldsymbol{\delta}, \mathbf{W}_{\text{op}}}(\cdot)$  the reward function conditional on the realization  $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$ . Using  $\mathbb{I}\{\cdot\}$  to denote the indicator function, it follows from the definition of  $\mathcal{R}(\cdot)$  in (1.17) that

$$\begin{aligned} \mathcal{R}_{\boldsymbol{\delta}, \mathbf{W}_{\text{op}}}(\mathbf{W}) &= \sum_{i=1}^N \sum_{d=1}^D (R_d - R_{d+1}) \mathbb{I} \left\{ \mathbf{w}_i^\top \boldsymbol{\delta} > G_{-i}^{(O+N-r_d)}(\mathbf{W}_{-i}, \mathbf{W}_{\text{op}}, \boldsymbol{\delta}) \right\} \\ &\leq \sum_{i=1}^N \sum_{d=1}^D (R_d - R_{d+1}) \mathbb{I} \left\{ \mathbf{w}_i^\top \boldsymbol{\delta} > G^{(O+1-r_d)}(\mathbf{W}_{\text{op}}, \boldsymbol{\delta}) \right\} \\ &= \sum_{i=1}^N \mathcal{R}_{\boldsymbol{\delta}, \mathbf{W}_{\text{op}}}(\mathbf{w}_i), \end{aligned}$$

where the inequality holds since  $G_{-i}^{(O+N-r_d)}(\mathbf{W}_{-i}, \mathbf{W}_{\text{op}}, \boldsymbol{\delta}) \geq G^{(O+1-r_d)}(\mathbf{W}_{\text{op}}, \boldsymbol{\delta})$  for all  $i \in \{1, \dots, N\}$  and for all  $d \in \{1, \dots, D\}$ . Taking an expectation over  $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$  allows us to conclude  $\mathcal{R}(\mathbf{W}) \leq \sum_{i=1}^N \mathcal{R}(\mathbf{w}_i)$ . ■



## *Why Skill Matters More for Top-Heavy Contests*

In the numerical experiments that we reported in Section 1.6, the performance of our strategic model was better in top-heavy contests than in double-up contests. While we would be reluctant to draw too many conclusions from this observation given the high variance of NFL games and the relatively few games in an NFL season, we do nonetheless believe skill is more important for top-heavy contests than for double-up contests. In fact, our numerical experiments also point to this. In particular, we report the optimal values of  $\lambda$  in Table A.3 in Appendix A.3.3 for the top-heavy, double-up and quintuple-up contests of Section 1.6. We see there that the top-heavy contests have a considerably higher value of  $\lambda^*$  than the double-up and quintuple-up contests whose values of  $\lambda^*$  are close to zero. This points to the fact that variance is important for top-heavy contests and that it plays a much smaller role for double-up contests. Moreover, because the variance of the fantasy points total includes the covariance term  $-2\mathbf{w}^\top \boldsymbol{\sigma}_{\delta, G(r'_d)}$  (see (1.14) for example), we know that our ability to estimate  $\boldsymbol{\sigma}_{\delta, G(r'_d)}$  is very important in determining the optimal entry  $\mathbf{w}^*$  for top-heavy contests. This was not the case for the double-up or quintuple-up contests we played, however, since  $\lambda^*$  was close to zero for them. This then can be viewed as a “structural” explanation for why top-heavy contests are more amenable to skill than double-up contests. (Of course, if the cutoff point for rewards in double-up contests was very high, e.g. the top 5% or 1% of entries, then we’d expect variance to start playing a more important role for these contests as well.)

## **A.2 Parimutuel Betting Markets and Their Relation to DFS**

In this appendix, we consider the setting of parimutuel betting markets, which can be viewed as a special case of our top-heavy DFS contests. Parimutuel betting is widespread in the horse-racing industry and has often been studied in the economics literature (Bayraktar and Munk 2017; Plott et al. 2003; Terrell and Farmer 1996; Thaler and Ziemba 1988) with the goal of studying the efficient markets hypothesis and the investment behavior of individuals in a simple and well-defined real-world setting. Our goal here is to use the simplified setting of parimutuel betting to

gain some insight into the structure of the optimal strategy for constructing multiple entries in a top-heavy DFS contest. The results we establish here are straightforward to obtain but are new to the best of our knowledge.

Consider then a horse race where there are  $H$  horses running and where there will be a single winner so there is no possibility of a tie. Each wager is for \$1 and we have  $\$N$  to wager. We let  $n_h$  denote the number of wagers, i.e., dollars, that we place on horse  $h$ . It therefore follows that  $\sum_{h=1}^H n_h = N$  and we use  $(n_1, n_2, n_3, \dots, n_H)$  to denote the allocation of our  $N$  wagers. We let  $q_h > 0$  denote the probability that horse  $h$  wins the race. We assume there are a total of  $O$  wagers made by our opponents so that  $\sum_{h=1}^H O_h = O$ , where  $O_h$  is the number of opponent wagers on horse  $h$ . We assume<sup>8</sup> the  $O_h$ 's are deterministic and known. The total dollar value of the wagers is then  $O + N$  and w.l.o.g. we assume the cut or “vig” taken by the race-track is zero. To make clear the connection between parimutuel and DFS contests, we can equate each horse to a feasible team in DFS.

### A.2.1 Parimutuel Winner-Takes-All Contests

In a parimutuel winner-takes-all (WTA) contest, the players that pick the winning horse share the total value wagered. In particular, if horse  $h$  wins, then our winnings are  $(O + N)n_h / (O_h + n_h)$  so that our share is proportional to the number of wagers we placed on  $h$ . If the winning horse is picked by no one, then we assume that none of the contestants receives a payoff. This is in contrast to the DFS setting where the reward  $O + N$  would be allocated to the highest ranked team that was submitted to the contest. This difference is quite significant and we will return to it later. For now, we note that it results in what we refer to as *reward independence* whereby the expected reward we earn from our wagers on horse  $h$  does not depend on  $n_{h'}$  for any  $h' \neq h$ .

**Definition A.3.** *Suppose our current portfolio of wagers has  $n_h = k$  with at least as yet one “unassigned” wager. Let  $\mu_h^{k+1}$  denote the expected gain we obtain from assigning this wager to horse  $h$ .*

---

<sup>8</sup>We could model the  $O_h$ 's as being stochastic but this makes the analysis unnecessarily complicated.

Reward independence allows us to easily compute  $\mu_h^{k+1}$ . In particular, we obtain

$$\mu_h^{k+1} = \underbrace{\frac{(k+1)q_h(O+N)}{O_h+k+1}}_{\text{“after”}} - \underbrace{\frac{kq_h(O+N)}{O_h+k}}_{\text{“before”}} = \frac{q_h(O+N)}{O_h+k} \times \frac{O_h}{O_h+k+1} \quad (\text{A.11})$$

It follows immediately from (A.11) that  $\mu_h^k$  is strictly decreasing in  $k$  for  $k = 1, \dots, N$  and for all horses  $h$ . We refer to this as the *saturation effect*. W.l.o.g., we assume hereafter that the horses have been sorted in decreasing order of the  $\mu_h^1$ 's so that  $\mu_1^1 \geq \mu_2^1 \geq \dots \geq \mu_H^1$ . This ordering and the saturation effect then imply the following partial ordering of the  $\mu_h^k$ 's:

$$\begin{array}{cccc} \mu_1^1 & \geq & \mu_2^1 & \geq \dots \geq & \mu_H^1 \\ \text{IV} & & \text{IV} & & \text{IV} \\ \mu_1^2 & & \mu_2^2 & & \mu_H^2 \\ \text{IV} & & \text{IV} & & \text{IV} \\ \vdots & & \vdots & & \vdots \\ \text{IV} & & \text{IV} & & \text{IV} \\ \mu_1^N & & \mu_2^N & & \mu_H^N \end{array}$$

This partial ordering suggests an approach for allocating the  $N$  wagers. We start by allocating the first wager to the first horse. (This is optimal in the  $N = 1$  case due to the presumed ordering of the horses.) We then consider allocating the second wager to either the first horse (and thereby replicating the first wager) or to the second horse. Because of the partial ordering, this must be optimal for the  $N = 2$  case. Suppose the optimal choice was to allocate the second wager to the second horse. Then the third wager should be allocated to either one of the first two horses (thereby replicating an earlier wager) or to the third horse. In contrast, if the optimal choice for the second wager was to replicate the first wager and place it on the first horse, then only the first and second horses need be considered for the third wager. These observations all follow from the partial ordering of the expected gains and they lead immediately to Algorithm 9, which handles the case of general  $N$ . It is a greedy algorithm where each successive wager is placed on the horse

with the highest expected gain given all previous wagers.

---

**Algorithm 9** Greedy Algorithm for Constructing a Portfolio of  $N$  Horses for Parimutuel WTA Contests

---

**Require:**  $\{\mu_h^k : 1 \leq h \leq H, 1 \leq k \leq N\}, N$

- 1:  $n_h = 0$  for all  $h = 1, \dots, H$  % initialize
  - 2:  $n_1 = 1$  % assign first wager to horse 1
  - 3: **for**  $j = 2 : N$
  - 4:  $\mathbb{A} = \{(h, n_h + 1) : n_h > 0\} \cup \{(h, 1) : n_h = 0, n_{h-1} > 0\}$  % next wager will be a replication or first horse
  - 5: % in ordering that has not yet been wagered upon
  - 6:  $h^* = \operatorname{argmax}_{\{h : (h,k) \in \mathbb{A}\}} \mu_h^k$  % horse in  $\mathbb{A}$  with highest expected gain
  - 7:  $n_{h^*} = n_{h^*} + 1$  % fill entry  $j$  with horse  $h^*$
  - 8: **end for**
  - 9: **return**  $(n_1, n_2, n_3, \dots, n_H)$
- 

The following proposition asserts the optimality of Algorithm 9.

**Proposition A.2.** *Algorithm 9 returns an optimal wager portfolio for the parimutuel WTA contest.*

**Proof.** Let  $\{n_h\}_{h=1}^H$  denote the output of Algorithm 9 and define  $\mathbb{H} := \{(h, k) : 1 \leq h \leq H, 1 \leq k \leq n_h\}$ . Note that  $|\mathbb{H}| = N$ . The expected reward of this wager allocation is  $\sum_{(h,k) \in \mathbb{H}} \mu_h^k$ . Let  $\mathbb{H}^c$  denote the complement of  $\mathbb{H}$  so that  $\mathbb{H}^c := \{(h, k) : 1 \leq h \leq H, 1 \leq k, (h, k) \notin \mathbb{H}\}$ . By construction of our greedy algorithm (which follows the partial ordering described after (A.11)) we have

$$\mu_h^k \geq \mu_{h'}^{k'} \text{ for all } (h, k) \in \mathbb{H} \text{ and for all } (h', k') \in \mathbb{H}^c. \quad (\text{A.12})$$

Consider now any alternative wager allocation  $\{n_h^{\text{alt}}\}_{h=1}^H$  where  $\sum_{h=1}^H n_h^{\text{alt}} = N$ . Define  $\mathbb{H}^{\text{alt}} := \{(h, k) : 1 \leq h \leq H, 1 \leq k \leq n_h^{\text{alt}}\}$  and note that  $\mathbb{H}^{\text{alt}} = \mathbb{H}_1 \cup \mathbb{H}_2$  where  $\mathbb{H}_1 := \mathbb{H}^{\text{alt}} \cap \mathbb{H}$  and  $\mathbb{H}_2 := \mathbb{H}^{\text{alt}} \cap \mathbb{H}^c$ . Since  $\mathbb{H}_1 \cap \mathbb{H}_2 = \emptyset$  the expected reward of the alternative wager allocation can be

written as

$$\begin{aligned} \sum_{(h,k) \in \mathbb{H}^{\text{alt}}} \mu_h^k &= \sum_{(h,k) \in \mathbb{H}_1} \mu_h^k + \sum_{(h',k') \in \mathbb{H}_2} \mu_{h'}^{k'} \\ &= \sum_{(h,k) \in \mathbb{H}} \mu_h^k - \left( \sum_{(h,k) \in \mathbb{H} \setminus \mathbb{H}_1} \mu_h^k - \sum_{(h',k') \in \mathbb{H}_2} \mu_{h'}^{k'} \right) \end{aligned} \quad (\text{A.13})$$

$$\leq \sum_{(h,k) \in \mathbb{H}} \mu_h^k. \quad (\text{A.14})$$

where (A.14) follows because the term in parentheses in (A.13) is non-negative which itself follows from (A.12) and the fact that  $|\mathbb{H} \setminus \mathbb{H}_1| = |\mathbb{H}_2|$ . (To see that  $|\mathbb{H} \setminus \mathbb{H}_1| = |\mathbb{H}_2|$  observe that  $|\mathbb{H} \setminus \mathbb{H}_1| = N - |\mathbb{H}_1|$  and  $\mathbb{H}_2 = \mathbb{H}^{\text{alt}} \setminus \mathbb{H}_1$  so that  $|\mathbb{H}_2| = |\mathbb{H}^{\text{alt}} \setminus \mathbb{H}_1| = N - |\mathbb{H}_1|$ .) The result now follows. ■

A natural question that arises when solving the  $N > 1$  problem is whether to replicate or diversify our wagers. Some insight into this issue can be provided in the  $N = 2$  case. From Proposition A.2, we know the optimal portfolio of wagers  $(n_1, n_2, n_3, \dots, n_H)$  is of the form  $(2, 0, 0, \dots, 0)$  or  $(1, 1, 0, \dots, 0)$ . A simple calculation that compares the expected values of these portfolios then implies

$$(n_1, n_2, n_3, \dots, n_H) = \begin{cases} (2, 0, 0, \dots, 0) & \text{if } \frac{\mu_1^1}{\mu_2^1} > \frac{O_1+2}{O_1} \\ (1, 1, 0, \dots, 0) & \text{otherwise.} \end{cases} \quad (\text{A.15})$$

We see from the condition in (A.15) that diversification becomes relatively more favorable when  $O_1$  is small so that horse 1 is not very popular among opponents. Our expected gain from replicating our wager on this horse declines as  $O_1$  decreases. For example, if  $O_1 = 0$ , then we would have made  $O + N$  if this horse won and the expected gain from replicating our wager on this horse equals 0. Diversification (by applying our second wager on horse 2) is clearly optimal in this case and this is reflected by the fact that  $\mu_1^1/\mu_2^1 > (O_1 + 2)/O_1 = \infty$  can never be satisfied.

In contrast, replication becomes relatively more favorable when  $O_1$  is large and horse 1 is

therefore very popular among opponents. This horse has a good chance of winning the race (since it has the highest  $\mu_1^1$ ) and by replicating our wager on it we can almost double our share of the total reward should the horse win. This follows because replicating our wager on horse 1 increases our total expected reward from  $(O+N)/(O_1+1)$  to  $(O+N)2/(O_1+2)$ , which is an approximate doubling when  $O_1$  is large. This must be close to optimal given that it was optimal to place our initial wager on horse 1 in the first place. Indeed this is reflected in the condition  $\mu_1^1/\mu_2^1 > (O_1+2)/O_1$  from (A.15), which will typically be satisfied when  $O_1$  is large since it is always the case that  $\mu_1^1/\mu_2^1 \geq 1$ .

It is perhaps worth mentioning at this point that unlike the parimutuel setting, there will typically be far more feasible teams than contestants in the DFS setting. This will be the case even for contests with several hundred thousand contestants. As such, in DFS contests we are invariably in the setting of small  $O_h$ 's, which results in diversification being favored.

## A.2.2 Extension to Parimutuel Top-Heavy Contests

The previous analysis for parimutuel WTA contests can be easily extended to more general parimutuel top-heavy contests. Suppose the horse that places  $d^{th}$  in the race carries a reward  $R_d$  for  $d = 1, \dots, D \leq H$ . This reward is then allocated to all contestants who placed wagers on this horse. Again, we assume that if no wagers were placed on it, then the reward is not allocated. We let  $q_h^d := \mathbb{P}\{\text{horse } h \text{ places } d^{th} \text{ in race}\}$  and then update our expression for the expected gain  $\mu_h^{k+1}$ . A simple calculation leads to

$$\mu_h^{k+1} = \frac{q_h^1 R_1 + q_h^2 R_2 + \dots + q_h^D R_D}{O_h + k} \times \frac{O_h}{O_h + k + 1}.$$

To maintain consistency with our earlier WTA setting, we can assume  $\sum_{d=1}^D R_d = O + N$ . Everything now goes through as before. In particular, Algorithm 9 still applies as does the proof of Proposition A.2, which guarantees its optimality. (We note that this also applies to double-up style parimutuel contests by simply assuming that for  $d = 1, \dots, D$  we have  $R_d = R$ , a constant.)

### A.2.3 Difference Between Parimutuel and DFS Contests

The key difference between DFS contests and our parimutuel setup is that in the DFS contests, the reward  $R_d$  is allocated to the submitted entry that has the  $d^{\text{th}}$  highest ranking among the *submitted* entries. As such, the prize is *always* awarded for each  $d = 1, \dots, D$ . To make the distinction concrete, suppose<sup>9</sup> there are 10 billion feasible teams (“horses”) in a given DFS contest with 500,000 entries and a WTA payoff structure. In this case, at most 0.005% of the feasible teams will have had wagers placed on them and so it’s very unlikely<sup>10</sup> that the ex-post best team will have received a wager. In our parimutuel setup, the  $O + N$  would simply not be allocated in that case. It is allocated in the DFS contest, however, and is allocated to the best performing team among the teams that were wagered upon. This might appear like a minor distinction but it is significant. In particular, reward independence no longer holds. To see this, consider a team that we have wagered upon and suppose it is ex-post the third ranked team out of the 10 billion possible entries. Again assuming a WTA structure, then that wager will win the reward of  $O + N$  only if there were no wagers placed by anyone else and ourselves in particular, on the first two horses. Our expected gain from the wager therefore depends on the other wagers we have placed. Because reward independence no longer holds, it means Proposition A.2 no longer holds even with updated  $\mu_h^k$ s.

Nonetheless, it is easy to see this loss of reward independence points towards a strategy of even greater diversification than that provided<sup>11</sup> by Algorithm 9. To see this, consider the following stylized setting. Suppose the space of feasible teams for the DFS contest can be partitioned into  $M$  “clusters” where  $M$  is “large”. The clustering is such that the fantasy points scores of teams

---

<sup>9</sup>To give these numbers some perspective, the typical top-heavy DFS contest that we entered had 24 NFL teams playing in a series of 12 games. We calculated the number of feasible entries for these contests to be approx.  $2 \times 10^{13}$  of which approx.  $7 \times 10^{10}$  utilized 99% of the budget. (In our experience, the vast majority of DFS contestants like to use > 98% of their budget when constructing their entries.)

