

University of Bath



PHD

Artificial Prediction Markets for Online Prediction of Continuous Variables

Jahedpari, Fatemeh

Award date:
2016

Awarding institution:
University of Bath

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 22. May. 2019

Artificial Prediction Markets for Online Prediction of Continuous Variables

submitted by

Fatemeh Jahedpari

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Computer Science

March 2016

COPYRIGHT

Attention is drawn to the fact that copyright of this thesis rests with its author. This copy of the thesis has been supplied on the condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the prior written consent of the author.

This thesis may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation.

Signature of Author

Fatemeh Jahedpari

Abstract

In this dissertation, we propose an online machine learning technique – named *Artificial Continuous Prediction Market (ACPM)* – to predict the value of a continuous variable by (i) integrating a set of data streams from heterogeneous sources with time varying compositions such as changing the quality of data streams, (ii) integrating the results of several analysis models for each data source when the most suitable model for a given data source is not known *a priori*, (iii) dynamically weighting the prediction of each analysis model and data source to form the system prediction.

We adapt the concept of prediction market, motivated by their success in forecasting accurately the outcome of many events [Nikolova and Sami, 2007]. Our proposed model instantiates a sequence of prediction markets in which artificial agents play the role of market participants. Agents participate in the markets with the objective of increasing their own utility and hence indirectly cause the markets to aggregate their knowledge. Each market is run in a number of rounds in which agents have the opportunity to send their prediction and bet to the market. At the end of each round, the aggregated prediction of the crowd is announced to all agents, which provides a signal to agents about the private information of other agents so they can adjust their beliefs accordingly. Once the true value of the record is known, agents are rewarded according to accuracy of their prediction. Using this information, agents update their models and knowledge, with the aim of improving their performance in future markets.

This thesis proposes two trading strategies to be utilised by agents when participating in a market. While the first one is a naive constant strategy, the second one is an adaptive strategy based on Q-Learning technique [Watkins, 1989].

We evaluate the performance of our model in different situations using real-world and synthetic data sets. Our results suggest that ACPM: i) is either better or very close to the best performing agents, ii) is resilient to the addition of agents with low performance, iii) outperforms many well-known machine learning models, iv) is resilient to quality drop-out in the best performing agents, v) adapts to changes in quality of agents predictions.

Acknowledgements

I thank God for keeping me in his tender loving care and never abandoning me.

I would like to thank my supervisors Dr Marina De Vos and Dr Julian Padget for providing help, direction and insight during my PhD. I wish to express my sincere gratitude to all those who advised me during the course of this work, specially Dr. Peyman Faratin, Dr Benjamin Hirsch, Dr Sattar Hashemi, Dr Tomasz Michalak and Dr Talal Rahwan. I also truly thank my bachelor and master supervisors Prof Iyad Rahwan and Dr Farhad Orumchian, who were the first to teach me what research is all about.

I am deeply and forever indebted to my parents, Elaheh Latifi and Karim Jahedpari for their endless love and support. My dear parents, thank you for your continuous encouragement and support throughout my entire life. Your love and support enabled me to reach this point.

A big thank goes to my beloved husband. Dear Ali, thank you for always believing in me, supporting me to fulfil my dreams, being patient, caring and cheerful. Thank you for listening to my worries and standing by my side. This achievement would not have been possible without your support. I will never forget all your kindness before and during my PhD.

I would like to thank my brother and his wife for their continuous love and support. Dear Saeed and Sara, thank you for supporting me spiritually throughout this journey and my life in general. I am so grateful to have you in my life.

Special thanks also goes to my friends in University of Bath, particularly Esra, Neda, Zohreh, Asieh, Shaghayegh, Fabio, Natalya, Maryam, Shahad and Latifa. Thank you for listening to my worries and bringing joy and happiness to this journey. I learnt a lot from each of you.

*To my husband, Ali,
for his love, kindness and never-ending support.*

Contents

List of Figures	5
List of Tables	9
1 Introduction	11
1.1 Problem Statement and Thesis Motivation	11
1.2 Prediction Markets	12
1.3 Thesis Objectives and Contributions	14
1.4 Thesis Outline	16
1.5 Related Publications	18
2 Literature Review	19
2.1 Machine Learning Background	19
2.1.1 Machine Learning Approaches	19
2.1.2 Ensemble Methods	21
2.1.3 Online Learning	23
2.1.3.1 Prediction with Expert Advice	23
2.1.3.2 Exponentially Weighted Average Forecaster	26
2.1.3.3 Tracking the Best Expert	26
2.1.3.4 The Exponentiated Gradient algorithm	27
2.1.3.5 Follow the Best Expert	28
2.1.4 Reinforcement Learning	28
2.1.4.1 Reward Function	29
2.1.4.2 Policy	30
2.1.4.3 Value Function	31
2.1.4.4 Model	31
2.1.4.5 Q-learning	31
2.1.4.6 Reinforcement Learning in Financial Markets	33
2.2 Prediction Market Background	34
2.2.1 Market Participants	37
2.2.2 Incentive	37
2.2.3 Trading Protocols	38
2.2.4 Prediction Market Applications	41
2.3 Connections Between Prediction Markets and Machine Learning	42
2.3.1 Connecting Prediction Markets and Online Learning	42
2.3.2 Artificial Prediction Markets	43

2.4	Summary	46
3	Artificial Continuous Prediction Market	49
3.1	ACPM Overview	50
3.2	ACPM Mechanism	52
3.2.1	Reward Function	53
3.2.2	Aggregation Function	56
3.2.3	Rate Per Transaction Parameters	56
3.2.4	Game Theory properties	58
3.3	Trading Strategy	59
3.3.1	Constant Trading Strategy	59
3.3.2	Q-Learning Trading Strategy	59
3.3.2.1	States	60
3.3.2.2	Actions	62
3.3.2.3	Updating Q-values	63
3.3.2.4	Updating the Confidence in the Crowd:	64
3.4	Trading Strategy Examples	65
3.4.1	Constant Trading Strategy Example	65
3.4.1.1	Round 1	65
3.4.1.2	Round 2	66
3.4.1.3	Final ACPM Prediction	66
3.4.2	Q-Learning Trading Strategy Example	66
3.4.2.1	Round 1	67
3.4.2.2	Round 2	67
3.4.2.3	Final Market Prediction	71
3.4.2.4	Revenues	71
3.4.2.5	Round 1 Revenues	71
3.4.2.6	Round 2 Revenues	71
3.4.2.7	Updating Q-Tables	72
3.5	Discussion	73
4	System Evaluation	77
4.1	Hypotheses	78
4.2	Data Sets	79
4.2.1	Syndromic Surveillance Data Sets	79
4.2.2	Artificial Data Sets	80
4.2.3	UCI Data Sets	80
4.3	ACPM Parameter Setting	80
4.3.1	Market Duration	81
4.3.2	Rate Per Transaction Parameters	81
4.4	Experimental Setup	82
4.4.1	Setting for Experiments A1-A4	83

4.4.2	Setting for Experiments A5-A8	84
4.4.3	Setting for Experiment B1	85
4.4.4	Setting for Experiments B2-B7	85
4.4.5	Setting for Experiments C1-C2	86
4.5	Experiments	88
4.5.1	Experiment A1	88
4.5.2	Experiment A2	89
4.5.3	Experiment A3	90
4.5.4	Experiment A4	91
4.5.5	Experiment A5	92
4.5.6	Experiment A6	93
4.5.7	Experiment A7	94
4.5.8	Experiment A8	95
4.5.9	Experiment B1	96
4.5.10	Experiment B2	97
4.5.11	Experiment B3	97
4.5.12	Experiment B4	98
4.5.13	Experiment B5	99
4.5.14	Experiment B6	100
4.5.15	Experiment B7	101
4.5.16	Experiment C1	102
4.5.17	Experiment C2	106
4.6	Analysis	106
5	Syndromic Surveillance	111
5.1	Introduction	111
5.2	Syndromic Surveillance Data Sources	112
5.3	Existing Syndromic Surveillance Systems	114
5.3.1	Traditional Syndromic Surveillance Systems	115
5.3.2	Internet-Based Syndromic Surveillance Systems	117
5.4	Statement of the Problem	118
5.5	Google Flu Trends Case Study	120
5.5.1	Comparison of ACPM and GFT	121
5.5.1.1	Experimental Setup	121
5.5.1.2	Experimental Results	123
5.6	GP Case Study	124
5.6.1	Comparison of ACPM and GP	125
5.6.1.1	Experimental Setup	125
5.6.1.2	Experimental Results	126
5.7	Analysis	127
6	Conclusions and Further Work	129

6.1	Contributions	130
6.2	ACPM Limitations	132
6.3	Future Works	133
Appendices		137
A	An Alternative Reward Function	138
B	ACPM Parameter Setting	142
B.1	Number of Round Parameter	142
B.2	MaxRPT Parameter	144
C	Evaluation of ACPM Using Syndromic Surveillance Data Set	146
D	Evaluation of ACPM Using Artificial Data	150
E	Evaluation of ACPM on UCI Data Sets	153
F	Evaluation of ACPM Using Syndromic Surveillance Data Set	158
G	Comparing ACPM and Google Flu Trend	159
H	Comparing ACPM and GP	172
Bibliography		178

List of Figures

2-1	Online Learning Process.	24
2-2	The agent and environment interaction in reinforcement learning [Sutton and Barto, 2011].	30
3-1	ACPM Architecture with three agents of A, B and C. For each event to predict, a market instance is instantiated, where each market instance comprises a number of rounds. In each round, agents send their predictions and bets to the market maker. At the end of each round, the market maker calculates and announces the aggregated prediction of all agents. Agents can use this information to update their prediction and bets for the subsequent rounds. Once the true outcome is revealed, the agents are notified of the true outcome and receive their payoff. Finally, each agent updates its trading strategy based on the received payoffs, and re-trains its analysis model taking into account the newly-revealed outcome, to prepare for any potential future markets.	54
4-1	Experiment A1. Comparing ACPM performance with the mean performance of participants for each market type. The figure shows that the performance of ACPM is higher than the mean performance of all agents in each market type.	88
4-2	Experiment A2. Comparing ACPM performance with the best performing participant performance for each market type. ACPM outperforms best performing agent in each market type.	89
4-3	Experiment A3. Popularity of each action for agents with different quality. Action “ChangePr” is the most popular action for low quality agents and “PreservePr” is the most popular action for high quality agents.	90
4-4	Experiment A4. Comparison of ACPM’s performance with Q-learning and without. Adopting the Q-learning by participants increases the performance of ACPM compared to the constant trading strategy in each market type.	91
4-5	Experiment A5. How participating in ACPM and utilising Q-learning strategy improves the performance of each classifier.	92
4-6	Experiment A6. Comparing ACPM performance with well known machine learning models.	93
4-7	Experiment A7. Performance comparison of ACPM against high quality data participants, over a range of time frames.	94
4-8	Experiment A8. Demonstrating the average time consumption for each market in ACPM. The y-axis shows the consumed time in nano seconds.	95

4-9	Experiment B1. Comparing ACPM performance. Type A: the market has 100 agents with random data. Type B: the market has agents of the type A market and additionally one agent with high quality data. Type C: the market has only the high quality agent.	96
4-10	Experiment B2. Comparing accumulated error of ACPM and its 40 participants. One agent accesses data with 10% noise and 39 agents access data with 50% noise. Each solid line represents the accumulated error of a participating agent and the multi-coloured line represents the accumulated error of ACPM.	97
4-11	Experiment B3. Comparing accumulated error of ACPM and its 40 participants. 5 agents access data with 10% noise and 35 agents access data with 50% noise. Each solid line represents the accumulated error of a participating agent and the multi-coloured line represents the accumulated error of ACPM.	98
4-12	Experiment B4. Comparing accumulated error of ACPM and its 40 participants each accessing data with 10% noise. Each solid line represents the accumulated error of a participating agent and the multi-coloured line represents the accumulated error of ACPM.	99
4-13	Experiment B5. Comparing accumulated error of ACPM and its 40 participants receiving random data with uniform distribution. Each solid line represents the accumulated error of a participating agent and the multi-coloured line represents the accumulated error of ACPM.	100
4-14	Experiment B6. Comparing accumulated error of ACPM and its 40 participants each accessing random data with uniform distribution. For each set of 10 markets, one of the agents has access to the correct answer. Each solid line represents the accumulated error of a participating agent and the multi-coloured line represents the accumulated error of ACPM.	101
4-15	Experiment B7 - Comparing error of ACPM and its 5 participants with varying performance. Each solid line represents the accumulated error of a participating agent and the multi-coloured line represents the accumulated error of ACPM.	102
5-1	Conceptual timeline of pre-diagnosis data types and sources for syndromic surveillance [Chen et al., 2010a].	113
5-2	Distribution of different data source usage in existing syndromic surveillance systems in the USA [Buehler et al., 2008].	115
5-3	Comparing the Performance of ACPM and GFT for different periods using Mean Absolute Error (MAE).	123
5-4	ACPM and GFT error in predicting ILI rate from 2004 to 2015	124
5-5	Comparing the Performance of ACPM and GP for different periods using Mean Absolute Error (MAE).	126
5-6	ACPM and GP error in predicting ILI rate from 2004 to 2015	127
A-1	The reward functions for $C = 1$, $\beta = 1$ and $P \in (\frac{1}{10}, \frac{1}{9}, \dots, \frac{1}{2}, 1, 2, \dots, 10)$. Note that errors greater than the cut-off ($C = 1$) receive zero rewards.	139

D-1	Scenario B2 - Comparing accumulated error of ACPM and its 40 participants. 39 agents access data with 50% noise and one agents access data with 10% noise.	150
D-2	Scenario B3- Comparing accumulated error of ACPM and its 40 participants. 35 agents access data with 50% noise and 5 agents access data with 10% noise.	151
D-3	Scenario B4- Comparing accumulated error of ACPM and its 40 participants each accessing data with 10% noise.	151
D-4	Scenario B5- Comparing accumulated error of ACPM and its 40 participants each accessing random data with uniform distribution.	152
D-5	Scenario B6 -Comparing accumulated error of ACPM and its 40 participants each accessing random data with uniform distribution. For each 10 consecutive markets one of the agents accesses the correct answer.	152
E-1	Part I. MSE of ACPM and the Benchmarks	154
E-2	Part II. MSE of ACPM and the Benchmarks	155
E-3	Part I. MAE of ACPM and the Benchmarks	156
E-4	Part II. MAE of ACPM and the Benchmarks	157
G-1	Comparing Prediction of ACPM and Google Flu Trend in 2004	160
G-2	Comparing ACPM Error and Google Flu Trend Error in 2004	160
G-3	Comparing Prediction of ACPM and Google Flu Trend in 2005	161
G-4	Comparing ACPM Error and Google Flu Trend Error in 2005	161
G-5	Comparing Prediction of ACPM and Google Flu Trend in 2006	162
G-6	Comparing ACPM Error and Google Flu Trend Error in 2006	162
G-7	Comparing Prediction of ACPM and Google Flu Trend in 2007	163
G-8	Comparing ACPM error and Google Flu Trend Error in 2007	163
G-9	Comparing Prediction of ACPM and Google Flu Trend in 2008	164
G-10	Comparing ACPM Error and Google Flu Trend Error in 2008	164
G-11	Comparing Prediction of ACPM and Google Flu Trend in 2009	165
G-12	Comparing ACPM Error and Google Flu Trend Error in 2009	165
G-13	Comparing Prediction of ACPM and Google Flu Trend in 2010	166
G-14	Comparing ACPM Error and Google Flu Trend Error in 2010	166
G-15	Comparing Prediction of ACPM and Google Flu Trend in 2011	167
G-16	Comparing ACPM Error and Google Flu Trend Error in 2011	167
G-17	Comparing Prediction of ACPM and Google Flu Trend in 2012	168
G-18	Comparing ACPM Error and Google Flu Trend Error in 2012	168
G-19	Comparing Prediction of ACPM and Google Flu Trend in 2013	169
G-20	Comparing ACPM Error and Google Flu Trend Error in 2013	169
G-21	Comparing Prediction of ACPM and Google Flu Trend in 2014	170
G-22	Comparing ACPM Error and Google Flu Trend Error in 2014	170
G-23	Comparing Prediction of ACPM and Google Flu Trend in 2015	171
G-24	Comparing ACPM Error and Google Flu Trend Error in 2015	171
H-1	Comparing Prediction of ACPM and GP in Flu Season 2008-9	173

H-2	Comparing ACPM and GP error in Flu Season 2008-9	173
H-3	Comparing Prediction of ACPM and GP in Flu Season 2009-10	174
H-4	Comparing ACPM and GP error in Flu Season 2009-10	174
H-5	Comparing Prediction of ACPM and GP in Flu Season 2010-11	175
H-6	Comparing ACPM and GP error in Flu Season 2010-11	175
H-7	Comparing Prediction of ACPM and GP in Flu Season 2011-12	176
H-8	Comparing ACPM and GP Error in Flu Season 2011-12	176
H-9	Comparing Prediction of ACPM and GP in Flu Season 2012-13	177
H-10	Comparing ACPM and GP Error in Flu Season 2012-13	177

List of Tables

2.1	Lay and Barbu [2010] model.	44
2.2	Existing Artificial Prediction Markets (APM). The first and second columns show the APMs which are designed for online setting problems and regression task, respectively. Third and fourth columns show the APMs in which agents utilise an adaptive trading strategy and the wisdom of the crowd, respectively.	47
3.1	Q-Table Example. The first and second columns show state information. The third and fourth columns show Q-value for “Change” and “Preserve” actions respectively and the last column shows the best value of δ (i.e confidence of the agent in the crowd) for the corresponding state.	69
3.2	Updated Q-Table for Agent A.	72
3.3	Updated Q-Table for Agent B.	73
3.4	Updated Q-Table for Agent C.	73
4.1	UCI Data Sets Used for Experiments	81
4.2	Our four market types. Data sources are divided into three categories of low, medium and high quality based on their MAE as determined by several regression models. Table rows describe market types according to data quality of their participants.	84
4.3	Experiment C1 - MSE of ACPM and Benchmarks. ACPM outperforms most benchmarks for many data sets. The value in the parentheses gives the ratio by which ACPM performs better. The best, second best and third best models are highlighted by red, yellow and grey colour respectively.	104
4.4	Experiment C1 - The p-values of paired t-test for ACPM and benchmarks.	105
4.5	Experiment C2 - MSE of ACPM and Forecasters. ACPM outperforms most forecasters in each data set, except a few cases which are marked by *.	106
4.6	Experiment C2 - The p-values of paired t-test for ACPM and PEA models.	107
5.1	R’s caret package models. ACPM instantiates one participant for each of these models.	122
5.2	Performance of ACPM and GFT in predicting ILI rate using Mean Absolute Error (MAE) and p-values of paired t-test.	125
5.3	Performance of ACPM and GP in predicting ILI rate using Mean Absolute Error (MAE) and p-values of paired t-test.	128

Chapter 1

Introduction

Machine learning, a core subarea of artificial intelligence, has become an indispensable part of many application areas, ranging from politics to engineering. Prediction is a key task in machine learning such as predicting political events, best selling books, tomorrow's temperature, customer loyalty, and etc. To make a prediction, a machine learning model can learn either in offline mode (i.e batch mode) or online mode. In offline mode, the model is first presented with a number of training examples to learn from and make a hypothesis which maps the input data to their corresponding outcomes, then the model is presented with test examples (examples without the outcome) to predict their outcomes. This type of learning is suitable when the quality of data does not change over time. However, in the online mode of learning, examples arrive in sequence and after the model makes the prediction, the true outcome of the example is revealed, and the model re-trains accordingly. Machine learning tasks are divided into classification and regression. When the subject of prediction is discrete, it is a '*classification*' task. On the other hand, '*regression*' refers to when the subject of prediction is continuous. In this dissertation, we are interested in making predictions for regression task in the online mode of learning.

1.1 Problem Statement and Thesis Motivation

New machine learning models are frequently added to the literature. Choosing which model to use is very arbitrary and is dependent on the diversity of models the data analyser is familiar and convenient with. Fernández-Delgado et al. [2014] state that there are a large number of machine learning models and it is not practical to examine all models to find the best model to suit a specific application domain problem. Therefore, they experimented with 179 classifiers on 121 data sets and concluded that the random forest [Breiman, 2001] achieve the highest performance on average and can be used by a data analyser as the first choice. Although their conclusion provides a fast track, random forest cannot achieve the highest performance in each domain. According to the No-Free-Lunch theorem [Wolpert, 1996], the best classifier will not be the same for all the data sets.

A second complicating factor is that, in some cases a data analyser may resort to several data sources to predict an event. One example of such a situation is predicting the influenza rate in the process of detecting an outbreak. Various data sources such as medical absentee rates at

schools, over-the-counter pharmacy sales, Internet queries and open source information can be used to predict influenza rate. In such situations, the process of prediction is even harder since different models may better suit each data source and the data analyst needs to find the most effective model for each data source and then combine their results in a way which maximises the accuracy of the prediction. In sum, given that the most suitable algorithm for a given data source is not known *a priori*, a reasonable mechanism is to analyse each data source with a variety of algorithms and integrate their results.

Furthermore, data source quality may change over time and hence the suitability of models may vary accordingly. In addition to losing or gaining quality, data source availability can also change, in that a particular data source might become available or unavailable for whatever circumstances. Hence, a data analyst is required to repeat the process of model selection regularly to adapt to environmental variations.

Against this background, the motivation of this thesis is automating the process of identifying the most effective models and data sets from a candidate set and aggregating their results while adapting to environmental variations. In particular, this thesis suggests a technique which not only identifies the best models from a candidate set for each data source and aggregates their predictions, but also dynamically adapts to environmental variations such as data sources' availability and quality changes.

In this thesis, we apply our model to the application domain of predicting influenza rate. Annual influenza epidemics cause approximately 3 to 5 million cases of severe illness, and up to 500,000 deaths worldwide [World Health Organization, 2014]. In this context, the event to predict is the disease activity level of influenza-like illnesses on a specific date and place. We evaluate the performance of our model by predicting influenza rates in the USA using publicly available data sources. We then compare the performance of ACPM with the two benchmarks of Google Flu Trend¹ (GFT) and a model proposed by Lamos et al. [2015], which is built upon Google Flu Trend and improves GFT performance. Our results show that our model outperforms GFT and GP in most cases. Our model is inspired by features of real world prediction markets, which we briefly review in the next section.

1.2 Prediction Markets

Prediction markets utilise the aggregated wisdom of the crowd in order to predict the outcome of a future event [Ray, 2006]. They have been used to forecast accurately the outcome of various events [Snowberg et al., 2012]. In these markets, trader behaviour reveals private information and beliefs about possible outcomes, and can be used to forecast an event accurately [Nikolova and Sami, 2007]. Prediction markets are increasingly being considered as approaches for collecting, summarising and aggregating dispersed information by governments and corporations [Hanson et al., 2006].

In prediction markets, participants purchase and sell assets whose payoffs are tied to the realisation of future events [Bray et al., 2008]. In other words, traders bet their money on the

¹<http://www.google.com/flutrends/> (Retrieved April 14, 2015)

outcome of future events by trading securities. A security is a financial instrument like a financial stock whose profit payments are based on the outcome of the event. Each event outcome has a security associated with it. Before the market is closed, traders can trade in each day of the market according to the current security prices. To illustrate, a prediction market can be used to predict “ whether candidate ‘X’ will win the election” by offering two securities: ‘Yes’ and ‘No’. Assuming the market finally ends by candidate ‘X’ winning in the election, all traders will receive \$1 payoff for each ‘Yes’ security they hold and \$0 for their ‘No’ security, which means they lose the money they spent on buying ‘No’ securities.

A prediction market is usually run by a market maker who is the company or individual that interacts with traders and determines the market prices using a market trading protocol. The aggregated monetary bets made by market traders can be used to dynamically determine the price of each security. The market price of a security not only represents the price at which the security can be bought or sold, but also shows the probability of the occurrence of its corresponding outcome by fusing all trader beliefs. Arguably, the price that a trader would pay to buy a security indicates how confident she is in the outcome of the event.

The Iowa Electronic Marketplace (IEM), one of the best known online prediction markets, has precisely predicted the winner of every presidential election since its inception and it has predicted the percentages of votes acquired by presidential candidates more accurately than other forecasting techniques including expert opinion and exit polls [Ray, 2006]. A number of other prediction markets such as TradeSports.com, Newsfutures and the Hollywood Stock Exchange emerged following the success of IEM. Likewise, a number of corporations including Hewlett Packard, Google and Yahoo exploit prediction markets as an alternative mechanism to polls, staff surveying, committee and work group in order to aggregate decision-relevant information, which is often widely dispersed across company staff [Bray et al., 2008].

The so-called ‘wisdom of the crowd’, in prediction markets, comes from aggregated information such as security prices. These provide signals to traders about the private information of other traders, so they can adjust their beliefs accordingly [Dimitrov and Sami, 2008]. Ideally, this process should result in a situation in which all traders reach a consensus belief that reflects all available information. The successful aggregation of information in prediction markets relies “critically on traders adjusting their beliefs in response to other traders’ trades” [Dimitrov and Sami, 2008]. In prediction markets, more accurate forecasters obtain more capital over time with which to participate in the subsequent prediction markets. On the other hand, less accurate forecasters will lack the funds to have a large influence on the consensus probabilities [Powell et al., 2013].

Artificial prediction markets are developed upon the basic idea of real prediction markets, however, humans are replaced with computer programs (i.e participants are machine learning models). These markets have been advocated as powerful machine learning techniques [Chen and Vaughan, 2010]. They can be used to automate the process of building an ensemble from (in)homogeneous models.

Relatively little research on artificial prediction markets as a machine learning technique exists [Chen and Vaughan, 2010, Lay and Barbu, 2010, Barbu and Lay, 2012, Storkey, 2011, Storkey et al., 2012, Hu and Storkey, 2014], which we discuss them in Chapter 2 (page 19).

While these existing artificial prediction markets have many advantages, they suffer from some deficiencies. Specifically:

- The current artificial prediction markets only address the integration of various machine learning models and hence propose artificial prediction markets as an ensemble. None uses artificial prediction markets to combine a variety of data sources of different and varying quality.
- These models are offline techniques and none of these works investigates how an artificial prediction market can be used as an *online* technique. The participants in these models are classifiers that are trained offline and never retrain their hypothesis once they join the markets.
- Participants are assumed to use fixed strategies, such as constant betting functions [Barbu and Lay, 2012], utility functions [Storkey, 2011, Storkey et al., 2012] or static risk measures [Hu and Storkey, 2014], unlike human traders in real prediction markets. Therefore, they cannot learn from their past experience to improve their trading performance for subsequent markets. According to [Jian and Sami, 2012], trader strategies are an important factor in prediction markets, as market efficiency depends on appropriate behaviour by traders.
- In real prediction markets, aggregated information such as security prices provide signals to traders about the private information of other traders so they can adjust their beliefs accordingly. While the successful aggregation of information in these markets critically relies on traders adjusting their beliefs in response to other traders actions [Dimitrov and Sami, 2008], such a mechanism is not explicitly modelled in the existing artificial prediction markets. In other words, the participants in these models do not revise their beliefs and predictions once trading starts. In particular, they ignore wisdom of the surrounding crowd to improve their prediction and maximise their rewards.