<sup>10</sup>In Section 1.6, we describe our results from playing various DFS contests during the 2017 NFL regular season. In the top-heavy contests of each of the 17 weeks of the season, we found that the ex-post best performing team was *not* wagered upon!

<sup>11</sup>Algorithm 9 can yield portfolios anywhere on the spectrum from complete replication to complete diversification but, as mentioned earlier, the  $O_h$ ’s tend to be very small and often 0 in DFS contests and this strongly encourages diversification.

in the same cluster are strongly positively correlated (owing to significant player overlap in these teams) while teams in different clusters are only weakly correlated. Suppose cluster 1 is ex-ante the “best” cluster in that the teams in cluster 1 have the highest expected reward. Clearly, in the  $N = 1$  case, it would be optimal to wager on the best team in cluster 1. In the  $N = 2$  case, however, it may not be optimal to place the second wager on a team from cluster 1 even if this team has the second highest expected reward when considered by itself. This is because in some sense, the first wager “covers” cluster 1. To see this, suppose none of our opponents wagered on a team from cluster 1 and that ex-post, the best team was another team from cluster 1. While we did not wager on the ex-post best team, neither did anyone else and as we were the only contestant to wager on a team from cluster 1, there’s a good chance our team will win the reward of  $O + N$  (assuming again a WTA structure) due to the strong positive correlation among teams within a cluster. It therefore may make more sense to select a team from another cluster for our second wager.

#### A.2.4 Our Initial Approach for Top-Heavy DFS Contests

We now discuss our initial approach for tackling the top-heavy DFS problem where we must submit  $N$  entries to the contest. As mentioned towards the end of Section 1.5.1, our discussion here provides further support for Algorithm 3 and in particular, the imposition of diversification. In that direction, this section sheds light on how we arrived at Algorithm 3 by discussing our initial approach which allows for some replication.

Recalling the problem formulation from Section 1.5.1, we must solve for

$$\max_{\mathbf{W} \in \mathbb{W}^{|\mathbf{W}|}, |\mathbf{W}|=N} \mathcal{R}(\mathbf{W})$$

where

$$\mathcal{R}(\mathbf{W}) := \sum_{i=1}^{|\mathbf{W}|} \sum_{d=1}^D (R_d - R_{d+1}) \mathbb{P} \left\{ \mathbf{w}_i^\top \boldsymbol{\delta} > G_{-i}^{(r'_d)}(\mathbf{W}_{-i}, \mathbf{W}_{\text{op}}, \boldsymbol{\delta}) \right\}, \quad (\text{A.16})$$

$r'_d := O + |\mathbf{W}| - r_d$ ,  $G_{-i}^{(r)}$  is the  $r^{\text{th}}$  order statistic of  $\{G_o\}_{o=1}^O \cup \{F_j\}_{j=1}^{|\mathbf{W}|} \setminus F_i$  and  $\mathbf{W}_{-i} := \mathbf{W} \setminus \mathbf{w}_i$ . Our initial algorithm (which was not discussed in the main text) for solving (A.16) was a greedy-style





Hence, Algorithm 10 allows for the possibility of replication even if  $\gamma < C$ . On the other hand, in Algorithm 3, we force  $\mathbf{w}_i^*$  to be  $\mathbf{w}_{\lambda^*}$  and the output portfolio is completely diversified if  $\gamma < C$ . To gain intuition behind our advocacy of Algorithm 3 (over Algorithm 10), recall from Section A.2.3 that we do not have reward independence in DFS contests. This is why even an idealized greedy algorithm (where we could find an entry with the highest value-add) would not be optimal in general. This is in contrast to our parimutuel setup and led to us arguing that even more diversification might be called for in the DFS setting. An easy way to test this is to set  $\gamma < C$  and to simply add entry  $\mathbf{w}_{\lambda^*}$  to the portfolio  $\mathbf{W}$  without considering replicating one of the previously chosen entries. This then results in full diversification and the selection of  $N$  distinct entries (Algorithm 3). In all of our numerical experiments, we found that  $\gamma = C - 3 = 6$  was an optimal choice in both Algorithms 3 and 10 in that it led to final portfolios with the highest expected value in each case. We also found that for any fixed value of  $\gamma$ , the portfolio resulting from Algorithm 3 was approximately 5% to 20% better (in expected value terms) than the portfolio resulting from Algorithm 10. In light of our earlier comments, this was not very surprising and so Algorithm 3 is our preferred algorithm and the one we used in our numerical experiments.

### A.3 Further Details of Numerical Experiments

#### A.3.1 Benchmark Models for the Numerical Experiments of Section 1.6

Our two benchmark models do not model opponents and in fact, they (implicitly) assume the benchmarks  $G^{(r')}$  or  $G^{(r'_d)}$  are deterministic.

##### *Benchmark Model 1 (For Double-Up Contests)*

To optimize in the  $N = 1$  case, our first benchmark model simply maximizes the expected points total subject to the feasibility constraints on the portfolio. The resulting optimization model is a binary program (BP):

$$\max_{\mathbf{w} \in \mathbb{W}} \mathbf{w}^\top \boldsymbol{\mu}_\delta.$$

For  $N > 1$  (which is the case in our numerical experiments), we employ the greedy diversification strategy discussed in Section 1.5.1 but suitably adapted for the case where we do not model opponents. In particular, when optimizing over the  $i^{th}$  entry, we add the constraints that ensure the  $i^{th}$  entry can not have more than  $\gamma$  athletes in common with any of the previous  $i - 1$  entries. We use this benchmark model for the double-up contest because, according to our calibrated model, we are comfortably in the case (ii) scenario of Proposition 1.1 where, other things being equal, we prefer less variance to more variance.

### *Benchmark Model 2 (For Top-Heavy and Quintuple-Up Contests)*

The second benchmark model is similar to the first and indeed the objective functions are identical. The only difference is that we add a stacking constraint to force the model to pick the QB and main WR from the same team. We denote this constraint as “QB-WR”. Mathematically, the resulting BP for  $N = 1$  is:

$$\max_{\mathbf{w} \in \mathbb{W}, \text{QB-WR}} \mathbf{w}^\top \boldsymbol{\mu}_\delta.$$

Again for  $N > 1$ , we employ a suitably adapted version of the greedy diversification strategy from Section 1.5.1, i.e., the  $i^{th}$  entry can not have more than  $\gamma$  athletes in common with any of the previous  $i - 1$  entries. As discussed in Appendix A.1.1, the purpose of the stacking constraint is to increase the portfolio’s variance. This is because we are invariably “out-of-the-money” in these contests as we noted in Appendix A.1.4 and so variance is preferred all other things, that is, expected number of points, being equal. We note this model is very similar to the model proposed by Hunter et al. 2016 for hockey contests. They presented several variations of their model typically along the lines of including more stacking (or anti-stacking<sup>12</sup>) constraints, e.g. choosing athletes from exactly 3 teams to increase portfolio variance. We note that we could easily construct and back-test other similar benchmark strategies as well but for the purposes of

---

<sup>12</sup>An example of an anti-stacking constraint in hockey is that the goalie of team A cannot be selected if the attacker of team B was selected and teams A and B are playing each other in the series of games underlying the DFS contest in question. Such an anti-stacking constraint is also designed to increase variance by avoiding athletes whose fantasy points would naturally be negatively correlated.

our experiments, the two benchmarks above seemed reasonable points of comparison.

### A.3.2 Parameters and Other Inputs for the Numerical Experiments of Section 1.6

Our models rely on the following five input “parameters”: the expected fantasy points of the real-world athletes  $\boldsymbol{\mu}_\delta$ , the corresponding variance-covariance matrix  $\boldsymbol{\Sigma}_\delta$ , the stacking probability  $q$  from Section 1.3.2, the diversification parameter  $\gamma$  from Section 1.5.1 and the lower bound on the budget for accepting an opponent’s portfolio  $B_b$  from Section 1.3.3.

We obtain the estimate of  $\boldsymbol{\mu}_\delta$  from FantasyPros (FantasyPros 2017). This estimate is specific to each week’s games and we normally obtained it a day before the NFL games were played. We decompose the variance-covariance matrix  $\boldsymbol{\Sigma}_\delta$  into the correlation matrix  $\boldsymbol{\rho}_\delta$  and the standard deviations of the individual athletes  $\boldsymbol{\sigma}_\delta \in \mathbb{R}^P$ . The estimate of  $\boldsymbol{\rho}_\delta$  was obtained from RotoViz (RotoViz 2017) and  $\boldsymbol{\sigma}_\delta$  is estimated using the realized  $\boldsymbol{\delta}$  values from the 2016 and 2017 seasons. In particular, RotoViz provides correlations pegged to positions. For instance, using historical data, RotoViz has estimated the average correlation between the kicker of a team and the defense of the *opposing* team to be  $-0.50$  and the average correlation between the kicker of a team and the defense of the *same* team to be  $0.35$ . These estimates are not specific to any teams or athletes but are averages. (As a sanity check, we verified that the resulting correlation matrix  $\boldsymbol{\rho}_\delta$  is positive semi-definite.) Hence,  $\boldsymbol{\rho}_\delta$  does not change from week to week whereas  $\boldsymbol{\sigma}_\delta$  is updated weekly using the realized  $\boldsymbol{\delta}$  from the previous week.

It was also necessary to assume a distribution for  $\boldsymbol{\delta}$  as we needed to generate samples of this random vector. We therefore simply assumed that  $\boldsymbol{\delta} \sim \text{MVN}_P(\boldsymbol{\mu}_\delta, \boldsymbol{\Sigma}_\delta)$  where  $\text{MVN}_P$  denotes the  $P$ -dimensional multivariate normal distribution. Other distributions may have worked just as well (or better) as long as they had the same first and second moments, that is, the same  $\boldsymbol{\mu}_\delta$  and  $\boldsymbol{\Sigma}_\delta$ .

We also needed the input features  $\mathbf{X}$  and the realized  $\boldsymbol{p}$  values for the Dirichlet regressions. Such data is available on the internet. For example, the  $\boldsymbol{f}$  feature (point estimate of  $\boldsymbol{p}$ ) was available at the FantasyPros website and FanDuel contains the cost vector  $\boldsymbol{c}$  (before a contest starts) and the realized positional marginals  $\boldsymbol{p}$  (after a contest is over). We note that accessing the positional

marginals data at FantasyPros required us to create an account and pay for a six-month subscription costing \$65.94.

For the stacking probability  $q$ , we first note that we expect it to be contest-specific as we anticipate more stacking to occur in top-heavy style contests where variance is relatively more important than in double-up contests. Accordingly, we empirically checked the proportion of opponents who stacked using data<sup>13</sup> from the 2016-17 season for each contest-type. We then calibrated  $q$  to ensure that our Dirichlet-multinomial model for generating opponents implied the same proportion (on average). We estimated  $q$  to be 0.35, 0.25 and 0.20 for top-heavy, quintuple-up and double-up contests, respectively. In principle, one can perform out-of-sample testing to pick the “best”  $q$  in order to avoid in-sample over-fitting. However, given we are estimating a one-dimensional parameter using a reasonably large (and random) dataset, over-fitting was not our concern.

We set  $B_{ib} = 0.99B$  using a straightforward moment matching technique. In particular, we observed that most of our opponents used 100% of the budget and the average budget usage was around 99.5%. Using our Dirichlet-multinomial model for generating opponent portfolios, we simply calibrated  $B_{ib}$  so that the average budget usage was approximately 99.5%, which resulted in an estimate of  $B_{ib} = 0.99B$ .

We used  $\gamma = 6$  for the strategic and benchmark models across all contests since we found this value of  $\gamma$  to produce a near maximum within-model expected P&L. We note that the sensitivity of the expected P&L with respect to  $\gamma$  (around  $\gamma = 6$ ) is relatively low in all contest types for both strategic and benchmark portfolios. For instance, in the top-heavy contest with  $N = 50$ , the average weekly expected P&L (averaged over 17 weeks of the 2017-18 NFL season) for the strategic portfolio equals USD 342, 357, and 344 for  $\gamma$  equals 5, 6, and 7, respectively. Furthermore, if we allow  $\gamma$  to vary from week-to-week, i.e., in week  $t$ , pick  $\gamma_t \in \{5, 6, 7\}$  that results in the

---

<sup>13</sup>Because of the user interface of FanDuel.com, collecting entry-level data on each opponent was very challenging and had to be done manually. Instead of collecting data for each entry (which would have been too time consuming), we therefore collected data on 300 entries for each reward structure type. We also ensured the 300 entries were spread out in terms of their ranks so that they formed a representative sample of the entire population. For each contest type, we then estimated  $q$  by inspecting the 300 data-points and checking to see whether or not stacking of the QB and main WR was present.

maximum expected P&L in week  $t$ , then the average weekly expected P&L changes to USD 358 (an increase of only 1). This indicates the robustness of setting  $\gamma = 6$ .

We used a data sample from DFS contests in the 2016 NFL season to select an appropriate choice for  $\Lambda$  (the grid of  $\lambda$  values required for our optimization algorithms). In all of our contests, we set  $\Lambda = (0.00, 0.01, \dots, 0.20)$ . We could of course have reduced the computational burden by allowing  $\Lambda$  to be contest-specific. For example, in the case of quintuple-up contests, a choice of  $\Lambda = (0.00, 0.01, \dots, 0.05)$  would probably have sufficed since  $\lambda^*$  for quintuple-up was usually close to zero as discussed in Appendix A.3.3.

All of our experiments were performed on a shared high-performance computing (HPC) cluster with 2.6 GHz Intel E5 processor cores. Each week, we first estimated the parameters  $\mu_{G(r')}$ ,  $\sigma_{G(r')}^2$  and  $\sigma_{\delta, G(r')}$  (as required by Algorithms 1 and 3) via Monte Carlo simulation. We typically ran the Monte-Carlo for one hour each week on just a single core and this was sufficient to obtain very accurate estimates of the parameters. We note there is considerable scope here for developing more sophisticated variance reduction algorithms which could prove very useful in practical settings when portfolios need to be re-optimized when significant late-breaking news arrives. In addition, it would of course also be easy to parallelize the Monte-Carlo by sharing the work across multiple cores.

The BQPs were solved using Gurobi's (Gurobi Optimization 2016) default BQP solver and all problem instances were successfully solved to optimality with the required computation time varying with  $P$  (the number of real-world athletes),  $\lambda$  (see Algorithms 1 and 3) and the contest structure (double-up, quintuple-up or top-heavy). A typical BQP problem instance took anywhere from a fraction of a second to a few hundred seconds to solve. It was possible to parallelize with respect to  $\lambda$  and so we used 4 cores for double-up and quintuple-up contests and 8 cores for the top-heavy contests where the BQPs required more time to solve. Our experiments required up to 8 GB of RAM for double-up and quintuple-up contests and up to 16 GB of RAM for top-heavy contests.

### A.3.3 Additional Results from the 2017 NFL Season

Here we present some additional numerical results from the 2017 NFL season. First, we discuss results related to the P&L and portfolio characteristics and second, we discuss results related to the performance of the Dirichlet regressions.

#### *Additional P&L Results and Portfolio Characteristics*

Table A.1 displays the cumulative realized P&L for both models across the three contest structures during the 2017 season; see the discussion around Figure 1.1 in Section 1.6.

Table A.1: Cumulative realized dollar P&L for the strategic and benchmark models across the three contest structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. We invested \$50 per week per model in top-heavy series with each entry costing \$1. In quintuple-up the numbers were \$50 per week per model with each entry costing \$2 and in double-up we invested \$20 per week per model with each entry costing \$2. (We were unable to participate in the quintuple-up contest in week 1 due to logistical reasons.)

Week	Top-heavy		Quintuple-up		Double-up	
	Strategic	Benchmark	Strategic	Benchmark	Strategic	Benchmark
1	25.5	-39.5	-	-	3.13	15.13
2	-18.5	-77	-50	-50	-16.87	-4.87
3	85.24	-97	30	-60	-8.87	3.13
4	61.74	12.5	80	20	11.13	15.13
5	15.74	-34.5	30	-30	-8.87	-4.87
6	-7.26	-52.5	30	-70	-16.87	-8.87
7	92.74	-6.5	-10	20	-36.87	-24.87
8	170.74	0.5	-10	20	-28.87	-24.87
9	290.74	32.5	40	110	-8.87	-4.87
10	437.74	-17.5	0	60	-8.87	-4.87
11	406.74	189.5	20	130	-8.87	7.13
12	384.74	154.5	-30	80	-20.87	-8.87
13	347.74	112.5	10	40	-28.87	-8.87
14	341.74	87.5	-30	-10	-44.87	-24.87
15	320.74	59.5	-80	-30	-60.87	-44.87
16	317.74	29.5	-90	-50	-76.87	-60.87
17	280.74	91.5	-40	20	-60.87	-40.87

In Table A.2, we display the performance of each week's best realized entry (out of the 50 that were submitted) for the strategic and benchmark models corresponding to the top-heavy contests for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. Perhaps the most notable feature of Table A.2 is the variability of our highest rank entry from week to week.

Table A.2: Performance of each week’s best realized entry for the strategic and benchmark models corresponding to the top-heavy contests for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season.

Week	Total # of entries	Rank		Percentile		Reward (USD)	
		Strategic	Benchmark	Strategic	Benchmark	Strategic	Benchmark
1	235,294	2,851	40,421	1.21%	17.18%	8	1.5
2	235,294	46,909	26,728	19.94%	11.36%	1.5	2
3	235,294	429	5,715	0.18%	2.43%	25	5
4	238,095	2,566	864	1.08%	0.36%	8	15
5	208,333	46,709	24,695	22.42%	11.85%	2	3
6	208,333	10,466	59,45	5.02%	2.85%	4	5
7	208,333	139	647	0.07%	0.31%	20	10
8	208,333	550	5,767	0.26%	2.77%	10	5
9	178,571	138	3,103	0.08%	1.74%	25	5
10	178,571	211	60,938	0.12%	34.13%	20	0
11	178,571	7,480	24	4.19%	0.01%	4	100
12	148,809	4,301	11,994	2.89%	8.06%	4	3
13	190,476	8,263	6,759	4.34%	3.55%	4	4
14	166,666	5,503	8,566	3.30%	5.14%	4	3
15	142,857	5,189	7,601	3.63%	5.32%	4	3
16	142,857	1,424	6,835	1.00%	4.78%	6	3
17	142,857	11,920	87	8.34%	0.06%	3	30

This reflects the considerable uncertainty that is inherent to these contests. While the best strategic entry did well, we are confident that it could do much better (at least in expectation) by being more vigilant in updating parameter and feature estimates each week.