1.3 Thesis Objectives and Contributions

Our goal in this dissertation is to develop an artificial prediction market which:

- 1) Is consist of a market mechanism and autonomous participants, where market and all participants perform as a unified entity with the global objective of predicting the outcome of an event accurately.
- 2) Integrates multiple data sources each analysed by multiple analysis models by instantiating an artificial participant –called an *agent*– for each combination of datasource and analysis model.
- 3) Works as an *online* technique, i.e. a prediction market is created every time a new prediction needs to be made and agents update their analysis models and trading strategies once the correct outcome is revealed.

- 4) Its participants have an adaptive trading strategy to mimic the behaviour of human traders in real world prediction markets in the sense that they learn from past experiences in order to improve their performance in future markets.
- 5) Its participants use wisdom of the crowd to revise their predictions in response to other traders' trades, during a market.
- 6) Shifts focus in response to changes in quality and availability of each individual agent (combination of a data source and analysis model) prediction. In other words, the technique adapts to dynamic environments, where the availability and quality of data sources and suitability of a model for each data set may change over time.
- 7) Is resilient to different proportions of low-and high-performing participants.
- 8) Its overall performance is close to or better than the best agent performance.
- 9) Is independent of each agent. Therefore, temporary or permanent deletion of an agent, for whatever circumstance, does not affect system performance significantly.

To achieve our objective, we put forward a technique, called the Artificial Continuous Prediction Market (ACPM). The model comprises of market participants and a market maker. Market participants are autonomous agents that have a budget, a trading strategy, a data source and an analysis model² to analyse their data and derive a prediction. The market maker runs the market, deals with agent transactions and determines the market prediction. Each market includes a number of rounds, where each agent sends its prediction and a bet on its prediction to the market maker. Subsequently, the market maker calculates the market prediction by combining all the individual predictions weighted by their corresponding bets. Once the true value of the record is known and the market is over, agents are informed of the correct outcome (the true value of the record). Then, the market maker rewards participants based on their prediction accuracy and their invested amount. Consequently, agents are incentivised to submit accurate prediction, and invest according to their estimate on quality of their prediction. Each agent learns from each market, based on the revenue they receive and the losses they make, in addition to finding out the correct answer. Consequently, they can update their trading strategy and analysis model for future markets.

We propose two trading strategies to be utilised by agents when participating in a market. The first one is the *Constant Strategy* which suggests agents simply dedicate a fixed ratio of their budget to invest in each round. The second one is the *Q-Learning based strategy* [Watkins, 1989]. This strategy advises the agent to what extent to rely on the market prediction, i.e wisdom of the crowd, as another source of information in order to improve its prediction in the subsequent rounds of a market. In other words, high performing participants may learn to rely on their own prediction more than market predictions but low performing participants learn to change their prediction based on market prediction. Hence, the amount of noise which is typically sent to the market maker by low performing participants is minimised.

²Analysis model and (machine) learning model terms are interchangeably used through this thesis.

Using a number of experiments, we empirically evaluate the performance of our model using various data sets from real-world syndromic surveillance domain and the widely-used repository of the University of California at Irvine (UCI). We also artificially generated some data sets to simulate the edge cases which are not presented in our real-world data sets. The experiments are designed and performed to answer the following questions:

- Can agents use the market prediction, the aggregated prediction of all market participants, as another source of information to improve their prediction in subsequent rounds of a market?
- Can ACPM shift focus in response to changes in quality of individual participants?
- Is ACPM performance resilient to different proportions of low- and high-performing participants?
- How does ACPM perform compared to other well known alternatives in the literature?
- How does ACPM perform in different application domains?
- How much time does ACPM require for each additional participant?
- How does ACPM perform in relation to its best performing agent in different situations?
- How many markets does ACPM require to recover in situations where the best performing agent suddenly exhibit a drop in quality or become unavailable?

1.4 Thesis Outline

This thesis is structured as follows:

Chapter 1: this chapter provides thesis motivation. In particular, it highlights the necessity of a meta model which can automate the process of aggregating different data sources and combining various analysis models, especially in this era in which we witness the increasing discovery of data sources and model innovations.

Chapter 2: related literature to this work are surveyed in this chapter. First, the chapter explains machine learning including ensembles, online learning and popular methods in these fields. Next, we cover reinforcement learning which is one of the branches of machine learning in which an agent learns by receiving feedback from interaction with the environment. In this thesis, we use reinforcement learning to propose a trade strategy for our agents. Afterwards, prediction markets and various works in this field are mentioned to elucidate the backbone of our proposed model. The chapter continues by discussing artificial prediction markets, which are similar to our proposed model, and utilise the concept of prediction market to propose a new machine learning technique. Finally, the chapter concludes by discussing the differences between our model and related works.

Chapter 3: we cover the detailed mechanism of our proposed model including its key components. In particular:

- (1) Agents: which play the role of prediction market participants. They predict the outcome by using their data source and analysis model and then participate in the market using their adaptive trading strategy.
- (2) Market maker: which runs the prediction markets and interacts with agents.
- (3) Integration function: which is the mechanism used by the market maker to aggregate the prediction of market participants in each market.
- (4) Reward function: which is the mechanism used by the market maker at the end of each market to reward agents based on their predictions and bets.
- (5) Rate per transaction parameters: which constrains the amount of agents' investments in order to (i) prevent ACPM from being unresponsive in cases where none of the participating agents have enough incentive to invest and (ii) prevent unsuccessful agents from bankrupting themselves and being eliminated from the system.
- (6) Agents' trading strategy: which advises an agent how to trade in the market. In this thesis, we propose two trading strategies, namely:
 - (i) Constant trading strategy which suggests the agent simply invest a portion of its budget in the market without considering neither the agent performance nor the environment conditions.
 - (ii) Q-learning trading strategy which suggests the agent constantly observes and compares its own and the market performance and participate in the market according to its observation.

Chapter 4: in this chapter we evaluate ACPM performance in different situations against various hypotheses. We use the real-world data set of syndromic surveillance and artificially generated data sets. The latter serves to explore the edge cases which are not present in the syndromic surveillance data set. Also, we evaluate ACPM with several data sets from the University of California at Irvine (UCI) repository in order to assess the performance of ACPM in different domains. Through these experiments, we investigate various capabilities of the system.

Chapter 5: in this chapter, we evaluate the performance of our model by applying it to the application domain of predicting influenza rate, which is known as syndromic surveillance. In this context, we predict the disease activity level of influenza-like illnesses on a specific date in the whole USA using publicly available data sources. The data used contains more than 100 real data sources from different sources including Google Flu Trends and Centers for Disease Control and Prevention (CDC), Google Trend and etc. The prediction of the system is then

compared with CDC ILI rate as ground truth. We also compare the performance of our model with two benchmarks of Google Flu Trend (GFT) and GP model [Lampos et al., 2015].

Chapter 6: conclusions and contributions of this thesis are addressed in this chapter, along with directions for further research.

1.5 Related Publications

Some parts of this thesis is published in the following papers:

- Fatemeh Jahedpari. Artificial prediction markets for online prediction. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Doctoral Consortium, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015, pages 43714372. AAAI Press, 2015. ISBN 978-1-57735-738-4.

This paper provides a summary of the thesis.

- Fatemeh Jahedpari, Marina De Vos, Sattar Hashemi, Benjamin Hirsch, and Julian Padget. Artificial prediction markets for online prediction of continuous variables, accepted for publication in *Proceedings of second workshop on Synergies between multiagent systems, machine learning and complex systems (TRI 2015)*, held together with IJCAI 2015 Buenos Aires, Argentina, page 1, 2015.

In this paper, we propose an Artificial Continuous Prediction Market (ACPM) for predicting a continuous variable based on the integration of diverse data sources with different varying quality. We demonstrate that ACPM acts as an adaptive ensemble algorithm which is capable of shifting focus in response to changes in individuals predictions.

- Fatemeh Jahedpari, Julian Padget, Marina De Vos, and Benjamin Hirsch. Artificial prediction markets as a tool for syndromic surveillance. *Crowd Intelligence: Foundations, Methods and Practices*. CEUR Workshop Proceedings, 2014.

In this paper, we focus on the problem of how to identify emerging trends after the original textual data has been processed into a quantitative form suitable for the application of machine learning techniques. We present some preliminary ideas, including an agent-based implementation and some early results, about the application of artificial prediction markets to such data, taking the specific domain of syndromic surveillance (early stage recognition of epidemics) as an example, using publicly available data sets.

Chapter 2

Literature Review

In this thesis, we adapt the concept of prediction market to propose a novel machine learning technique, namely ACPM. Our proposed model is an online technique in which a prediction market is instantiated whenever a prediction needs to be made. The participants of the markets are artificial agents which have a machine learning model to arrive at a prediction. By assigning a different learning model to each agent, ACPM functions as an ensemble. Agents are also equipped with an adaptive trading strategy – based on reinforcement learning – to guide their participation in the market.

In this chapter, we provide an overview of the necessary background for our work. In particular, Section 2.1 covers machine learning background including various machine learning models, ensemble learning, online learning and reinforcement learning. Next, Section 2.2 provide the prediction market background by discussing its key elements and applications. After that, Section 2.3 discusses the works which links prediction markets to machine learning techniques. Finally we conclude in Section 2.4 by discussing the motivation behind this work and the differences between our model and those appearing in the literature.

2.1 Machine Learning Background

“A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .” [Mitchell, 1997]

Machine learning is the field of scientific study that develops algorithms which automatically extract information from data. In the next section, we review different machine learning approaches and models.

2.1.1 Machine Learning Approaches

Machine learning is usually divided into four types: (i) supervised, (ii) unsupervised, (iii) semi-supervised and (iv) reinforcement learning. In this section, we briefly explain each of these types.

I) Supervised Learning

In supervised learning, the training data comprise a number of training records (instances) where each record includes the input vectors and their corresponding target value (correct answer) [Mohri et al., 2012]. The objective of the algorithm is learning a hypothesis to map the training input to the corresponding target value, so that the algorithm can use the hypothesis to predict the data records whose target values are not available. Based on the target value types, supervised learning is further divided into two categories, namely classification and regression. If the target value is a finite number of discrete categories then the task is a *classification* problem. Otherwise, if the target value is a continuous value, then the task is known as a *regression* problem.

For example, classifying customers credit risk into “low-risk” and “high-risk” group is a classification problem. In this problem, a training example may include information about various customers including their age, profession, saving, monthly income, number of open loans, number of mortgage, number of dependents in family and so forth. The training example, also, includes the correct answer for each customer, i.e. whether the customer is a “low-risk” or “high-risk” customer. An example of regression task is predicting the temperature of a region based on previous days’ temperatures.

II) Unsupervised Learning

In unsupervised learning, which is also known as knowledge discovery, the training data do not include the target value and the objective is finding an interesting structure within the data [Murphy, 2012]. One classic unsupervised learning task is *clustering*. Clustering refers to discovering groups of similar examples or hidden patterns in data. The other common task in unsupervised learning is *discovering latent factors* which means reducing the dimensionality of high dimensional data, while preserving the essence of the data. *Discovering graph structure* is another case of unsupervised learning, in which one can find the most correlated variables by presenting them as a graph where nodes refer to variables and the links indicate the direct dependences between variables.

III) Semi-Supervised Learning

Semi-supervised learning falls between supervised learning and unsupervised learning [Chapelle et al., 2010]. Most of the training data in this class of learning do not include target values and only a small percentage of data is labeled with the correct answer. For example, an algorithm is presented with a data set of movies and their ranking, but not all the users rank all the movies and hence the training data have a lot of missing information.

IV) Reinforcement Learning

Reinforcement learning [Sutton and Barto, 1998] is another type of machine learning where the algorithm learns based on the feedback it receives from the environment. For example, a computer can learn to play the game of backgammon to a high standard using rein-

forcement learning [Tesauro, 1994]. Section 2.1.4 (page 28) provides more explanation of reinforcement learning.

Our technique is a supervised machine learning model which concentrates on regression problems. Having covered different machine learning types and models, we now describe ensemble methods in the next section in which the prediction of different models are combined in some way.

2.1.2 Ensemble Methods

This section covers some popular ensemble models. In ensemble learning, various homogeneous or inhomogeneous models – which are called base learners or base models – are trained independently; then their prediction for a given data record combined somehow to be returned as the overall prediction for that data record.

Ensembles are divided into two categories of hybrid and non-hybrid. While the hybrid ensembles combine different types of models, non-hybrid ensembles choose one model as the main model and then replicate the model using different training schemes (i.e. with different initialisation values, different training data records or features). We recall, our proposed model in this thesis can function as an hybrid or non-hybrid ensemble by granting different homogeneous or inhomogeneous models to its agents.

A simple ensemble model is voting which has been used as early as 1974 [Spanjersberg, 1974]. Voting models count the votes of each base model to make the prediction. In simple majority vote the prediction of base models have equal weights. However, in a weighted combination, the base models are assigned different weights in a way to maximise the performance of the ensembles on the training data. In a weighted combination, the overall prediction is constructed by calculating weighted sums of the base models' predictions.

One of the earliest work on ensembles refers to Neural Network Ensembles. Hansen and Salamon [1990] show that an ensemble of neural networks, which are also known as committees, performs far better than using a single neural networks. In these models, different neural networks are trained independently and then their prediction is combined using a simple averaging (in regression tasks) or a majority vote (in classification tasks) [Spanjersberg, 1974] or a weighted combination scheme.

The most common ensemble models in machine learning are bagging and boosting and usually their base learners are decision trees. Bagging [Breiman, 1996] stands for “bootstrap aggregating” in which different training data subsets are chosen randomly with replacement from the entire training data set. Classifiers of the same type are trained on each subset and their decisions are combined by a simple majority vote. Several variations of bagging models [Bauer and Kohavi, 1999, Canuto et al., 2007, Hothorn and Lausen, 2005] are proposed since its introduction, however the original model [Breiman, 1996] is still widely used [De Bock et al., 2010].

One popular type of bagging is random forest proposed by Breiman [2001]. A random forest builds various decision trees using dissimilar samples of training data set. This model changes how the trees are constructed by adding randomness when choosing the splitting node.

According to Liaw and Wiener [2002], the random forest technique is very user-friendly as it takes only two parameters – which are the number of variables in the random subset at each node and the number of trees in the forest – and the performance of the model is typically not very sensitive to these parameter values. Random forests are shown to be very effective compared to many other models including discriminant analysis, support vector machines and neural networks [Breiman, 2001, Fernández-Delgado et al., 2014].

Boosting [Schapire, 1990, Freund et al., 1996] is another ensemble algorithm which builds a series of base models consecutively. The training set used for each base model is chosen according to the performance of the preceding base models. For example, the algorithm first randomly chooses a subset of training data set from the entire training data set with uniform distribution and trains a classifier on that. It then chooses a second set of data in a way that a subset of data includes the training examples which were misclassified by the first classifier. The next classifier is then trained with instances on which the first two classifiers disagree. Finally, the decision of these classifiers for the new data are combined by majority voting.

A well-known extension of the boosting algorithm is AdaBoost [Freund and Schapire, 1997] which is an adaptive boosting algorithm and extends boosting to multi-class and regression problems. Gradient boosting models, which are also called Gradient Boosting Machines (GBMs), are boosting methods based on a gradient-descent formulation [Freund and Schapire, 1997, Friedman, 2001]. Friedman [2002] extends gradient boosting by proposing Stochastic Gradient Boosting, in which a base learner trains on a subsample of the training set drawn at random without replacement, at each iteration of the training process. This modification results in a substantial improvement compared to the original gradient boosting. Cubist [Holmes et al., 1999] is a type of boosting ensemble and it works by developing a series of trees consecutively with adjusted weights. The final prediction is determined by averaging the predictions of all base learners. Gradient Boosted Regression Trees (GBRT) [Zheng et al., 2008] is another ensemble method in which each base model is a simple decision tree and is based on gradient boosting. Long and Servedio [2010] demonstrate that boosting algorithms cannot perform effectively if a percentage of the training data is mislabeled as the boosting algorithm concentrates extensively on properly classifying these training examples.

The difference between boosting methods and bagging methods lies in their training set. In boosting methods, distribution of the training sets are adaptively altered based on the performance of base models, but not in bagging models. In boosting, consecutive generated trees put further weight on instances which are incorrectly predicted by previous base models. However, in bagging, sequential trees are independently built using a bootstrap sample of the training data and do not depend on the former base models.

Stacked generalisation [Wolpert, 1992] is another well known ensemble model which is different from bagging and boosting in a sense that stacking does not normally combine base learners of the same type, but combines base learners constructed by different learning algorithms. In stacked generalisation, base models are independently trained on the training data. After that, another model is trained to combine the predictions of base models.

According to Natekin and Knoll [2013], the ensemble approaches are successful only when they combine a number of relatively weak models to achieve a stronger prediction rather than

combining a set of strong models to yield a superior prediction.

So far in this section, we mentioned various machine learning approaches and types. Regardless of their types, machine learning models can learn either in an online or offline setting (i.e. batch training). While in the former data becomes available sequentially, in the latter one the entire training data is available at once. More specifically, in offline learning, an algorithm takes the entire training data set and returns a model (hypothesis) in order to determine a mapping from the input vector to the corresponding target value. In this type of learning, a new example that the learner encounters, is assumed to be similar to the training set; In other words, all the training and test examples are identically distributed. However, when the nature of data may change over time, online learning tackles the problem by updating the model every time a new example arrives in order to adapt to the non-stationary environment.

Our proposed model in this dissertation is an online learning algorithm. The next section gives a general overview of online learning models.

2.1.3 Online Learning

This section explains online learning and covers several well-known algorithms in the literature. In online learning, a learner sequentially make predictions as data becomes available. That is, at each time t , using past instances x_1, \dots, x_{t-1} and the arrived data x_t , the learner predicts the target value of x_t (i.e. p_t). Once the true outcome y_t is revealed by the environment, the learner suffers a loss $L(p_t, y_t)$ which is the difference of the learner prediction and the true outcome. Then, the learner is allowed to improve its model (See Figure 2-1). The overall performance of the learner is measured based on the sum of all these losses – known as the cumulative loss.

Online learning usually is used in cases occurring over time. One example is predicting the price of a stock. In this example the target variable is price. Once the true price of the stock is revealed, the quality of the learner’s prediction is measured by the loss function. The learner has the chance to improve its hypothesis so as to be more accurate in later rounds. In this context, the goal of a learner is minimising the cumulative loss suffered along its run, as more and more samples become available over time.

Different online learning methods have been developed in variety of research areas. The seminal book “Prediction, learning, and games” [Cesa-Bianchi and Lugosi, 2006] studies the connections between online learning, universal prediction, and repeated games. In this book, results from the various fields are described under the umbrella of the prediction with expert advice framework, which we discuss in the next section.

2.1.3.1 Prediction with Expert Advice

The paradigm of Prediction with Expert Advice (PEA) was first introduced as a model of online learning in the pioneering paper by Littlestone and Warmuth [1994]. Gradually, the (similar) problem has been (extensively) explored by other studies in the literature, including [Haussler et al., 1995, Cesa-Bianchi et al., 1997, Freund and Schapire, 1997, Cesa-Bianchi and Lugosi, 2006].

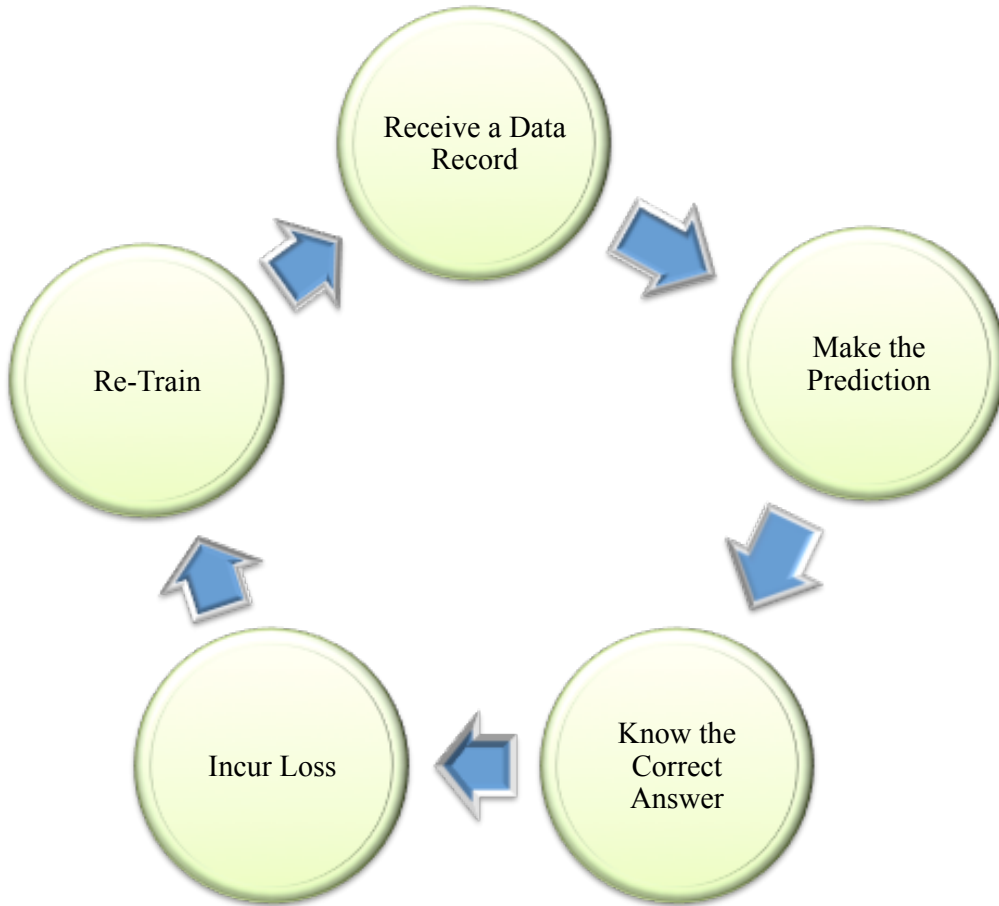


Figure 2-1: Online Learning Process.

Predicting with expert advice and many variations and extensions have been addressed in a number of different communities, under names such as the “sequential compound decision problem” [Robbins, 1985, Blackwell, 1956], “universal prediction” [Feder et al., 1992], “universal coding” [Shtar’kov, 1987], “universal portfolios” [Cutland et al., 1991].

In prediction with expert advice framework (see algorithm 1), the forecaster is required to predict an unknown sequence y_1, y_2, \dots of an outcome space Y and the forecaster’s prediction is p_1, p_2, \dots which belong to a convex subset of the decision space D . The forecaster has access to predictions of a set of n experts $E_1 \dots E_n \in \mathcal{E}$.

At each time t , the forecaster receives an incoming training instance and computes its prediction p_t on the basis of expert predictions $f_{E_1, t} \dots f_{E_n, t} \in D$. The environment discloses the real outcome y_t of the instance after the forecaster has made the prediction. After that, the performance of the forecaster and experts are measured using a loss function $L : D \times Y \rightarrow \mathbb{R}$.

It is irrational to expect the algorithm to achieve a small cumulative loss if none of the experts perform well [Cesa-Bianchi and Lugosi, 2006]. Hence, it is common in the literature of PEA to measure the performance of the model by comparing it to the performance of the best performing expert. Therefore, the discrepancy between the forecaster’s accumulated loss and that of an expert is called regret R . In hindsight it shows how much the forecaster regrets not

Algorithm 1 Prediction With Expert Advice

Input: decision space D , outcome space Y , loss function L , set of experts \mathcal{E} .

- 1: **for** $t = 1, 2, \dots, T$ **do**
 - 2: the environment chooses the next outcome $y_t \in Y$
 - 3: the experts reveal their predictions $f_{E,t} \in D : E \in \mathcal{E}$ to the forecaster
 - 4: the forecaster computes its prediction $p_t \in D$
 - 5: the environment reveals the outcome y_t
 - 6: the forecaster suffers loss $L(p_t, y_t)$
 - 7: each expert E suffers loss $L(f_{E,t}, y_t)$
 - 8: **end for**
-

having listened to the advice of an expert. The forecaster's goal is to keep the cumulative regret as small as possible. The number of experts and the loss function have a key role in keeping the regrets small [Cesa-Bianchi and Lugosi, 2006]. The forecaster cumulative regret is defined as:

$$R_T = \sum_{t=1}^T (L(p_t, y_t) - \min_{i \in 1 \dots n} \sum_{t=1}^T L(f_{i,t}, y_t)), \quad (2.1)$$

so that:

$$R_T = \widehat{L}_T - \min L_{E,T} \quad (2.2)$$

where $\widehat{L}_T = \sum_{t=1}^T L(p_t, y_t)$ denotes the forecaster's cumulative loss and $L_{E,T} = \sum_{t=1}^T L(f_{E,t}, y_t)$ denote cumulative loss of expert E . The instantaneous regret with respect to expert E at time t is defined as:

$$r_{E,t} = L(p_t, y_t) - L(f_{E,t}, y_t) \quad (2.3)$$

and the cumulative regret with respect to expert E at time t is defined as:

$$R_{E,T} = \sum_{t=1}^T r_{E,t} \quad (2.4)$$

A forecast who can achieve a vanishing per-round regret is said to be a Hannan-consistent forecaster [Hannan, 1957, Cesa-Bianchi and Lugosi, 2006]. Hannan consistency is defined as:

$$\frac{1}{T} \left(\widehat{L}_T - \min_{i=1, \dots, N} L_{i,T} \right) \xrightarrow{T \rightarrow \infty} 0 \quad (2.5)$$

Experts can be seen as a black box of unknown computational power. Indeed, a small regret assures that, even when the model does not define the state of nature perfectly, the forecaster performs roughly as well as the best available model [Cesa-Bianchi and Lugosi, 2006].

2.1.3.2 Exponentially Weighted Average Forecaster

The Exponentially Weighted Average Forecaster (EWA) [Cesa-Bianchi and Lugosi, 2006] is a popular technique in the PEA literature. The intuitive idea of EWA algorithms is assigning large weights to experts with low regrets and small weights for expert with high regrets. For any sequence of length T , the average number of prediction mistakes of EWA is bounded above by the average number of prediction mistakes made by the best expert plus a constant. Thus, the EWA algorithm will perform well when for every sequence there exists an expert that performs well on it and when the sequence is long enough. The forecaster computes its prediction p_t at time t according to:

$$p_t = \frac{\sum_{i=1}^n w_{i,t-1} f_{i,t}}{\sum_{j=1}^n w_{j,t-1}} \quad (2.6)$$

where $f_{i,t}$ and $w_{i,t}$ denotes the prediction and weight of expert i at time t respectively. Once the outcome is revealed and the losses $L(f_{i,t}, y_t)$ are calculated, the expert weights are updated as follows:

$$w_{i,t} = w_{i,t-1} \exp(-\eta L(f_{i,t}, y_t)) \quad (2.7)$$

where η is the learning rate parameter. Therefore, EWA makes the following prediction:

$$p_t = \frac{\sum_{i=1}^n \exp(-\eta L_{i,t-1}) f_{i,t}}{\sum_{i=1}^n \exp(-\eta L_{i,t-1})} \quad (2.8)$$

Exponential Weighted Average Forecaster with having $\eta = \sqrt{2 \ln n / T}$, is guaranteed to achieve:

$$R_T \leq \sqrt{2 T \ln n} \quad (2.9)$$

The disadvantage of the exponential weighting is that optimal tuning of the parameter η requires advance knowledge of the horizon T [Cesa-Bianchi and Lugosi, 2006].

2.1.3.3 Tracking the Best Expert

Another method based on PEA is Tracking the Best Expert proposed by Herbster and Warmuth [1998]. This model partitions the sequence of trials into segments with the goal of bounding the forecaster's regret in each segment. The method models the situations which nature of data may change and different experts are better for a certain subsequence of trials (i.e each segment).

In this setting, the best expert may change over a series of trials, and the weight of an expert, which may be the next best expert, might happen to be very small for the current trial. Therefore, each expert share a proportion of its weight with other experts after each trial to make sure any expert can recover quickly; this process is called Share Update. The authors propose two algorithms called Fixed-Share and Variable-Share each with exclusive share update which happens after updating the weight of experts. In both of them, the forecaster computes its prediction using Equation 2.8 and after the outcome is revealed, weight of experts are updated

using Equation 2.10:

$$w_{i,t} = w_{i,t-1} \exp(-\eta L(f_{i,t}, y_t)). \quad (2.10)$$

For squared loss function, the author suggests Parameter η should be set to $1/2$. After that, a specific share update happens, which we explain below.

In the Fixed-Share Algorithm, the experts share a fixed fraction of their weights with each other and the weight of each expert is updated using Equation 2.11.

$$w_{t,i} = (1 - \alpha) w_{t,i} + \sum_{j \neq i}^n \frac{\alpha}{n-1} w_{t,j} \quad (2.11)$$

where n is number of experts, α quantifies the rate of shifting that is expected to occur per trial (i.e number of times the best expert changes over the number of trials).

Variable-Share algorithm assumes that the loss per trial is in $[0, 1]$. In this algorithm, the share is commensurate with the loss of the expert in the current trial, which means the expert with no loss does not share any weight. In Variable-Share algorithm, the weight of each expert is updated using Equation 2.12.

$$w_{t,i} = (1 - \alpha)^{L(f_{i,t}, y_t)} w_{t,i} + \sum_{j \neq i}^n \frac{(1 - (1 - \alpha)^{L(f_{i,t}, y_t)})}{n-1} w_{t,j} \quad (2.12)$$

where n is number of experts, α quantify the rate of shifting that is expected to occur and hence α is roughly the rate of shifting per unit of loss of the best partition (i.e the sequence of segments with the smallest loss).

They show that algorithm prediction is close to those of the best expert for each segment. In particular, in a single segment case, the model regret is equal to the best expert regret plus $[c \log n]$, where n is the number of experts and c depends on the loss function.

2.1.3.4 The Exponentiated Gradient algorithm

The Exponentiated Gradient algorithm (EG) is proposed by Kivinen and Warmuth [1997] in the context of PEA. In this model, the forecaster computes its prediction using Equation 2.8. Once the outcome is known, the forecaster updates the weights of expert using Equation 2.13.

$$w_{i,t} = w_{i,t-1} \exp(-\eta L'(p_t, y_t) f_{i,t}) \quad (2.13)$$

Hence, for the square loss function, Equation 2.13 becomes:

$$w_{i,t} = w_{i,t-1} \exp(-2 \eta (p_t - y_t) f_{i,t}) \quad (2.14)$$

where η is the learning rate parameter. According to the authors, the typical choice for initial expert weights is $(1/n)$ and a common learning rate could be $\eta = 2/(3R^2)$, where R is an upper bound for the maximum difference between expert predictions at time t .

2.1.3.5 Follow the Best Expert

Follow the Best Expert (FBE) is another well-known PEA model proposed by Robbins [1985]. Regarding the performance of FBE, Cesa-Bianchi and Lugosi [2006] state “Perhaps surprisingly, this simple predictor has a good performance under general conditions on the loss function and the experts”. At each time t , FBE forecaster predicts the same as the expert who has the minimum cumulative loss for time $= 1..t - 1$, hence:

$$p_t = f_{E,t} \quad \text{if} \quad E = \arg \min_{E' \in \mathcal{E}} \sum_{t=1}^{T-1} L(f_{E',t}, y_t) \quad (2.15)$$

Having covered well-known online learning models, now we move to explain reinforcement learning, which we use in this thesis to propose a trade strategy to guide agents’ participation in the market.

2.1.4 Reinforcement Learning

Reinforcement learning is one of the adaptive and nonlinear algorithms that is independent of environmental conditions [Sutton and Barto, 1998]. According to Sutton and Barto [2011] “This framework is abstract and flexible and can be applied to many different problems in many different ways.” Notable historical examples of reinforcement include Neuro-gammon [Tesauro, 1989], the financial traders [Neuneier, 1996], an elevator scheduler [Barto and Crites, 1996] and a space-shuttle payload scheduler [Zhang and Dietterich, 1995].

The automated learning scheme of reinforcement learning results in less need for a human expert who knows about the domain application of a problem [Baziukaite, 2007]. According to Khuen et al. [2005] “Much less time will be spent designing a solution, since there is no need for hand-crafting complex sets of rules as with Expert Systems, and all that is required is someone familiar with Reinforcement Learning”.

Reinforcement learning differs from standard supervised learning in the sense that the input and the correct output pairs are never given. In this context, the learner is called the agent and the environment is everything outside the agent with which agent interacts. Anything that cannot be altered by the agent is considered to be external to the agent and part of the environment. As the agent interacts with the environment and performs actions, the environment responds to the actions and moves to a new state. An agent learns from its history of interaction with the environment, in order to find the optimal behaviour in future. More specifically, an agent collects valuable knowledge about the available environmental states, actions, transitions and rewards in order to act optimally. In other words, an agent attempts to discover the optimal policy by interacting with the environment and receiving feedback.

For example, one of the application of reinforcement learning is when a mobile robot learns to clean a room. In cases when the robot has a low level of battery, the robot is required to decide whether it should find the way to return back to its battery recharging station or should enter a new room to resume its cleaning task. The robot can make its decision based on how much time and power it consumed in past to find the recharger. The robot seeks to interact

with its environment to achieve a goal, despite uncertainty about its environment. The robot actions affect the future state of the environment, which is the next location of the robot and hence impact the options accessible to the agent at later times. The robot should take into account indirect and delayed consequences of actions. When the consequences of actions cannot be fully predicted, the robot must monitor its environment regularly and respond accordingly. The robot interaction with the environment is indispensable for adapting its behaviour to the environment and the robot uses its experience to improve its performance over time.

Figure 2-2 shows the general framework for the reinforcement learning process as described in [Sutton and Barto, 2011]. In reinforcement learning, an agent interacts with an environment in discrete time steps. The behaviour of the agent is described as a sequence of actions the agent follows over a period of time. At each time t , the agent chooses an action a_t from the set of available actions. As a consequence of the agent action, the environment moves to a new state s_{t+1} . According to this transition (s_t, a_t, s_{t+1}) the agent receives reward r_{t+1} . In the most challenging cases, the actions may not only affect the immediate reward but also all the subsequent states and correspondingly their rewards [Sutton and Barto, 1998].

The action with the highest value in a particular state is called the greedy action and the policy of choosing the greedy action is called the greedy policy. The agent who follows the greedy policy *exploits* its current knowledge of the values of the actions. Inversely, the agent who chooses one of its non greedy actions is *exploring* to improve its estimate of other actions value. While exploitation maximises the expected reward on one state, exploration potentially produces the greater total reward in the long run.

The objective of an agent in reinforcement learning is studying how to map situations into actions (i.e the best action to execute in each situation) [Sutton and Barto, 2011]. In particular, agents must learn to choose actions in a way to maximise their cumulative reward. Unlike supervised forms of machine learning, the agent is never given what is the best action to choose in each state, but must learn which actions yield the most reward by experiencing them. For instance, a dog can learn to bring a ball by the reward or punishment it receives. If the dog brings the ball, it receives a cookie as an award or may get nothing as a punishment. Therefore, the dog gradually realises that fetching the ball leads into positive reward and learns to do so.

In addition to agent and environment, a reinforcement learning system includes a reward function, a policy, a value function, and, optionally, a model of the environment.

2.1.4.1 Reward Function

A *reward function* maps each perceived state (or state-action pair) of the environment to a reward value, a single number which describes how desirable the state is. Rewards are given to the agents immediately and describe features of the problem. The reward function must essentially be unchangeable by the agent. The main objective of an agent is maximising the total accumulated reward in the long run.

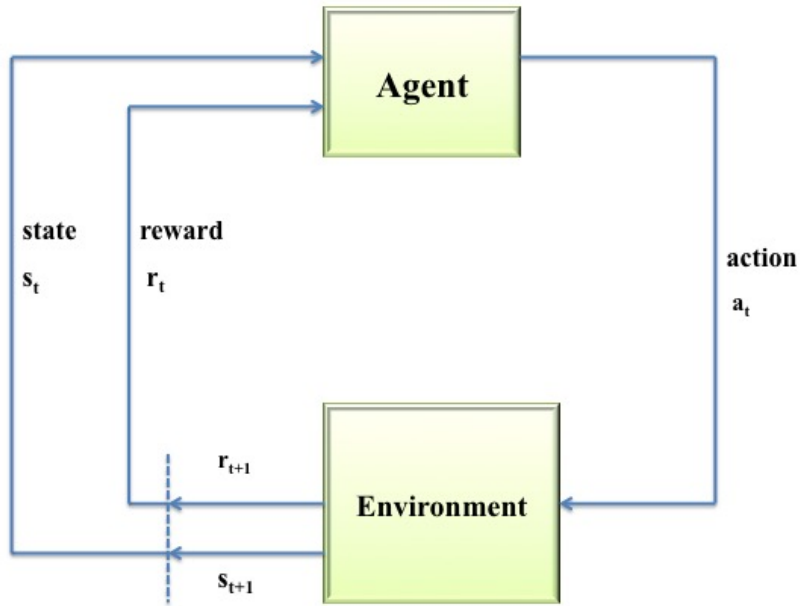


Figure 2-2: The agent and environment interaction in reinforcement learning [Sutton and Barto, 2011].

2.1.4.2 Policy

A *policy* defines how an agent should behave in a specific situation. In other words, it maps the states of the environment to one of the actions which are available to the agent in that state. A policy may be a simple function, a lookup table or involve extensive computation such as a search process. Policy is the core of a reinforcement learning agent in the sense that it solely is adequate to determine behaviour [Sutton and Barto, 2011]. The optimal policy corresponds to determining the best action in each state. The optimal policy π maximises the expectation of accumulated discounted rewards over all time steps R , which is defined as:

$$R = \sum_{t=0}^{\infty} \gamma^t r_{t+1} \quad (2.16)$$

The parameter $\gamma \in [0, 1]$ is a discount factor to specify whether the agent cares about short term or long term rewards. If $\gamma = 0$, the agent looks forward to short term reward and hence only considers the reward of the current time step. However, with $\gamma = 1$, the reward in each time step has the same significance for the agent. One approach of finding a good policy is to first learn a value function, which is explained in the next section, for all states and then choose the policy that maximizes this value function in every state.

2.1.4.3 Value Function

The converse of the reward function which specifies which action brings rewards immediately, a *value function* states what is beneficial in the long run. The value of a state is the amount of reward an agent can anticipate to accrue over the future, starting from that state. Value functions indicate the long-term benefits of states after taking into account each state it is probable to reach after the current state and the rewards obtainable in those states [Sutton and Barto, 2011]. For example, a state which is regularly followed by other states that yield high rewards should have a high value even though it might have a low immediate reward.

Agents should seek actions to achieve the highest value and not the highest reward because these actions obtain the utmost amount of reward for agents in the long run. Determining values is much more challenging than determining rewards. Values must be estimated from the history of experiences that an agent gains over its lifetime, but rewards are given directly by the environment. In order to calculate the optimal value function analytically, a perfect model of the environment is required. Otherwise, the next states cannot be determined for actions of the state.

While the *state-value function* specifies the value of a state which is the overall reward that it is expected to obtain when choosing that state as starting point, the *action-value function* specifies the overall return for choosing a particular action in a state and following an optimal policy thereafter [Sutton and Barto, 1998].

2.1.4.4 Model

The model of the environment imitates the behaviour of the environment. Given a state and an action, a model predicts the subsequent next state and next reward. Models can be used for planning as they help agents to determine a sequence of actions by anticipating likely future situations before they are experienced.

Having covered the general overview of the reinforcement learning, we now move to the next section to present a well-known reinforcement learning model, namely Q-learning.

2.1.4.5 Q-learning

In some situations, an agent has no information about the states transitions and their expected rewards, hence does not have a model of the environment but can learn by interacting with the environment. In such situations, the agent can use a model free reinforcement learning method. According to Sutton and Barto [2011], these type of methods are ‘undoubtedly’ the most central and novel to reinforcement learning. Q-learning [Watkins, 1989] is a popular model free reinforcement learning method.

To compare the expected utility of the available actions without requiring a model of the environment, Q-learning learns an action-value function to estimate the utility of taking an action in a particular state and following the optimal policy – a policy which achieves the highest expected return from any initial state – thereafter.

In Q-learning, the problem contains a set of states and a set of actions per state. By performing an action, the agent moves from one state to another and receives a reward. To achieve its goal, which is maximising its total reward, an agent chooses the optimal action in each state. The optimal action in each state is the action that has the highest long-term reward.

To put it simply, an agent is given a representation of the environmental states and possible actions in each state. The agent learns the value of performing each action in each state, which is called *action-value* or *Q-value*, by experiencing them. In the beginning, the agent assumes an arbitrary Q-value, for example 0, for all actions. Gradually, the agent explores the environment and experiences the action-value space.

In doing so, after trying an action in a state, the agent evaluates how good the action was according to the reward it received or the transited state desirability. Based on this calculated value, the agent updates the Q-value for that action in that state. In each state, the actions which leads to better outcome get a higher value of Q than the actions which result in less desirable outcomes. Therefore, if the agent is confronted with the same state in future, it chooses the action with higher Q-value.

It is important to note that an agent updates previous state-action combinations, as the agent can only know the Q-value of each state action combination only after seeing the outcome of taking that action in that state.

One of the challenges for an agent in this setting is the trade-off between exploration and exploitation. An agent prefers to choose (exploit) actions which are already tried and return high rewards, but an agent should also try (explore) other actions to discover actions with potentially higher rewards. One way to trade off exploration and exploitation is the \mathcal{E} -greedy strategy, which selects the greedy action by probability $(1 - \mathcal{E})$ and selects a random action otherwise. The Q-learning algorithm is shown in Algorithm 2. This model updates the Q-value for performing an action in a state using the following formula.

$$Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (2.17)$$

where α is the learning rate and γ is the discount factor.

The learning rate α states to what extent the newly acquired information should override the old information. Parameter α equal to 0 makes the agent not learn anything from the new experience, while α equal to 1 makes the agent override all previous information. Parameter γ is a value between 0 and 1 and states how less a delayed rewards worth.

In this thesis, we equip our agents with an adaptive trading strategy based on Q-learning, which is a popular reinforcement learning method. Reinforcement learning has been extensively used to develop intelligent trading strategies and has demonstrated significant promise in trading and asset allocation [Gorse, 2011]. In the next section, we cover some trading strategies based on reinforcement learning.

Algorithm 2 Q-learning Algorithm

```
1: initialise  $Q(s, a)$  arbitrarily
2: for each episode do
3:   initialise  $s$ 
4:   while state  $s$  is not terminal do
5:     choose action  $a$  in state  $s$  using policy derived from  $Q$  (using  $\mathcal{E}$ -greedy policy, i.e select
       the greedy action by probability  $(1 - \mathcal{E})$  and select a random action otherwise.)
6:     take action  $a$ , observe reward  $r$  and new state  $s'$ 
7:      $Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$ 
8:      $s \leftarrow s'$ 
9:   end while
10: end for
```

2.1.4.6 Reinforcement Learning in Financial Markets

In financial markets, human traders usually use some forms of computational techniques to determine how to participate in a market. Some human traders prefer to use some algorithmic computer program to guide them on how to trade. Reinforcement learning based trading systems have been shown to maximise traders profits [Moody and Wu, 1997]. We now describe some of the important works in this field.

Neuneier [1996] formalise asset allocation as a Markovian decision problems and optimise it using a reinforcement learning algorithm. Later, Neuneier [1997] extends his work by incorporating Q-learning. Moody and Wu [1997] demonstrate the efficacy of training trading systems using reinforcement learning with the goal of maximising traders' profits in trading system. Moody and Saffell [1998] propose to use recurrent reinforcement learning to directly optimise different measures of trading system performance such as profit, economic utility, risk-adjusted return. They also propose a new measure for financial markets which is called the differential Sharpe ratio. Moody and Saffell [1998] state that a trading system based on recurrent reinforcement learning "significantly outperforms systems trained using supervised methods for traders of both single securities and portfolios."

Gao and Chan [2000] propose a portfolio management system based on Q-learning and show that the performance of a trader can be enhanced by considering absolute profit and relative risk-adjusted profit as performance indicators. Mihatsch and Neuneier [2002] utilise a risk sensitivity measure to incorporate in Q-learning model. Lee and O [2002] propose a trading system which is based on Q-learning models and includes four agents where each agent controls different aspects of trading.

Dempster and Romahi [2002] suggest an automated trading system based on a hybrid approach which combines a genetic algorithm with a reinforcement learning. Later, they extend their work by including risk management, automatic parameter tuning and dynamic utility optimisation in a layered system [Dempster and Leemans, 2006]. Using a large financial data sets

from NASDAQ online market¹, Nevmyvaka et al. [2006] demonstrate the promise of reinforcement learning methods by applying reinforcement learning to the problem of optimised trade execution.

Lee et al. [2007] propose a Q-learning algorithm based on a multi agent system where each agent has its own capability and specialty. They demonstrate that reinforcement learning can successfully address the problem of combining stock price prediction models with dynamic trading strategies in order to develop an automatic trading system. Gorse [2011] compares the performance of a stochastic adaptation of the recurrent reinforcement learning and a genetic programming approach based on daily, weekly, and monthly stock index data, and demonstrate that recurrent reinforcement learning can reliably outperform the genetic programming approach for higher frequency data.

Having discussed machine learning background in this section, now we move to the next section to cover prediction markets whose concept is adapted by our proposed model.

2.2 Prediction Market Background

Prediction markets have been used to forecast accurately the outcome of numerous events such as political contests, sporting events, and even economic outcomes [Nikolova and Sami, 2007]. According to Credit Suisse First Boston, prediction markets have “proven to be uncannily accurate in predicting all types of events.” (Wall Street Journal, July 30, 2003, p. C1.). These markets are increasingly being applied by governments and corporations as a means of collecting, summarising and aggregating dispersed, private information [Hanson et al., 2006]. Moreover, various studies show prediction markets outperform experts, opinion polls and group consensus in different application domains [Forsythe et al., 1992, 1999, Oliven and Rietz, 2004, Berg et al., 2008, Berg and Rietz, 2003, Gandar et al., 1998, Plott and Chen, 2002].

Prediction markets utilise the ‘wisdom of the crowd’ to predict the outcome of a future event [Ray, 2006]. The principle of the ‘wisdom of the crowd’ states that the collective opinion of a group is wiser than the opinion of individuals. The concept of wisdom of the crowd originates from an experiment performed one hundred years ago in England, where 800 people participated in an annual show to estimate the weight of an ox. After calculating the median average of all individuals’ predictions, Galton [1907] found out that the crowd prediction is very accurate and can even be better than every single individual prediction.

A success criterion for the wisdom of the crowd is including participants with accurate and diverse information. According to Hong and Page [2004] “groups of diverse problem solvers can outperform groups of high-ability problem solvers”. As a formal example, assume we have a number of predictors each with near-zero bias, hence each predictor reports a prediction, which is close to the ground truth. If each predictor gives the same prediction, then averaging of the predictions will not gain anything more than a single predictor. However, if the predictions are well spread around the ground truth, then the average of the predictions will be closer to the ground truth than most of the individual predictors. Therefore, the wisdom of crowd occurs

¹<http://www.nasdaq.com> (Retrieved Feb 5, 2016).

when there is enough variance between participant predictions.

It is well known that market prices reflect the aggregated knowledge of all the traders participating in the market, since market prices are influenced by all the trades in the market. The efficient markets hypothesis [Fama, 1970] states that as soon as information become available to traders, that information is immediately reflected in prices. Prediction markets are a type financial markets with the aim of forecasting the outcome of future events. The principal purpose of prediction markets is predicting future events as market prices reflect aggregated information [Berg and Rietz, 2003].

Prediction markets have recently gained importance in the area of forecasting as a new approach to aggregate dispersed information. In these markets, a group of participants trade securities (contracts) whose payoff depends on the realisation of an uncertain future event according to their expectations of the future event likelihood. Each contract price is a bet on the outcome of the future event of interest, reflecting the traders aggregated expectations on the outcome of the event, and that is why contract price can be used to predict the likelihood of the uncertain future event. When the outcome is revealed, the traders receive revenue in exchange for the contracts they hold.

For example, with the goal of predicting which candidate will win the presidential election, a prediction market offers a security for each of the candidates. If one thinks that the current estimate for a candidate is too low, a trader can buy the corresponding security of that candidate. Consequently, the participants, for the sake of gaining money, have an incentive to become active in trading, whenever they expect the market estimate is inaccurate [Graefe and Armstrong, 2011].

As participants trade the securities, they disclose their private information. Therefore, contract prices move to aggregate all relevant information and constitute a collective forecast about the likelihood of the outcome of the event. According to Conitzer [2009], the security prices can be interpreted as the consensus probability of the outcome of the event. Traders trade securities until the deadline of the prediction market, which in our example is obviously before the outcome of the election is revealed. Once the outcome of the election is known, the corresponding security of the realised event pays out some amount of money (say, 1 unit of currency), which means, participants can either win or lose money based on their individual performances.

Luckner [2008] states “prediction markets motivate participation and well-designed incentive schemes motivate traders to reveal their beliefs instead of their preferences”. In the context of the above example, even an enthusiastic supporter of a candidate in the presidential election rather not try to purchase securities of his favourite candidate since he would lose money in case his favourite candidate does not win the election.

The roots of prediction markets originate in the Hayek Hypothesis [Hayek, 1945], that in a competitive market, the prices system is a very efficient mechanism to combine information which is distributed among market participants. Probably, the most popular prediction market is the Iowa political market², which was set up in 1988 by Iowa university academics to forecast US presidential elections. Since its establishment, a wide range of events have been predicted

²<http://www.biz.uiowa.edu/iem> (Retrieved Feb 1, 2016)

using the Iowa prediction market. For example, University of Iowa researchers have established a prediction market for forecasting outbreaks of avian flu. The predictions from Iowa markets have beaten opinion polls and political pundits remarkably consistently over the years [Berg et al., 2008].

A special kind of prediction markets are combinatorial prediction markets that allow participants to invest on conditional events or boolean combinations of events. For instance, a prediction market with objective of ranking n football teams may wish to accept bets over permutations (Team A will beat Team B) and hence include up to $n!$ possible permutations of the football teams. Combinatorial outcomes increase the complexity of prediction market. In combinatorial prediction markets traders' attention is divided among a large number of outcomes and hence these markets must be designed carefully otherwise low liquidity can be a serious problem. This results in a growing interest [Pennock and Sami, 2007, Chen et al., 2008b,a, Gao et al., 2009, Othman and Sandholm, 2012, Chen et al., 2013] in proposing and analysing prediction markets with combinatorial outcomes. For example, Abernethy et al. [2011] propose a general framework for the design of securities over combinatorial outcome spaces with the objective of developing efficient automated market makers for markets with enormous outcome spaces.

Due to increasing success of prediction markets in practice, prediction markets are becoming the focus of much research. Chan [2001] studies markets and market participants using agent-based simulations in order to verify the results from the experiments using humans in double-auction markets. In particular, agents are modelled using “zero-intelligence (ZI) [Gode and Sunder, 1993] to replicate actual mechanisms that human traders use to analyse their obtained information and learn from their experiences acquired in the market. Using reinforcement learning, Chan [2001] proposes an adaptive market maker to perform effectively under different noisy market environments. Manski [2006] presents a formal analysis of price determination in prediction markets with traders who possesses heterogeneous beliefs.

Jumadinova and Dasgupta [2011] developed a multi-agent system to understand the influence of information on the performance of prediction markets and on the strategies used by market participants. Later, Jumadinova [2013] studies the dynamics of prediction markets under various conditions. She also suggests a mathematical model for a prediction market using a boolean network and approaches from statistical physics, which was used in [Jumadinova and Dasgupta, 2013] to aggregate information for object classification. Jian and Sami [2012] show that prediction markets with a pre-determined sequence of trading opportunities compared to unstructured markets with an endogenous trading sequence work more effectively and aggregate trader information more efficiently. Beygelzimer et al. [2012] study the dynamics of wealth in a prediction market in which all participants use Kelly betting [Kelly Jr, 1956] – a strategy that maximises the compound growth rate of wealth during an infinite number of interactions with the market – and shows that such a market achieves a prediction which is a wealth-weighted average of the individual participants' belief.

The weight of available evidence suggests that prediction markets can provide a “remarkable crystal ball into the future”, if they are carefully designed and sensitively implemented [Bray et al., 2008]. When public information is inaccurate, misleading, or little useful intelligence is available to be aggregated in a prediction market is unlikely to perform well [Wolfers and Zitze-

witz, 2004]. Hence, prediction markets are required be designed and implemented carefully to be able to successfully aggregate traders' information [Weinhardt et al., 2006]. According to Wolfers and Zitzewitz [2004], key elements that needs to be considered in designing a prediction market are:

- Market participants: the prediction market participants and diversity of information between them,
- Incentive: incentivising scheme of the prediction market (i.e the incentives provided to the participants to reveal their private information),
- Prediction market trading protocols: how participants trade in these markets.

Now, we discuss these elements in the next three subsections.

2.2.1 Market Participants

Who is participating in the market can affect the effectiveness of the market. Few participants results in a 'thin markets' – in which few transactions take place, prices are volatile and assets are less liquid– and hence, cannot yield accurate predictions [Williams, 2011].

Most prediction markets with the aim of predicting public events are typically open to the general public. However, Iowa Health markets³ is an exception in which only authenticated health professionals are authorised to trade. Similarly, prediction market run by some corporates choose only a group of their employees to trade who seems more expert on the subject of the prediction.

Besides the number of market participants, diversity of beliefs held by the market participants is another factor affecting the quality of a prediction market. Diversity of information is a requirement constituting a basis for trading in prediction markets [Wolfers and Zitzewitz, 2004]. Tziralis and Tatsiopoulou [2007] state disagreement among participants is desirable and accordingly the selection of traders is a key design concern. Wisdom of the crowd, which is the underlying presumption for prediction markets, depends not only on the forecasting capability of the markets participants, but also depends on their intellectual dissimilarities [Page, 2007].

Traders can effect other participants' beliefs through their trades. Some traders may seek to achieve gains by *manipulating* the market by first misleading other traders about the outcome of the event, and then exploit their inaccurate information in later trades. Awareness and reaction to this problem may result in weakening the aggregative powers of the market, as traders may become excessively cautious about making inferences from market prices [Dimitrov and Sami, 2008]. Therefore prediction markets characterisation should be designed in a way to limits traders temptation for manipulation.

2.2.2 Incentive

The traders' rewards in exchange for disclosing their information is essential in prediction markets, as market participants are assuming to be rational – they consistently make decisions in

³ <http://iehm.uiowa.edu>

order to maximise their own expected payoff. In such markets, participants should be motivated in such a way to contribute accurate information to the market, directly report all their private information, and have no incentive to misreport their private information [Conitzer, 2009]. Such a mechanism is known as *incentive compatible*. When information holders do not have enough incentive, traders will not participate in the market and will not disclose their information and, in extreme cases, traders will not trade at all which is referred to the so-called ‘no trade theorems’ [Milgrom and Stokey, 1982].