In Table A.3, we present various statistics of interest for the ex-ante optimal entry  $w_1^*$  of the strategic model across all three reward structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. It is interesting to note that none of the numbers vary much from week to week. It is also interesting to see how the top-heavy entry  $w_1^*$  has a lower mean and higher standard deviation than the corresponding entries in the double-up and quintuple-up contests. This is not surprising and is reflected by the fact that the top-heavy contests have a higher value of  $\lambda^*$  than the double-up and quintuple-up contests. This is as expected since variance is clearly more desirable in top-heavy contests and the optimization over  $\lambda$  recognizes this. It is also interesting to see that  $\lambda^*$  for the quintuple-up contests is approximately 0.

Table A.4 below shows the same results as Table A.3 except this time for the benchmark portfolios. It is interesting to see how similar<sup>14</sup> the statistics are for all three contest-types in Table A.4.

<sup>14</sup>Recall the benchmark quintuple-up model enforces stacking and hence, the optimal strategic quintuple-up entry



Table A.3: Various statistics of interest for the ex-ante optimal entry  $w_1^*$  of the strategic model across all three reward structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. Mean and StDev refer to the expected fantasy points and its standard deviation. (We were unable to participate in the quintuple-up contest in week 1 due to logistical reasons.)

Week	Top-heavy			Quintuple-up			Double-up		
	Mean	StDev	$\lambda^*$	Mean	StDev	$\lambda^*$	Mean	StDev	$\lambda^*$
1	124.45	24.76	0.03	-	-	-	127.35	20.00	0.00
2	120.70	26.94	0.05	124.22	23.69	0.01	123.58	21.50	0.01
3	115.08	27.54	0.04	121.15	22.34	0.00	120.54	21.03	0.01
4	114.18	27.54	0.04	121.85	21.67	0.00	121.85	21.67	0.00
5	115.48	23.65	0.05	123.22	20.49	0.00	123.22	20.49	0.00
6	106.45	27.96	0.05	118.82	21.37	0.00	118.28	17.53	0.02
7	108.69	29.82	0.06	120.53	22.08	0.00	119.34	21.02	0.02
8	107.61	28.26	0.04	120.73	20.22	0.00	120.61	20.11	0.01
9	105.16	28.52	0.05	116.42	21.83	0.01	115.94	19.48	0.01
10	110.25	28.99	0.05	123.36	21.49	0.00	122.74	19.42	0.02
11	107.79	29.43	0.04	123.28	20.88	0.00	122.44	19.88	0.02
12	117.60	25.47	0.03	124.90	19.72	0.00	124.90	19.72	0.00
13	116.70	29.30	0.03	123.10	22.65	0.00	122.20	19.44	0.02
14	111.50	28.15	0.04	119.70	20.68	0.00	119.40	19.33	0.01
15	116.80	27.79	0.06	129.30	19.30	0.00	129.30	19.30	0.00
16	117.60	26.38	0.04	122.20	23.40	0.01	120.90	16.96	0.02
17	110.70	27.68	0.07	126.80	19.02	0.00	126.40	17.29	0.01
<b>Average</b>	<b>113.34</b>	<b>27.54</b>	<b>0.05</b>	<b>122.47</b>	<b>21.30</b>	<b>0.00</b>	<b>122.29</b>	<b>19.66</b>	<b>0.01</b>

Indeed these statistics are similar to the statistics for the double-up and quintuple-up strategic portfolios in Table A.3 which is not surprising because the value of  $\lambda^*$  in those contests was close to 0. (We know that when  $\lambda^*$  is close to 0, then there is less value to being able to model opponents accurately. This merely reinforces the view that our strategic model adds more value in top-heavy style contests.)

In Table A.5, we present some information on the QB selected by the best performing entry of the strategic and benchmark models in the top-heavy contests for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. It's clear that, on average, the strategic model picks less popular QBs than the benchmark model - an average  $p_{QB}$  of 6.88% for the strategic model versus an average of 12.74% for the benchmark model. In addition, QBs picked by the strategic model cost less (approx. 3% lower on average) and have lower expected fantasy points

might differ from the optimal benchmark portfolio even if  $\lambda^* = 0$ . There is no enforced stacking in the benchmark model for double-up and so if  $\lambda^* = 0$  for the strategic double-up model, then the two entries will coincide and have same expected fantasy points.

Table A.4: Mean fantasy points and its standard deviation for the first optimal entry  $w_1^*$  of the benchmark model across all three reward structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. (We were unable to participate in the quintuple-up contest in week 1 due to logistical reasons.)

Week	Top-heavy		Quintuple-up		Double-up	
	Mean	StDev	Mean	StDev	Mean	StDev
1	125.71	19.28	-	-	127.35	20.00
2	122.65	23.70	122.65	23.70	124.24	23.19
3	117.28	23.53	117.28	23.53	121.15	22.34
4	118.07	21.67	118.07	21.67	121.85	21.67
5	120.17	20.16	120.17	20.16	123.22	20.49
6	116.71	18.97	116.71	18.97	118.82	21.37
7	118.35	22.37	118.35	22.37	120.53	22.08
8	119.12	20.04	119.12	20.04	120.73	20.22
9	113.60	21.78	113.60	21.78	116.51	21.40
10	121.85	22.80	121.85	22.80	123.36	21.49
11	121.00	21.94	121.00	21.94	123.28	20.88
12	121.40	22.03	121.40	22.03	124.90	19.72
13	120.40	23.39	120.40	23.39	123.10	22.65
14	117.80	20.81	117.80	20.81	119.70	20.68
15	126.40	20.96	126.40	20.96	129.30	19.30
16	119.90	22.10	119.90	22.10	122.30	22.70
17	125.30	18.88	125.30	18.88	126.80	19.02
<b>Average</b>	<b>120.34</b>	<b>21.44</b>	<b>120.00</b>	<b>21.57</b>	<b>122.77</b>	<b>21.13</b>

(approx. 9% lower on average) than the QBs picked by the benchmark model. To put the cost numbers in perspective, the budget that was available for entry was set by the contest organizers at  $B = 60,000$ .

We end this appendix with two anecdotes highlighting top-heavy contests where our strategic portfolio went against the “crowd” and was successful in doing so. In week 3, our strategic model selected an entry that consisted of some crowd favorites, in particular Tom Brady (QB) and A.J. Green as one of the three WRs. The entry also included four underdog picks from the Minnesota Vikings: two WRs (S. Diggs and A. Thielen), the kicker K. Forbath and the defense. Each of these four picks were expected to be chosen by less than 5% of our opponents and by choosing four players from the same team, the entry was stacked which resulted in a reasonably high variance. The Minnesota Vikings ended up having a good game, winning 34-17 at home against Tampa Bay. Our entry ended up ranking 429<sup>th</sup> out of 235,294 entries in total. In contrast, none of the benchmark entries were similar to this team. While some of them picked Thielen, none of them

Table A.5: Characteristics of the QB picked by the best performing entry of the strategic and benchmark models in the top-heavy contests for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season.

Week	Strategic				Benchmark			
	QB	$p_{QB}$	Cost	$\mu_\delta$	QB	$p_{QB}$	Cost	$\mu_\delta$
1	D. Carr (OAK)	9.80%	7700	18.63	R. Wilson (SEA)	7.30%	8000	20.23
2	A. Rodgers (GB)	9.80%	9100	26.19	M. Ryan (ATL)	10.90%	8200	24.28
3	T. Brady (NE)	7.20%	9400	20.72	A. Dalton (CIN)	2.40%	6800	15.85
4	R. Wilson (SEA)	13.20%	7900	21.19	A. Dalton (CIN)	3.50%	7100	17.19
5	J. Brissett (IND)	4.40%	7000	15.79	A. Rodgers (GB)	18.10%	9500	24.67
6	D. Carr (OAK)	1.10%	7500	16.48	D. Watson (HOU)	29.70%	7900	20.76
7	D. Brees (NO)	8.50%	8300	22.75	D. Brees (NO)	8.50%	8300	22.75
8	D. Watson (HOU)	3.70%	8000	17.30	A. Dalton (CIN)	9.70%	7600	19.02
9	R. Wilson (SEA)	16.30%	8500	24.52	R. Wilson (SEA)	16.30%	8500	24.52
10	C. Keenum (MIN)	0.70%	6800	15.50	B. Roethlisberger (PIT)	12.70%	7600	18.53
11	C. Keenum (MIN)	2.80%	7300	15.26	T. Brady (NE)	20.50%	8600	24.60
12	M. Ryan (ATL)	8.80%	7600	19.20	M. Ryan (ATL)	8.80%	7600	19.20
13	C. Keenum (MIN)	6.50%	7600	17.80	R. Wilson (SEA)	8.50%	8200	21.90
14	C. Keenum (MIN)	3.40%	7500	17.20	P. Rivers (LAC)	12.80%	8100	20.30
15	C. Keenum (MIN)	8.00%	7400	18.30	B. Roethlisberger (PIT)	13.70%	8000	21.10
16	A. Smith (KC)	7.50%	7800	19.20	C. Newton (CAR)	24.00%	8300	22.30
17	M. Ryan (ATL)	5.20%	7400	18.40	P. Rivers (LAC)	9.10%	8300	19.90
<b>Average</b>		<b>6.88%</b>	<b>7812</b>	<b>19.08</b>		<b>12.74%</b>	<b>8035</b>	<b>21.01</b>

picked Diggs, Forbath or the Vikings defense and so there was no strategic stacking.

Another such example can be found in week 10. One of the entries selected by the strategic model included an underdog QB (C. Keenum) again from the Minnesota Vikings. Keenum was predicted to be chosen by fewer than 3% of our opponents. This could be explained by his low  $\delta$ , i.e., his low expected fantasy points, and his low expected return. Choosing Keenum was quite a bold choice since the QB position is particularly important as QBs typically have the highest expected points and expected returns among all positions. In that particular week, Matthew Stafford was predicted to be the most popular QB with approx. 25% of opponents expected to pick him; see Figure A.2(a) in Appendix A.3.3. In addition to Keenum, our strategic entry was also stacked with 2 WRs (A. Thielen and S. Diggs) and the kicker (K. Forbath) all chosen from the Vikings and all predicted to be chosen by only approx. 5% of opponents. In the NFL game itself, the Vikings won 38-30 away to the Redskins with Keenum, Thielen, Diggs and Forbath scoring 26.06, 26.6, 15.8 and 10 fantasy points, respectively. Our entry ended up ranking 211<sup>th</sup> out of 178,571 entries. In contrast, all 50 benchmark entries chose Ben Roethlisberger as the QB, who was considerably

more popular than Keenum.

### *Dirichlet Regression Results*

Before concluding this section, we shed light on the performance of our Dirichlet-multinomial data generating process for modeling team selections of opponents and the corresponding Dirichlet regression introduced in Section 1.3 in terms of how well they predict the marginals  $\mathbf{p}_{\text{QB}}, \dots, \mathbf{p}_D$  and how well they predict the benchmark fantasy points  $G^{(r')}$  (double-up) and  $G^{(r'_d)}$  for  $d = 1, \dots, D$  (top-heavy). Our Dirichlet regression models used the features described in (1.5) and we validated this choice of features by evaluating its goodness-of-fit and comparing its out-of-sample performance against two “simpler” variations of the Dirichlet regression model. Specific details are deferred to Appendix A.3.4. In this subsection we focus instead on the key results and anecdotes we witnessed during the 2017-18 NFL season.

In Figure A.2, we show the performance of our approach in terms of predicting the QB marginals  $\mathbf{p}_{\text{QB}}$  for the top-heavy and double-up contests<sup>15</sup> in week 10 of the 2017 NFL season. First, we observe that in both top-heavy and double-up contests, our model correctly forecasted one of the top-picked QBs in week 10, namely Matthew Stafford. Second, we observe that our 95% prediction intervals (PI) contain around 95% of the realizations. This speaks to the predictive power of our statistical model. Of course, we expect roughly 5% of the realizations to lie outside the 95% intervals and we do indeed see this in our results. For example, in Figure A.2, out of a total of 24 QBs, the number of QBs that lie outside the intervals for top-heavy and double-up equal 2 and 1, respectively.

Of course, we did not do as well across all seventeen weeks as we did in week 10 but in general, our 95% prediction intervals contained 95% of the realizations. Over the course of the season, we did witness instances where our models under-predicted or over-predicted the ownerships by a relatively large margin. See Ryan Fitzpatrick in Figure A.2(b), for example. Accordingly, there is room for improvement, specifically in the quality of features provided to our Dirichlet regression.

---

<sup>15</sup>We do not present Dirichlet regression results corresponding to quintuple-up contests for brevity. We note that the results in quintuple-up are very similar.

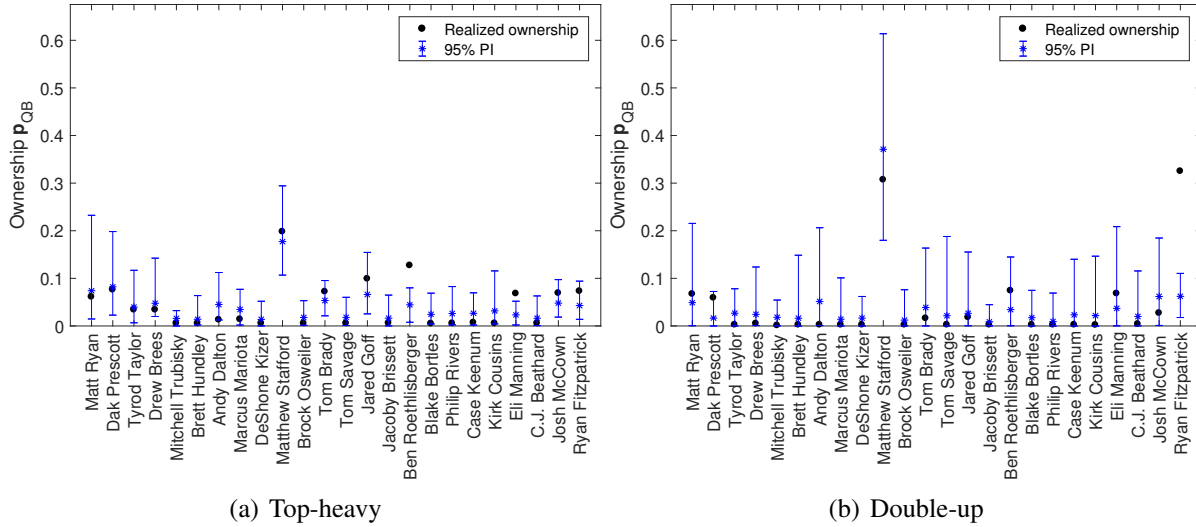


Figure A.2: Predicted and realized QB ownerships ( $p_{QB}$ ) for week 10 contests of the 2017 NFL season.

Retrospectively speaking, including a feature capturing the “momentum” of athletes, that is, how well they performed in the previous few weeks, would have been beneficial in terms of predicting opponents’ behavior. This statement is supported by multiple cases we noticed in the 2017 season. To give but one example, in week 9, Ezekiel Elliott (Dallas Cowboys) was picked by around 80% of our opponents in double-up but our 95% interval predicted 0% to 10%. It turns out that Elliott had performed extremely well in the two weeks prior to week 9. In fact, he was the top-scoring RB in week 7 and the second highest scoring RB in week 8.

We also would expect a significant improvement in predicting the player selections of our opponents if we were more proactive in responding to late developing news as discussed in Section 1.6. Such late developing news would typically impact our estimate of  $\mu_{\delta}$  which in turn would change both our optimal portfolios as well as our opponents’ team selections. Continuing on with the week 7 Fournette-Ivory case study from Section 1.6, due to our low estimate of the expected points of Ivory, we predicted his ownership to be below 5% with high probability (in top-heavy). In reality, around 15% of fantasy players in the top-heavy contest picked Ivory, which aligns with the sequence of events we discussed earlier. If we had updated our expected points estimate cor-

responding to Ivory to a value of 15 (from 6.78)<sup>16</sup>, we would have fared better. This is illustrated in Figure A.3(a) where we plot our original predictions (“before”) in blue and updated predictions (“after”)<sup>17</sup> in red. It’s clear that our original over-prediction of the ownership of Le’Veon Bell can be largely attributed to our stale  $\mu_\delta$  estimate of Ivory. As a side note, we can also observe that both our “before” and “after” predictions in Figure A.3(a) under-predict the ownership of Adrian Peterson (first point on the x-axis). We believe the reason for this is “momentum” (a feature we omitted from our Dirichlet regressions) as Peterson scored over 25 fantasy points in week 6 but was expected to score only 7 points, making him the RB with the highest points to cost ratio (among all week 6 RBs) that week.

An interesting illustration of the importance of good features can be found in Figure A.3(b) where we display the positional marginals for QBs in week 12’s double-up contest. We clearly over-predicted Tom Brady’s ownership and under-predicted Russell Wilson’s ownership. Perhaps the main reason for this was the point estimate  $f$  provided by FantasyPros (29.5% for Brady and 20.9% for Wilson) which was a feature in our Dirichlet regression. FantasyPros therefore severely overestimated the ownership of Brady and underestimated the ownership of Wilson and our regression model followed suit. However, it is well known in football that Tom Brady (arguably the greatest QB of all time) and the New England Patriots generally perform very poorly in Miami where his team were playing in week 12. It is no surprise then that the realized ownership of Tom Brady that week was very low. Unfortunately FantasyPros did not account for this in their prediction and so none of our features captured this well known Tom Brady - Miami issue. To confirm that it was indeed the FantasyPros point estimate that skewed our predictions we re-ran the regression after deducting 11% from Brady’s FantasyPros’ estimate (making it 18.5%) and adding it to Wilson’s estimate. The resulting fit is displayed in red in Figure A.3(b) and it’s clear that it does a much better job of predicting the realized ownerships.

In Figure A.4(a), we plot the realized fantasy points total against the rank  $r_d$  in the top-heavy

---

<sup>16</sup>An estimate of 15 for the expected points of a “main” RB is quite reasonable.

<sup>17</sup>For our updated predictions, we did not include the FantasyPros point estimate feature  $f$  in our Dirichlet regression model since we do not know how FantasyPros would have updated their estimate.