One interesting issue in designing prediction markets is deciding whether agents should be rewarded by real money or some form of play-money. Gruca et al. [2003] suggest that real money is necessary to ensure truth-revealing incentives but Rosenbloom and Notz [2006] believe real-money markets are only more accurate for non-sports events. Prediction markets with real money rewarding mechanism creates regulatory complications depending on the precise context and geography of the initiative [Bray et al., 2008]. The U.S. Commodities Futures Trading Commission has strictly regulated prediction markets, because of concerns about speculation and manipulation [Arrow et al., 2008]. Therefore, many prediction markets organiser wish to side step potential complications and establish prediction markets using play money [Bell, 2009]. For example, the Hollywood Stock Exchange⁴ runs prediction markets to forecast entertainment related outcomes and reward their participants using play money that can either be exchanged for prizes or amassed for prestige.

2.2.3 Trading Protocols

The essential aspects of any trading platform is how transactions are committed, buyers and sellers interact and demand and supply is matched. This section explains a number of popular trading protocols used for prediction markets.

Continuous Double Auction (CDA)

Early prediction markets adopted continuous double auction (CDA) protocols and this is still a preferred market structure in many professional financial markets [Williams, 2011]. In CDA markets, buyers and seller can call out the price that they are willing to sell (ask price) and the price they are willing to buy (bid price) at any anytime. If the highest bid (outstanding bid) is equal or more than the lowest ask (outstanding ask), a transaction is made at the price of the former. Since it merely matches agreeable traders, all markets can be considered as a zero-sum game [Spann and Skiera, 2003].

As traders use news to improve their decisions in order to maximise their potential profit, the market prices which indicate the market forecast are constantly updated. Therefore, CDA markets allow for continuous information incorporation into prices [Luckner, 2008]. Lack of financial risk for market operators in CDA is one of the main advantages of CDA markets and this is the reason of their popularity among real money markets [Luckner, 2008]. For instance, the Iowa Electronic Markets (IEM) began using the CDA mechanism in their prediction markets in the late 1980s [Forsythe et al., 1999].

⁴<http://www.hsx.com> (Retrieved Jan 25, 2016).

However, CDA markets suffer from insufficient liquidity and may fail in *thin markets* – where there are few number of sellers and buyers – and consequently the bids and ask prices cannot be easily matched, the bid-ask spread are large and order queues become vacant [Spann and Skiera, 2003].

Call Auction (CA)

The call auction (CA) very similar to CDA apart from the fact that in CDA transactions are executed immediately, but in CA orders are amassed for simultaneous execution at a pre-arranged time [Madhavan, 1992]. Similar to CDA, CA pose no financial risk for the market operator. However, since trading happens periodically, new information is not reflected instantly in trading prices.

Pari-Mutuel Market (PMM)

Pari-Mutuel Market (PMM) is another type of prediction market structures which was originally designed for sports and horse betting. In PMM, participants pool all of the money placed on individual contracts. The proportion of the amount bet on each contract to the entire pool of bets can be inferred as the probability of a contract winning. Once the market is closed and the real outcome is known, the entire amount wagered is divided between winners proportional to the amount they wagered [Peters et al., 2007]. An advantage of the PMM is that it does not have the liquidity problem, as participants bet into an expandable pool. On the other hand, PMM suffers from the fact that it cannot aggregate crowd information continuously. In addition, rational traders postpone their participation until all other participants submit their bet and all possible information is known. This causes the probability of the outcomes do not be precise until the last moment.

Dynamic Pari-Mutuel Markets (DPMM)

Dynamic Pari-Mutuel Markets (DPMM) is another protocol for prediction markets proposed by Pennock [2004] to combine a PMM and CDA in order to achieve PMM's infinite buy-in liquidity and CDA's ability to react continuously to new information.

In this mechanism, similar to pari-mutuel market, the market offers n securities each corresponding to a possible outcome and the participants invest in an outcome by buying its corresponding security. After the true outcome is known, the total wagered amount is divided proportionally between the traders who own the correct security. Participants who invested in the wrong security lose their invested amount. The difference between DPMM and pari-mutuel markets is that the prices of securities in DPMM are decided using a price function which is based on traders' demands.

Since agents can purchase any security at any time, buy-in liquidity is not a problem in these markets. Also, the market operator is free from financial risk. However, the problem with this market is the fact that the market operator does not accept sell offers and agents can only trade their previous investment using CDA protocol among themselves [Luckner, 2008]. An example of DPM is Yahoo! Buzz market [Mangold et al., 2005]

Market Scoring Rules (MSR)

Scoring rules are techniques to reward participants in a way to motivate them to give the most accurate prediction. Based on scoring rules, Hanson developed market scoring rules (MSR) which provide infinite liquidity and “produces an accuracy like that of information markets when many people make the same kind of estimates, and like that of scoring rules when only one person makes each estimate” [Hanson, 2003]. Market scoring rules, like standard scoring rules, require a *market maker* to reward accurate predictions. Since the automated market maker is always willing to trade, MSR provides infinite liquidity and hence it can easily deal with problems that typically arise from thin markets.

One important feature of market scoring rules is that it is myopically incentive compatible, which means traders under market scoring rules benefit the most when they report their true beliefs when participating in the market and ignore the profit that other traders may gain based on their report [Hanson, 2003].

MSR maintains a probability distribution over all potential outcomes. Hanson [2003] believes that any trader who believes the probabilities are incorrect can modify the existing prediction by updating it with a new prediction, as long as the trader is willing to pay off the the predictors responsible for the current prediction. If the traders move the market prices into the right direction and hence improve the prediction, they can expect a positive payoff. Alternatively, they will lose if they move the market prices in a worse direction. MSR provides infinite liquidity for the sell side of the market and prices in MSR reflect new information immediately. However, the market operator is not risk-free and has a bounded maximum loss.

A popular type of MSR is the Logarithmic Market Scoring Rule (LMSR) [Hanson, 2007] which is used by a number of companies including Inkling Markets, Consensus Point, Yahoo!, Microsoft, and the large-scale non-commercial Gates Hillman Prediction Market at Carnegie Mellon [Othman and Sandholm, 2010]. It also has been the focus of many research papers [Agrawal et al., 2011, Chen and Pennock, 2012, Nikolova and Sami, 2007].

In LMSR, the market maker sells a number of securities, each corresponding to a possible outcome, and returns \$1 for the security whose corresponding outcome is realised. These types of prediction markets are also called *cost function based markets* as automated market makers use cost functions to determine the security prices. For example, in LMSR, the cost and price of a security is calculated as follows:

$$C(q_i) = b * \log\left(\sum_{i=1}^m e^{q_i/b}\right), \text{ and,} \quad (2.18)$$

$$P(q_i) = \frac{\exp(q_i/b)}{\sum_{j=1}^m \exp(q_j/b)}, \text{ respectively,} \quad (2.19)$$

where m is number of securities that market offers, each for one possible outcome, $q_i \in (q_1, q_2, \dots, q_m)$ represents the quantity of security i held by market traders and b is the liquidity

rate chosen by the market maker. The bigger the value of b , the more money the market maker can lose. It also means that traders can purchase additional shares without causing price swings largely.

Note that the price of a security only applies to buying an infinitesimal number of shares and the price of the security immediately changes as soon as a traders start trading. In order to calculate the cost of trading X securities, the market makers must calculate $C(q + X) - C(q)$.

Having covered different key elements of the prediction market in this section, we now highlight the popularity of prediction markets by presenting popular applications of such markets in the next section.

2.2.4 Prediction Market Applications

Prediction markets have been widely used to predict the outcome of various events in different domains. Various online prediction markets in different domains exist. One of the most popular one is the Iowa Electronic Markets (IEM)⁵, which are real-money markets with the goal of predicting economic and political events such as presidential elections. Iowa electronic markets are also investigated as a decision support systems where decisions are made according to trading prices [Berg and Rietz, 2003].

Hollywood Stock Exchange (HSX)⁶ is another popular online prediction market which aims to forecast future box office profits for new movies. Another instance of online prediction markets are Tech Buzz Game with the goal of forecasting high-tech trends.

Sport prediction markets are very popular prediction markets with the goal of forecasting the outcome of tournaments and events of various sports such as baseball, soccer, football, hockey, basketball, tennis and horse racing. Some works such as [Chen et al., 2005, Servan-Schreiber et al., 2004] show that sports prediction markets provide as accurate predictions as expert polls.

Companies use prediction markets to make their internal forecasts since valuable decision relevant information is often widely dispersed among company members [Bray et al., 2008]. For instance, Hewlett-Packard (commonly referred to as HP⁷), which is an early pioneer in utilising prediction markets, established a number of prediction markets to predict important internal events such as quarterly printer sales. Their results reveal that the prediction markets are more precise than the companys official forecasts [Plott and Chen, 2002].

Google predicts a range of strategic importance events such as product launch dates by running internal prediction markets within their company. Starwoods⁸ deploys prediction markets with the aim of developing and selecting marketing campaigns. Many pharmaceutical companies such as Pfizer, GSK, Eli Lilly and Novartis use prediction markets to predict many company related events such as production of new product forecasts and competitor intelligence. Further fields of application within companies are new product development [Dahan et al., 2010] and the identification of lead users for consumer products [Spann et al., 2009].

Another application of prediction market within companies is the generation of ideas [Soukhoroukova

⁵ <http://www.biz.uiowa.edu/iem/> (Retrieved Feb 5, 2016).

⁶ <http://www.hsx.com> (Retrieved Feb 5, 2016)

⁷ <http://www.hp.com/> (Retrieved Feb 5, 2016)

⁸ <http://www.starwoodhotels.com/> (Retrieved Feb 5, 2016)

et al., 2012]. Idea markets, which are sometimes described as “preference markets”, determine which idea(s) are likely to be the most advantageous for the company where participants of the markets are employees who trade stock in particular ideas. Some works [Spears et al., 2009] reveal that the ideas’ creators trade aggressively to increase the price of their idea and drop the price of other’s ideas.

Prediction markets are also utilised in the healthcare sector [Graefe et al., 2010, Rajakovich and Vladimirov, 2009]. Polgreen et al. [2007] use a real prediction market, with health care expert participants, to forecast infectious disease activity 2-4 weeks earlier than being published by formal reports.

Abernethy and Frongillo [2011] propose the framework of a prediction market as a tool to aggregate predictors or classifiers to solve a given learning problem, where the participants of the markets are crowd of human machine learning researchers and domain experts. The objective of the works is decentralising the process of solving a task by building a consensus hypothesis which reflects all the knowledge of the experts, while the individual gains a benefit for applying their expertise on the learning problem.

So far this chapter, we reviewed machine learning and prediction market research lines. Next, we discuss how various research emerged in the last decade by connecting these two research fields.

2.3 Connections Between Prediction Markets and Machine Learning

Some works applied machine learning – more specifically online learning – concepts to explain the reasons behind the success of prediction markets. These works are discussed below.

2.3.1 Connecting Prediction Markets and Online Learning

This research justifies the success of prediction markets in practice by linking prediction markets to prediction with expert advice models (PEA), i.e. a field of online learning. These works link the market outcomes and transactions happening in a prediction market to experts and training instances in PEA, respectively. Now, we discuss some of these works.

Chen et al. [2008a] were the first to establish a formal connection between prediction markets – in particular LMSR market makers [Hanson, 2003] – and PEA. They show that standard randomised weighted majority regret bound [Littlestone and Warmuth, 1994] can be used to deduce the worst-case loss bound of a marker maker. They further discuss that extension of the weighted majority algorithm to permutation learning [Helmbold and Warmuth, 2009] can be used to design an algorithm which approximates market prices in LMSR-based markets with combinatorial outcomes.

Chen and Vaughan [2010] explain the mathematical connections between prediction markets and PEA. Their main goal is to explain why prediction markets achieve accurate estimates in practice. They show that the market maker can be seen as a model which learns a probability distribution over outcomes, similar to PEA models which learn a distribution over experts by

observing losses. In particular, they show: i) cost function based prediction markets with bounded loss can be understood as a PEA algorithm, ii) the weighted majority regret bound can be deduced from the bound on the worst-case loss of a market maker based on LMSR, iii) the class of convex cost function based markets parallels to the class of follow the regularised leader learning algorithms – an algorithm within the framework of PEA, and iv) market scoring rules are equivalent to convex cost function based markets and hence market scoring rules can be inferred as follow the regularised leader algorithms.

Frongillo et al. [2012] show the market process as a stochastic mirror descent when traders use Kelly strategy [Kelly Jr, 1956]. They demonstrate that market prices can be inferred as a summary of the market’s belief distribution.

Later, Abernethy et al. [2013] show a relationship between market design and machine learning. In particular, they demonstrate that cost function based prediction markets and PEA are arithmetically parallel. In their model, the securities correspond to the experts in PEA and each trading happens in the market corresponds to a training example in PEA – similar to all other works mentioned in this subsection.

While these works link prediction market to machine learning to explain the reasons behind success of prediction markets, some researchers have borrowed the concept of prediction markets to advocate a new machine learning technique. We focus on the latter line of research in the next subsection.

2.3.2 Artificial Prediction Markets

To our knowledge, there is relatively little research on proposing new machine learning techniques using prediction market concept – which we refer to them as *artificial prediction markets* in this dissertation. The existing works mainly concentrate on providing useful mechanisms to integrate the result various models and act as a machine learning ensemble method. Market participants, in these works, are trained machine learning models.

These works are initiated by Perols et al. [2009] who focus on parimutuel betting mechanisms for combining classifiers. This model only handles binary outcomes. After that, Lay and Barbu [2010] propose their own artificial prediction market as a new machine learning ensemble for multi-class formulation. In their model, the market simulates the Iowa electronic market in that the winning contract pays 1 unit currency if realised. The market offers contracts for each possible outcomes. The market includes participants which are trained classifiers. The participants have a betting function which says what percentage of its budget this participant will allocate to for each contract. The contract price does not fluctuate, but is calculated in a way that the total amount won by the winning contracts is equal to the total amount received from selling contracts to the participants, independent of the outcome. Table 2.1 demonstrates the mechanism of their model.

They conduct an evaluation for a number of data sets from the UCI machine learning repository⁹. For each data set, 50 random trees are trained on bootstrap samples of the training data. These trained random trees are used as market participants to construct a random forest.

⁹<https://archive.ics.uci.edu/ml/datasets.html>

1. Train classifiers in offline mode based on the training examples.
2. Assign the same budget to each market participant (i.e trained classifiers).
3. Assign a betting function to each participant. For each class, the betting function states what proportion of the budget must be allocated for each price.
4. For each training example:
 - (a) Participants give the probability of each class being the correct answer.
 - (b) Calculate the price of each contract by solving:
 - The amount which will be invested by all participants in this contract, for a certain price, must be equal to the amount won by winners.
 - The sum of all contracts' prices must be equal to 1.

In binary class problems with linear betting function and cases which participants use constant betting functions, the equation can be solved analytically. In all other cases it must be solved numerically. The authors offered double bisection method and Mann Iteration algorithm [Mann, 1953] for this purpose. However, they are not guaranteed to find a unique price.
 - (c) Calculate how much each participant has invested for each class. So far, the market prices had not been decided and only participant betting functions were known. However, after knowing market prices, the whole amount invested by the participant for each class can be calculated using their betting function.
 - (d) Update budget of all participants.
5. After the training is performed, the obtained budget of participants can be used as weights of classifiers to integrate their results in offline mode.

Table 2.1: Lay and Barbu [2010] model.

The authors claim that their model is often significantly more accurate than a standard random forest implementation. They also suggest when each market participant purchases the contract for the class it predicts, regardless of the contract prices, their model generalises linear aggregation of classifiers. They further demonstrate that their model can be equivalent to a logistic regression or a kernel-based classifier using special betting functions. Their results show that this model significantly outperforms the implicit online learning [Kulis and Bartlett, 2010] on some of UCI data sets and is never outperformed by it.

Lay and Barbu [2012] extend this research line to perform regression in order to predict a continuous variable. In this model, participants are rewarded using a reward kernel which calculate the distance between the participants prediction and the correct answer.

Analogously, Storkey [2011] proposes an artificial prediction market with no global market maker. In this work, participants are independent agents defined by their respective utility functions. Each agent has an associated utility function indicating the utility of a wealth it gains in the market. Agents also have a belief which is a probability measure for each possible outcome of the event. The model is based on equilibrium markets, where market equilibrium is defined by a price and an allocation, such that none of the participants has any incentive to trade and there is no surplus demand of any goods. Each market participant purchases securities for the possible outcomes to maximise its own utility function. Then, using an optimisation procedure, the equilibrium prices of the securities are computed. Storkey demonstrates that the market can find a probability distribution over the future events by aggregating agents' beliefs. He also show that different model combination methods such as weighted median of experts, weighted mixture of experts, and product of expert models are obtained via different utility functions for the individual agents. He mentions that the focus of this work is the investigation of model combination methods for agents that have previously learnt beliefs and investigating agent learning in this context is left as matter of future work.

Storkey et al. [2012] extends [Storkey, 2011] by introducing agents with their proposed isoelastic utilities. They show that Bayesian model averaging and mixture model learning can be implemented using this market mechanisms. The paper demonstrates that various utility functions have a substantial consequence for market results. In addition, they show that isoelastic utility functions are usually better than the logarithmic utility functions proposed in [Storkey, 2011]. They further demonstrate that inhomogeneous markets of agents with isoelastic utilities outperform classifiers such as random forests, neural networks, decision tree on numerous machine learning benchmarks.

They mention two future directions of i) investigating the adaptability of markets in the context of data set shift or non-stationary environments, and ii) enabling agents to learn about their own performance in the market to assess on which situations a positive return is more probable.

Hu and Storkey [2014] present a model for prediction markets and analyse the market by modelling agents using risk measures rather than utility functions. Risk measures are a common theory in finance literature and denote the potential loss of choosing a particular asset. In this work, the prices of securities are decided using a global market maker and agents decide their optimal purchase such that minimise their risk measure. The market is run in number of rounds

where, in each round, only one agent trades with the market maker. This agent traded based on the prices at that round, which reflect the opinions of the agents who traded in previous rounds. As such, only the last agent can infer wisdom of the entire crowd.

2.4 Summary

In this chapter, we discussed various machine learning and prediction market research, followed by covering the works which link these two research fields. We reviewed artificial prediction markets (APMs) which are similar to real ones, except for replacing human traders with computer programs. These models turned out to be an effective approach for aggregating the predictions of classifiers [Barbu and Lay, 2012, Storkey, 2011]. Despite their many advantages, current artificial prediction markets have the following limitations:

- The current APMs are proposed and studied only for offline problems, i.e. the participants only learn before entering the market, and never revise their hypothesis once prediction process starts.
- Their participants are assumed to use fixed strategies, such as constant betting functions [Barbu and Lay, 2012], utility functions [Storkey, 2011, Storkey et al., 2012] or static risk measures [Hu and Storkey, 2014], unlike human traders in real prediction markets.
- In real prediction markets, aggregated information provides signals to each trader about the private information of other traders so the trader can adjust its prediction accordingly. However, this mechanism is not explicitly modelled in current artificial prediction markets.

Against this background, in this thesis, we propose the *Artificial Continuous Prediction Market (ACPM)*: a multiagent system in which a market is created once a new example is received. ACPM has the following advantages:

- ACPM is designed to tackle online problems, i.e. the agents update their hypothesis once the true outcome of each example is revealed. This results in adapting the agents, and consequently ACPM, to dynamic environments in which the quality of data source may change over time.
- Each agent has a private trading strategy based on reinforcement learning to dynamically identify the actions which maximise the agent's reward.
- Each agent is explicitly armed with the ability to revise its predictions and bet in response to those of other agents, thus taking advantage of *wisdom of the crowd*.

In addition to the aforementioned points, most of the current APMs are designed for discrete classification, unlike ACPM which is designed to perform regression (i.e predict a continuous variable). The only artificial prediction market in the literature which performs regression is due to Lay and Barbu [2012]; in this work, the authors designed their own specific prediction

Paper	Online Problem	Regression	Adaptive Trading Strategy	Agents Utilise Wisdom of the Crowd
[Perols et al., 2009]	✗	✗	✗	✗
[Lay and Barbu, 2010]	✗	✗	✗	✗
[Lay and Barbu, 2012]	✗	✓	✗	✗
[Storkey, 2011]	✗	✗	✗	✗
[Storkey, 2011]	✗	✗	✗	✗
[Storkey et al., 2012]	✗	✗	✗	✗
[Hu and Storkey, 2014]	✗	✗	✗	✓
ACPM	✓	✓	✓	✓

Table 2.2: Existing Artificial Prediction Markets (APM). The first and second columns show the APMs which are designed for online setting problems and regression task, respectively. Third and fourth columns show the APMs in which agents utilise an adaptive trading strategy and the wisdom of the crowd, respectively.

market mechanism in which the participants have constant betting functions, that define what percentage of their budget should be dedicated for each security price. The price of securities are decided using a price equation based on market equilibria. Therefore, the participants cannot decide how much they will invest and have no scope for autonomy.

Table 2.2 compares ACPM with the existing artificial prediction markets with regards to four criteria.

1. Suitability for online setting problems

As the table shows none of the existing APMs is explicitly designed for online setting problems.

2. Suitability for regression tasks

The table shows that the only existing APM suggested for regression is [Lay and Barbu, 2012], whereby the algorithms are required to report a conditional density on the possible responses of the target variable. However, ACPM relaxes this requirement.

3. Facilitating the participating agents with an adaptive trading strategy

As demonstrated by the table, none of the existing APMs provide an adaptive trading strategy for their agents.

4. Enabling agents to use the wisdom of the crowd to revise their beliefs in response to other traders' trades

As the table shows the only existing APM in which agents use wisdom of the crowd is [Hu and Storkey, 2014]. In their model, in each round of the market, only one agent trades with the market maker. This agent observes the prices at that round, which reflect the opinions of the agents who traded before him. As such, only the last agent can infer the

wisdom of the entire crowd. In contrast, all the agents in ACPM trade at each round of the market and update their prediction and bets based on the wisdom of the entire crowd.

Chapter 3

Artificial Continuous Prediction Market

Prediction markets are increasingly being considered by governments and corporations as an approach for collecting, summarising and aggregating dispersed information [Hanson et al., 2006]. They have been used to forecast accurately the outcome of many events such as political contests, sporting events, and on some occasions, even economic outcomes [Nikolova and Sami, 2007]. In these markets, participants buy and sell instruments, called securities, whose payoffs are tied to the outcome of the specified future event. Some types of prediction markets are run by a market maker who deals with traders to buy and sell the securities.

Real world prediction markets utilise the aggregated “wisdom of the crowd” to predict the outcome of a future event [Ray, 2006]. In these markets, wisdom of the crowds comes from aggregated information such as security prices. They provide signals to traders about the private information of other traders so they can adjust their beliefs accordingly. Ideally, this process should result in a situation in which all traders reach a consensus belief that reflects all available information. Adjusting participants beliefs in response to other participants’ trades has an important role in the successful aggregation of information in prediction markets [Dimitrov and Sami, 2008].

In this chapter, we introduce the Artificial Continuous Prediction Market (ACPM), a multi-agent system which adapts the concept of real prediction markets with the objective of predicting the outcome of an event by integrating a range of data sources and aggregating the results of different analysis models. In particular:

- ACPM is *online*, i.e., a prediction market is instantiated every time a prediction needs to be made (i.e for every record in the data set) and agents update their hypotheses once the true outcome of each event is revealed.
- Once the true outcome is revealed, agents update their hypothesis at the end of each market to prepare for the next market.
- ACPM is designed to predict a real value in a continuous domain;

- The agents have a trading strategy which learn from their past experiences in order to improve their performance in future, similar to human traders.
- Each agent is explicitly provided with the ability to use wisdom of the crowds—one of the key factors in the effectiveness of real-life prediction markets. In particular, each agent revises its predictions in response to those of other agents. This results in implicit message passing happens between agents before the final prediction of the system is determined.
- ACPM uses reinforcement learning and performs as a supervised learning algorithm.
 - ACPM performs as a supervised learning algorithm. ACPM instantiates a prediction market instance for each record of a given data set, where each record includes the input vectors and its corresponding target value (true outcome). Agents use the input data to predict the true outcome. Once the prediction of ACPM is determined, the true outcome is revealed to the agents and then agents re-train their hypotheses.
 - ACPM agents use reinforcement learning (i.e Q-learning) to adapts their trading strategy by learning from their experience of interacting and receiving feedback from the market. In particular, agents utilise reinforcement learning to learn which strategy is the most rewarding based on the payoffs they receive from the market.

In this chapter, we explain the technical aspects of ACPM. Firstly, we provide a general overview and scheme of ACPM in Section 3.1 and 3.2. After that, in Sections 3.2.1 to 3.2.3, we provide a detailed description of ACPM components, namely: (i) reward function, (ii) aggregation function, and (iii) rate per transaction parameters. Since each ACPM agent is facilitated by a trading strategy which guides its participation in markets, we propose two different trading strategies for our agents in Sections 3.3. We demonstrate the performance of these proposed trading strategies using two examples in Section 3.4. ACPM’s parameter settings are covered in Section 4.3, and finally, Section 3.5 assesses ACPM capabilities.

3.1 ACPM Overview

ACPM creates a prediction market instance for each prediction that needs to be made. All the data needed for this, including the correct prediction, is referred to as record in accordance with the machine learning literature. Each market includes a number of rounds. The participants, which we refer to as agents, predict the value of the record using data from their assigned source and their analysis model and accordingly participate in the market. Subsequently, the market maker calculates the market prediction by combining all the individual predictions and announces it to all agents. While market duration (number of rounds) are not over, agents participate in the market. Once the true value of the record is known, the market maker computes the reward for each agent and informs the agents about the correct outcome, hence they can update their analysis model and their trading strategy with the aim of improving their performance in future markets. Inspired by real prediction markets, ACPM includes the following components:

- 1) **Agents (market participants):** which are artificial agents¹. Agents are endowed with (i) a data source, (ii) an analysis model, (iii) a budget and, (iv) a trading strategy. Each agent receives data from its designated source and analyses this data with its designated analysis model to make the prediction. An agent's *analysis model* can be a simple algorithm or a complex machine-learning model such as neural network and random forest. The purpose of the analysis model is to receive data as input and return the prediction as output. Then, based on its trading strategy, each agent places a bet on its predicted outcome; this bet reflects the agent's beliefs about the quality of its prediction, and must never exceed the agent's allocated budget. An agent's **trading strategy** guides the agent in how to participate in the market by identifying those situations that generate greater returns. ACPM can aggregate multiple data sources by assigning them to different agents and function as an ensemble by assigning a different analysis models to each agent.
- 2) **Market Maker:** runs the market, deals with agent transactions and establishes the market prediction.
- 3) **Market Rounds:** real prediction markets usually take over a number of days before the market is closed and participants can participate for each day of the market. In the context of an artificial prediction market, where the participants are software agents and sidereal time is effectively meaningless, we use the time-neutral term 'round' to correspond to the day of real prediction markets. Hence, each ACPM prediction market comprises a number of rounds, during which each agent sends a **bid** to the market maker. Each bid comprises: (i) a prediction value, (ii) the amount the agent is betting on its prediction. At the end of each round, agents can use wisdom of the crowd to update their beliefs to prepare for the next round transaction. In other words, they update their beliefs in response to other traders' trades.
- 4) **Aggregation Function:** the market maker aggregates agents' predictions using an aggregation function at the end of each round of a market (see Section 3.2.2, page 56).
- 5) **Reward Function:** Once the market is over and the correct answer is known, the market maker use the reward function to reward agents based on the bids they sent in each round of that market (see Section 3.2.1, page 53).
- 6) **Rate Per Transaction Parameters:** constrains the amount an agent can invest in each transaction. These parameters are *Maximum Rate Per Transaction* (MaxRPT) which specifies the maximum percentage of the budget that each participant can invest and *Minimum Rate Per Transaction* (MinRPT) which is the minimum percentage of the budget that each participant can invest (see Section 3.2.3, page 56).