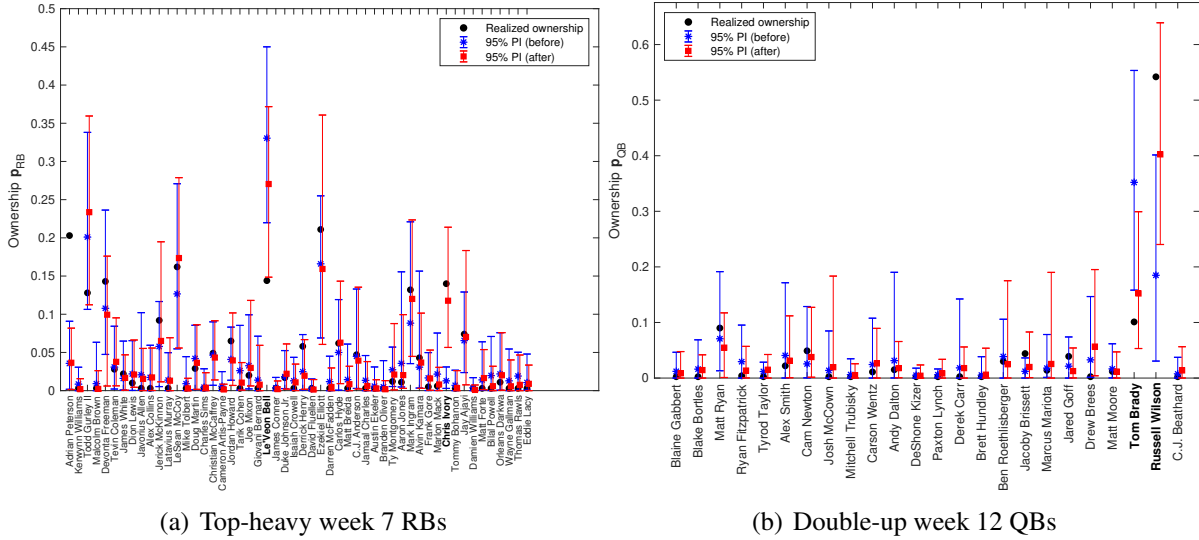


Figure A.3: Highlighting instances where the Dirichlet regression either under-predicted or over-predicted ownerships of some athletes (“before”) and what would have happened (“after”) if we had (a) reacted to breaking news or (b) access to better quality features that accounted for historical factors such as Brady’s poor track record in Miami.

contest of week 10. We also show our 95% prediction intervals for these totals as well as our 95% prediction intervals *conditional* on the realized value of  $\delta$ . These conditional prediction intervals provide a better approach to evaluate the quality of our Dirichlet-multinomial model for  $\mathbf{W}_{op}$  as they depend only on our model for  $\mathbf{W}_{op}$ . Not surprisingly, the interval widths shrink considerably when we condition on  $\delta$  and it is clear from the figure that we do an excellent job in week 10. In Figure A.4(b), we display the results for the double-up contests across the entire 17 weeks of the 2017 NFL season. While our model appears to perform well overall, there were some weeks where the realized points total was perhaps 3 conditional standard deviations away from the conditional mean. This largely reflects the issues outlined in our earlier discussions, in particular the need to better monitor player developments in the day and hours immediately preceding the NFL games.

### A.3.4 Model Checking and Goodness-of-Fit of the Dirichlet Regressions

In our numerical experiments of Section 1.6, we used the Dirichlet regression model of Section 1.3.1 to predict our opponents’ behavior and in particular, the positional marginals  $\mathbf{p}_{QB}, \dots, \mathbf{p}_D$

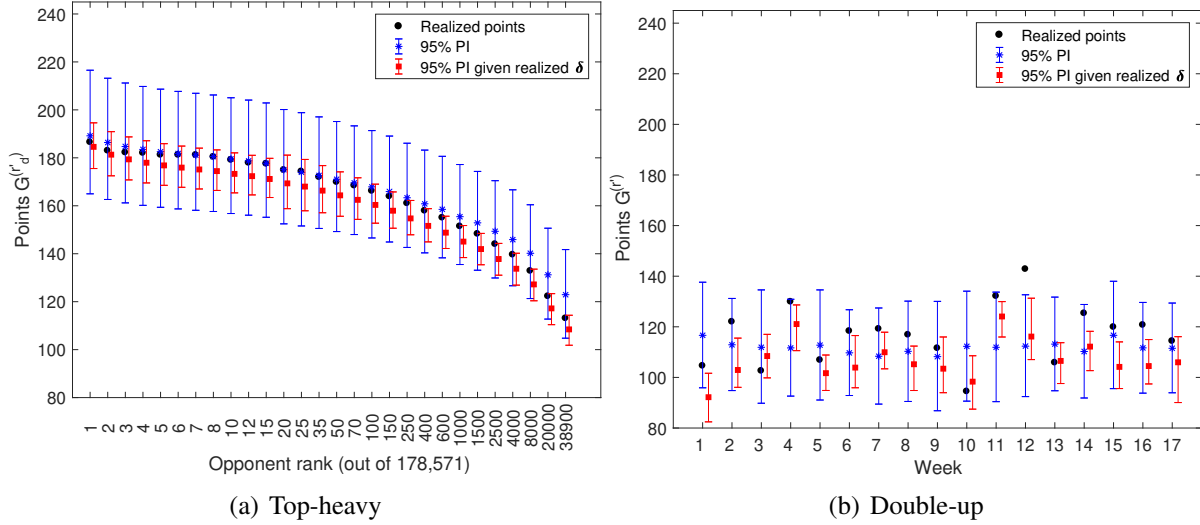


Figure A.4: Predicted and realized portfolio points total of various opponent ranks for (a) the top-heavy contest in week 10 and (b) all weeks of the double-up series during the 2017 NFL season. For double-up, the rank of interest for each week was around 13,000 and the number of opponents was around 30,000.

of the players in our opponents' team selections. In Appendix A.3.3, we discussed the performance of these Dirichlet regressions but not in a systematic fashion. In this appendix, we revisit this issue and evaluate the goodness-of-fit of our particular model and benchmark it against two simpler variations using the data from the 2017-18 NFL season. We first state the three variations of the Dirichlet regression model that we considered.

1. Variation 1 has 2 features: the player cost vector  $\mathbf{c}$  and the expected points vector  $\boldsymbol{\mu}_\delta$ .
2. Variation 2 has 1 feature: the point estimate  $\mathbf{f}$  of the positional marginals from FantasyPros.
3. Variation 3 has 3 features: the player cost vector  $\mathbf{c}$ , the expected points vector  $\boldsymbol{\mu}_\delta$ , and the point estimate  $\mathbf{f}$  of the positional marginals from FantasyPros.

All three variations also include the constant vector  $\mathbf{1}$  as an intercept. Variation 3 is therefore the model proposed in (1.5) and used in the numerical experiments of Section 1.6. Clearly, variations 1 and 2 are simpler versions of variation 3. We also note that we scaled the features to ensure they were roughly on the same scale as this helped STAN in fitting the model. In particular, we



divided the costs by 10,000 and divided the expected points by 25 so that all the feature values were typically in  $[0, 2]$ .

Before proceeding, we note our Dirichlet regression model is very simple and that modern software packages such as STAN can fit such models within seconds. One could therefore easily include additional features as well as interaction / higher-order terms with the goal of increasing the predictive power of the model. Our goal here was not to find the best set of features, however, but simply to find features that explain the data reasonably well. As we will show later in this appendix, the three variations listed above all explain the data quite well while variation 3 performed best in the cross-validation tests. These results justified the use of variation 3 in our numerical experiments in Section 1.6. The plots and discussion in Appendix A.3.3 also provide further support for the model. That said, the anecdotes from Appendix A.3.3 (which are reflected in the results in this subsection below) suggest how the performance of the model could have been significantly improved had we focussed more on the correctness of the features particularly in the light of new player developments before the games.

Finally, we note that the model checking and the cross-validation results done here are standard Bayesian techniques and are discussed for example in Gelman et al. 2013. We used data from the 2017-18 NFL season for both of these tasks.

### *Data Collection*

We were able to obtain complete data on the features  $\{\mathbf{f}_{\text{QB},t}, \mathbf{c}_{\text{QB},t}, \boldsymbol{\mu}_{\text{QB},t}\}_{t=1}^T$  where  $T = 17$  was the number of weeks in the season. There was a minor issue with obtaining the realized positional marginals and to explain this issue, we will focus on the QB position whose realized marginals are  $\{\mathbf{p}_{\text{QB},t}\}_{t=1}^{T-1}$ . Consider now week  $t$  with  $\mathbf{p}_{\text{QB},t} = \{p_{\text{QB},t}^k\}_{k=1}^{P_{\text{QB}}}$  and note that  $\sum_{k=1}^{P_{\text{QB}}} p_{\text{QB},t}^k = 1$ . If there were  $O$  opponents in a given contest in week  $t$ , then we would like to inspect their lineups to determine  $\mathbf{p}_{\text{QB},t}$ . Unfortunately, there was no way to do this in an automated fashion and so we had to resort to sampling their lineups. Fortunately, however, if a particular QB appeared in a lineup, then the web-site listed the realized positional marginal for that particular QB in the underlying DFS

contest. See Figure A.5. As a result, it would only be necessary to sample lineups until each QB appeared at least once. Unfortunately, some QBs were selected very rarely if at all and sampling sufficient lineups to find them proved too time consuming. Instead, we typically sampled approx. 100 lineups each week and we let  $\mathbb{C}_{\text{QB},t} \subseteq \{1, \dots, P_{\text{QB}}\}$  denote the set of QBs for which we collect the marginal in week  $t$ . Since  $\mathbb{C}_{\text{QB},t}$  is a subset of  $\{1, \dots, P_{\text{QB}}\}$ , it follows that  $\sum_{k \in \mathbb{C}_{\text{QB},t}} p_{\text{QB},t}^k \leq 1$ . Typical values of  $\sum_{k \in \mathbb{C}_{\text{QB},t}} p_{\text{QB},t}^k$  were 95% to 99%.










	QB Matthew Stafford QB 11 @ DET 35 FINAL \$7,800 SALARY	8.9% OWNED	27.12
	RB Alex Collins CIN 31 @ BAL 27 FINAL \$6,800 SALARY	9.6% OWNED	16.6
	RB Dion Lewis NYJ 6 @ NE 26 FINAL \$7,200 SALARY	25.3% OWNED	28.3
	WR JuJu Smith-Schuster CLE 24 @ PIT 28 FINAL \$7,300 SALARY	8.8% OWNED	30.8
	WR Marvin Jones Jr. GB 11 @ DET 35 FINAL \$7,300 SALARY	12.3% OWNED	16.1
	WR Keenan Allen OAK 10 @ LAC 30 FINAL \$8,600 SALARY	27% OWNED	29.8
	TE Jack Doyle HOU 13 @ IND 22 FINAL \$5,400 SALARY	10% OWNED	11.8
	K Matt Bryant CAR 10 @ ATL 22 FINAL \$5,000 SALARY	6.3% OWNED	19
	D Washington Redskins WAS 10 @ NYG 18 FINAL \$4,600 SALARY	4% OWNED	7

Figure A.5: Screenshot of a web-page from FanDuel.com when we click on an opponent lineup. The lineup has 9 athletes. For each athlete selected in the lineup we can observe the realized positional marginal of that athlete in the underlying contest. For example in the contest corresponding to this screenshot the realized positional marginal of Matthew Stafford equals 8.9%.

We defined the vector of *collected* marginals for week  $t$  as  $\hat{p}_{\text{QB},t} := [p_{\text{QB},t}^k]_{k \in \mathbb{C}_{\text{QB},t}}$  and the corresponding vector of FantasyPros estimates  $\hat{f}_{\text{QB},t} := [f_{\text{QB},t}^k]_{k \in \mathbb{C}_{\text{QB},t}}$ . Both of these vectors are then re-scaled so that they sum to 1. We similarly define  $\hat{c}_{\text{QB},t} := [c_{\text{QB},t}^k]_{k \in \mathbb{C}_{\text{QB},t}}$  and  $\hat{\mu}_{\text{QB},t} := [\mu_{\text{QB},t}^k]_{k \in \mathbb{C}_{\text{QB},t}}$  and then use these features to fit the three Dirichlet regression models using non-informative priors

for  $\beta_{\text{QB}}$ ; see (1.6). While this data collection procedure might introduce some bias in the estimation of  $\beta_{\text{QB}}$ , we expect this to be a second order effect as  $\sum_{k \in \mathbb{C}_{\text{QB},t}} P_{\text{QB},t}^k$  and  $\sum_{k \in \mathbb{C}_{\text{QB},t}} \hat{f}_{\text{QB},t}$  were (before scaling) always close to 1. That said, further investigation of this may be worth pursuing.

### *Model Checking and Posterior Predictive Checks*

The purpose of model checking and posterior predictive checks is to obtain a general idea of how well the model in question explains the data and what its weaknesses are. It may be viewed as a form of checking for internal consistency; see Gelman et al. 2013. For each possible tuple of the form (model variation, reward structure, position), we fit a Bayesian Dirichlet regression model with STAN. We ran each of 4 MCMC chains for 1,000 iterations and then discarded the first 500 iterations from each one. This left us with 2,000 posterior samples of the corresponding  $\beta$  parameter. All of our  $\hat{R}$  values<sup>18</sup> were between 1.00 and 1.05, indicating the MCMC chains had mixed well.

### *Marginal Posterior Predictive Checks*

For each of the 2,000 posterior samples of  $\beta$ , we generated a sample of all of the positional marginals using the appropriate Dirichlet distribution and then used these samples to construct 95% posterior intervals for the marginals. We then computed the proportion of times the 95% posterior intervals contain the true realized values. For each (variation, reward structure, position) tuple, we computed a summary statistic as follows.

As before, we will use the QB position to explain our procedure. There were  $T = 17$  weeks and for each week  $t$ , there were  $P_{\text{QB},t}$  QBs available for selection. Hence there were  $\sum_{t=1}^T P_{\text{QB},t}$  QB “instances” in total and for each such instance, we know the true realized marginal from the real-world data that we used to fit the model. We also have the posterior samples for that instance and therefore a 95% posterior interval for it. If the realized marginal is in the 95% posterior interval, then we assign the instance the value 1. Otherwise we assign it the value 0. The summary statistic

---

<sup>18</sup> $\hat{R}$  is a commonly used metric to determine how well the Markov chains have mixed. The closer  $\hat{R}$  is to 1 the better and a common rule of thumb is that an  $\hat{R}$  between 1.00 and 1.05 indicates that the chains have mixed sufficiently well.

Table A.6: Posterior predictive test summary statistic for each variation (denoted by V1, V2, and V3) of the Dirichlet regression model across all reward structures corresponding to the QB, RB, and WR positions.

	QB			RB			WR		
	V1	V2	V3	V1	V2	V3	V1	V2	V3
Top-heavy	0.96	0.97	0.96	0.95	0.97	0.95	0.96	0.96	0.96
Quintuple-up	0.95	0.93	0.95	0.95	0.95	0.95	0.94	0.95	0.95
Double-up	0.93	0.93	0.94	0.94	0.95	0.95	0.94	0.94	0.94

is then the average of these binary indicators over all  $\sum_{t=1}^T P_{QB,t}$  instances. The summary statistic for each combination of model variation and reward structures is shown in Table A.6 for the QB, RB, and WR positions<sup>19</sup>. The three model variations seem to pass this check (at least at the aggregate level) as each of the summary statistics lie between 93% and 97%.

### *Most-Picked Athlete Predictive Checks*

We also computed predictive checks of the test quantity “most-picked athlete” for each combination of model variation, reward structure, position, and week. To see how these p-values were computed, consider a specific combination where the position (as usual) is the QB and the week is week  $t$ . Each posterior sample of  $\mathbf{p}_{QB,t}$  has a maximum value corresponding to the most popular QB that week in that posterior sample. We take all of these maximum values and compute the percentage of them that exceeded the *realized*<sup>20</sup> maximum. Ideally, the resulting percentile should be away from the extremes, i.e., 0 and 1. They are reported in Tables A.7 (top-heavy) and A.8 (double-up) below for the QB, RB and WR positions. We omitted the other positions and quintuple-up contests for the sake of brevity. Highlighted instances correspond to percentiles less than 2.5% (blue) or greater than 97.5% (red). While we would expect to see extreme values approx. 5% of the time even if the model in question was correct, we see such extreme values approx. 12.5% of the time for the top-heavy contests and 19.5% of the time for the double-up contests. While variation 3 does perform the best of the models on this test, there is clearly some room for improvement here. It is

<sup>19</sup>The results are similar for the other positions and are not shown for the sake of brevity.

<sup>20</sup>The realized maximum is the percentage of people who picked the most popular QB that week in that contest. So for example, if Matthew Stafford was the most popular QB in the week 10 top-heavy contest with 23% of contestants picking him, then the realized maximum for that week was 23%.

interesting to note that in the double-up contests, the extreme values (when they occur) are almost invariably on the low end, i.e., less than 2.5%. This means that in these instances, the Dirichlet regression model is predicting that the most popular player in the given position will be considerably less popular among opponents than the realized most popular player in that position.

There are two obvious directions for improving the model performance in light of these results. First of all, we have outlined in Section 1.6 some of our occasional failures to obtain accurate data for the features or to adjust features to account for relevant information that would be known to most DFS players and in particular, our DFS opponents. For example, in Appendix A.3.3, we discussed the failure of the FantasyPros feature  $f$  to account for Russell Wilson’s popularity in week 12 double-up – in part because it also failed to account for Tom Brady’s well-known difficulties with playing in Miami. As depicted in Figure A.3(b), Wilson’s realized ownership that week was over 50% and so this was the realized maximum in week 12 for the QB position in double-up contests. Given our feature values that week and in particular the point estimate  $f$ , our fitted models were generally unable to produce such a high value in the posterior samples of the most-picked QB. As a result, we observed the low values that we see for the QB position in week 12 in Table A.8. In contrast, we can see from Figure A.2 that we had no difficulty in predicting the popularity of the most popular QB in the week 10 double-up and top-heavy contests. It is not surprising then to see that the week 10 QB results in Tables A.7 and A.8 are not<sup>21</sup> extreme. It is no surprise then that our explanation for the less than perfect results here are explained by a combination of occasionally inaccurate feature data as well as possibly missing other useful features, e.g. momentum. It should be clear, however, that these are issues with the features and occasional accuracy of the features rather than a problem with our Dirichlet regression, which certainly seems to be the right way to model this problem.

---

<sup>21</sup>While it is clear from Figure A.2(b) that we severely underestimated the ownership of Ryan Fitzpatrick in week 10 double-up, this isn’t reflected in Table A.8 because we are focusing on the most popular player in that table and we did predict correctly a very popular player that week, i.e., Matthew Stafford.

Table A.7: Bayesian  $p$ -values for the test statistic “most-picked athlete” for each variation of the Dirichlet regression model and each week corresponding to the QB, RB, and WR positions in the top-heavy reward structure.

Week	QB			RB			WR		
	V1	V2	V3	V1	V2	V3	V1	V2	V3
1	0.77	0.88	0.77	<b>0.98</b>	0.95	<b>0.99</b>	0.90	0.86	0.90
2	0.92	0.95	0.94	0.46	0.48	0.48	0.37	0.71	0.46
3	0.37	0.91	0.46	0.34	0.90	0.57	0.27	0.84	0.35
4	0.22	0.34	0.21	0.61	0.90	0.74	0.15	0.33	0.22
5	0.78	0.95	0.87	0.28	0.87	0.40	0.14	0.81	0.20
6	<b>0.01</b>	<b>0.02</b>	<b>0.01</b>	0.61	0.64	0.65	0.04	0.44	0.07
7	0.74	0.36	0.71	0.92	0.96	0.96	0.44	0.93	0.60
8	<b>0.00</b>	<b>0.02</b>	<b>0.00</b>	<b>0.98</b>	<b>0.98</b>	<b>0.99</b>	0.14	0.13	0.21
9	0.87	0.87	0.91	0.20	0.32	0.24	0.49	0.91	0.57
10	0.09	0.87	0.14	0.65	0.27	0.70	<b>1.00</b>	0.44	<b>1.00</b>
11	0.68	0.73	0.75	0.15	0.17	0.17	0.80	0.95	0.80
12	0.74	0.80	0.82	0.84	<b>1.00</b>	0.89	0.94	0.84	0.89
13	0.30	0.51	0.39	0.43	0.83	0.44	0.06	0.16	0.06
14	0.07	0.21	0.09	0.72	0.63	0.70	0.12	<b>0.98</b>	0.16
15	0.96	0.96	0.96	0.80	0.89	0.88	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>
16	0.03	0.22	0.04	0.11	0.33	0.11	0.35	0.77	0.46
17	0.78	0.78	0.77	<b>0.99</b>	0.29	0.97	0.09	0.07	0.09

Table A.8: Bayesian  $p$ -values for the test statistic “most-picked athlete” for each variation of the Dirichlet regression model and each week corresponding to the QB, RB, and WR positions in the double-up reward structure.

Week	QB			RB			WR		
	V1	V2	V3	V1	V2	V3	V1	V2	V3
1	0.41	0.49	0.40	0.07	0.54	0.59	<b>0.02</b>	0.08	0.07
2	0.15	0.40	0.35	0.04	<b>0.02</b>	0.03	<b>0.01</b>	0.08	0.06
3	0.12	0.22	0.11	0.05	0.72	0.65	<b>0.00</b>	0.09	0.07
4	0.18	0.15	0.18	0.03	0.15	0.11	0.07	0.13	0.11
5	0.03	0.21	0.06	0.03	0.65	0.39	0.15	0.67	0.58
6	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.36	0.45	0.33	<b>0.02</b>	0.12	0.04
7	0.64	0.55	0.53	0.09	0.24	0.15	0.32	0.73	0.47
8	0.25	0.18	0.27	0.29	0.46	0.48	0.03	<b>0.02</b>	<b>0.02</b>
9	0.80	0.66	0.86	0.05	0.04	0.05	0.19	0.25	0.24
10	0.27	0.70	0.54	0.03	0.03	0.05	0.58	0.11	0.13
11	<b>0.01</b>	<b>0.00</b>	<b>0.01</b>	<b>0.02</b>	<b>0.02</b>	<b>0.02</b>	0.36	0.36	0.32
12	<b>0.01</b>	0.06	0.06	0.07	0.93	0.88	0.24	0.21	0.22
13	0.04	0.11	0.07	0.09	0.37	0.27	0.04	0.08	0.07
14	0.08	0.09	0.06	<b>0.01</b>	<b>0.01</b>	<b>0.01</b>	<b>0.01</b>	0.47	0.25
15	0.41	0.43	0.56	0.09	0.44	0.44	0.97	<b>1.00</b>	<b>1.00</b>
16	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.02</b>	0.05	0.03	0.32	0.58	0.43
17	0.30	0.22	0.28	0.14	0.05	0.13	<b>0.00</b>	<b>0.01</b>	<b>0.00</b>

Table A.9: Comparing the three variations of the Dirichlet regression model using normalized cross-validation scores for each position and each reward structure.

Position	Top-heavy			Quintuple-up			Double-up		
	V1	V2	V3	V1	V2	V3	V1	V2	V3
D	<b>1.0000</b>	0.9376	0.9822	<b>1.0000</b>	0.9910	0.9998	0.9961	0.9895	<b>1.0000</b>
K	0.9700	0.9565	<b>1.0000</b>	0.9747	0.9843	<b>1.0000</b>	0.9649	0.9863	<b>1.0000</b>
QB	0.9998	0.9299	<b>1.0000</b>	0.9852	0.9808	<b>1.0000</b>	0.9708	0.9792	<b>1.0000</b>
RB	0.9978	0.9260	<b>1.0000</b>	0.9737	0.9878	<b>1.0000</b>	0.9607	0.9977	<b>1.0000</b>
TE	0.9983	0.9023	<b>1.0000</b>	0.9693	0.9655	<b>1.0000</b>	0.9744	0.9682	<b>1.0000</b>
WR	<b>1.0000</b>	0.9593	0.9911	0.9870	<b>1.0000</b>	0.9832	0.9806	<b>1.0000</b>	0.9887

### *Cross-Validation*

In order to compare the models in terms of out-of-sample performance, we perform leave-one-out cross-validation. For each combination of (model variation, reward structure, position), we do the following. We pick 16 weeks (the training set) out of the 17 available. We then fit the model on the data from those 16 weeks and use it to generate posterior samples of  $\beta$ . We then compute the log-likelihood on the data for the holdout week (the test set). We repeat this 17 times, each time with a different holdout week, and sum the 17 log-likelihoods to get a “raw” cross-validation score. See Chapter 7 (page 175) of Gelman et al. 2013 for further details.

The results are displayed in Table A.9 for all positions across all variations and reward structures. In the table we report a “normalized” cross-validation score to make the results easier to interpret. Consider the QB position and the top-heavy reward structure for example. The “raw” cross-validation scores are 742.71, 690.79, and 742.88 for variations 1, 2, and 3, respectively and we normalize these scores by simply dividing across by their maximum. Hence, a normalized score of 1 denotes the “winner” and this winner is displayed in bold font in Table A.9. Variation 3 clearly performs the best among the three models.

## Appendix B: Additional Details for Chapter 2

### B.1 Coalition Value

A key ingredient to use the concept of Shapley value is to define the value of a coalition, i.e., the value that a subset of players generates. The key decision here is with regards to the players that are *not* in the coalition, and this decision can be very context dependent.

We illustrate the issues with a simple example. An organization consists of  $\mathbb{P}$  players, and each player  $s$  takes the baseline action  $a = 1$ , or an enhanced action  $a = 2$ . Let  $\mathbf{a} = [a_s]_{s \in \mathbb{P}}$  denote the action of all the players. Let  $f(\mathbf{a}_S, S)$  denote the value generated by the organization when only  $S \subseteq \mathbb{P}$  players are present, and they take actions  $\mathbf{a}_S$ , with  $f(\mathbf{a}_\emptyset, \emptyset) = 0$ , and  $f(\mathbf{1}, \mathbb{P}) > 0$ , i.e., there is value to the organization even if all players take the baseline action. Next, consider a coalition  $\mathcal{X} \subseteq \mathbb{P}$  of players. We have two options for treating the players  $\mathbb{P} \setminus \mathcal{X}$ :

1. A player  $p \notin \mathcal{X}$  is treated as absent from the organization. Thus, the value  $v(\mathcal{X})$  is only generated by the players in  $\mathcal{X}$ , i.e.  $v(\mathcal{X}) = f(\mathbf{a}_\mathcal{X}, \mathcal{X})$ . This is the typical specifications in SV applications. In this setting,  $v(\emptyset) = 0$  since, with no players, there is no organization, and therefore, no value.
2. A player  $p \notin \mathcal{X}$  is assumed to be still present in the organization and taking the baseline action  $a = 1$ , i.e.  $v(\mathcal{X}) = f((\mathbf{a}_\mathcal{X}, \mathbf{1}_{\mathbb{P} \setminus \mathcal{X}}), \mathbb{P})$ . In this case,  $v(\emptyset) = f(\mathbf{1}, \mathbb{P}) > 0$ , the value generated when all players in the organization take the baseline action  $a = 1$ . Next, consider any player  $s \in \mathbb{P}$  that takes the baseline action  $a_s = 1$ , then  $v(\mathcal{X} \cup \{s\}) = v(\mathcal{X}) = f((\mathbf{a}_\mathcal{X}, \mathbf{1}_{\mathbb{P} \setminus \mathcal{X}}), \mathbb{P})$ ; consequently, the SV allocated to any such player is, identically, zero. Thus, all of the additional value  $f(\mathbf{a}, \mathbb{P}) - f(\mathbf{1}, \mathbb{P})$  is allocated to players taking action 2, i.e. players taking action 2 are able to “free ride” on the effort of the players taking the baseline action. The players taking the baseline action  $a = 1$  at most get a share of the baseline value  $f(\mathbf{1}, \mathbb{P})$ , which can be arbitrarily small. Note that this approach implicitly attributes the counterfactual value of



taking  $a = 2$ ; however, it attributes action  $a = 2$  too much of the value. To further highlight the “unfairness” of this free-riding, consider a setting  $f(\mathbf{1}, \mathbb{P}) = 0$ , i.e. the value is zero if *all* players take the baseline action, and, in addition, player  $s_0$  is essential for the organization, i.e.,  $f(\mathbf{a}_{\mathcal{X}}, \mathcal{X}) = 0$  if  $s_0 \notin \mathcal{X}$ . Suppose, the player  $s_0$  plays the baseline action  $a = 1$ , then  $s_0$  gets zero attribution even though she is essential.

This simple setting shows that if there is “value” in the baseline action, and we want to be able to attribute to the baseline action, we must define coalitions by ensuring that players  $\mathbb{P} \setminus \mathcal{X}$  are not contributing to value generation. In the online advertising setting, we can achieve this by assuming that the traffic exits the system with probability 1 when one employs a state-action pair  $(s, a) \notin \mathcal{X}$ .

To illustrate the implications of the two choices in a Markov chain setting, consider the network in Figure B.1. The total value in the system is 1. States 1 and 2 are both essential for completion; however, in state 2 it does not matter whether we take the baseline no-ad-action or the ad-action. Thus, any reasonable attribution scheme should assign a value of  $1/2$  to each of the states, and further assign a counterfactual adjusted value of  $1/2$  to the ad-action in state 1, and a counterfactual adjusted value of 0 to the ad-action in state 2.

Under the construct in the setting 2 where the “empty” coalition consists of taking the baseline “no-ad” in each state, the ad-action at state 1 receives the entire value, i.e. the baseline action in state 2 does not receive any attribution.

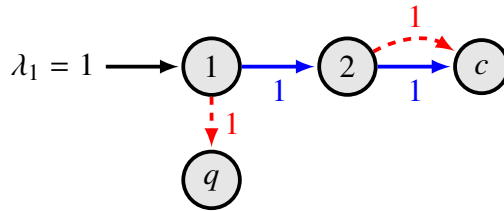


Figure B.1: The action space consists of two actions: no-ad-action ( $a = 1$ ) and ad-action ( $a = 2$ ). Solid blue (dashed red) lines denote transitions if the ad-action (no-ad-action) is taken. The advertiser takes the ad-action at state 1 w.p. 1 and takes the no-ad-action in state 2 w.p. 1, i.e.,  $\beta_1^2 = \beta_2^1 = 1$ .

## B.2 Proof of Theorem 2.1

**Proof of Theorem 2.1.** Counterfactual efficiency and linearity follow from (2.6) when used with the efficiency and linearity of SV, respectively. Counterfactual null player follows from (2.5). For counterfactual symmetry, consider  $(s, a) \in \mathbb{S} \times \mathbb{A}$  and  $(s', a') \in \mathbb{S} \times \mathbb{A}$  satisfying (A) and (B) and observe that

$$\begin{aligned}
\psi_s^{a, \text{Shap}} &= \sum_{\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s, a)\}} w_{|\mathcal{X}|} \times \{v(\mathcal{X} \cup \{(s, a)\}) - v(\mathcal{X} \cup \{(s, 1)^a\})\} \\
&= \sum_{\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s, a), (s', a')\}} w_{|\mathcal{X}|} \times \{v(\mathcal{X} \cup \{(s, a)\}) - v(\mathcal{X} \cup \{(s, 1)^a\})\} \\
&+ \sum_{\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s, a), (s', a')\}} w_{|\mathcal{X}|+1} \times \{v(\mathcal{X} \cup \{(s, a), (s', a')\}) - v(\mathcal{X} \cup \{(s, 1)^a, (s', a')\})\} \\
&= \sum_{\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s, a), (s', a')\}} w_{|\mathcal{X}|} \times \{v(\mathcal{X} \cup \{(s', a')\}) - v(\mathcal{X} \cup \{(s', 1)^{a'}\})\} \\
&+ \sum_{\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s, a), (s', a')\}} w_{|\mathcal{X}|+1} \times \{v(\mathcal{X} \cup \{(s, a), (s', a')\}) - v(\mathcal{X} \cup \{(s', 1)^{a'}, (s, a)\})\} \\
&= \sum_{\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s', a')\}} w_{|\mathcal{X}|} \times \{v(\mathcal{X} \cup \{(s', a')\}) - v(\mathcal{X} \cup \{(s', 1)^{a'}\})\} \\
&= \psi_{s'}^{a', \text{Shap}}.
\end{aligned}$$

To show uniqueness, consider  $\{\psi_s^a\}_{(s, a) \in \mathbb{S} \times \mathbb{A}}$  such that it satisfies the counterfactual axioms.

Motivated by (2.6), express  $\psi_s^a$  as

$$\psi_s^a = \underbrace{\psi_s^a + \pi_s^{a, \text{Shap}}(\mathcal{M}_{(s, a)}) - \pi_s^{a, \text{Shap}}(\mathcal{M}_{(s, a)})}_{=: \pi_s^a}$$

for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ . To show  $\psi_s^a = \psi_s^{a, \text{Shap}}$  for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ , it suffices to show that  $\pi_s^a = \pi_s^{a, \text{Shap}}$  for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ . We do so by proving  $\{\pi_s^a\}_{(s, a) \in \mathbb{S} \times \mathbb{A}}$  satisfies the four desirable properties of SV from Section 2.4.1 (recall  $\{\pi_s^{a, \text{Shap}}\}_{(s, a) \in \mathbb{S} \times \mathbb{A}}$  is the unique solution to those properties).

**Efficiency:** Since  $\pi_s^a = \psi_s^a + \pi_s^{a, \text{Shap}}(\mathcal{M}_{(s, a)})$  for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$  and  $\{\psi_s^a\}_{(s, a) \in \mathbb{S} \times \mathbb{A}}$  satisfies

counterfactual efficiency, we get

$$\sum_{(s,a) \in \mathbb{S} \times \mathbb{A}} \pi_s^a = \sum_{(s,a) \in \mathbb{S} \times \mathbb{A}} \psi_s^a + \sum_{(s,a) \in \mathbb{S} \times \mathbb{A}} \pi_s^{a, \text{Shap}}(\mathcal{M}_{(s,a)}) = v(\mathbb{S} \times \mathbb{A}).$$

**Symmetry:** Consider  $(s, a) \in \mathbb{S} \times \mathbb{A}$  and  $(s', a') \in \mathbb{S} \times \mathbb{A}$ . We need to show that

$$\underbrace{v(\mathcal{X} \cup \{(s, a)\}) = v(\mathcal{X} \cup \{(s', a')\}) \quad \forall \mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s, a), (s', a')\}}_{(\diamond)} \implies \pi_s^a = \pi_{s'}^{a'}.$$

We can use the fact that  $\{\psi_s^a, \psi_{s'}^{a'}\}$  satisfy counterfactual symmetry, i.e., if (A) and (B) hold for all  $\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s, a), (s', a')\}$ , then

$$\pi_s^a - \pi_s^{a, \text{Shap}}(\mathcal{M}_{(s,a)}) = \pi_{s'}^{a'} - \pi_{s'}^{a', \text{Shap}}(\mathcal{M}_{(s',a')}).$$

For the purposes of a contradiction, suppose that  $(\diamond)$  holds but  $\pi_s^a \neq \pi_{s'}^{a'}$ . It suffices to show that this results in  $\{\psi_s^a, \psi_{s'}^{a'}\}$  violating counterfactual symmetry. Suppose (A) and (B) hold. Given  $(\diamond)$ , statement (A) implies

$$v(\mathcal{X} \cup \{(s, 1)^a\}) = v(\mathcal{X} \cup \{(s', 1)^{a'}\}) \quad \forall \mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s, a), (s', a')\}.$$

Together with (B), this implies  $\pi_s^{a, \text{Shap}}(\mathcal{M}_{(s,a)}) = \pi_{s'}^{a', \text{Shap}}(\mathcal{M}_{(s',a')})$ :

$$\begin{aligned}
\pi_s^{a, \text{Shap}}(\mathcal{M}_{(s,a)}) &= \sum_{\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s,a)\}} w_{|\mathcal{X}|} \times \{v(\mathcal{X} \cup \{(s, 1)^a\}) - v(\mathcal{X})\} \\
&= \sum_{\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s,a), (s',a')\}} w_{|\mathcal{X}|} \times \{v(\mathcal{X} \cup \{(s, 1)^a\}) - v(\mathcal{X})\} \\
&+ \sum_{\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s,a), (s',a')\}} w_{|\mathcal{X}|+1} \times \{v(\mathcal{X} \cup \{(s, 1)^a, (s', a')\}) - v(\mathcal{X} \cup \{(s', a')\})\} \\
&= \sum_{\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s,a), (s',a')\}} w_{|\mathcal{X}|} \times \{v(\mathcal{X} \cup \{(s', 1)^{a'}\}) - v(\mathcal{X})\} \\
&+ \sum_{\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s,a), (s',a')\}} w_{|\mathcal{X}|+1} \times \{v(\mathcal{X} \cup \{(s', 1)^{a'}, (s, a)\}) - v(\mathcal{X} \cup \{(s, a)\})\} \\
&= \sum_{\mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s',a')\}} w_{|\mathcal{X}|} \times \{v(\mathcal{X} \cup \{(s', 1)^{a'}\}) - v(\mathcal{X})\} \\
&= \pi_{s'}^{a', \text{Shap}}(\mathcal{M}_{(s',a')}).
\end{aligned}$$

Hence, we require  $\pi_s^a = \pi_{s'}^{a'}$  for  $\{\psi_s^a, \psi_{s'}^{a'}\}$  to satisfy counterfactual symmetry, which contradicts  $\pi_s^a \neq \pi_{s'}^{a'}$ .