Algorithm 3 ACPM Model

```
1: An initial  $budget_i$  is given to each agent  $a_i$ ;  
2: for every record  $x$  in the data set do  
3:   The market maker instantiates a prediction market instance for  $x$ ;  
4:   Each agent observes (some) features of  $x$ ;  
5:   for each round do  
6:     Each agent  $a_i$  submits  $\langle prediction_i, bet_i \rangle$ ;  
7:     Each agent  $a_i$  updates  $budget_i$ ;  
8:     The market maker announces the market prediction;  
9:     Each agent  $a_i$  updates  $\langle prediction_i, bet_i \rangle$ ;  
10:  end for  
11:  ACPM prediction  $\leftarrow$  market prediction of the final round;  
12:  The market maker reveals the true outcome;  
13:  Each agent receives its payoff and updates its budget;  
14:  Each agent updates its analysis model & trading strategy  
15: end for
```

3.2 ACPM Mechanism

We now explain how ACPM operates. ACPM adapts the mechanism of the *pari-mutuel* prediction market. Original pari-mutuel prediction market works as follows: first, participants pool the money placed on each possible outcome [Peters et al., 2007]. Then, the probability of each outcome is taken as the total bet on that outcome, divided by the total bet on all outcomes. Finally, once the market is closed and the real outcome is revealed, the entire wagered amount is divided between the winners proportional to the amount they each wagered.

While this mechanism was originally proposed for prediction given a predetermined discrete set of outcomes, we use it for predicting a continuous variable, without placing any assumptions on the range of possible values that it may have. Importantly, to incorporate wisdom of the crowd, we generalise the pari-mutuel mechanism from one round to multiple rounds, to allow the agents to update their prediction and investment using crowd-sourced information. More specifically, in each round, the agents send their predictions and bets to the market maker, which then combines the corresponding predictions using an *aggregation function* and computes the payouts using a *reward function*. These two functions will be discussed later, but first we give an overview of how ACPM works.

The main steps of ACPM architecture are outlined in Algorithm 3. In more detail, the

¹We use the terms *participant* and *agent* interchangeably.

agents are first given equal initial budgets (line 1). After that, for each record x in the data set, the market maker instantiates a prediction market instance (lines 2 and 3). Each agent then observes (some²) features from x (line 4). The market then proceeds in a number rounds (line 5), each consisting of the following steps: first, each agent makes a prediction based on the data it receives and the knowledge it has accumulated thus far, then the agent determines its bet based on its trading strategy and its current budget (line 6). The next step involves the agent subtracting its bet from its budget (line 7). After that, the market maker aggregates all predictions, thus producing the *market prediction* for that round (line 8), which is announced to all the agents. The round ends with the agents preparing themselves for the next round, by updating their prediction and bet based on their trading strategy as well as the market prediction announced earlier (line 9); this is the part of ACPM where each agent may capitalise on wisdom of the crowd. The market prediction of the final round is reported as ACPM prediction for the current example, x (line 11). Once the true outcome is revealed (line 12), each agent receives its payoff and updates its budget by adding to it any revenue it may have earned (line 13). Here, to motivate the agents to participate in every round—and not just the last round—the agents receive a revenue for each bet they place at each round. Finally, to prepare for any potential future markets, each agent updates its trading strategy based on the received payoff, and re-trains its analysis model taking into account the newly-revealed outcome (line 14).

The process set out above is repeated for each record, which means the agents predict the outcome of a record, agents' budgets, analysis models and trading strategies are updated and then the system moves to the next record. Figure 3-1 illustrates ACPM architecture.

After a few market instances³, the differences between agent budgets becomes apparent as some agents with superior data, analysis models and trade strategies are likely to accumulate greater budget, while other agents lose a proportion of their budget as a result of their poor performance. The larger the budget of an agent, the larger the investment it can make and the larger its influence on the market prediction. In other words, high performing agents are identified and their influence in predicting the outcome of the events is increased. This increased influence of the more successful agents is expected to increase the performance of the system overall. The agents are not in competition *per se*, so we do not care which agents are better, but we do want the better ones to have more impact on the prediction mechanism.

Following this overview of ACPM architecture, we now provide more details on some of its components, namely (i) the *reward function*; (ii) the *aggregation function*; (iii) the *rate per transaction parameters* (which constrain the size of agent investment); and (iv) the *agent trading strategies*.

3.2.1 Reward Function

As mentioned in Section 3.2, once the true outcome is revealed, each agent receives its revenue, which is determined by the reward function.

A reward function can impact the influence of agents in the subsequent markets by the

²We say “(some)” because the data source of each agent may be different from that of another.

³Recall ACPM creates a market instance for each record in the data set

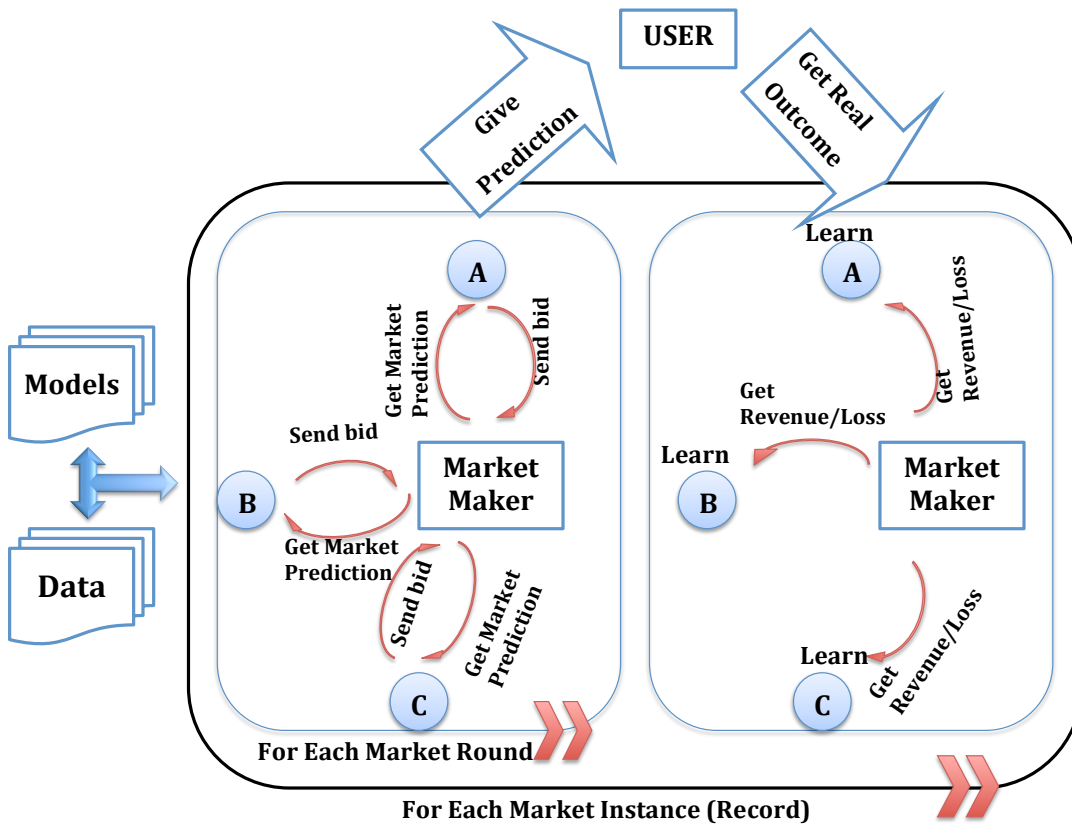


Figure 3-1: ACPM Architecture with three agents of A, B and C. For each event to predict, a market instance is instantiated, where each market instance comprises a number of rounds. In each round, agents send their predictions and bets to the market maker. At the end of each round, the market maker calculates and announces the aggregated prediction of all agents. Agents can use this information to update their prediction and bets for the subsequent rounds. Once the true outcome is revealed, the agents are notified of the true outcome and receive their payoff. Finally, each agent updates its trading strategy based on the received payoffs, and re-trains its analysis model taking into account the newly-revealed outcome, to prepare for any potential future markets.

amount of revenue it returns to them. If the reward function returns nothing to an agent for the bid it submitted to the market maker, then the agent loses a portion of its budget for the investment it made, therefore its budget will be decreased. On the other hand, if an agent receives a large reward in a market, then the budget is increased. The more budget an agent has, the more investment it can make on its predictions. Moreover, the market maker calculates the market prediction by aggregating prediction of all agents, each weighted by its corresponding bet (see Section 3.2.2, page 56). Therefore, the budget of an agent can reflect the influence of the agent on the aggregated prediction of the entire market (i.e the market prediction).

A legitimate reward function ensures that the agents' revenues depend both on their prediction accuracy and their investment. In this way agents are motivated to make accurate predictions and invest according to their beliefs about the quality of their prediction. In this work, for each bet of agent a_i , the revenue is computed as follows⁴:

$$\text{revenue}_i = \text{score}_i \times \text{bet}_i, \quad (3.1)$$

where

$$\text{score}_i = \max \{ \ln(\text{accuracy}_i), 0 \}; \quad (3.2)$$

$$\text{accuracy}_i = \max \left\{ 100 \left(1 - \frac{|\text{outcome} - \text{prediction}_i|}{\text{error outlier threshold}} \right), \varepsilon \right\} \quad (3.3)$$

where ε is epsilon.

Before explaining the rationale behind assigning a score to each agent (as per Equation 3.2), let us first explain how the accuracy of a_i 's prediction is calculated (i.e., Equation 3.3). According to this equation, accuracy_i is a real number in $(0, 100]$, which decreases linearly as the prediction error of a_i increases. In particular, whenever the error is equal to 0, we have: $\text{accuracy}_i = 100$. In contrast, whenever the error is large enough that it is identified as an outlier according to the *interquartile-range* measure [Upton and Cook, 1996], we have: $\text{accuracy}_i = \varepsilon$.⁵ Importantly, by ensuring that the accuracy is always greater than 0, we can use it in a logarithmic function (Equation (3.2)).

Having explained how the accuracy is computed, let us now explain how it is mapped to a score (see Equation 3.2). Here, we adapt the *logarithmic scoring rule* [Good, 1952]: an incentive-compatible scoring function used widely in the literature, see, e.g., [Hanson, 2003, 2007]. This way, any accuracy greater than ε receives a score greater than 1 (as $\ln(\varepsilon) = 1$), resulting in a

⁴ We also proposed an alternative reward function which is presented in Appendix A. Based on the experiments covered in this thesis, we note both produced qualitatively-similar results

⁵ We compute accuracy in this particular way to ensure that it falls within a bounded interval; this allows us to pass it to a logarithmic function when computing the score later on. However, in principle, any other alternative for computing the accuracy may also work just as well, provided that it also ensures that the accuracy falls within a bounded interval.

positive reward, i.e., a revenue greater than the investment (see Equation 3.1). Conversely, any accuracy smaller than e receives a score smaller than 1, resulting in a negative reward (i.e. the revenue is less than the investment).

By rewarding the participants according to their performance in a given market and updating their budgets accordingly, ACPM can keep track of the overall performance of each of the participants according to their accumulated budgets. Furthermore, since better performing participants gain more revenue and accumulate bigger budgets, they can invest more in subsequent markets and hence have more influence on those markets' predictions (see Formula 3.4).

ACPM is not a zero-sum game as the market maker rewards the agents based on their own prediction accuracy and not in proportion to other agents prediction accuracy. This characteristic secures ACPM from non-myopic agents temptations to manipulate the market. In other words, ACPM is myopically incentive compatible, which means the traders benefit the most when they participate in the market truthfully and ignore the profit that other traders may gain based on their trade.

3.2.2 Aggregation Function

The market maker uses the aggregation function to aggregate n bets—one for each agent—resulting in the *market prediction*. The aggregation function, we use in this work, works by assigning more weight to predictions that are backed by higher investments. Formally, the market prediction, denoted by *Prediction*, is defined as:

$$\text{Prediction} = \frac{\sum_{i=1}^n \text{prediction}_i \times \text{bet}_i}{\sum_{i=1}^n \text{bet}_i} \quad (3.4)$$

A corollary of the aggregation function is that participants that consistently make more accurate predictions accrue more budget over time and are able consequently to make bigger investments. Recall that the reward function (i.e Equation (3.1)) allows agents with accurate predictions to accrue more revenue. Since this revenue is added to the agent budget (line 13 of Algorithm 3), agents with relatively high performance accumulate greater budgets, which in turn allows them to make greater investments. Hence, since the aggregation function gives more weight to predictions backed by larger investments, the participants with a history of good prediction are able to feed this quality indirectly into the market prediction.

In sum, the designed reward and aggregation functions of ACPM are a set of connected mechanisms which work with one another to allow high quality agents be able to feed their proficiency into the market prediction.

3.2.3 Rate Per Transaction Parameters

In ACPMs, every bet is constrained by two parameters, namely the *Minimum Rate Per Transaction* (MinRPT) and the *Maximum Rate Per Transaction* (MaxRPT), which specify the minimum, and maximum percentage of the agent's budget that is allowed to be placed in a single

bet. Next, we explain the reason behind having each of these parameters.

Without a minimum rate per transaction, the agents may find themselves in a situation where none of them has any incentive to invest, meaning that they each place a bet of 0. In such a case, ACPM would simply not return any outcome.

Without a maximum rate per transaction, some agents may go bankrupt. This is undesirable since it could mean the loss of those agents' data sources, and the loss of any insight that those agents' analysis models provide. Note that the bankruptcy of an agent, a_i , does not necessarily imply that a_i brings no value to the collective performance, since a_i 's prediction can always improve in subsequent markets. Thus, it is desirable to ensure that no matter how small a_i 's budget becomes, it never reaches 0. This way, the door remains open for a_i to regain its competitiveness in the market whenever the opportunity arises. To do so, it suffices to simply set MaxRPT to a value less than 100%.

While we do not want the agents to go bankrupt, their budget needs to be changed as soon as their quality change. To achieve this objective, agents are required to invest majority of their budget in each market. For example, imagine that the quality of a high performing agent suddenly drops. If the agent invests only 10% of its budget in each market, then the agent loses most of its budget only after several number of markets, which infers that, during these markets, the agent influence on the market is not changed significantly. However, if the agent invests, for example, 90% of its budget in each market, then once the agent makes a poor prediction, it loses 90% of its budget in one market, remaining with only 10% for the next market. In such a case, the agent loses its influence on the subsequent market as soon as its quality drops. Analogously, when the agent regains its quality back, the agent budget will be increased more rapidly if the agent invest more percentage of its budget in a market – recall Formula 3.1, which determines the agent rewards both on their investment as well as their prediction accuracy.

In addition to the above, the MinRPT and MaxRPT parameters serve another important purpose: they control how much the agent original prediction quality influences its budget. Recall each market includes a number of rounds. In the first rounds, agents have no information about the other agents' predictions, hence their prediction reflects their original performance. In the subsequent rounds, on the other hand, agents are aware of the crowd prediction. Therefore, low quality agents, who know their prediction is usually worse than the crowd prediction, may report market prediction instead of their original prediction with the aim of obtaining higher rewards. This behaviour is desirable as it improves the system performance by minimising the amount of noise that low quality agents may send to the market maker. However, this behaviour gets problematic once it comes to rewarding process, as we explain now.

We do not want low quality agents to accumulate budgets as large as high quality agents. The budget of an agent affects the influence of the agent on market prediction, as the more budget the agent has the more investment it can put on its prediction and impact the market prediction (see Formula 3.4). As a result, agent revenue needs to be mostly determined based on their original predictions (i.e first round prediction) which reflects their original quality rather than the predictions altered after knowing the crowd prediction. On the other hand, the agents' predictions on the last round are improved using wisdom of the crowd and hence the final system prediction should be based on these improved predictions rather than the agents' first round

predictions.

A solution to this paradox is that agents invest the majority of their budget in the first round which includes their original prediction but still participate in other market rounds to send their revised bids to the market maker. In this way, the majority of the revenue or the loss agents receive in each market corresponds to their first round bids and hence agent budgets are mostly changed based on these bids which reflect their original quality. However, the agents still revise predictions according to crowd prediction and send it to the market maker. Therefore, the market maker can calculate the system prediction based on the revised predictions (i.e final round bids). With this in mind, first round MaxRPT should be much higher than the MaxRPT of other rounds. For example, for a market with two rounds, MaxRPT can be 90% for the first round and 1% for the second round. Recall the market maker integrates the prediction of agents weighted by their investments (see Section 3.2.2, page 56) and since all agents must follow MaxRPT constraints, increasing or decreasing this parameter has no effect on the market prediction.

3.2.4 Game Theory properties

In this subsection, we discuss ACPM with regard to some of the game theory properties, as follows.

1. ACPM provides a platform for agents to participate autonomously in the market, although it forces agents to follow the rate per transaction constrains. Given that the agent are autonomous, they can decide how much to invest, however, the MinRPT specifies the minimum amount that an agent must invest in each market round. Accordingly, ACPM is not individually rational if the $\text{MinRPT} > 1$, since an agent which is sure that it will lose in a market round, is still required to invest MinRPT percentage of its budget in that market round.
2. ACPM is allocatively efficient as agents are rewarded based on their prediction accuracy and their bet. Hence agents with more accurate predictions and larger investments receive more revenue than agents with less accurate and smaller investments.
3. ACPM is not budget balanced as the total receipt of the market is not equal to the total payment. In other words, system is not a closed one as the market maker provides infinity budget and does set a maximum bound on its loss.
4. Since the market maker rewards the agents based on their own prediction accuracy and not in proportion to other agents' prediction accuracy (i.e ACPM is not a zero-sum game), ACPM is secured from non-myopic agents' temptations to manipulate the market. In other words, ACPM is incentive compatible, which means the traders benefit the most when they participate in the market truthfully.

3.3 Trading Strategy

In ACPM, agents participate in the market based on their trading strategy. In this section, we examine two strategies: one that remains the same (i.e constant trading strategy) and one that changes using the Q-learning technique (i.e Q-Learning trading strategy). While the former ignores the advantage of wisdom of the crowd, the latter uses the market prediction at the end of each round to update the bids in the subsequent rounds.

3.3.1 Constant Trading Strategy

In the constant trading strategy, agents simply dedicate a fixed ratio of their budget to bid in each round. In this work, we set this percentage equal to MaxRPT. This naïve strategy ignores the advantage of wisdom of the crowd, i.e updating the prediction and bet in response to what the crowd believe.

3.3.2 Q-Learning Trading Strategy

Instead of having naïve agents who simply invest a portion of their budget on their prediction while considering neither their own performance nor that of the market, ACPM agents can constantly observe the environment and participate in the market according to their observations using Q-learning trading strategy.

In particular, agents can learn from their experience of interacting and receiving feedback from the environment. This type of learning falls within the area of reinforcement learning. ACPM agents can utilise reinforcement learning to learn which strategy is the most rewarding based on the feedback they receive from the market. The success of using reinforcement learning as a trading strategy in other form of financial markets is highlighted by a number of works (see Section 2.1.4.6, page 33).

In a prediction market, agents do not have complete information about the transition between states – how their actions change their environmental situations and move them into new states – and hence a model free type of reinforcement learning such as Q-learning technique is appropriate. Section 2.1.4.5 (page 31) explains Q-learning. In Q-learning algorithm [Watkins, 1989], an agent learns an *action-value function* to estimate the utility of taking an action in a particular state, called *Q-value*. The agent objective is to maximise its total reward. Therefore, the agent chooses action a in state s which has the highest Q-value.

First, we briefly explain our proposed Q-learning trading strategy and then we provide more details on different components of this strategy. An overview of this strategy can be found in Algorithm 4. Specifically, in line 1, the strategy checks whether the current market happens to be the first one (recall that a market is created for every instance from the data set) or if it is the first round of a market. If not, then the following happens: first, the agent estimates its error, as shown in line 2, where *Prediction* denotes the market prediction. As can be seen, the error estimation is done by comparing one’s prediction against wisdom of the crowd (i.e., the market prediction) rather than comparing it against the true outcome (simply because the true outcome is unknown to the agent at this stage). After that, the agent identifies its own state

based on its estimated error and the current round number (line 3). The agent then considers two actions: (i) *Change*, which involves changing the agent’s prediction, and (ii) *Preserve*, which involves sticking to the current prediction; the one with the higher Q-value is adopted by the agent (line 4). If *Change* is adopted, the agent modifies its prediction to minimise its estimated error according to a parameter, δ , that reflects the agent’s level of confidence in wisdom of the crowd (lines 5 and 6). Later on in this section, we show how the agent adjusts this level of confidence over time. On the other hand, if *Preserve* is adopted, then the agent’s prediction remains unchanged (lines 7 and 8). In the next step, the agent estimates its score using the following equations (line 10)

$$\text{score}'_i = \max \{ \ln(\text{accuracy}'_i), 0 \}, \quad (3.5)$$

where

$$\text{accuracy}'_i = \max \left\{ 100 \left(1 - \frac{|\text{Prediction} - \text{prediction}_i|}{\text{error outlier threshold}} \right), \varepsilon \right\} \quad (3.6)$$

Essentially, the above two equations are the same as equations (3.2) and (3.3), except that the true outcome is replaced by the market prediction (since the true outcome is not yet known at this stage) and the error outlier threshold now refers to the previous market, rather than the current one (since the agent never knows the exact prediction of every other agent⁶). Since the agent’s estimated revenue equals $\text{score}'_i \times \text{bet}_i$, then if $\text{score}'_i < 1$, the agent would make a loss, and so must set its bet to be as small as possible (lines 11 and 12). On the other hand, if $\text{score}'_i \geq 1$, the agent sets its bet to be as large as possible (lines 13 and 14).

Finally, Algorithm 4 handles the special case where the market happens to be the first one or the first round of a market. In the first market, the agent has no information about the market performance (i.e Q-values are void), hence the agent preserves its prediction value (line 17). Also, in this market, the agent cannot use Equation 3.6 (since it requires the error outlier threshold of the previous market), and so it cannot estimate its score. Consequently, we allow the agent to make only the smallest possible investment (line 18). Similarly, in the first round of each market, the agent does not have access to wisdom of the crowd because market prediction is not yet known, hence the agent cannot estimate its error and score. Therefore, the agent preserves its prediction value and invest MinRPT percentage of its budget.

Now that we have provided an overview of the Q-learning trading strategy, we move to give further details about (i) *states*, (ii) *actions*, (iii) updating the Q-values, and (iv) updating δ —the agent’s confidence in the crowd.

3.3.2.1 States

In a prediction market, the main objective for each participant is to maximise its revenue, which is not possible unless the agent submits an accurate prediction. The more accurate the prediction

⁶This is not an issue in Equation (3.3) since that equation is used solely by the market maker which knows every agent’s prediction.

Algorithm 4 Q-learning Trading Strategy of agent a_i

Input: $Prediction$, $prediction_i$, $round$

Output: $prediction_i$, bet_i .

```
1: if this is neither the first market nor the first round then
2:    $estError \leftarrow Prediction - prediction_i$ ;
3:    $state \leftarrow identifyState(estError, round)$ ;
4:    $action \leftarrow \operatorname{argmax}_{a \in \{Change, Preserve\}} Q(state, a)$ ;
5:   if  $action = Change$  then
6:      $prediction_i \leftarrow prediction_i + (\delta_{state} \times estError)$ ;
7:   else if  $action = Preserve$  then
8:      $prediction_i \leftarrow prediction_i$ ;
9:   end if
10:  calculate  $score'_i$  using Equation 3.5;
11:  if  $score'_i < 1$  then
12:     $bet_i \leftarrow budget_i \times MinRPT$ ;
13:  else
14:     $bet_i \leftarrow budget_i \times MaxRPT$ ;
15:  end if
16: else
17:   $prediction_i \leftarrow prediction_i$ ;
18:   $bet_i \leftarrow budget_i \times MinRPT$ ;
19: end if
20: return  $\langle prediction_i, bet_i \rangle$ 
```

of an agent, the higher the revenue it gains. Since the main concern of agents is accuracy of their prediction, the agents recognise their state by estimating their prediction error (i.e the absolute difference of agent and market predictions) and the round number.

However, the correct answer is not known until the market ends and hence agents cannot use this information to calculate their accuracy. Instead, agents can utilise the so-called “wisdom of the crowd” which is reflected in the market prediction (i.e the aggregated prediction of all participants in the market). Accordingly, an agent can assume that the correct answer is equal to the market prediction of the previous round and calculate its *estimated error* (i.e the absolute difference of agent’s and market’s predictions) instead.

Since the estimated error is a continuous variable, it is practically impossible for an agent

to store and update the Q-values for each combination of state and action. Hence, following the common approach in the Q-learning literature, we discretise estimated error to the three clusters, namely “Small”, “Medium” and “Large”. Discretisation has been shown to be very effective for one dimensional continuous states [Ng, 2011] in reinforcement learning problems. Since the scale of errors may change from one market to another and from one agent to another, each agent recomputes the decision boundaries of these clusters at the end of each market based on its own experiences.

More specifically, inspired by the online version of k-means clustering [MacQueen, 1967], the boundaries are computed as follows. First, each cluster is represented by its centroid (i.e the average of the estimated errors which are matched to the cluster). Agents map each estimated error to a cluster whose centroid has the lowest Euclidean distance to the estimated error. Cluster centroids are initially set to zero. However, each agent updates the clusters boundaries at the end of each market. In particular, the agent refers back to each round of the market and its estimated error in that round. Then, the agent matches the estimated error to a cluster and updates the centroid by averaging it with the estimated error. In cases where the distance of the estimated error to any cluster centroid is greater than the distance between any two cluster centroids, the two closest clusters are merged and their values (i.e centroids and corresponding Q-values) are averaged. Then, a new cluster with the centroid equal to the new estimated error is formed.

3.3.2.2 Actions

Considering human traders in real prediction markets, it is plausible that traders who have successful history in trading are more confident in their prediction. On the other side, traders with a less successful history have lower confidence in their prediction than the market prediction, especially if they have regretted that they could earn much more if they had listened to the wisdom of the crowd (market prediction) in previous markets. Henceforth, while the former type of traders may ignore market prediction when they participate in the market, the latter type of traders use market prediction as another source of information and shift their prediction by a percentage towards the market prediction, if they observe a high difference between their own and that of the market.

Our proposed Q-learning trading strategy attempts to emulate this behaviour of human traders. Therefore, the agents following this trading strategy have two actions of ‘Preserve’ and ‘Change’, which differ in whether the agent uses the market prediction as an additional source of information to modify its prediction. While ‘Preserve’ action suggests the agent ignores the market prediction of the previous round, the ‘Change’ action suggests the agent shifts its prediction by a percentage, δ , towards the market prediction. In each state, agents choose the action which has shown to be the most rewarding action in the past, or in other words, the action with the highest Q-value.