**Linearity:** Since  $\pi_s^a = \psi_s^a + \pi_s^{a, \text{Shap}}(\mathcal{M}_{(s,a)})$  for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$  and both  $\psi_s^a$  and  $\pi_s^{a, \text{Shap}}(\mathcal{M}_{(s,a)})$  satisfy linearity, it follows that  $\pi_s^a$  does so too.

**Null player:** Consider  $(s, a) \in \mathbb{S} \times \mathbb{A}$ . We need to show that

$$\underbrace{v(\mathcal{X} \cup \{(s, a)\}) = v(\mathcal{X}) \quad \forall \mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s, a)\}}_{(*)} \implies \pi_s^a = 0.$$

We can use the fact that  $\psi_s^a$  satisfies counterfactual null player, i.e.,

$$\underbrace{v(\mathcal{X} \cup \{(s, a)\}) = v(\mathcal{X} \cup \{(s, 1)^a\}) \quad \forall \mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s, a)\}}_{(\square)} \implies \pi_s^a = \pi_s^{a, \text{Shap}}(\mathcal{M}_{(s,a)}).$$

For the purposes of a contradiction, suppose that (\*) holds but  $\pi_s^a \neq 0$ . It suffices to show that this results in  $\psi_s^a$  violating counterfactual null player. Suppose (□) holds. Subtract  $v(\mathcal{X})$  from both

sides of  $(\square)$  to get

$$v(\mathcal{X} \cup \{(s, a)\}) - v(\mathcal{X}) = v(\mathcal{X} \cup \{(s, 1)^a\}) - v(\mathcal{X}) \quad \forall \mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s, a)\},$$

which combined with  $(*)$  yields

$$0 = v(\mathcal{X} \cup \{(s, 1)^a\}) - v(\mathcal{X}) \quad \forall \mathcal{X} \subseteq \{\mathbb{S} \times \mathbb{A}\} \setminus \{(s, a)\}.$$

This implies  $\pi_s^{a, \text{Shap}}(\mathcal{M}_{(s,a)}) = 0$  and hence, we require  $\pi_s^a = 0$  for  $\psi_s^a$  to satisfy counterfactual null player, which contradicts  $\pi_s^a \neq 0$ .

This completes the proof. ■

### B.3 Sensitivity of CASV Due to Finite Size of Data

In this appendix, we discuss the sensitivity of CASV due to the finite amount of data one observes in practice. Throughout the discussion, we assume there exists a true underlying Markov model  $\mathcal{M}$  but we only observe a finite dataset of  $D$  sample paths<sup>1</sup>  $\mathcal{D} := \{\mathcal{P}_i\}_{i=1}^D$ , where each path is sampled independently from  $\mathcal{M}$ . The goal is to estimate CASV using the data  $\mathcal{D}$  and understand the difference between the estimated CASV and the true CASV.

We use the data  $\mathcal{D}$  to compute an estimate  $\widehat{\mathcal{M}}$  for the Markov model  $\mathcal{M}$  (step 1) and then, use  $\widehat{\mathcal{M}}$  to estimate CASV (step 2). We followed this process for the results presented in Section 2.6. For step 2, we used Algorithm 4 with the inputs  $\widehat{\mathcal{M}}$  (estimated model) and  $N = 100,000$  (number of Monte Carlo samples). As discussed in Section 2.5.2,  $N = 100,000$  is large enough to obtain a stable estimate of CASV and hence, we do not discuss Monte Carlo error here. For ease of notation, we use  $\{\psi_s^{a, \text{Shap}}(\mathcal{M})\}_{(s,a) \in \mathbb{S} \times \mathbb{A}}$  and  $\{\psi_s^{a, \text{Shap}}(\widehat{\mathcal{M}})\}_{(s,a) \in \mathbb{S} \times \mathbb{A}}$  to denote the output of Algorithm 4 when the input model to the algorithm is  $\mathcal{M}$  and  $\widehat{\mathcal{M}}$ , respectively<sup>2</sup>.

<sup>1</sup>The notation  $D$  should not be confused with the number of Monte Carlo samples  $N$  used in Algorithm 4. Instead,  $D$  corresponds to the number of user paths we have in our real-world dataset, which is several million.

<sup>2</sup>To be technically correct, we should use the notation  $\widehat{\psi}_s^{a, \text{Shap}}(\mathcal{M})$  and  $\widehat{\psi}_s^{a, \text{Shap}}(\widehat{\mathcal{M}})$  to acknowledge the Monte Carlo

In order to understand the difference between the true CASV  $\{\psi_s^{a,\text{Shap}}(\mathcal{M})\}_{(s,a)\in\mathbb{S}\times\mathbb{A}}$  and the estimated CASV  $\{\psi_s^{a,\text{Shap}}(\widehat{\mathcal{M}})\}_{(s,a)\in\mathbb{S}\times\mathbb{A}}$ , we conducted the following simple yet informative experiment. We fix  $\mathcal{M}$  to be the model we estimated using real-world data in Section 2.6 and then repeat the following steps 1,000 times:

1. Sample a dataset  $\mathcal{D} := \{\mathcal{P}_i\}_{i=1}^D$  of  $D$  independent paths from  $\mathcal{M}$ .
2. Use  $\mathcal{D}$  to obtain the estimated Markov model  $\widehat{\mathcal{M}}$ .
3. Use  $\widehat{\mathcal{M}}$  in Algorithm 4 with  $N = 100,000$  to obtain an estimate of CASV.

We experiment with  $D \in \{100, 1000, 10000\}$  to understand how the finite sample size of data affects the estimate of CASV. In particular, we expect the distribution of the estimator to become more concentrated around the true value as we increase the sample size  $D$ .

The results are summarized in Figure B.2. As in Figure 2.4, we have condensed the state-action attributions into attributions to only actions by aggregating over states *after* computing state-action attributions. Note that the dotted red line in each subfigure of Figure B.2 is the attribution corresponding to the true model  $\mathcal{M}$ , i.e., it corresponds to the attribution displayed under CASV in Figure 2.4. For instance, the dotted red line in the no-ad column corresponds to an attribution of 50%, which matches the attribution to no-ad under CASV in Figure 2.4. The histogram in each subfigure characterizes the distribution of the estimator and the solid black line is the empirical mean of the distribution. Each column corresponds to a different ad action and each row corresponds to a different value of  $D$  (number of paths used to estimate the Markov model).

Clearly, as we expected, the distribution of the estimator concentrates around the true value as we increase the sample size  $D$ . This is perhaps not too surprising since the estimator is asymptotically consistent (because  $\widehat{\mathcal{M}} \rightarrow \mathcal{M}$  as  $D \rightarrow \infty$ ). However, it is remarkable that the estimator almost converges to the true value with a value of  $D$  as low as 10,000. Contrasting this sample size of 10,000 to the size of our real-world data (several million) highlights the robustness of our numerics presented in Section 2.6.3.

---

error in Algorithm 4 but as discussed earlier, we found the Monte Carlo error to be negligible with  $N = 100,000$ .

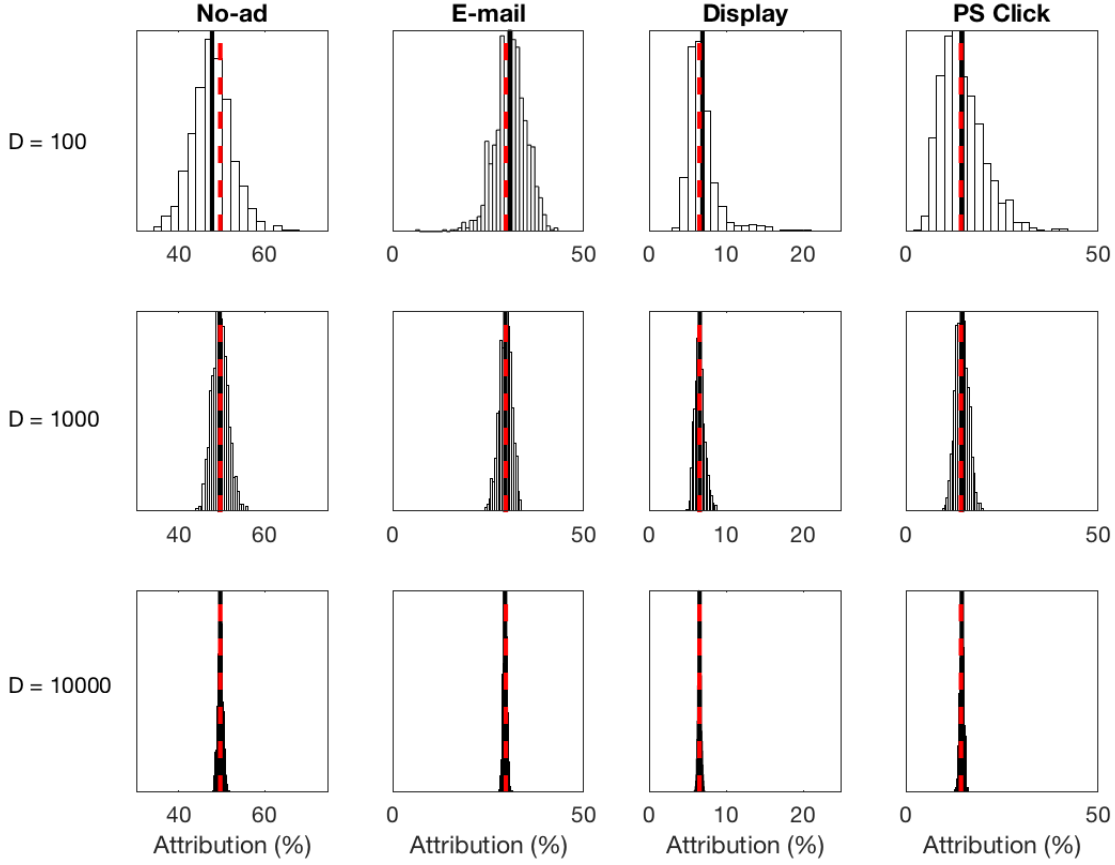


Figure B.2: Summary of the sensitivity experiment. Each column corresponds to an ad action and each row corresponds to a different value of  $D$ . The dotted red line denotes the true CASV and the histogram denotes the distribution of the estimator. The solid black line equals the empirical mean of the distribution. (We report the percentage attributions to each action by aggregating over states, i.e.,  $\sum_s \psi_s^a / \sum_{(s', a')} \psi_{s'}^{a'}$  for each  $a \in \mathbb{A}$ .)

Furthermore, the difference between the solid black line (empirical mean) and the dotted red line (true value) in Figure B.2 for  $D = 100$  suggests that CASV estimator might be biased. In fact, we believe that to be quite likely. Even if  $\widehat{\mathcal{M}}$  is an unbiased estimator of  $\mathcal{M}$ , it is not necessarily the case that  $\psi_s^{a, \text{Shap}}(\widehat{\mathcal{M}})$  is an unbiased estimator of  $\psi_s^{a, \text{Shap}}(\mathcal{M})$  since the mapping from  $\mathcal{M}$  to  $\psi_s^{a, \text{Shap}}$  is non-linear for an arbitrary  $(s, a) \in \mathbb{S} \times \mathbb{A}$ . Furthermore, characterizing such bias analytically is challenging due to the complicated nature of this mapping. Fortunately, given the enormous size of our real-world dataset (which is quite common in the context of online advertising where data

is relatively cheap to collect), we do not have to be concerned about this bias.

## B.4 Proofs of Theorem 2.3 and Proposition 2.2

**Proof of Theorem 2.3.** We split the proof into four parts.

**Step 1:** We use the fact that CASV equals the difference between two SVs

$$\begin{aligned}\psi_s^{a,\text{Shap}}(\mathcal{M}) &= \pi_s^{a,\text{Shap}}(\mathcal{M}) - \pi_s^{a,\text{Shap}}(\mathcal{M}_{(s,a)}) \quad \forall (s,a) \in \mathbb{S} \times \mathbb{A} \\ \bar{\psi}^{a,\text{Shap}}(\mathcal{M}) &= \bar{\pi}^{a,\text{Shap}}(\mathcal{M}) - \bar{\pi}^{a,\text{Shap}}(\bar{\mathcal{M}}^a) \quad \forall a \in \mathbb{A}\end{aligned}$$

to express the difference  $\sum_{s \in \mathbb{S}} \psi_s^{a,\text{Shap}} - \bar{\psi}^{a,\text{Shap}}$  as

$$\underbrace{\left[ \sum_{s \in \mathbb{S}} \pi_s^{a,\text{Shap}}(\mathcal{M}) - \bar{\pi}^{a,\text{Shap}}(\mathcal{M}) \right]}_{=: (\diamond)} - \underbrace{\left[ \sum_{s \in \mathbb{S}} \pi_s^{a,\text{Shap}}(\mathcal{M}_{(s,a)}) - \bar{\pi}^{a,\text{Shap}}(\bar{\mathcal{M}}^a) \right]}_{=: (\bullet)}.$$

**Step 2:** To analyze  $(\diamond)$ , we observe the state-specific and aggregated SVs (see (2.3) and (2.10))

equal

$$\begin{aligned}\pi_s^{a,\text{Shap}}(\mathcal{M}) &= \sum_{\mathcal{X} \in \mathbb{T}_1 \cup \mathbb{T}_2} c_s^a(\mathcal{X}) v_{\mathcal{M}}(\mathcal{X}) \\ \bar{\pi}^{a,\text{Shap}}(\mathcal{M}) &= \sum_{\mathcal{X} \in \mathbb{T}_1} \bar{c}^a(\mathcal{X}) v_{\mathcal{M}}(\mathcal{X}),\end{aligned}$$

where  $\mathbb{T}_1$  and  $\mathbb{T}_2$  are the sets of type 1 and 2 coalitions, respectively. This implies

$$(\diamond) = \sum_{\mathcal{X} \in \mathbb{T}_1} \{c^a(\mathcal{X}) - \bar{c}^a(\mathcal{X})\} v_{\mathcal{M}}(\mathcal{X}) + \sum_{\mathcal{X} \in \mathbb{T}_2} \sum_{s \in \mathbb{S}} c_s^a(\mathcal{X}) v_{\mathcal{M}}(\mathcal{X}).$$

**Step 3:** We analyze  $(\bullet)$  similarly to obtain

$$(\bullet) = \sum_{\mathcal{X} \in \mathbb{T}_1} \{c^a(\mathcal{X}) - \bar{c}^a(\mathcal{X})\} v_{\bar{\mathcal{M}}^a}(\mathcal{X}) + \sum_{\mathcal{X} \in \mathbb{T}_2} \sum_{s \in \mathbb{S}} c_s^a(\mathcal{X}) v_{\mathcal{M}_{(s,a)}}(\mathcal{X}),$$



where we use the fact that if  $\mathcal{X} \in \mathbb{T}_1$ , then  $v_{\mathcal{M}(s,a)}(\mathcal{X}) = v_{\tilde{\mathcal{M}}^a}(\mathcal{X})$  for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ .

**Step 4:** Putting steps 1, 2, and 3 together concludes the proof. ■

**Proof of Proposition 2.2.** It suffices to show the existence of one such instance of  $(\mathcal{M}, a)$ . Consider the network in Figure B.3. In the aggregated model, the two players ( $a = 1$  and  $a = 2$ ) are symmetric since a user can not convert if either one is absent. Accordingly, for  $a \in \{1, 2\}$ ,  $\bar{\psi}^{a, \text{Shap}} = 1/2$ . On the other hand, in the state-specific model, players  $(1, 2)$  and  $\{(s, 1)\}_{s=2}^m$  are null players and the remaining  $m$  players are symmetric and hence, each of them receives an attribution equal to the total value in the system divided by  $m$ . Accordingly,  $\sum_{s \in \mathbb{S}} \psi_s^{1, \text{Shap}} = 1/m$  and  $\sum_{s \in \mathbb{S}} \psi_s^{2, \text{Shap}} = (m-1)/m$ . Clearly,  $\lim_{m \rightarrow \infty} \bar{\psi}^{1, \text{Shap}} / \sum_{s \in \mathbb{S}} \psi_s^{1, \text{Shap}} = \infty$ , which concludes the proof.

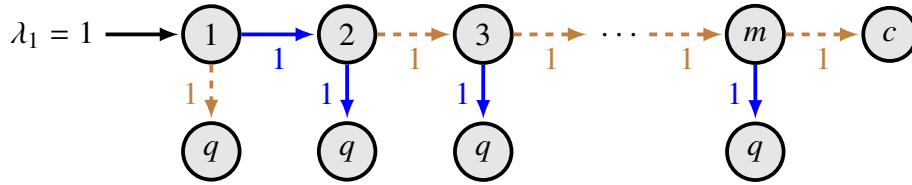


Figure B.3: Network for the proof of Proposition 2.2. The action space consists of three actions: show no-ad, show ad 1, and show ad 2. Solid blue (dashed brown) lines denote transitions if an ad 1 (ad 2) is shown. At state 1, taking action 1 moves the traffic to state 2 w.p. 1 whereas taking action 2 directs the traffic to quit state. At state  $s \in \{2, \dots, m\}$ , taking action 1 results in a transition to the quit state whereas action 2 moves the users to the “next” state. For brevity, we do not show the transitions if an ad is not shown (to quit state w.p. 1). The advertiser shows ad 1 at state 1 w.p. 1 and ad 2 at all other states w.p. 1. ■

## B.5 Counterfactuals: Local, Global, and Forward-Looking

In Section 2.4.3, we defined the counterfactual of player  $(s, a)$  as  $(s, 1)^a$ , which is equivalent to replacing the transition probabilities of  $(s, a)$  by that of  $(s, 1)$ . Given the original Markov chain  $\mathcal{M}$  (which corresponds to players  $\{(s', a')\}_{(s', a') \in \mathbb{S} \times \mathbb{A}}$ ) and an arbitrary player  $(s, a) \in \mathbb{S} \times \mathbb{A}$ , we

defined the *counterfactual network*  $\mathcal{M}_{(s,a)}$  such that we “replace” the player  $(s, a)$  by its counterfactual player  $(s, 1)^a$ , *keeping everything else as it is* and stated that our notion of CASV equals the difference between two SVs:

$$\psi_s^{\alpha, \text{Shap}}(\mathcal{M}) = \pi_s^{\alpha, \text{Shap}}(\mathcal{M}) - \pi_s^{\alpha, \text{Shap}}(\mathcal{M}_{(s,a)}). \quad (\text{B.1})$$

In this appendix, we will call such a view of a counterfactual as *local* since we replace a player  $(s, a)$  by its counterfactual  $(s, 1)^a$  “locally” and keep everything else as it is. Furthermore, we will call  $\mathcal{M}_{(s,a)}$  as the *local counterfactual network*. Though our local counterfactual construct is axiomatically appealing (recall Theorem 2.1), there are alternative ways to construct a counterfactual in the Markovian network  $\mathcal{M}$  and we discuss some possibilities in this appendix. In this direction, we first revisit our local counterfactual construct in Appendix B.5.1 and show its potential limitation. We then introduce two alternative constructs of counterfactuals in Appendices B.5.2 (“global”) and B.5.3 (“forward-looking”) and contrast these three different constructs. Note that the discussion in this appendix is applicable to non-Markovian models as well but we ground ourselves to Markovian models for ease of exposition.