Since the agents’ primary goal is maximising their utility, both actions suggest the agent to maximise its utility according to its beliefs, hence they suggest the agent: (i) invest the possible maximum percentage (MaxRPT) of budget, if the agent believes that it will earn more

than what it invests, and (ii) invest the minimum possible percentage (MinRPT) of budget, if the agent believes that it will earn less than what it invests.

Agents can determine whether they will earn more or less than they invest by calculating their expected score as explained above (see equation 3.5). An agent expects to earn less than it invests, if the expected score is less than one. An agent expects to earn more than what it invests if the expected score is more than one (recall equation 3.1).

3.3.2.3 Updating Q-values

In Q-learning, an agent updates the Q-values of each state-action combination after knowing the resulting reward of taking that action in that particular state. Similarly in ACPM, once the market is closed and the correct answer is revealed, agents update the Q-value of the actions they executed in each of the states they have been through, during the market. In Q-learning techniques, Q-values are updated using the following formula:

$$Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (3.7)$$

where α is the learning rate and γ is the discount factor.

In ACPM, agents do not receive a reward except the time when the true outcome is revealed. Accordingly, in this work, we set the immediate reward to zero (i.e $r_t = 0$) and the discount factor parameter γ – which reflects how much the delayed reward worth less than the immediate reward – to 1. Since one of the main objectives of ACPM, in this thesis, is adapting to dynamic and non-stationary environment rapidly, the agent needs to update their Q-value fast and hence we set $\alpha = 1$. Hence, in ACPM, Formula 3.7 becomes:

$$Q(s, a) \leftarrow \max_{a'} Q(s', a') \quad (3.8)$$

which means agents set the Q-value of state s and action a (i.e $Q(s, a)$) equal to maximum return that results from taking action a in state s . Hence, ACPM agents assign $Q(s, a)$ equal to the amount of net revenue – its revenue minus the investment amount – they receive as a result of performing action a in state s .

One of the issues that an agent using reinforcement learning must address is the trade-off between exploration and exploitation. An agent prefers to exploit the actions which are already tried and result in the highest reward, however, agents need to explore other actions, in case those actions can potentially achieve higher rewards. ACPM agents not only can update the Q-value of the executed actions of the visited states, but also they can easily calculate the Q-value of other actions on those states. Agents can trace back to each state they visited during the market and calculate how much net revenue they could earn if they followed that certain action in that state. By knowing the true outcome, each agent, a_i , can now compute the exact, rather than the expected, revenue that could earn from each action. Consequently, for each state that the agent has visited, and each action that the agent could have made in that state, the Q-value is set to the computed revenue. For example, an agent can calculate the final net revenue it could obtain by considering:

- how much it would have invested on its prediction, if it had followed that particular action,
- how its prediction would have changed, if it had followed that particular action,
- how much would have been its error based on its updated prediction, using the correct outcome which is now revealed by the market maker, and
- how much net revenue it would have obtained according to earlier points and using the exact formula that the market maker used to reward participants in this market.

In this way, the agents do not need to trade off between exploration and exploitation and can exploit their attained knowledge without concerning for exploring other actions. Therefore, agents can follow the *greedy policy*, which is to select the action with the highest Q-value in each state. The main reason that agents do not need to explore is that ACPM is a supervised learning technique and once a market is over, the agents can access the correct answer to update their beliefs. In other words, the information that would be gained by exploration is provided by a post hoc supervised learning update.

3.3.2.4 Updating the Confidence in the Crowd:

At the end of each market, after the true outcome has been revealed, each agent updates δ —the parameter reflecting the level of confidence in the wisdom of the crowd. This parameter is computed for any given state as follows:

$$\delta_{\text{state}} = \text{truncate} \left(\frac{\text{outcome} - \text{prediction}_{i,\text{state}}}{\text{Prediction}_{\text{state}} - \text{prediction}_{i,\text{state}}} \right), \quad (3.9)$$

where $\text{prediction}_{i,\text{round}}$ and $\text{Prediction}_{\text{round}}$ denote a_i 's prediction and the market prediction at that round, respectively; and $\text{truncate} : \mathbb{R} \rightarrow [0, 1]$ is defined for every real value, $r \in \mathbb{R}$ as follows:

$$\text{truncate}(r) = \begin{cases} 0, & \text{if } r \leq 0 \\ 1, & \text{if } r > 1 \\ r, & \text{otherwise} \end{cases}$$

The truncate function limits δ between 0 and 1. When the agent error is equal to zero or less than the market error, the truncate function returns 0, which means the agent should not change its prediction toward the market prediction. The function returns 1 when the agent should have 100% confidence in wisdom of the crowd, which means that the agent should ignore its own prediction and only rely to market prediction.

3.4 Trading Strategy Examples

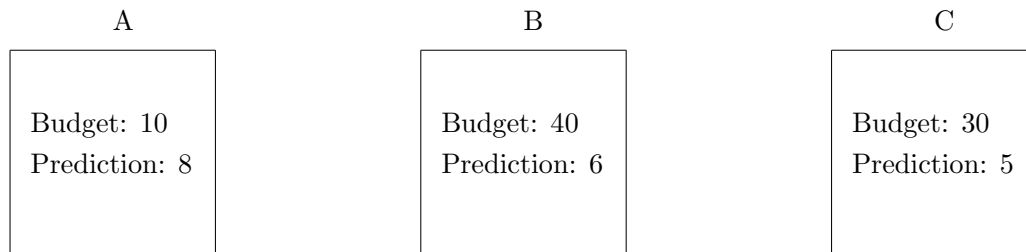
In this section, we demonstrate how agents and the market maker derive their predictions in a market instance with two examples corresponding to two cases: (a) agents using the constant trading strategy, and (b) agents using the Q-learning trading strategy. Given that the agents do not have equal budgets, these examples may demonstrate any market instance except the first one.

For both examples, imagine:

- We have 3 agents A, B and C,
- The true outcome of the event is 5,
- Number of rounds in each market is set to 2,
- MaxRPT and MinRPT is set to 90% for the first round, and
- MaxRPT is set to 1% and MinRPT is set to 0.01% for the second round.

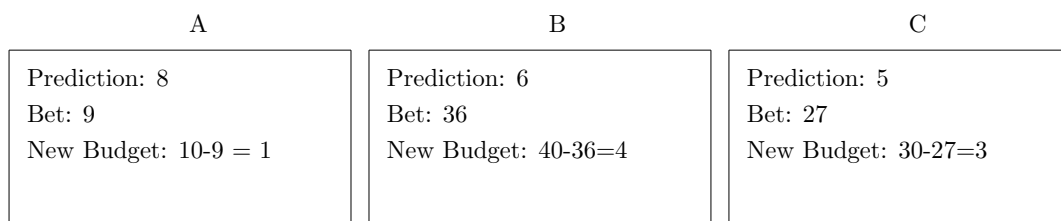
3.4.1 Constant Trading Strategy Example

In this example, ACPM participants use the constant trading strategy. The budget and prediction of each agent are assumed as shown below.



3.4.1.1 Round 1

By using the constant strategy, agents always invest MaxRPT percentage of their budgets and MaxRPT is set to 90% in the first round. Therefore, agent data are updated as below:



Using the aggregation function (see Formula 3.4), ACPM calculates the market prediction for first round:

$$\text{Market Prediction: } \frac{8 \times 9 + 6 \times 36 + 5 \times 27}{9 + 36 + 27} = 5.87$$

3.4.1.2 Round 2

In the second round, MaxRPT is set to 1% and accordingly agent bids are updated as shown below:

A	B	C
Prediction: 8 Bet: 0.01 New Budget: $1 - 0.01 = 0.99$	Prediction: 6 Bet: 0.04 New Budget: $4 - 0.04 = 3.96$	Prediction: 5 Bet: 0.03 New Budget: $3 - 0.03 = 2.97$

Then, ACPM calculates the market prediction for the second round:

$$\text{Market Prediction: } \frac{8 \times 0.01 + 6 \times 0.04 + 5 \times 0.03}{0.01 + 0.04 + 0.03} = 5.87$$

3.4.1.3 Final ACPM Prediction

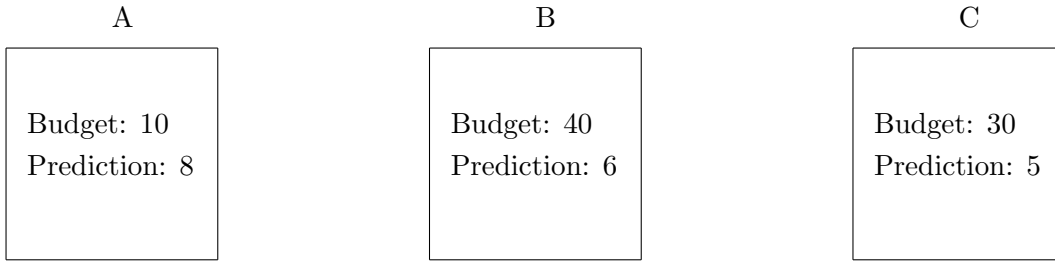
ACPM prediction for the record is equal to the final round market prediction.

$$\text{Final Market Prediction: } 5.87$$

As demonstrated by this example, by moving from the first round to the second, the agents do not change neither their prediction nor their investment, despite the fact that the market prediction calculated at the end of the first round informs wisdom of the crowd. Ignoring this information by agents, cause the market to arrive to the same prediction in both rounds.

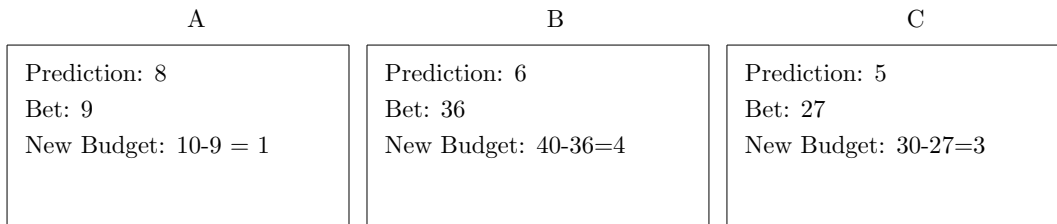
3.4.2 Q-Learning Trading Strategy Example

In this example, ACPM participants use the Q-learning trading strategy. The budget and prediction of each agent are as shown below:



3.4.2.1 Round 1

Using this strategy, agents invest MinRPT (which is set to 90%) percentage of their budget in the first round. Hence, agent data are updated as below:



Using the aggregation function (see Formula 3.4) , ACPM calculates the market prediction for first round:

Market Prediction: $\frac{8 \times 9 + 6 \times 36 + 5 \times 27}{9 + 36 + 27} = 5.87$

Recall, the market maker arrived to the same value of 5.87 in the previous example in which agents use the constant strategy. By moving to the next round, we see how Q-learning can decrease the market maker prediction error.

3.4.2.2 Round 2

Now, we demonstrate how agents decide their investment amount and their prediction in the second round using the Q-learning trading strategy.

I) Deciding Bets

Agents calculates their bets by calculating their accuracy and score using Equations 3.5 and 3.6, which we now restate:

$$score'_i = \max \{ \ln (accuracy'_i), 0 \},$$

where

$$accuracy'_i = \max \left\{ 100 \left(1 - \frac{|Prediction - prediction_i|}{\text{error outlier threshold}} \right), \varepsilon \right\}$$

As can be seen, to calculate $accuracy'$ an agent requires the error outlier threshold and since agents do not have access to this value until the market ends, the agents use that of the previous market, assuming that the previous market has the most similar characteristics to the current one. In this example, we assume the error outlier threshold of the previous market was 2.

Then, according to this strategy, agents invest MaxRPT percent of their budget if $score'$ is more than 1, and invest MinRPT of their budget, otherwise. Now, we present the calculation of each agent bet below:

A

$$\begin{aligned} accuracy'_A &= \varepsilon \\ score'_A &= 0 \\ \text{since } (score'_A < 1) \\ bet'_A &= \text{MinRPT} * \text{Budget} = 0.0001 * 1 = 0.0001 \end{aligned}$$

B

$$\begin{aligned} accuracy'_B &= 93.75 \\ score'_B &= 4.54 \\ \text{since } (score'_B > 1) \\ bet'_B &= \text{MaxRPT} * \text{Budget} = 0.01 * 4 = 0.04 \end{aligned}$$

C

$$\begin{aligned} accuracy'_C &= 56.25 \\ score'_C &= 4.02 \\ \text{since } (score'_C > 1) \\ bet'_C &= \text{MaxRPT} * \text{Budget} = 0.01 * 3 = 0.03 \end{aligned}$$

II) Deciding Predictions

State		Change Action Q-value	Preserve Action Q-value	Confidence in the Crowd(δ)
Round	Estimated Error			
2	Small (centroid=0)	0.0	0.25	0%
2	Medium (centroid=0.75)	0.10	0.15	70%
2	Large (centroid=2)	0.15	0.0	80%

Table 3.1: Q-Table Example. The first and second columns show state information. The third and fourth columns show Q-value for “Change” and “Preserve” actions respectively and the last column shows the best value of δ (i.e confidence of the agent in the crowd) for the corresponding state.

Agents following the Q-learning trading strategy create a Q-table from the first market and update their table at the end of each market based on the gained experience in that market. Although it is expected that agents have different Q-tables based on their own experience, in this example, we assume they all built similar Q-tables as shown in Table 3.1. In this example, imagine agents have built a Q-table as shown in Table 3.1: the first and second columns show state information. The third and fourth columns show Q-value for “Change” and “Preserve” actions respectively and the last column shows the best value of δ (i.e confidence of the agent in the crowd) for the corresponding state. Recall the agents require market prediction to estimate their estimated error and this information (i.e market prediction) is not available in the first round of the market. Hence, the agents cannot compute and store Q-value for the first round and that is why Table 3.1 only presents the states for each estimated error cluster in the second round.

A

Since agent A estimated error is $|5.875 - 8| = 2.125$, it realises that it is in large state (centroid = 2). As Table 3.1 shows, in this state, the Q-value for action “Change” is more than the Q-value for action “Preserve” ($0.15 > 0.0$). Therefore, agent A chooses the “Change” action and changes its prediction by δ percentage, which 80% for this state, toward the market prediction. Therefore, the agent changes its prediction to 6.30 using the following formula (see line 6 in Algorithm 4).

$$newprediction_i \leftarrow prediction_i + (\delta_{round} \times estError)$$

where,

$$estError \leftarrow Prediction - prediction_i;$$

hence,

$$New Prediction = 8 + 0.80 \times (5.875 - 8) = 6.30$$

B

Since agent B estimated error is $|5.875 - 5| = 0.875$, it realises that it is in small state (centroid = 0). As Table 3.1 shows, in this state, the Q-value for action “Change” is less than the Q-value for action “Preserve” ($0.0 < 0.25$). Therefore, agent B chooses the “Preserve” action and keeps its prediction at 6.

C

Since agent C estimated error is 0.875, it realises that it is in medium state (centroid = 0.75). As Table 3.1 shows, in this state, the Q-value for action “Change” is less than the Q-value for action “Preserve” ($0.10 < 0.15$). Therefore, agent C chooses the “Preserve” action and keeps its prediction at 5.

Thus, agent bids and market prediction in the second round are as shown below:

A	B	C
Prediction: 6.3 Bet: 0.0001 New Budget: $1 - 0.0001 = 0.999$	Prediction: 6.0 Bet: 0.04 New Budget: $4 - 0.04 = 3.960$	Prediction: 5.0 Bet: 0.03 New Budget: $3 - 0.03 = 2.970$

$$\text{Market Prediction: } \frac{6.30 \times 0.0001 + 6 \times 0.04 + 5 \times 0.03}{0.0001 + 0.04 + 0.03} = 5.57$$

3.4.2.3 Final Market Prediction

ACPM prediction for the record is equal to the final round market prediction.

$$\text{Final Market Prediction: } 5.57$$

As demonstrated by this example, Q-learning trading strategy leads market prediction from 5.87 to 5.57, by moving from first round to the second, and hence resulted in decreasing ACPM error from 0.875 ($|5 - 5.87| = 0.87$) to 0.57 ($|5 - 5.57| = 0.57$). In sum, Q-learning trading strategy caused ACPM error to be reduced by 0.30 (i.e $0.87 - 0.57 = 0.30$), from first round to the second, when its participants use Q-learning trading strategy. At this point, agents retrain their hypotheses and update their Q-tables to prepare for the next market instance. Then, the market is closed and a new market instance is initiated with the objective of predicting the next record.

3.4.2.4 Revenues

Once the number of rounds are over, the market maker reveals the correct answer and then gives revenue to the agents using Formula 3.1.

3.4.2.5 Round 1 Revenues

A	B	C
Error: 3 Revenue: 30.43 New Budget: 31.43	Error: 1 Revenue: 156.13 New Budget: 160.09	Error: 0 Revenue: 124.34 New Budget: 127.31

3.4.2.6 Round 2 Revenues

A	B	C
Error: 1.30 Revenue: 0.0004 New Budget: 31.43	Error: 1.00 Revenue: 0.16 New Budget: 160.25	Error: 0 Revenue: 0.14 New Budget: 127.45

3.4.2.7 Updating Q-Tables

After receiving revenues, agents update the Q-value of the actions they executed in each of the states⁷ they have been through (see page 63). Hence:

- Agent A updates Q-value of action “Change” in state (Round 2 and “Large” cluster) to the revenue it gained in that state, which is 0.0004.
- Agent B updates Q-value of action “Preserve” in state (Round 2 and “Small” cluster) to the revenue it gained in that state, which is 0.16.
- Agent C updates Q-value of action “Preserve” in state (Round 2 and “Medium” cluster) to the revenue it gained in that state, which is 0.14.

After that, agents calculate the revenue they could earn if they had executed the alternative action in the visited states based on the discussion on page 63. Hence,

- Agent A updates the Q-value of action “Preserve” in state (Round 2 and “Large” cluster) to the revenue it could gain by executing action “Preserve” in that state, which is 0.
- Agent B updates the Q-value of action “Change” in state (Round 2 and “Small” cluster) to the revenue it could gain by executing action “Change” in that state, which is 0.15.
- Agent C updates the Q-value of action “Change” in state (Round 2 and “Medium” cluster) to the revenue it could gain by executing action “Change” in that state, which is 0.12.

Accordingly agents’ Q-tables are updated as shown in Tables 3.2 to 3.4 and is carried forward to the next market instance.

State		Change Action	Preserve Action	Confidence
Round	Estimated Error	Q-value	Q-value	in the Crowd(δ)
2	Small (centroid=0)	0.0	0.25	0%
2	Medium (centroid=0.75)	0.10	0.15	70%
2	Large (centroid=2)	0.0004	0.0	100%

Table 3.2: Updated Q-Table for Agent A.

⁷For sake of simplicity, we assume agents do not update the decision boundaries of these clusters here.

State		Change Action	Preserve Action	Confidence in the Crowd(δ)
Round	Estimated Error	Q-value	Q-value	
2	Small (centroid=0)	0.15	0.16	100%
2	Medium (centroid=0.75)	0.10	0.15	70%
2	Large (centroid=2)	0.15	0.0	80%

Table 3.3: Updated Q-Table for Agent B.

State		Change Action	Preserve Action	Confidence in the Crowd(δ)
Round	Estimated Error	Q-value	Q-value	
2	Small (centroid=0)	0.0	0.25	0%
2	Medium (centroid=0.75)	0.12	0.14	0%
2	Large (centroid=2)	0.15	0.0	80%

Table 3.4: Updated Q-Table for Agent C.

3.5 Discussion

ACPM design provides a number of capabilities as listed below:

1) ACPM design is expected to bring high performance for two reasons:

- a) The system grants more influence to participants that have high quality data sources and effective analysis models. In the first market, all participants have equal amount of budget. Over time, by participating in different markets, agents with superior data and analysis models accumulate greater budget and low quality agents lose most of their budget after investing in several markets. In other words, the difference in agent budgets arises as a result of their performance. Consequently low quality agents cannot invest as much as well performing agents who accumulated high amount of budget. Since the aggregation function weights each prediction by the amount of investment, the high quality agents – who could acquired more budget over time and consequently can invest more on their prediction – have greater influence in predicting the outcome of the event.

In brief, the reward function and the aggregation function together give the system the power to associate more influence with distinguished agents that have high quality data sources and effective analysis models. This increased influence of the more successful agents is expected to increase the performance of the system overall.

- b) The other feature which increases the system performance is that the agents learn to improve their prediction by considering market prediction of the previous round as another source of information to update their bids. While high quality agents ignore market predictions, low quality agents minimise the amount of noise (low accurate prediction) they send to the market maker. Therefore, Q-learning does improve each agent’s performance

and consequently the system's performance by adding a further reduction in prediction error.

In brief, low quality agents only have a short-term negative impact – for the period of a few markets – after which those agents have either lost most of their budget or learnt to lessen the noise they send to the market.

- 2) The configuration of ACPM allows it to mimic and to hybridise several existing techniques:
 - a) Agents can have the same data source but different analysis models. In this case, ACPM is as an ensemble model. ACPM is a non-hybrid ensemble, if the agents use one type of analysis models but with different parameter setting, however, ACPM is a hybrid ensemble if agents use different types of analysis models.
 - b) Agents have distinct data source but same analysis models. In this case, ACPM acts as a data aggregation technique.
 - c) Agents have distinct data source and distinct analysis models. In this case, ACPM acts both as data aggregation technique and an ensemble model.
- 3) A key capability of the system is its dynamic adaptation to the changing environment where the quality of an agent's data source fluctuates over time. Once the quality of an agent prediction changes – either through loss or gain in data quality or the effectiveness of the analysis model for that data source, or both – its influence (i.e budget and investment) on the market prediction is adjusted according to its current quality.
- 4) ACPM can handle temporary or permanent addition and deletion of a data source at any point of time. Agents can be added or deleted at any point of time and ACPM dynamically adapts to the new situation.
- 5) ACPM achieves two objectives when its agents use our proposed Q-learning trading strategy:
 - a) Agents change their prediction and bet in response to other agents' trades. More specifically, agents estimate their error by comparing their prediction by that of the market (i.e considering other agents' trades). Based on this estimated error, agents decide what is the best action to execute (i.e maintain their prediction or change their prediction toward the market prediction by some degree).
 - b) Agents learn from their past experiences to improve their performance in future, similar to human traders. In particular, agents update their trading strategy at the end of each market by calculating the reward they could earn if executed different actions in each state.
- 6) ACPM is a novel machine learning technique.

As discussed in Chapter 2, there is relatively little research on artificial prediction markets as a machine learning technique [Chen and Vaughan, 2010, Lay and Barbu, 2010, Barbu and

Lay, 2012, Storkey, 2011, Storkey et al., 2012, Hu and Storkey, 2014]. We identify a number of shortcomings in these works, which we address in this thesis. First, they are presented as an offline technique, i.e the participants only learn before entering the market and never revise their hypothesis once trading starts. In contrast, ACPM is online and agents update their hypothesis at the end of each market. Second, their participants follow a fixed strategy, while in ACPM, agents use an adaptive strategy informed by their trading history. Third, when making predictions, their participants lack the ability to observe, and reflect upon, the wisdom of the crowd. However, ACPM agents have the ability to revise their predictions in response to those of other agents.

Having explained how our work extends the literature of artificial prediction market, the remainder of this section discusses how ACPM is different from the related work appearing in the literature of PEA (covered in Section 2.1.3, page 23) and the works which connect prediction markets and machine learning models (covered in Section 2.3.1, page 42).

ACPM bears some resemblance to the notion of Prediction with Expert Advice (PEA) [Cesa-Bianchi and Lugosi, 2006], where a *forecaster* collects predictions from a set of *experts* about a future outcome and then combines them to form a prediction. While the agents in ACPM can be seen as experts in PEA who provide predictions to the forecaster, there are key differences between ACPM and PEA, which we discuss below.

- PEA only advances a paradigm for a forecaster to weight and merge expert predictions, while ACPM offers a paradigm for both market maker (forecaster) and participating agents (experts). In PEA, an expert is viewed as a black box and is an entity that is external to the (PEA) model, whereas in ACPM, agents are explicitly modelled to modify their own performance over the course of the market, by being offered a level of autonomy through their trading strategies.
- The weighting process in ACPM is different from PEA. While in PEA the weightings of the experts are decided by the forecaster based on their previous performance, in ACPM, the agents independently decide how much to invest in each round, which determines their influence on the outcome. In ACPM, weights of agents (i.e experts in the context of PEA) are decided at two levels:
 - i) their budget which indirectly limits the weight of agent as it limits the amount of agent investment
 - ii) the amount they bet on their prediction which is decided by agent trading strategy.
 In other words, in ACPM, while maximum weight of an agent is determined by the budget they could earn over time, their actual weight is determined by the agents independently.
- While experts in PEA submit their prediction independently, ACPM agents revise their prediction and bet in response to other agents. Hence, ACPM agents leverage wisdom of the crowd, a notion which is missing in PEA,

Furthermore, ACPM is different from the works links prediction market to machine learning, in particular online learning. While the objective in those is formalising a connection between

PEA and prediction market, ACPM is a novel machine learning technique by adopting the concept of prediction markets.

In addition, those mentioned works make a connection between a prediction market and PEA by corresponding the outcomes of prediction markets to experts and each committed trade in the market to a training instance. However, if one wishes to describe ACPM as expert learning, agents correspond to experts and each prediction market can be referred to a training instance.

Finally, we mention the concept of artificial economy due to the number of common key words that they share with our work. Baum and Durdanovic [2000] propose artificial economy (named as Hayek) as a new form of reinforcement learning approach in order to solve very complex problems such as complex forms of Block World Problems. Their model consist of agents cooperating together to solve a very complex problem. Agents participate in different number of auctions, where each auction decides which agent will be the winner to execute the next action in order to solve the problem. The authors claim that their proposed model can solve almost all random Blocks World Problems with goal stacks 200 blocks high, where competing models can only solve up to goal stacks of maximum 8 blocks. This work is similar to ours in the sense that agents are embedded in an artificial market to coordinate their activities. However, their focus is on reinforcement learning, with the main application being to solve various forms of Blocks-World Problems. As such, they do not propose any prediction methods.

Chapter 4

System Evaluation

In order to test the performance of ACPM in different situations, we set out a number of hypotheses, for each of which we then designed an experiment to test through simulations. First, we investigate the general performance of ACPM using a real-world data set of syndromic surveillance. We use syndromic surveillance as our prime case study, for the reason which we discuss below.

The main objective of a syndromic surveillance system is the earliest possible detection of a disease outbreak within a population. To achieve this objective, much research has been done to discover potential data sources which contribute to disease outbreak identification. One of the main source of information is Internet and open source information such as search engine queries, posts on social media platforms and government websites [Buehler et al., 2008, Chen et al., 2010a]. However, the data quality of some of these data sources oscillates over time. Therefore, integrating available data sources according to an adaptive weighting scheme seems promising. In addition to availability of data sources, various analysis models are proposed. Given that the quality of data sources change over time, and the most suitable algorithm for a given data source is not known *a priori*, a reasonable response is to consider analysing each data source with a variety of algorithms and integrate their results. Hence, ACPM can effectively be applied to the problems of syndromic surveillance since ACPM can analyse each data source with a number of analysis models and combine their results according to their varying quality.¹

We further investigate the performance of ACPM in cases which are not presented in our real-world syndromic surveillance data set. For example, we investigate the performance of ACPM when an agent outperforms every other agents permanently (i.e for every record from the data set) and when the quality of such agent drops suddenly. To this end, we simulate these cases using synthetic data sets.

Finally, in order to investigate the performance of ACPM in other real-world application domains, we perform another set of experiments using a number of data sets from the widely-used repository of the University of California at Irvine (UCI).

We designed a number of hypotheses regarding the performance of ACPM which are pre-

¹In this chapter, we use syndromic surveillance data set to investigate the general performance of ACPM in different situations. However, in Chapter 5, we discuss syndromic surveillance systems in detail and investigate ACPM aspects as a syndromic surveillance system.

sented in Section 4.1. Next, Section 4.2 and 4.4 describe the data sets and the experiment settings we used to test our hypotheses. Finally, Section 4.5 demonstrates the result of our empirical experiments, followed by Section 4.6 which analyses our results.

4.1 Hypotheses

To investigate different aspects of ACPM performance in different cases, we set out three sets of hypotheses, namely A, B and C, which are:

A. Using a real-world data set of syndromic surveillance, this set of hypotheses tests the general performance of ACPM.

1. ACPM performance is higher than the mean performance across all participating agents.
2. ACPM performance is higher than its best performing agent.
3. The Q-learning trading strategy encourages low quality agents to change their prediction based on the aggregated prediction of other agents and encourages high quality agents to ignore market prediction as another source of information.
4. Adopting the Q-learning trading strategy, compared to the constant strategy, improves ACPM performance.
5. Adopting the Q-learning based trading strategy, compared to the constant trading strategy, improves the performance of each agent, and hence the performance of each analysis model.
6. ACPM outperforms well-known regression and ensemble models as ACPM can instantiate one agent to use each of those models.
7. ACPM changes focus between agents in response to changes in their quality.
8. ACPM requires relatively small additional time for including each additional agent.

B. Using synthetic data sets, this set of hypotheses investigates the performance of ACPM in edge cases.

1. ACPM is resilient to addition of low quality agents. In other words, addition of agents with low quality does not affect the performance of ACPM significantly.
2. ACPM performance is higher than its best performing agent in the situations where one of the agents permanently outperforms the other agents.
3. ACPM performance is higher than its best performing agent in the situations where a number of equally well performing agents permanently outperform other agents.
4. ACPM performance is higher than its best performing agent in the situations where all agents perform equally well.
5. ACPM performance is higher than its best performing agent in the situations where all agents perform equally poor (i.e access random data).

6. ACPM performance is higher than its best performing agent in situations where all agents access poor quality data, however, for each consecutive number of records, one of the agents performs perfectly (i.e obtains zero error).
 7. ACPM requires relatively a small number of markets to learn and recover in situations where the best performing agent suddenly loses quality or become unavailable.
- C. Using UCI data sets, this set of hypotheses examines the performance of ACPM in a variety of real-world domains. In particular, we compare ACPM performance with that of popular techniques in the literature of machine learning and prediction with expert advice.
1. ACPM performance is domain independent.
 2. ACPM outperforms well-known forecasters from the literature of prediction with expert advice.

4.2 Data Sets

In this section, we describe the data sets used for our experiments.

4.2.1 Syndromic Surveillance Data Sets

One of the popular syndromic surveillance applications is predicting the disease activity level of influenza-like illnesses (ILI) [Eysenbach, 2006, Ginsberg et al., 2009, Polgreen et al., 2008]. In these experiments, ACPM predicts the disease activity level of influenza-like illnesses (ILI) in a given week in the whole of the USA using publicly available data sources. The data used here contains more than 100 real data sources covering the period 4th January 2004 (when most of our data sources became publicly available) to 27th April 2014 (when these experiments were performed), from a variety of sources of Google Flu Trends (GFT), Centers for Disease Control and Prevention (CDC), and Google Trend.

Google Flu Trends² is a web service operated by Google which attempts to make accurate predictions about flu activity by aggregating Google search queries. We used weekly Google Flu Prediction for different areas of the United States including states, cities and regions, for which GFT data is available since January 2004.

The CDC Influenza Division³ produces a weekly report on influenza-like illness⁴ activity in the USA. We used CDC statistics⁵ including CDC ILI rate for different age groups (0-4 years, 5-24 years, 25-49 years, 50-64 years, and older than 65 years), USA national ILI rate, total number of patients and total number of outpatient healthcare providers in ILI network⁶. CDC reports ILI rates with a two-week time lag. Therefore, in order to align CDC data with the other

²<https://www.google.org/flutrends/>

³<http://www.cdc.gov/flu/>

⁴ILI is defined as fever (temperature of 100°F [37.8°C] or greater) and a cough and/or a sore throat without a known cause other than influenza (<http://www.cdc.gov/flu/weekly/overview.htm>)

⁵These data can be accessed from <http://gis.cdc.gov/grasp/fluview/fluportaldashboard.html>

⁶U.S. Outpatient Influenza-like Illness Surveillance Network.

data sources used, we take the ILI rate from two weeks earlier for each week of the experiment period. ACPM prediction is compared against the CDC ILI rate.

Google Trends⁷, another web service operated by Google, shows how often a particular search-term is entered relative to the total search-volume across various regions of the world. Google Trend statistics for different terms such as “flu”, “fever”, “cough”, “sore throat”, “flu symptoms” for each week of the mentioned period in USA are used as another set of data sources.

4.2.2 Artificial Data Sets

To investigate the performance of ACPM in edge cases, we create different noisy data sources to be accessed by ACPM agents. To this end, we use CDC ILI rate for whole United States⁸ as the correct answers. For ease of representation and comparison, we normalised the correct answer to the range of 0 and 1. Then, we create different data sources by generating some degree of noise to manipulate the correct answer. This degree of noise is decided based on the objective of the corresponding hypothesis (for more detail refer to Section 4.4.3 in page 85). To generate our noisy data, we use a simple model, in which, $x\%$ noisy data is constructed by sampling uniformly at random from the interval (the correct answer $-x\%$, the correct answer $+x\%$).

4.2.3 UCI Data Sets

To establish the general nature of ACPM, we investigate the performance of ACPM on a variety of other real-world data sets, in addition to our syndromic surveillance data set. Hence, we randomly chose ten UCI data sets⁹ from those identified as suitable for regression; these turned out to be: (i) Bike Sharing, (ii) Auto MPG, (iii) Yacht Hydro-dynamics, (iv) Istanbul Stock Exchange, (v) Servo, (vi) Forest Fires, (vii) Automobile, (viii) Housing, (ix) Airfoil Self Noise, and (x) Computer Hardware. Table 4.1 shows the data sets used in this set of experiments including their number of attributes and instances.

4.3 ACPM Parameter Setting

Like many other machine learning techniques, ACPM has a few parameters that need to be set. The optimisation of these parameters can be done experimentally using historical data. However, in this thesis, we assume we have no access to historical data and hence we cannot optimise these parameters. Therefore, we select some values for these parameters which we use here to evaluate ACPM. Though tuning these parameters for each particular experiment can improve the performance of ACPM in each case, we expect these values deliver effective results without requiring spending time on tuning the parameters and avoids risking overfitting to the training data.

⁷www.google.com/trends/

⁸This data covers 4th January 2004 to 5th February 2015 (when these experiments are performed)

⁹<https://archive.ics.uci.edu/ml/datasets.html>

Data Set	Number of Records	Number of Attributes
Bike Sharing (hourly basis)	730	16
Auto MPG	398	8
Yacht Hydrodynamics	360	7
Istanbul Stock Exchange	536	8
Servo	167	4
Forest Fires	517	13
Automobile	205	26
Housing	506	14
Airfoil Self Noise	1503	6
Computer Hardware	209	9

Table 4.1: UCI Data Sets Used for Experiments

4.3.1 Market Duration

In situations where agents use constant trading strategy (described in Section 3.3.1, page 59), variation of market duration (number of rounds) does not affect the market prediction, since agents change neither their prediction nor their investment over different rounds of a market.

In situations where agents use the Q-learning trading strategy (described in Section 3.3.2, page 59), the number of rounds must be equal or greater to 2, so that agents can update their bids using the market prediction of the previous round, otherwise Q-learning trading strategy is useless. Therefore, we set the value of market duration (number of rounds) to 2. However, the optimisation of this parameter for each experiment can be done experimentally using historical data, as mentioned earlier.

4.3.2 Rate Per Transaction Parameters

In Section 3.2.3 (page 56), we discuss the purpose of MinRPT and MaxRPT parameters. In here, we intuitively set these parameters with regard to a market consisting of two rounds. According to our discussion in Section 3.2.3 (page 56):

i) First round

- MaxRPT and MinRPT: while we want agents to bet less than 100% of their budget in each round in order to avoid agent bankruptcy, we prefer agents to bet most of their budget in the first round, so that as soon as their quality vary, this variation be reflected in their budgets. Hence, we set MaxRPT and MinRPT for first round to 90%.

ii) Second round

- MinRPT: the primary purpose of MinRPT is to make sure that ACPM is not unresponsive in situations where none of its agents has any incentive to invest (i.e intend

to place a bet of 0). Hence, any small value for MinRPT serves the purpose and accordingly we set MinRPT to 0.01%.

- MaxRPT: this parameter must be set to more than 0, so that agents be motivated to participate in the second round of a market and submit their revised prediction after listening to wisdom of the crowd. However, we do not want the budget of agents be largely affected by the revenue they receive by this round bids, simply because these round bids do not reflect their original quality. Accordingly, we limit the amount they can bet – and hence receive revenue¹⁰ – by setting MaxRPT for second round to 1%.

These values are suggested by concentration on the situations which quality of agents vary over time, which is one of the motivations of this work. In Section 4.5.16, we evaluate the performance of ACPM using a number of randomly chosen UCI¹¹ data sets, using these default values. While the assumption of variations of agent qualities may not be present in some of these experimented data sets, ACPM works effectively in all these data sets, as demonstrated in Section 4.5.16 (page 102).

Appendix B presents the performance of ACPM in these data sets for different values of MaxRPT and number of rounds. The appendix shows that the values we suggest here are not the best for all data sets and if we tune these parameters, the performance of ACPM can be even higher than the performance we presented in Section 4.5.16 (page 102).

4.4 Experimental Setup

Five different experiment settings are designed to create the market conditions required for various experiments to test the hypotheses described in Section 4.1, page 78.

Unless otherwise stated, in all the experiments:

- The results of the experiments are based on one run as they are deterministic,
- Agents use the Q-learning trading strategy,
- Our evaluation criterion is Mean Absolute Error (MAE), which is a common measure in the literature. MAE is calculated using the following formula:

$$\text{MAE} = 1/n \sum_{i=1}^n |(Y_i - P_i)|$$

where Y_i the true value and P_i is the prediction and n is the number of records (i.e markets),

- The experiments are performed using a MacBook Pro laptop (2.5 GHz Intel Core i5 processor, 8 GB Memory, Os X Yosemite).

¹⁰ Recall the reward function rewards agents based on their investment amount as well as their prediction accuracy (see Formula 3.1).

¹¹Repository of the University of California at Irvine (UCI).

4.4.1 Setting for Experiments A1-A4

In this section, we explain the setting used for experiments evaluating Hypotheses A1 to A4. Hypotheses A1 and A2 examine the performance of ACPM with regard to all agents' average performance and the best performing agent performance. Hypotheses A3 and A4 investigate how the Q-learning trading strategy is used by agents and whether this intelligent strategy can increase ACPM performance.

Therefore, we designed four experiments to test each of these hypotheses. Experiment A1 examines ACPM performance with average of all participant performance and Experiment A2 compares the performance of ACPM with the best performing agent. Experiment A3 examines the popularity of each action of the Q-learning trading strategy for agents with different quality. Finally, Experiment A4 investigates the effect of the Q-learning trading strategy, compares to the constant trading strategy, in the performance of ACPM.

To achieve our goal, we create agents accessing different quality data sources. To distinguish agent quality based on their data source, we set all agents to use the same analysis model. Then, four market types with different proportions of participant (with regard to their data source quality) are defined. Each market has 100 participants each having one distinct data sources. We distinguish the quality of data sources to three categories of low, medium and high quality, based on their mean absolute error (MAE) as calculated by a number of regression models¹². These categories are not absolute judgements, but relative ones confirmed through the use of the mentioned analysis models to cluster the data sources according to their mean absolute error (MAE) and hence identify threshold values that fall between the clusters. Categories thresholds are set at $(0.67, 1]$ for low quality, $[0.55, 0.67]$ for medium and $[0, 0.55)$ for high quality¹³. A summary of each market can be found in Table 4.2 and a detailed information of the data sources used for each market type is presented in Appendix C. Our four market types are described below.

Market Type 1: This market comprises agents with medium quality data only (100%).

Market Type 2: To investigate how the presence of a small number of high quality agents affect the ACPM performance, this market comprises mostly (97%) medium and a few (3%) high quality data agents.

Market Type 3: To investigate how the presence of a small number of low quality agents affect ACPM performance, this market contains mostly (88%) medium and several (12%) low quality data agents.

Market Type 4: This market contains several (13%) low quality data agents, a few (3%) high quality data agents and many (84%) medium quality data agents.

In these experiments, all agents use the same analysis model, namely SGD¹⁴ from the implemented Java Weka API (3-7-10) and configured with their default parameters. There is no

¹²The models are SGD, IBK, LinearRegression, SMOreg, REPTree, ZeroR, DecisionStump, SimpleLinearRegression, DecisionTable, LWL, Bagging, AdditiveRegression, Stacking and Vote.

¹³Based on the mean absolute error (MAE) of SGD (with Squared Loss function).

¹⁴SGD loss function is set to Squared Loss function for the purpose of performing regression.

Type	Mean Error (Variance)	Low Quality Agents		Medium Quality Agents		High Quality Agents	
		Percentage	Mean Error (Variance)	Percentage	Mean Error (Variance)	Percentage	Mean Error (Variance)
Type 1	0.6043 (0.0007)	0%	-	100%	0.6042 (0.0007)	0%	-
Type 2	0.6009 (0.0009)	0%	-	97%	0.6033 (0.0008)	3%	0.5243 (4.79002E-05)
Type 3	0.6214 (0.0030)	12%	0.7423 (0.0020)	88%	0.6048 (0.0008)	0%	-
Type 4	0.6198 (0.0033)	13%	0.7406 (0.0019)	84%	0.6044 (0.0008)	3%	0.5243 (4.79002E-05)

Table 4.2: Our four market types. Data sources are divided into three categories of low, medium and high quality based on their MAE as determined by several regression models. Table rows describe market types according to data quality of their participants.

specific reason for the use of SGD: it is just one of several models used for the initial clustering and other analysis models produce qualitatively-similar results.

4.4.2 Setting for Experiments A5-A8

In this section, we explain the setting used for experiments evaluating Hypotheses A5 to A8. In this set of experiments, we are interested to compare the performance or the time consumption of ACPM and different analysis models. Therefore, we instantiate one agent for each of the the following analysis models: SGD, IBK, LinearRegression, SMOreg, REPTree, ZeroR, DecisionStump, SimpleLinearRegression, DecisionTable, LWL, Bagging, AdditiveRegression, Stacking and Vote. This selection of models comprises those: i) available in Java Weka API (3-7-10), ii) capable of performing regression, iii) can be run on the data set without generating an error. Since the focus of these experiments are on agent analysis models, we set all agents to have the same data sources. In this way, we can investigate the behaviour of system and agents with regard to analysis models, while other variations are kept constant.

In the experiments evaluating Hypotheses A5 and A6, the data source for each agent is the entire data set, described in market type 4 of Table 4.2.

Hypothesis A7 investigates how ACPM shifts focus between agents, when the quality of agents vary. To investigate how varying the quality of a data source in different periods affects the performance of agents and ACPM, In Experiment A7, agents access to only one data source (Google Flu Prediction for the entire United States).

Hypothesis A8 and accordingly Experiment A8 examines the average time consumption of ACPM and different agents in each market. With this in mind, we require that all agents have the same data source. Therefore, we set all agents to access Google Flu Prediction for the entire United Sate¹⁵. Our evaluation criteria in this experiment is the amount of time taken by ACPM and each agent (i.e analysis model).

¹⁵By increasing the size of data source, the time consumption increases for analysis models, but not ACPM itself.

4.4.3 Setting for Experiment B1

In this section, we explain the setting used for experiments evaluating Hypothesis B1. The hypothesis investigates how presence of agents with low and high quality affects the performance of ACPM. Therefore, the experiment comprises three markets with three different populations with regards to their prediction quality.

Market Type A: The market has 100 agents receiving random data with a uniform distribution.

Market Type B: The market has agents that are same as the type A market and additionally one agent with high quality data, namely Google Flu Prediction for the whole of the United States, which has Mean Absolute Error (MAE) equal to 0.5275.

Market Type C: The market has only one agent which is the high quality agent of market type B (i.e the agent receiving data from Google Flu Prediction for whole United State.)

4.4.4 Setting for Experiments B2-B7

A set of experiments is designed to test Hypotheses B2 to B7. In these experiments, we use synthetic data to simulate the required market conditions. As mentioned in Section 4.2.2 (page 80), $x\%$ noisy data is constructed by sampling uniformly at random from the interval (the correct answer $-x\%$, the correct answer $+x\%$).

Setting for Experiment B2

To test Hypothesis B2, the experiment examines the performance of ACPM in the situation where one of the agents permanently outperforms the other agents. Hence, the experiment has one agent receiving a 10% noisy data and 39 agents accessing distinct 50% noisy data sources.

Setting for Experiment B3

To test Hypothesis B3, the experiment examines the performance of ACPM in situations where a number of equally well performing agents permanently outperform the other agents with similar quality. Therefore, the experiment with this setting has 5 agents accessing distinct 10% noisy data and 35 agents accessing distinct 50% noisy data sources.

Setting for Experiment B4

To test Hypothesis B4, the experiment examines the performance of ACPM when all agents perform equally well. Accordingly, the experiment includes 40 agents each accessing 40 different data sources with 10% noise.

Setting for Experiment B5

To test Hypothesis B5, the experiment examines the performance of ACPM when agents access very poor quality data such as random data. To achieve this, the experiment has 40 agents each access a distinct random data with uniform distribution.

Setting for Experiment B6

To test Hypothesis B6, the experiment examines the performance of ACPM when all agents access very poor quality data, however, for each number of consecutive markets, one agent predicts perfectly. Accordingly, the experiment includes 40 agents each access randomly generated data, but for each set of 10 records, one agent is chosen randomly to access the correct answer.

Setting for Experiment B7

To test Hypothesis B7, the experiment examines the performance of ACPM in situations where the quality of the agents' predictions vary over time. In particular, the experiment examines how ACPM responds in situations where the best performing agent suddenly loses its quality or become unavailable. Hence, the experiment has 5 agents with fixed errors of 0.1, 0.2, ..., 0.5. A data with fixed error of X, is the addition of X to the correct answer. After each 100 markets, the error of the best performing agent is set to 1 and kept unchanged until the end.

In all the above, agents simply return the data as the prediction (i.e no analysis model). In Experiments B2 to B6, we measure accumulated error of ACPM and its agents from the first market to the last market. However, since Hypothesis B7 investigates the immediate behaviour of ACPM due to a sudden change in the system, we measure and present error (rather than the accumulated error) of agents and ACPM in Experiment B7.

As the process of generating noise in experiment B2 to B6 are not deterministic, the experiments are run for 100 times. For sake of clear presentation, the figures in Section 4.5.10 to 4.5.14 (page 97 to 100) shows the results for one run, which are very similar to the averaged results for 100 runs. Please refer to Appendix D for the results based on 100 runs.

4.4.5 Setting for Experiments C1-C2

In this section, we explain the setting used for experiments evaluating Hypotheses C1 and C2. Hypothesis C1 examines the performance of ACPM in a variety of real-world domains and Hypothesis C2 claims ACPM outperforms some well-known models from the literature of prediction with expert advice.

In these experiments, we run a number of experiments using UCI data sets, which we mention in Section 4.2.3 (page 80). For every data set, the records were revealed to the different analysis models, one record at a time (according to their original order in the data set). Any records that include null values were excluded from this process. The first five records were used for the initial training of models, and were excluded from the evaluation process. We used a variety of models from R's widely-used *caret* package (version 6.0-37), namely:

- 1) Bagged CART (treebag),
- 2) Conditional Inference Random Forest (cforest),
- 3) Random Forest (rf),
- 4) Multi-Layer Perceptron (mlp),

- 5) Model Averaged Neural Network (avNNet),
- 6) Boosted Generalized Linear Model (glmboost),
- 7) Boosted Tree Linear Regression (blackboost),
- 8) Linear Regression (lm),
- 9) Gaussian Process (gaussprLinear),
- 10) CART (rpart),
- 11) Generalized Linear Model (glm),
- 12) K-Nearest Neighbors (knn), and
- 13) Gaussian Process with Polynomial Kernel (gaussprPoly).

Any parameters of those models were kept at their default values. This selection of models comprises those: i) available in R's *caret* package (version 6.0-37), ii) capable of performing regression, iii) can be run on all of the mentioned data sets without generating an error.

To test these hypotheses, we constructed an ACPM with 13 agents, each agent has a unique analysis model corresponding to one of the above analysis model. We experimented with two variations of ACPM; one in which the agents use a constant trading strategies, and the other in which the agents use Q-learning trading strategy. In Experiment C1, we compare the performance of ACPM with the same analysis models mentioned above as benchmarks.

We then experimented four well-known forecasters from the literature of Prediction with Expert Advice (PEA) [Cesa-Bianchi and Lugosi, 2006], namely:

- 1) Exponentially Weighted Average Forecaster (EWA) [Cesa-Bianchi and Lugosi, 2006],
- 2) Tracking the Best Expert¹⁶ (TBE) [Herbster and Warmuth, 1998],
- 3) Following the Best Expert (FBE) [Robbins, 1985],
- 4) Exponentiated Gradient algorithm (EG) [Kivinen and Warmuth, 1997].

In our experiments, each of these forecasters had one expert (i.e agent) for each of the aforementioned learning models, meaning that they were given the same input as our ACPM. The forecaster parameters are set according to the suggestions made by their own developers. Specifically:

- 1) EWA: $\eta = \sqrt{2 \ln(n)/T}$, where n is the number of experts and T is number of time steps (i.e number of records).
- 2) TBE: $\eta = 1/2$ and $\alpha = n/T$, where n is the number of experts and T is number of time steps (i.e number of records).
- 3) FBE: no parameter to be set.

¹⁶More specifically, the fixed-share algorithm.

4) EG: $\eta = 2/(3R^2)$, where R is an upper bound for the maximum difference between expert predictions at time T.

Experiment C2 benchmarks the performance of ACPM with each of the forecasters mentioned above. In these experiments, our evaluation criteria is Mean Squared Error (MSE) as MSE is a popular metric in the literature of PEA. MSE is calculated using the following formula:

$$\text{MSE} = 1/n \sum_{i=1}^n (Y_i - P_i)^2$$

where Y_i the true value and P_i is the prediction and n is the number of records (i.e markets).

4.5 Experiments

In order to test Hypotheses A1 to C2, a number of experiments are performed, the results of which are presented in this section.

4.5.1 Experiment A1

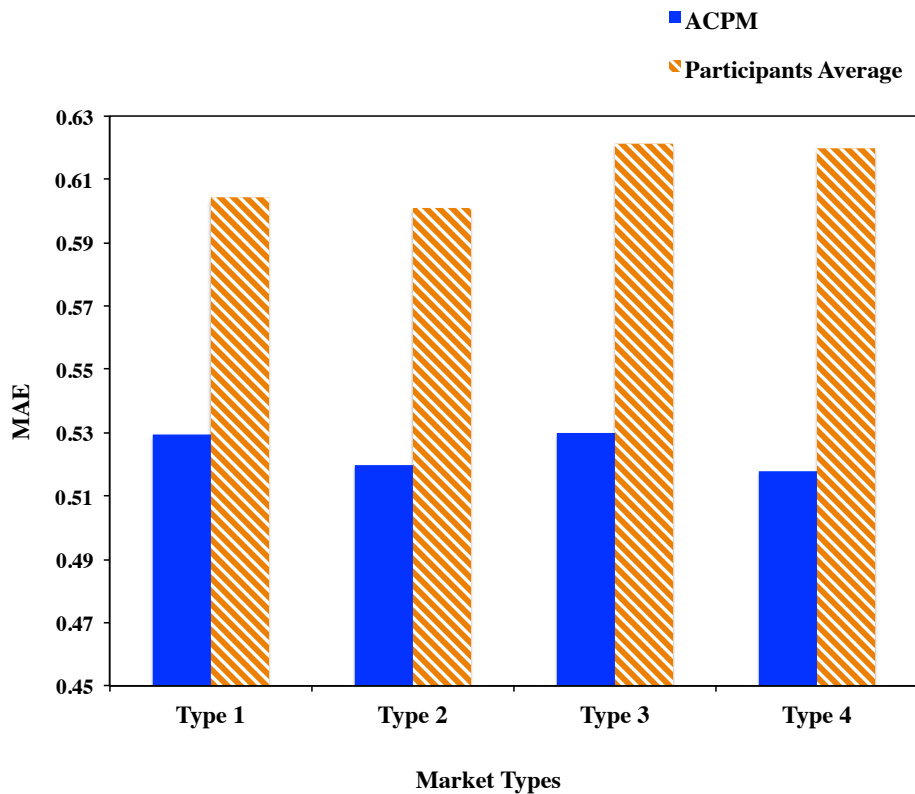


Figure 4-1: Experiment A1. Comparing ACPM performance with the mean performance of participants for each market type. The figure shows that the performance of ACPM is higher than the mean performance of all agents in each market type.

This experiment tests Hypothesis A1, which states ACPM performance is higher than the mean performance across all participating agents. The experiment explores the impact of different combinations of participating agents with regards to their quality on ACPM performance. As described in Table 4.2, we consider four markets with different proportions of data source quality participants.

Figure 4-1 compares the performance of ACPM with the average performance of all agents in each market type. The figure shows that the performance of ACPM is higher than the mean performance across all agents across all four market types. In particular, the figure shows, the MAE of ACPM is less than the average of agents MAE in market type 1 to type 4 by 12%, 13%, 14% and 16%, respectively. The results are statistically highly significant in all cases ($p\text{-value} < 2.20E - 16$).

4.5.2 Experiment A2

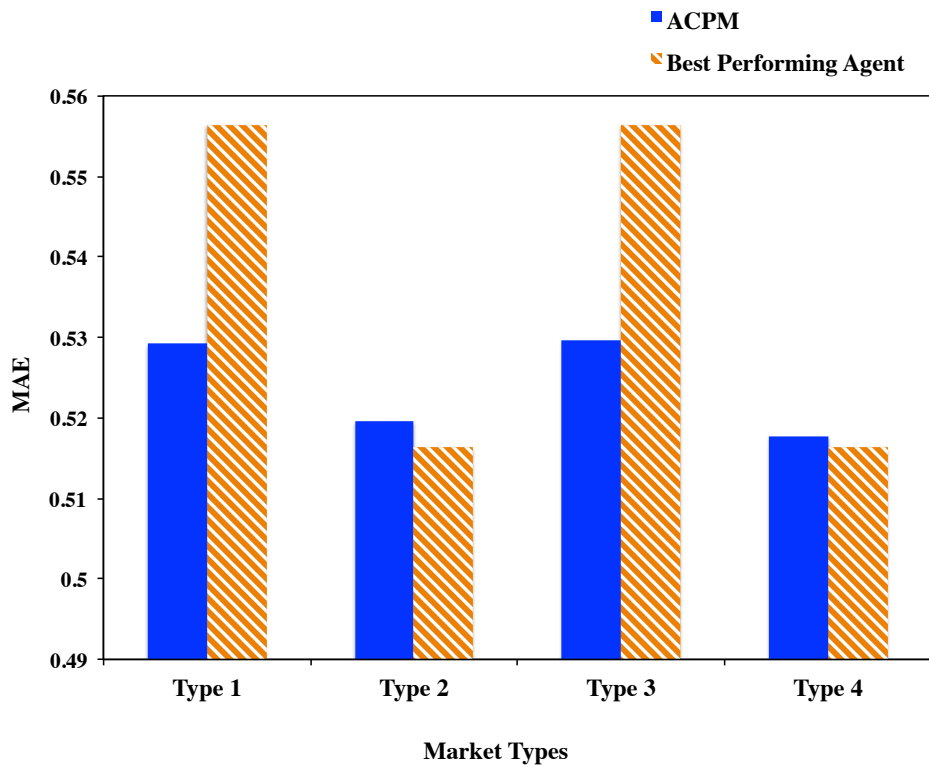


Figure 4-2: Experiment A2. Comparing ACPM performance with the best performing participant performance for each market type. ACPM outperforms best performing agent in each market type.