### B.5.1 Local Counterfactual

Although the “local” counterfactual has an axiomatic grounding, it has the potential to “take away” too much value from the player  $(s, a)$  since it allows the counterfactual player  $(s, 1)^a$  to “have access” to all of the remaining players that were present in the original network  $\mathcal{M}$ . To shed light on this limitation, we now discuss a simple example.

Consider the network in Figure B.4. This network is time-indexed, and a user converts if and only if she sees *at least* 1 ad in total. The current policy is to show ad on both the days and the path  $\mathcal{P}$  corresponding to the original network  $\mathcal{M}$  is always  $A0 \xrightarrow{\text{ad}} B1 \xrightarrow{\text{ad}} c$ . Then, CASV attributes zero to the ad action on both the days. To see this, observe that under the original network  $\mathcal{M}$ , since the path  $\mathcal{P}$  is always  $A0 \xrightarrow{\text{ad}} B1 \xrightarrow{\text{ad}} c$ , both the players  $(A0, \text{ad})$  and  $(B1, \text{ad})$  have a SV of

1/2 each. However, since “locally” replacing either of the players by their counterfactuals also results in a conversion w.p. 1, the SV under the counterfactual network  $\mathcal{M}_{(s,a)}$  is also half for  $(s, a) \in \{(A0, \text{ad}), (B1, \text{ad})\}$ . Hence, it follows from (B.1) that CASV equals zero for both the players.

Thus, we see that the “local” counterfactual construct has the potential to “take away” too much value from the players. The key reason here is that when turning ad #1 off, ad #2 is assumed to be on and when turning ad #2 off, ad #1 is assumed to be on. In other words, the counterfactual is local and it “gets access” to all of the remaining players that were present in the original network.

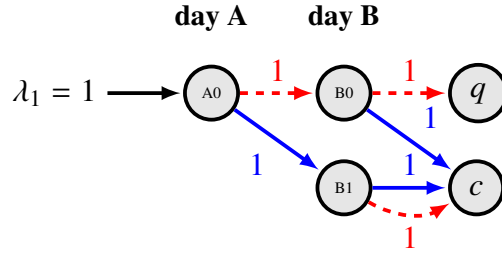


Figure B.4: There are two days (day A and day B) and the state equals (day, number of ads seen so far). For notational simplicity, we use “A0” to denote state (A, 0), “B0” to denote state (B, 0), and “B1” to denote state (B, 1). The action space consists of two actions: no-ad-action ( $a = 1$ ) and ad-action ( $a = 2$ ). Solid blue (dashed red) lines denote transitions if an ad-action (no-ad-action) is taken. The advertiser takes the ad-action on both the days w.p. 1. User’s decision to make a purchase is made after the second day and she converts if and only if she sees *at least* 1 ad in total.

### B.5.2 Global Counterfactual

In direct contrast to the local counterfactual, we now discuss the *global* CASV (G-CASV), where the no-ad-action in state  $s$  is not allowed to “exploit” the ads in states  $s' \neq s$ . We define the *global counterfactual network*  $\mathcal{M}^G$  as the network where *all* the players are replaced by their counterfactuals, i.e. no-ad-action is taken at all the states w.p. 1. The G-CASV attribution to player  $(s, a) \in \mathbb{S} \times \mathbb{A}$  is given by

$$\chi_s^{a, \text{Shap}}(\mathcal{M}) = \pi_s^{a, \text{Shap}}(\mathcal{M}) - \pi_s^{a, \text{Shap}}(\mathcal{M}^G). \quad (\text{B.2})$$

The G-CASV satisfies a (different) set of axioms but for brevity, we do not discuss such axioms here. Instead, we focus on the difference between G-CASV and (local) CASV.

For the network in Figure B.4, the SV  $\pi_s^{a, \text{Shap}}(\mathcal{M}^G)$  is identically equal to zero for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$  because the user converts if, and only if, she sees at least one ad. Hence, (B.2) implies that the ad on day 1 and the ad on day 2 will each receive an attribution of 1/2.

Clearly, G-CASV moves in the right direction in that it recognizes that there is value to showing ads and does not “take away” all the value from the ads. However, it appears that this global view may end up attributing too much value to ads. For instance, in Figure B.4, once the user has seen the first ad (which happens w.p. 1 in our setup), there is no *added* value in showing the second ad. This line of reasoning implies that the second ad should receive zero attribution but the global counterfactual still attributes a value of 1/2. Accordingly, it seems desirable to have a middleground between the two extremes of local and global counterfactuals, and we propose one such middleground next.

### B.5.3 Forward-Looking Counterfactual

The forward-looking counterfactual asks “*conditioned on being in state  $s$ , what would have happened if instead of taking the ad-action  $a$  at state  $s$ , no-ad-action was taken at state  $s$  and at all the states encountered from then on?*”. In contrast, the local counterfactual view asks the question “*conditioned on being in state  $s$ , what would have happened if instead of taking the ad-action  $a$  at state  $s$ , no-ad-action was taken at state  $s$  but other state-ad pairs were as in the original network?*” and the global counterfactual view does not even condition on state  $s$  and simply asks the question “*what would have happened if no-ad-action was taken at all the states?*”.

Define the “expanded” Markov chain  $\mathcal{M}^{\text{ext}}$  on the state space  $\mathbb{S}^{\text{ext}} = \mathbb{S} \times \mathbb{A}$ , and for each state  $(s, a) \in \mathbb{S}^{\text{ext}}$ , there is a single action, with the transition probability  $(s, a)$  to state  $(s', a') \in \mathbb{S}^{\text{ext}}$  equal to  $p_{ss'}^a \beta_{s'}^{a'}$ . Note the original Markov chain  $\mathcal{M}$  and  $\mathcal{M}^{\text{ext}}$  are equivalent in the sense that the sequence of state-action pairs have the same distribution. Define the Markov chain  $\mathcal{M}^G$  on the extended state space  $\mathbb{S}^{\text{ext}}$  by setting the transition from state  $(s, a)$  to state  $(s', a') \in \mathbb{S}^{\text{ext}}$  to be equal

to  $p_{ss}^1, \beta_{s'}^{a'}$ , i.e. we take the baseline no-ad action in each state  $(s, a) \in \mathbb{S}^{\text{ext}}$ .

For a given state-action pair  $(s, a)$ , define the “forward-looking” Markov chain  $\mathcal{M}_{(s,a)}^{\text{F}}$  as follows:

1. Start in the Markov chain  $\mathcal{M}^{\text{ext}}$  with the expanded state space  $\mathbb{S}^{\text{ext}}$ .
2. When the given  $(s, a)$  is encountered in  $\mathcal{M}^{\text{ext}}$ , transition to the corresponding state in  $\mathcal{M}^{\text{G}}$  and follow that chain until absorption.

Given these definitions, the forward-looking CASV (F-CASV) for the state  $(s, a)$  is given by

$$\phi_s^{a, \text{Shap}}(\mathcal{M}) = \pi_s^{a, \text{Shap}}(\mathcal{M}) - \pi_s^{a, \text{Shap}}(\mathcal{M}_{(s,a)}^{\text{F}}). \quad (\text{B.3})$$

This characterization can be converted into an appropriate axiomatic characterization for F-CASV. For the network in Figure B.4, F-CASV attributes 1/2 to the ad-action on day 1 and 0 to the ad-action on day 2. The remaining half is attributed to no-ad.

For all reasonable networks, we expect

$$\text{local CASV} \leq \text{forward-looking CASV} \leq \text{global CASV};$$

however, we are able to prove the statement for very strong assumptions on the underlying Markovian network.

## Appendix C: Additional Details for Chapter 3

### C.1 Proof of Theorem 3.1

In this appendix, we present the proof of Theorem 3.1. To do so, we leverage the stochastic approximation theory, for which, we provide a brief primer (Appendix C.1.1). We then establish a few supporting lemmas (Appendix C.1.2), which we invoke to prove Theorem 3.1 (Appendix C.1.3).

#### C.1.1 Primer on Asynchronous Stochastic Approximation

The contents of this subsection are based on Tsitsiklis 1994 and hence, we refer the reader to Tsitsiklis 1994 for further details. Here, we only present the results of Tsitsiklis 1994 that are relevant to us. To help the reader connect these results to our setup, we alter some of the notation in Tsitsiklis 1994 so that it matches our notation.

First, observe that the optimal  $Q$ -value vector  $\mathbf{Q}^* := [Q^*(s, a)]_{(s,a) \in \mathbb{S} \times \mathbb{A}}$  for the conversion funnel is *the* unique solution to the following system of equations (Bertsekas 1995; Puterman 2014; Sutton and Barto 2018):

$$Q^*(s, a) = \sum_{s' \in \mathbb{S}^+} p_{sas'} \times \max_{a' \in \mathbb{A}} Q^*(s', a') \quad \forall (s, a) \in \mathbb{S} \times \mathbb{A}, \quad (\text{C.1})$$

where we set  $Q^*(c, a) = 1$  and  $Q^*(q, a) = 0$  for all  $a \in \mathbb{A}$ . For ease of notation, we will denote the RHS of (C.1) by  $F_{sa}(\mathbf{Q}^*)$  and hence, we get:

$$Q^*(s, a) = F_{sa}(\mathbf{Q}^*) \quad \forall (s, a) \in \mathbb{S} \times \mathbb{A}. \quad (\text{C.2})$$

Second, we define an *asynchronous stochastic approximation* scheme with respect to the system of equations (C.2). Our goal here is to be able to find a vector  $\mathbf{X} := [X(s, a)]_{(s,a) \in \mathbb{S} \times \mathbb{A}}$  such that it satisfies (C.2) (and hence, equals  $\mathbf{Q}^*$ ). We initialize  $X(s, a)$  arbitrarily in  $[0, 1]$  for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$

and  $X(c, a) = 1$  and  $X(q, a) = 0$  for all  $a \in \mathbb{A}$ . We will update  $\mathbf{X}$  iteratively and use the notation  $\mathbf{X}_{t+1} := [X_{t+1}(s, a)]_{(s,a) \in \mathbb{S} \times \mathbb{A}}$  to denote the value of  $\mathbf{X}$  just *after* iteration  $t \in \{1, 2, \dots\}$ . The initialization corresponds to  $\mathbf{X}_1$ . The value corresponding to a state-action pair  $(s, a) \in \mathbb{S} \times \mathbb{A}$  might not be updated in each iteration and we denote by  $\mathbb{T}(s, a) \subseteq \{1, 2, \dots\}$  the iterations in which it is updated. In an asynchronous stochastic approximation scheme, the updates are as follows for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ :

$$X_{t+1}(s, a) \leftarrow \begin{cases} X_t(s, a) & \text{if } t \notin \mathbb{T}(s, a) \\ X_t(s, a) + \kappa_t(s, a) (F_{s,a}(\mathbf{X}_t) - X_t(s, a) + w_t(s, a)) & \text{if } t \in \mathbb{T}(s, a) \end{cases} \quad (\text{C.3})$$

Here,  $\kappa_t(s, a) \in [0, 1]$  is a stepsize parameter and  $w_t(s, a)$  is a noise term. For all  $t \in \{1, 2, \dots\}$ , the information set  $\mathcal{F}_t$  is defined such that it captures the history of the algorithm till the time the stepsizes  $\alpha_t(s, a)$  are selected, but does not include the noise information  $w_t(s, a)$ .

Finally, we state the result we will leverage, which follows Theorem 3 of Tsitsiklis 1994 and uses Assumption 3.1 (absorption) from Section 3.2.

**Proposition C.1.** *Under Assumption 3.1,  $\mathbf{X}_t$  converges to  $\mathbf{Q}^*$  w.p. 1 as  $t \rightarrow \infty$  if the following conditions hold:*

1. *For every  $(s, a) \in \mathbb{S} \times \mathbb{A}$ ,  $t \in \{1, 2, \dots\}$ ,  $w_t(s, a)$  is  $\mathcal{F}_{t+1}$ -measurable.*
2. *For every  $(s, a) \in \mathbb{S} \times \mathbb{A}$ ,  $t \in \{1, 2, \dots\}$ ,  $\kappa_t(s, a)$  is  $\mathcal{F}_t$ -measurable.*
3. *For every  $(s, a) \in \mathbb{S} \times \mathbb{A}$ ,  $t \in \{1, 2, \dots\}$ ,  $\mathbb{E}[w_t(s, a) | \mathcal{F}_t] = 0$ .*
4. *There exist deterministic constants  $A$  and  $B$  such that for every  $(s, a) \in \mathbb{S} \times \mathbb{A}$ ,  $t \in \{1, 2, \dots\}$ , we have*

$$\mathbb{E}[w_t^2(s, a) | \mathcal{F}_t] \leq A + B \max_{(s', a')} \max_{\tau \leq t} |Q_\tau(s', a')|^2.$$

5. For every  $(s, a) \in \mathbb{S} \times \mathbb{A}$ ,

$$\sum_{t=0}^{\infty} \kappa_t(s, a) = \infty \text{ w.p. } 1$$

$$\sum_{t=0}^{\infty} \kappa_t^2(s, a) < \infty \text{ w.p. } 1.$$

**Remark C.1.** *Theorem 3 of Tsitsiklis 1994 requires “Assumptions 1, 2, 3, and 5” (as stated in Tsitsiklis 1994). Assumption 1 of Tsitsiklis 1994 is trivially satisfied by the asynchronous stochastic approximation we defined above and our conditions in Proposition C.1 cover Assumptions 2 and 3 of Tsitsiklis 1994. Finally, our absorption assumption implies Assumption 5 of Tsitsiklis 1994 (see the discussion above Theorem 4 in Tsitsiklis 1994).*

### C.1.2 Supporting Lemmas

We now establish a few supporting lemmas that will help us in proving Theorem 3.1. In particular, we show that the *expected* update in our MFABL algorithm is an instance of the asynchronous stochastic approximation scheme above.

Consider an arbitrary iteration<sup>1</sup>  $i$  in Algorithm 6 such that the corresponding consumer was in state  $s \in \mathbb{S}$ , the firm took action  $a \in \mathbb{A}$ , and the consumer transitioned to state  $s' \in \mathbb{S}^+$ . Denote by  $Q_i(s, a) \stackrel{d}{=} \text{Beta}(\alpha_{sa}(i), \beta_{sa}(i))$  the belief over the value of  $(s, a)$  *before* the update and by  $Q_{i+1}(s, a) \stackrel{d}{=} \text{Beta}(\alpha_{sa}(i+1), \beta_{sa}(i+1))$  the belief *after* the update. Denote by  $\bar{Q}_i(s, a)$  and  $\bar{Q}_{i+1}(s, a)$  the expected values of  $Q_i(s, a)$  and  $Q_{i+1}(s, a)$ , respectively. Finally, define  $n_{sa}(i) := \alpha_{sa}(i) + \beta_{sa}(i)$  and  $n_{sa}(i+1) := \alpha_{sa}(i+1) + \beta_{sa}(i+1)$ . Recall that  $f_{s'}$  denotes the feedback generated from state  $s'$ :

$$f_{s'} \sim \text{Bernoulli} \left( \max_{a' \in \mathbb{A}} \frac{\alpha_{s'a'}(i)}{\alpha_{s'a'}(i) + \beta_{s'a'}(i)} \right).$$

The following lemma characterizes the *expected* update.

---

<sup>1</sup>By “iteration”, we refer to specific  $(t, n)$  pair in Algorithm 6.



**Lemma C.1.** *The expected update obeys the following equation:*

$$\bar{Q}_{i+1}(s, a) = \frac{n_{sa}(i)}{n_{sa}(i) + 1} \bar{Q}_i(s, a) + \frac{1}{n_{sa}(i) + 1} f_{s'}.$$

**Proof.** We split the proof into two parts: (1)  $f_{s'} = 0$  and (2)  $f_{s'} = 1$ .

**Case 1.** If  $f_{s'} = 0$ , MFABL increases  $\beta_{sa}$  by 1, i.e.,

$$\alpha_{sa}(i + 1) = \alpha_{sa}(i)$$

$$\beta_{sa}(i + 1) = \beta_{sa}(i) + 1.$$

Before the update, the expected value of  $Q_i(s, a) \stackrel{d}{=} \text{Beta}(\alpha_{sa}(i), \beta_{sa}(i))$  equals

$$\bar{Q}_i(s, a) = \frac{\alpha_{sa}(i)}{n_{sa}(i)}.$$

After the update, the expected value of  $Q_{i+1}(s, a) \stackrel{d}{=} \text{Beta}(\alpha_{sa}(i + 1), \beta_{sa}(i + 1))$  equals

$$\begin{aligned} \bar{Q}_{i+1}(s, a) &= \frac{\alpha_{sa}(i + 1)}{n_{sa}(i + 1)} \\ &= \frac{\alpha_{sa}(i)}{n_{sa}(i) + 1} \\ &= \frac{n_{sa}(i)}{n_{sa}(i) + 1} \frac{\alpha_{sa}(i)}{n_{sa}(i)} + \frac{1}{n_{sa}(i) + 1} 0 \\ &= \frac{n_{sa}(i)}{n_{sa}(i) + 1} \bar{Q}_i(s, a) + \frac{1}{n_{sa}(i) + 1} 0 \\ &= \frac{n_{sa}(i)}{n_{sa}(i) + 1} \bar{Q}_i(s, a) + \frac{1}{n_{sa}(i) + 1} f_{s'}. \end{aligned}$$

**Case 2.** If  $f_{s'} = 1$ , MFABL increases  $\alpha_{sa}$  by 1, i.e.,

$$\alpha_{sa}(i + 1) = \alpha_{sa}(i) + 1$$

$$\beta_{sa}(i + 1) = \beta_{sa}(i).$$

Before the update, the expected value of  $Q_i(s, a) \stackrel{d}{=} \text{Beta}(\alpha_{sa}(i), \beta_{sa}(i))$  equals

$$\bar{Q}_i(s, a) = \frac{\alpha_{sa}(i)}{n_{sa}(i)}.$$

After the update, the expected value of  $Q_{i+1}(s, a) \stackrel{d}{=} \text{Beta}(\alpha_{sa}(i+1), \beta_{sa}(i+1))$  equals

$$\begin{aligned} \bar{Q}_{i+1}(s, a) &= \frac{\alpha_{sa}(i+1)}{n_{sa}(i+1)} \\ &= \frac{\alpha_{sa}(i) + 1}{n_{sa}(i) + 1} \\ &= \frac{\alpha_{sa}(i)}{n_{sa}(i) + 1} + \frac{1}{n_{sa}(i) + 1} \\ &= \frac{n_{sa}(i)}{n_{sa}(i) + 1} \frac{\alpha_{sa}(i)}{n_{sa}(i)} + \frac{1}{n_{sa}(i) + 1} \\ &= \frac{n_{sa}(i)}{n_{sa}(i) + 1} \bar{Q}_i(s, a) + \frac{1}{n_{sa}(i) + 1} f_{s'}. \end{aligned}$$

This completes the proof. ■

In iteration  $i$ , MFABL updates belief  $Q_i(s, a)$  to  $Q_{i+1}(s, a)$  by updating the corresponding parameters  $\alpha_i(s, a)$  and  $\beta_i(s, a)$  to  $\alpha_{i+1}(s, a)$  and  $\beta_{i+1}(s, a)$ . The belief over the value of all other state-action pairs  $(s', a') \neq (s, a)$  remains unchanged. This process repeats for multiple iterations and the corresponding process is denoted by  $\{\mathbf{Q}_i\}_i$  where  $\mathbf{Q}_i := [Q_i(s, a)]_{(s,a) \in \mathbb{S} \times \mathbb{A}}$  for all  $i$ . By the ‘‘counterpart of MFABL in the expectation space’’, we refer to the process  $\{\bar{\mathbf{Q}}_i\}_i$  where  $\bar{\mathbf{Q}}_i := [\bar{Q}_i(s, a)]_{(s,a) \in \mathbb{S} \times \mathbb{A}}$  for all  $i$ . Lemma C.1 enables us to draw a connection between MFABL and the asynchronous stochastic approximation scheme above. In particular, we claim that the ‘‘counterpart of MFABL in the expectation space’’ is an instance of the asynchronous stochastic approximation scheme.

**Lemma C.2.** *The process  $\{\bar{\mathbf{Q}}_i\}_i$  is an asynchronous stochastic approximation scheme with respect to the system of equations (C.2).*

**Proof.** Given any prior counts as an input to MFABL,  $\bar{Q}_1(s, a) \in [0, 1]$  for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ . Line

1 of Algorithm 6 ensures  $\bar{Q}_1(c, a) = 1$  and  $\bar{Q}_1(q, a) = 0$  for all  $a \in \mathbb{A}$ . Lemma C.1 implies (C.3) is satisfied with the stepsize parameter equal to  $\kappa_i(s, a) = \frac{1}{n_{sa}(i)+1}$ , which is in  $[0,1]$  since  $n_{sa}(i) \geq 0$  for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$  and for all  $i$ . To see this, observe that

$$\begin{aligned}\bar{Q}_{i+1}(s, a) &= \frac{n_{sa}(i)}{n_{sa}(i) + 1} \bar{Q}_i(s, a) + \frac{1}{n_{sa}(i) + 1} f_{s'} \\ &= \bar{Q}_i(s, a) + \frac{1}{n_{sa}(i) + 1} \left( f_{s'} - \bar{Q}_i(s, a) \right) \\ &= \bar{Q}_i(s, a) + \frac{1}{n_{sa}(i) + 1} \left( F_{s,a}(\bar{\mathbf{Q}}_i) - \bar{Q}_i(s, a) + f_{s'} - F_{s,a}(\bar{\mathbf{Q}}_i) \right) \\ &= \bar{Q}_i(s, a) + \frac{1}{n_{sa}(i) + 1} \left( F_{s,a}(\bar{\mathbf{Q}}_i) - \bar{Q}_i(s, a) + w_i(s, a) \right),\end{aligned}$$

where  $w_i(s, a) := f_{s'} - F_{s,a}(\bar{\mathbf{Q}}_i)$  represents the noise term:

$$\begin{aligned}\mathbb{E}[w_i(s, a) | \mathcal{F}_i] &= \mathbb{E} \left[ f_{s'} - F_{s,a}(\bar{\mathbf{Q}}_i) | \mathcal{F}_i \right] \\ &= \mathbb{E} [f_{s'} | \mathcal{F}_i] - \mathbb{E} \left[ F_{s,a}(\bar{\mathbf{Q}}_i) | \mathcal{F}_i \right] \\ &= \mathbb{E}_{s' \in \mathbb{S}^+} \left[ \text{Bernoulli} \left( \max_{a' \in \mathbb{A}} \frac{\alpha_{s'a'}(i)}{\alpha_{s'a'}(i) + \beta_{s'a'}(i)} \right) \middle| a \right] - \sum_{s' \in \mathbb{S}^+} p_{sas'} \max_{a' \in \mathbb{A}} \bar{Q}_i(s', a') \\ &= \sum_{s' \in \mathbb{S}^+} p_{sas'} \max_{a' \in \mathbb{A}} \bar{Q}_i(s', a') - \sum_{s' \in \mathbb{S}^+} p_{sas'} \max_{a' \in \mathbb{A}} \bar{Q}_i(s', a') \\ &= 0.\end{aligned}$$

Note that  $\mathcal{F}_i$  includes the information that MFABL played action  $a$  in iteration  $i$  but it does not include the information that the consumer transitioned to state  $s'$ . This completes the proof. ■

### C.1.3 Proof of Theorem 3.1

To prove Theorem 3.1, we first establish that the expectation counterpart  $\bar{\mathbf{Q}}_i$  converges to  $\mathbf{Q}^*$  w.p. 1 as  $i$  goes to infinity. Then, we show that the variance of the Beta belief  $\mathbf{Q}_i$  goes to zero as  $i$  goes to infinity, and hence,  $\mathbf{Q}_i$  converges to  $\mathbf{Q}^*$ . The following lemma claims the first part.

**Lemma C.3.** *Under Assumption 3.1,  $\bar{\mathbf{Q}}_i$  converges to  $\mathbf{Q}^*$  w.p. 1 as  $i \rightarrow \infty$ .*

**Proof.** Given Lemma C.2, it suffices to show that the conditions in Proposition C.1 are satisfied. For condition 1, as shown in the proof of Lemma C.1, the noise term equals  $w_i(s, a) = f_{s'} - F_{s,a}(\bar{\mathbf{Q}}_i)$ , which is  $\mathcal{F}_{i+1}$ -measurable for every  $(s, a) \in \mathbb{S} \times \mathbb{A}$ ,  $i \in \{1, 2, \dots\}$ . From the proof of Lemma C.1, the stepsize equals  $\kappa_i(s, a) = \frac{1}{n_{sa}(i)+1}$ , which is  $\mathcal{F}_i$ -measurable for every  $(s, a) \in \mathbb{S} \times \mathbb{A}$ ,  $i \in \{1, 2, \dots\}$ . We verified condition 3 (noise is mean-zero) in the proof of Lemma C.1. For condition 4, note that  $A = 1$  and  $B = 0$  works since for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ ,  $i \in \{1, 2, \dots\}$ , and  $s' \in \mathbb{S}^+$ , we have

$$\begin{aligned} w_i^2(s, a) &= \left( f_{s'} - F_{s,a}(\bar{\mathbf{Q}}_i) \right)^2 \\ &= \left( f_{s'} - \sum_{s' \in \mathbb{S}^+} p_{sas'} \max_{a' \in \mathbb{A}} \bar{Q}_i(s', a') \right)^2 \\ &\leq 1. \end{aligned}$$

Final inequality is true because  $f_{s'} \in [0, 1]$  and  $\bar{Q}_i(s', a') \in [0, 1]$ . Finally, condition 5 is true because the stepsize sequence corresponding to  $\kappa_i(s, a) = \frac{1}{n_{sa}(i)+1}$  forms a harmonic series and each state-action pair is visited infinitely often due to the  $\epsilon$ -greedy construction of MFABL and the “connectedness” assumption (recall the discussion when defining initial state probabilities in Section 3.2). ■

**Proof of Theorem 3.1.** Given Lemma C.3, it suffices to show that the variance of the Beta belief  $\mathbf{Q}_i$  goes to zero as  $i$  goes to infinity. Observe that in each visit to  $(s, a) \in \mathbb{S} \times \mathbb{A}$ , the “count”  $n(s, a)$  increases by 1 since either  $\alpha(s, a)$  is increased by 1 or  $\beta(s, a)$  is increased by 1. Furthermore, due to the  $\epsilon$ -greedy construction of MFABL and the “connectedness” assumption (recall the discussion when defining initial state probabilities in Section 3.2), each state-action pair is visited infinitely often and hence  $n_i(s, a) \rightarrow \infty$  as  $i$  goes to infinity for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ . This implies that the

variance of  $Q_i(s, a)$  goes to 0 because

$$\begin{aligned}
\text{Var}(Q_i(s, a)) &= \text{Var}(\text{Beta}(\alpha_i(s, a), \beta_i(s, a))) \\
&= \frac{\alpha_i(s, a) \times \beta_i(s, a)}{n_i^2(s, a) \times (n_i(s, a) + 1)} \\
&\leq \frac{n_i(s, a) \times n_i(s, a)}{n_i^2(s, a) \times (n_i(s, a) + 1)} \\
&= \frac{1}{n_i(s, a) + 1}.
\end{aligned}$$

■

## C.2 Algorithms for Model Extensions

In this appendix, we discuss the algorithms corresponding to the model extensions discussed in Section 3.5. Since the algorithm for extension #3 (consumer features) is the same as the original MFABL (Algorithm 6), we skip that extension here and only discuss the first two extensions: (1) multiple products (Appendix C.2.1) and (2) actions with costs (Appendix C.2.2).

### C.2.1 Multiple Products

The MFABL algorithm for the conversion funnel with multiple products is essentially the same as the one corresponding to a single product (Algorithm 6). The only change is in line 1 of Algorithm 6, i.e., the initialization of the counts corresponding to the absorbing states. In particular, the initialization in line 1 is as follows:

$$\begin{aligned}
(\alpha_{qa}, \beta_{qa}) &= (1, \infty) \text{ for all } a \in \mathbb{A} \\
(\alpha_{c_ma}, \beta_{c_ma}) &= (1 + L, (1 + L)(1 - r_m)/r_m) \text{ for all } a \in \mathbb{A}, \forall m \in \{1, \dots, M\},
\end{aligned}$$

where  $L$  is a “large” constant. Such an initialization of the conversion states ensures the belief in MFABL is such that

$$Q(c_m, a) = r_m \text{ w.p. } 1 \forall m \in \{1, \dots, M\} \text{ as } L \rightarrow \infty.$$

Hence, a transition to a conversion state generates the “true” reward. The convergence result (Theorem 3.1) for this generalization mimics the same arguments as in Appendix C.1.

### C.2.2 Actions with Costs

The MFABL algorithm for actions with costs is presented as Algorithm 11. The key change is that instead of maintaining a Beta belief on the value of a state-action pair, we now maintain a Gaussian belief  $\text{Normal}(\mu_{sa}, \tau_{sa})$  where  $\mu_{sa}$  denotes the mean and  $\tau_{sa}$  denotes the precision for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ . The action selection remains the same as in MFABL (pick maximum sample value with  $\epsilon$ -greedy). The parameters update is done to mimic the  $\bar{\mathbf{Q}}$ -update in the original MFABL algorithm (Algorithm 6). In particular, compare line 11 of Algorithm 11 to Lemma C.1 in Appendix C.1. Due to such construction, convergence of Algorithm 11 to  $\mathbf{Q}^*$  can be shown in a similar manner as the convergence of Algorithm 6 (Theorem 3.1). To be precise, (C.1) from Appendix C.1 changes to the following:

$$Q^*(s, a) = \mathbb{E}[c_{sa}] + \sum_{s' \in \mathbb{S}^+} p_{sas'} \max_{a' \in \mathbb{A}} Q^*(s', a') \forall (s, a) \in \mathbb{S} \times \mathbb{A}, \quad (\text{C.5})$$

where we set  $Q^*(c, a) = r$  and  $Q^*(q, a) = 0$  for all  $a \in \mathbb{A}$ . As in Appendix C.1, we denote the RHS of (C.5) by  $F_{sa}(\mathbf{Q}^*)$ :

$$Q^*(s, a) = F_{sa}(\mathbf{Q}^*) \forall (s, a) \in \mathbb{S} \times \mathbb{A}, \quad (\text{C.6})$$

The definition of an asynchronous stochastic approximation scheme remains the same as in Appendix C.1 (see (C.3)) but now  $F_{sa}(\cdot)$  is as defined above and the initialization  $\mathbf{X}_1$  need not be in

[0, 1]. Proposition C.1 still holds due to Tsitsiklis 1994. By construction of Algorithm 11 (lines 11 and 12 in particular), Lemma C.1 still holds, and so does Lemma C.2 with  $\kappa_i(s, a) = \frac{1}{\tau_{sa}(i)+1}$ , i.e.,  $\tau_{sa}(i)$  replaces  $n_{sa}(i)$ . The noise  $w_i(s, a)$  is still mean-zero as shown below:

$$\begin{aligned}
\mathbb{E}[w_i(s, a)|\mathcal{F}_i] &= \mathbb{E}\left[f_{s'} - F_{s,a}(\bar{Q}_i)|\mathcal{F}_i\right] \\
&= \mathbb{E}[f_{s'}|\mathcal{F}_i] - \mathbb{E}\left[F_{s,a}(\bar{Q}_i)|\mathcal{F}_i\right] \\
&= \mathbb{E}_{s' \in \mathbb{S}^+} \left[ \tilde{c}_{sa} + \text{Normal}(\mu_{s'a'}, \tau_{s'a'}) \Big| a, a' = \arg \max_{\hat{a} \in \mathbb{A}} \mu_{s'\hat{a}} \right] - \mathbb{E}[c_{sa}] - \sum_{s' \in \mathbb{S}^+} p_{sas'} \max_{a' \in \mathbb{A}} \bar{Q}_i(s', a') \\
&= \mathbb{E}[\tilde{c}_{sa}] + \sum_{s' \in \mathbb{S}^+} p_{sas'} \max_{a' \in \mathbb{A}} \mu_{s'a'} - \mathbb{E}[c_{sa}] - \sum_{s' \in \mathbb{S}^+} p_{sas'} \max_{a' \in \mathbb{A}} \mu_{s'a'} \\
&= 0.
\end{aligned}$$

To ensure the validity of Lemma C.3, we need to re-establish condition 4 of Proposition C.1, which is known to hold (see the discussion above Theorem 4 in Tsitsiklis 1994). All other conditions of Proposition C.1 hold for similar reasons as in Appendix C.1. Finally, to conclude the analysis, we need to ensure the variance of our Beta belief  $Q(s, a)$  goes to zero for all  $(s, a) \in \mathbb{S} \times \mathbb{A}$ . This is true since our  $\epsilon$ -greedy construction of Algorithm 11 and the “connectedness” assumption (recall the discussion when defining initial state probabilities in Section 3.2) ensures each state-action pair is visited infinitely often and hence the precision  $\tau_{sa}$  (which equals the inverse of variance) goes to infinity for each  $(s, a) \in \mathbb{S} \times \mathbb{A}$  (follows line 12 of Algorithm 11).

---

**Algorithm 11** MFABL with Non-Zero Cost Actions ( $N$  Consumers)
 

---

**Require:** Prior parameters  $(\mu_{sa}, \tau_{sa}) \forall (s, a) \in \mathbb{S} \times \mathbb{A}$ ,  $\epsilon$

- 1:  $(\mu_{ca}, \tau_{ca}) = (r, \infty)$  and  $(\mu_{qa}, \tau_{qa}) = (0, \infty)$  for all  $a \in \mathbb{A}$       % parameters for states  $c$  and  $q$
  - 2: **for**  $t = 1, 2, \dots$       % until there exists an “active” consumer (has not converted or quit)
  - 3:   **for**  $n = 1$  to  $N$
  - 4:     Observe state  $s = s_{nt}$  of consumer  $n$  at time  $t$
  - 5:     **if**  $s \in \mathbb{S}$       % filter for “active” consumers
  - 6:        $q_{sa} \sim \text{Normal}(\mu_{sa}, \tau_{sa}) \forall a \in \mathbb{A}$       % generate samples
  - 7:        $a^* = \arg \max_a q_{sa}$  with  $\epsilon$ -greedy      % highest sample value with  $\epsilon$ -greedy
  - 8:       Firm incurs a cost  $\tilde{c}_{sa^*} \sim c_{sa^*}$       %  $c_{sa^*}$  possibly random
  - 9:       Consumer transitions to state  $s' = s_{n,t+1} \in \mathbb{S}^+$
  - 10:        $f_{s'} \sim \tilde{c}_{sa^*} + \text{Normal}(\mu_{s'a'}, \tau_{s'a'})$  where  $a' = \arg \max_{a \in \mathbb{A}} \mu_{s'a}$       % generate feedback
  - 11:        $\mu_{sa^*} \leftarrow \frac{\tau_{sa^*}}{\tau_{sa^*} + 1} \mu_{sa^*} + \frac{1}{\tau_{sa^*} + 1} f_{s'}$       % update mean
  - 12:        $\tau_{sa^*} \leftarrow \tau_{sa^*} + 1$       % update precision
  - 13:     **end if**
  - 14:   **end for**
  - 15: **end for**
  - 16: **return**  $\mathbf{Q} := [Q(s, a)]_{(s,a) \in \mathbb{S} \times \mathbb{A}}$  where  $Q(s, a) \stackrel{d}{=} \text{Normal}(\mu_{sa}, \tau_{sa})$
-