This experiment tests Hypothesis A2, which states ACPM performance is higher than its best performing agent. Since ACPM performance is dependent on participating agents' performance, one reasonable measure for evaluating ACPM performance is comparing the performance of ACPM with its best performing agent.

Figure 4-2 compares ACPM performance with the best performing agent performance in each market type. In here, we consider the original prediction of agents and not the improved prediction using the Q-learning trading strategy since the improved prediction does not present the original performance of an agent but the enhanced performance of an agent through ACPM mechanism. The MAE of ACPM is 4% less than the best performing agent in type 1 (p-value = $1.32E - 04$) and 4% less in type 3 (p-value = $1.40E - 04$). However, ACPM performance is slightly worse than the best performing agent in type 2 and 4 (i.e 0.6% in type 2 and 0.2% in type 4). This is due to the presence of high quality agents in market type 2 and 4, while market types 1 and 3 only have medium and low quality agents.

4.5.3 Experiment A3

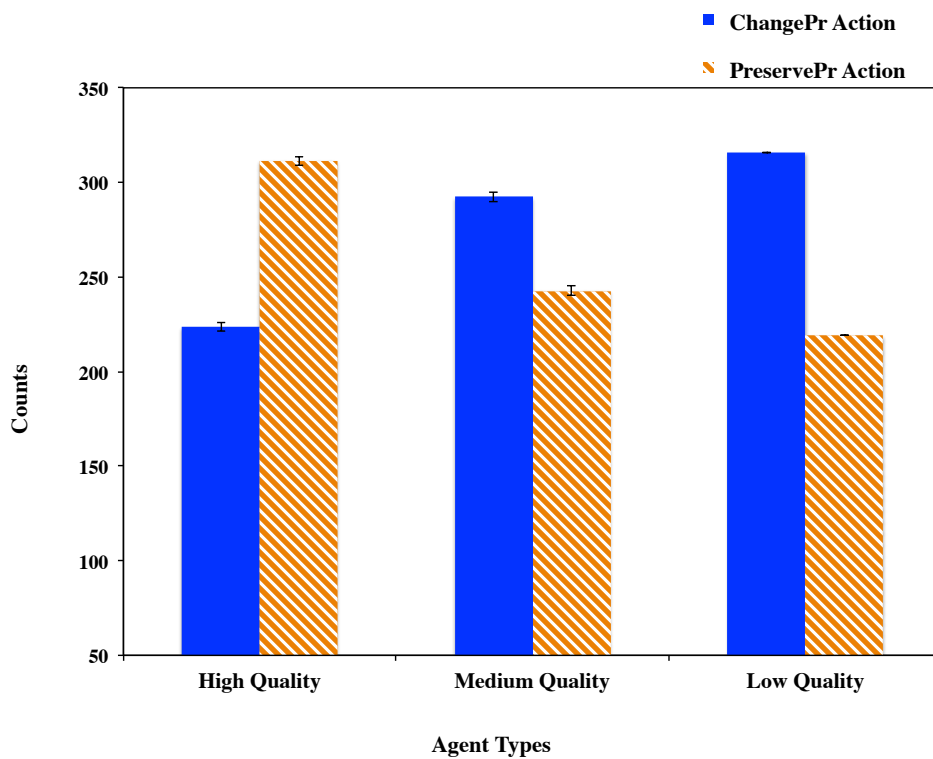


Figure 4-3: Experiment A3. Popularity of each action for agents with different quality. Action “ChangePr” is the most popular action for low quality agents and “PreservePr” is the most popular action for high quality agents.

This experiment tests Hypothesis A3. The Q-learning trading strategy, proposed in Section 3.3.2 (page 59), offers two actions: “ChangePr” and “PreservePr”. Hypothesis and Experiment A3 investigates which action is more popular for agents with different quality. As explained in Section 4.4.1 (page 83), we distinguish the data sources into three categories of low, medium and high quality. In this experiment, each agent is fed by one of these data source qualities and all agents use the same analysis models. Therefore, agents are distinguished by low quality , medium quality or high quality according to the data they have.

As can be seen from Figure 4-3, Action “PreservePr” which suggests the agent does not change its prediction, based on the previous round market prediction, is the most popular action for agents with high quality data and the least popular action for agents with low quality data. Conversely, Action “ChangePr”, which suggests the agent changes its prediction based on the previous round market prediction, is the most popular action in agents with low quality data and the least popular action in agents with high quality data. The graph also shows that high quality agents choose “ChangePr” action and low quality agent choose “PreservePr” action occasionally.

4.5.4 Experiment A4

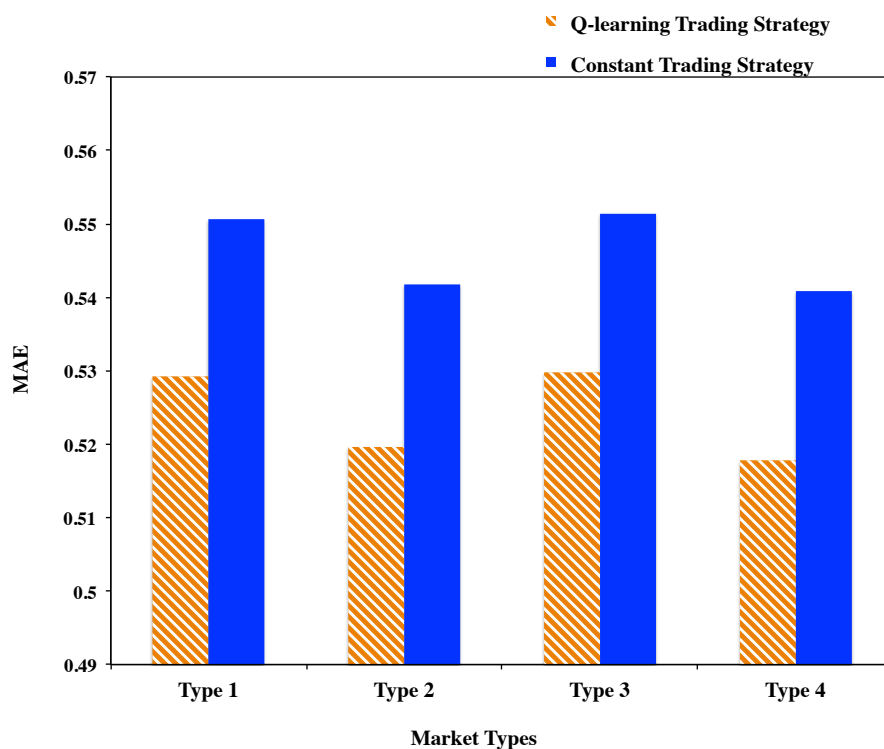


Figure 4-4: Experiment A4. Comparison of ACPM’s performance with Q-learning and without. Adopting the Q-learning by participants increases the performance of ACPM compared to the constant trading strategy in each market type.

This experiment tests Hypothesis A4, which states adopting the Q-learning trading strategy, compared to the constant strategy, improves ACPM performance. As discussed in Section 3.3 (page 59), we designed: constant trading strategy and Q-learning trading strategy. While the first one ignores the previous round market prediction, the latter one uses this information, to update the bids for the next trade.

This trading strategy causes low quality agents to lessen the amount of noise they send to the market by updating their bids based on the wisdom of the crowd (i.e market prediction). However, high quality agents preserve their prediction and ignore the market prediction as they

believe in themselves more than then the crowd. Experiment A4 compares the performance of ACPM in two situations:

- 1) all participants use the constant trading strategy,
- 2) all participants adopts the Q-learning trading strategy.

Figure 4-4 shows that adopting the Q-learning trading strategy by participants increases the performance of ACPM compared to the constant trading strategy in each market type, as expected. The results are highly significant in all market types (p-value for Type 1= $2.77E-39$, p-value for Type 2= $5.78E-42$, p-value for Type 3= $1.26E-48$, p-value for Type 4= $5.05E-53$).

4.5.5 Experiment A5

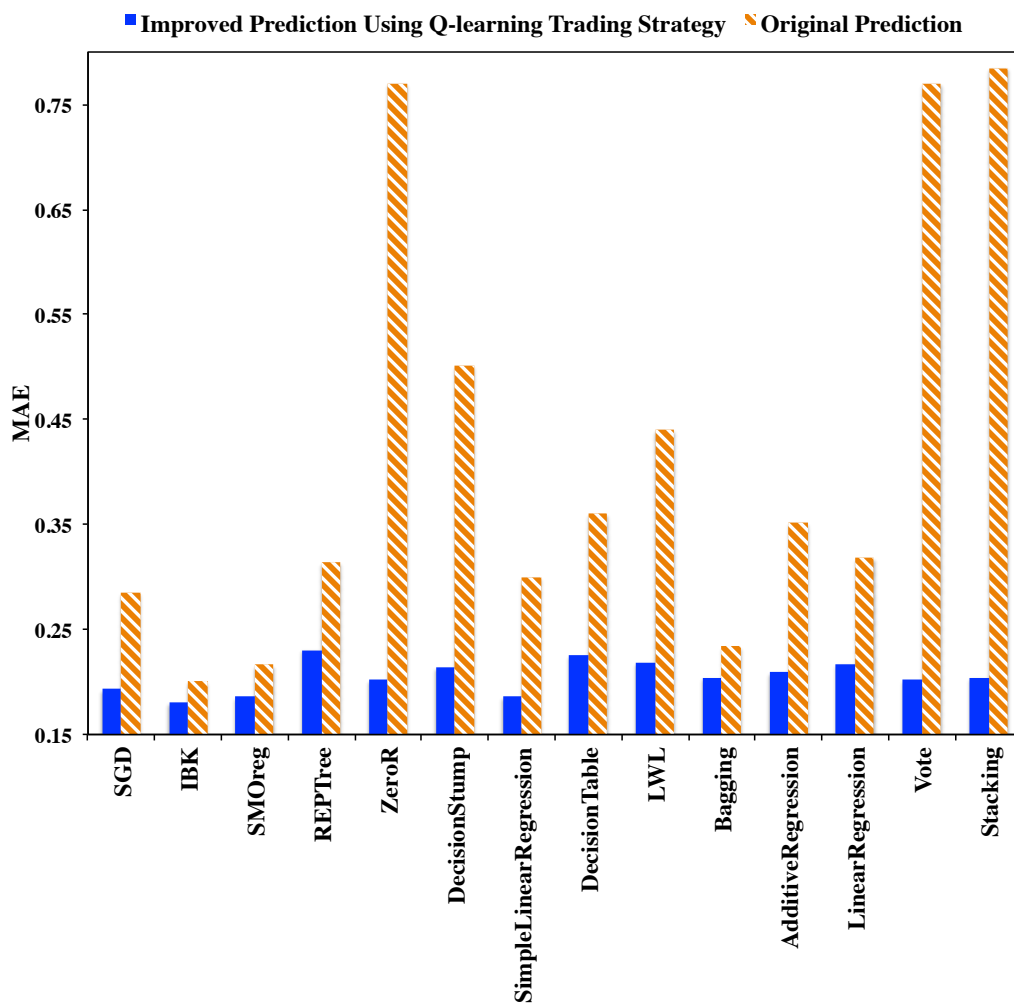


Figure 4-5: Experiment A5. How participating in ACPM and utilising Q-learning strategy improves the performance of each classifier.

This experiment tests Hypothesis A5, which investigates whether each agent performance can be improved by adopting the Q-learning trading strategy compared to adopting the constant trading strategy.

Figure 4-5 compares the difference of MAE between models if being used by an agent who is employing Q-learning or constant strategy. The performance of agents using the constant strategy is equal to performance of their assigned models which are run independently without the concept of ACPM. Therefore, the graph also shows how much each model performance is improved by participating in ACPM.

Figure 4-5 demonstrates that the performance of every analysis model is improved by participating in the market and using the Q-learning trading strategy. As can be seen from the figure, the amount of improvement is larger with low performing agents such as ZeroR, Vote and Stacking than the high performing agents such as IBk and SMOreg.

4.5.6 Experiment A6

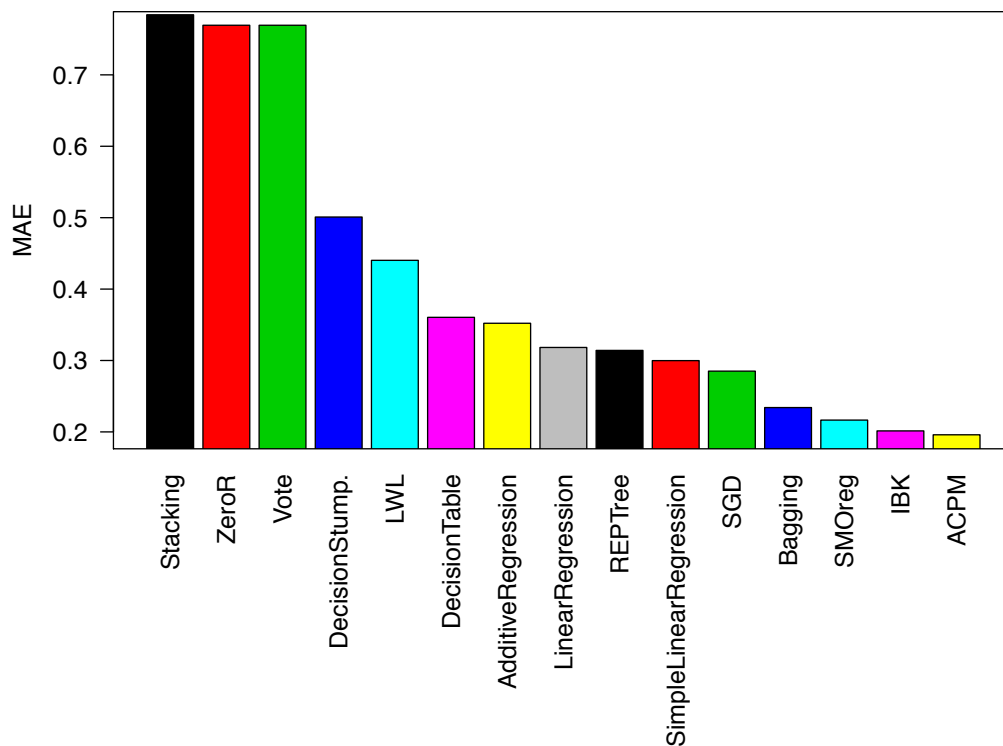


Figure 4-6: Experiment A6. Comparing ACPM performance with well known machine learning models.

This experiment is designed to test Hypothesis A6, which states ACPM outperforms well-known regression and ensemble models. The experiment compares the performance of ACPM with well-known regression models. In this experiment, ACPM has one agent for each of the models presented in Figure 4-6 and the data source of each agent is the entire data set. Then the performance of ACPM is compared with the performance of each of these models as the benchmarks. The benchmarks use the same data as ACPM agents do, and similar to ACPM are run incrementally, which means for each available record, they predict the true value and then are retrained again with the correct answer and all seen records. Therefore, the performance of each benchmark is equal to the performance of the agent uses the same model.

Figure 4-6 compares ACPM’s MAE with the MAE of well-known regression models and ensemble methods. The figure shows that ACPM has a lower MAE than all the models. The p-value is less than $5.19E-10$ for all except IBK (p-value= 0.31) and SMOReg (p-value= 0.17).

4.5.7 Experiment A7

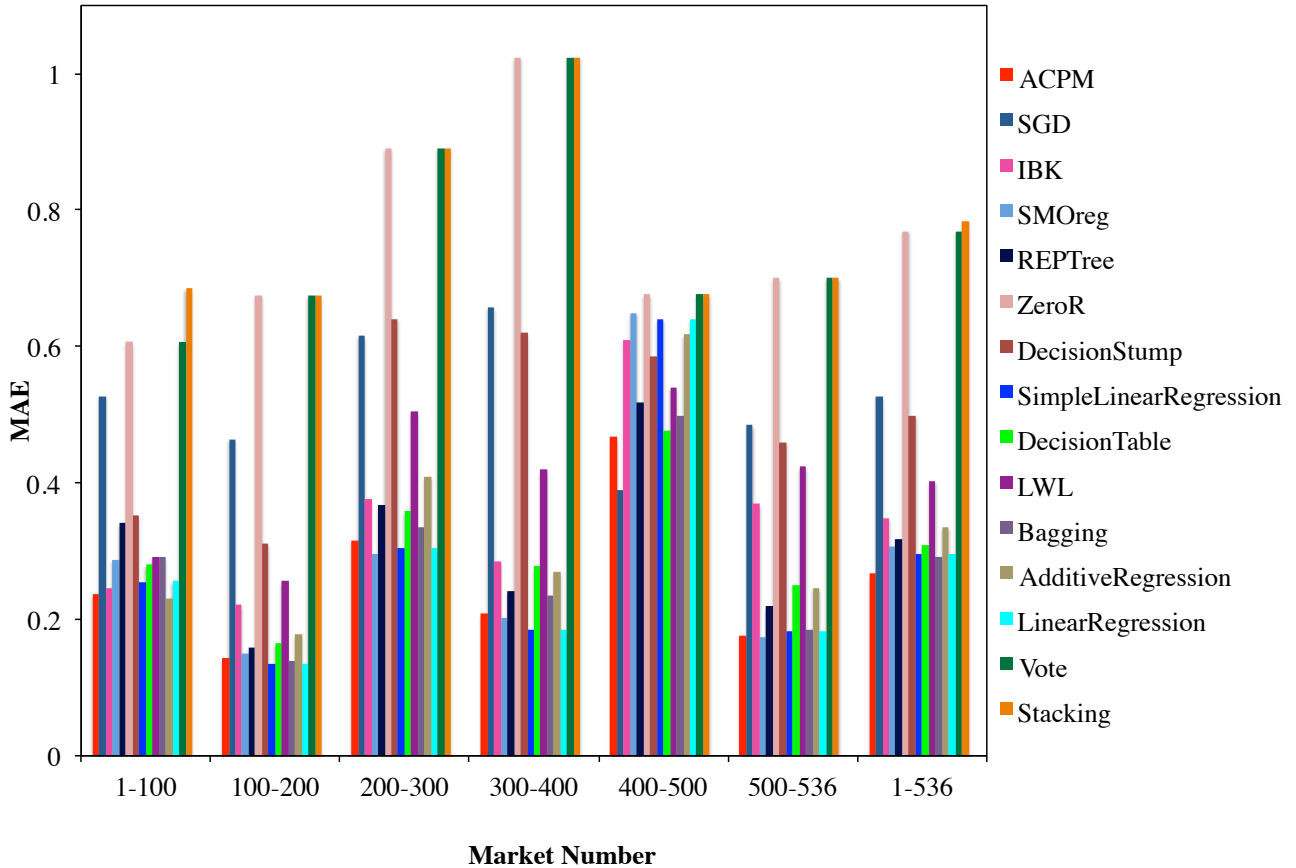


Figure 4-7: Experiment A7. Performance comparison of ACPM against high quality data participants, over a range of time frames.

This experiment tests Hypothesis A7. One of the main benefits of ACPM is that it shifts its focus between agents once their quality vary. Therefore, Hypothesis A7 examines this claim by creating an agent, for each of the models listed in the Figure 4-7. All agents have the same single data source whose quality varies over time. Therefore, the performance of each model and hence the performance of each agent changes from time to time.

Figure 4-7 presents the performance of ACPM and agents in different periods of time (i.e every 100 markets) as shown in x-axes of the figure. The figure demonstrates that ACPM performance shifts focus in response to changes in quality of individual agent performance. As can be seen from the graph, different agents outperform the others in each period. For instance, IBK agent has relatively very low MAE in Market number 1-100, but has relatively high MAE in market numbers 400-500. The graph shows that ACPM is close to the best performing agent performance in each period, as expected. ACPM performance cannot always achieve

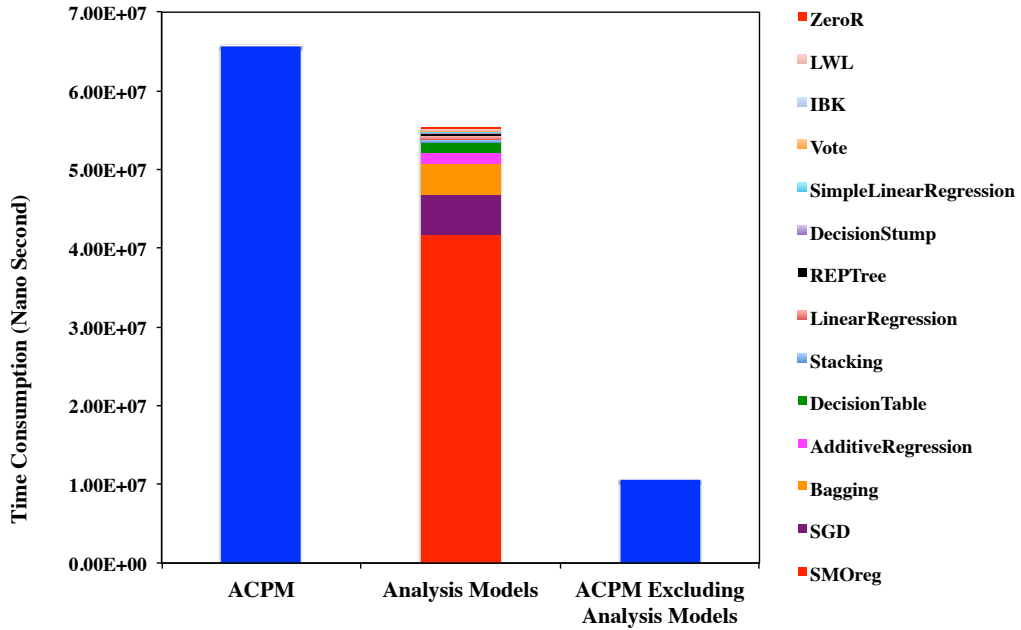


Figure 4-8: Experiment A8. Demonstrating the average time consumption for each market in ACPM. The y-axis shows the consumed time in nano seconds.

the highest performance in each period as ACPM requires time to adapt to the environmental changes. However, when compared across all market numbers (market numbers 1-536), ACPM outperforms all agents, as shown by the final cluster of results.

4.5.8 Experiment A8

This experiment tests Hypothesis A8 by investigating the time taken by ACPM. The experiment compares the time taken by ACPM including and excluding time used by the agents' analysis models. In this experiment, ACPM includes 14 agents, each having a distinct analysis model but having the same data set.

Figure 4-8 demonstrates the average time taken for each market (i.e per record). The y-axis shows the time taken in nano seconds. The first bar, from left side of the figure, represents the amount of time which ACPM takes for each market on average. The second bar demonstrates the average time only taken by the models in each market, clustered into different colours to show time taken by each analysis model. The third bar shows the average time taken by ACPM for each market, exclusive of the time used by the analysis models. With regards to the last case, ACPM takes approximately $5.60E + 06$ nano seconds for including the first agent to the system and $3.50E + 05$ nano second for each additional agents, per record¹⁷.

As Figure 4-8 shows the analysis models used 84% of the total time taken of the entire simulation, where 76% of this time is used by SMOreg model, followed by SGD and Bagging

¹⁷We ran this experiment with different number of agents between [1...14] and we calculated the average time taken by ACPM using this formula: $\frac{\text{total time taken for the entire simulation excluding the time used by the analysis models}}{\text{number of records} \times \text{number of agents}}$. We obtained the same results whenever the market had more than one agent

models, each consuming 9% and 7% respectively. In this experiment, the data set used by models is small (has only one attribute), however, as the size of the data set increases the amount of time required by the analysis models increases as well.

4.5.9 Experiment B1

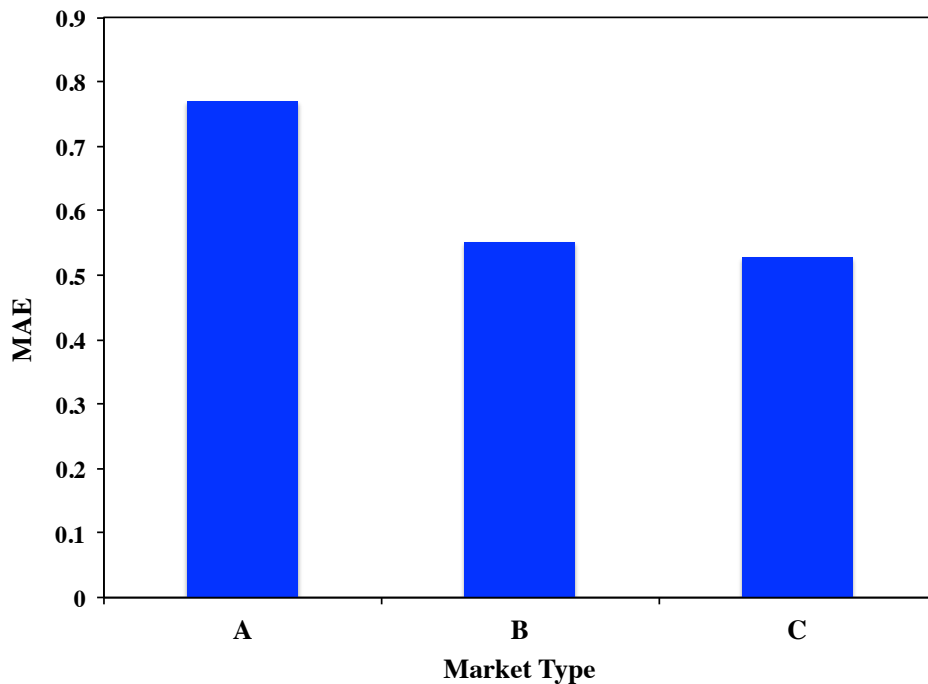


Figure 4-9: Experiment B1. Comparing ACPM performance. Type A: the market has 100 agents with random data. Type B: the market has agents of the type A market and additionally one agent with high quality data. Type C: the market has only the high quality agent.

This experiment tests Hypothesis B1, which investigates how addition of low and high quality agents affects the performance of ACPM. Figure 4-9 compares performance of ACPM in different markets:

- (i) Type A: has 100 agents with very low quality data.
- (ii) Type B: has type A market population and additionally one agent with high quality data (i.e has 101 agents).
- (iii) Type C: has only the high quality agent in market type B (i.e has 1 agent).

The graph shows that market type A has a MAE of 0.77 and types B and C have MAE of 0.55 and 0.53 respectively. The result suggests that addition of one high quality agent to 100 very low quality agents decreases the error by 0.22 (i.e by changing market composition from type A with MAE=0.77 to type B with MAE=0.55), while, the addition of 100 very low quality

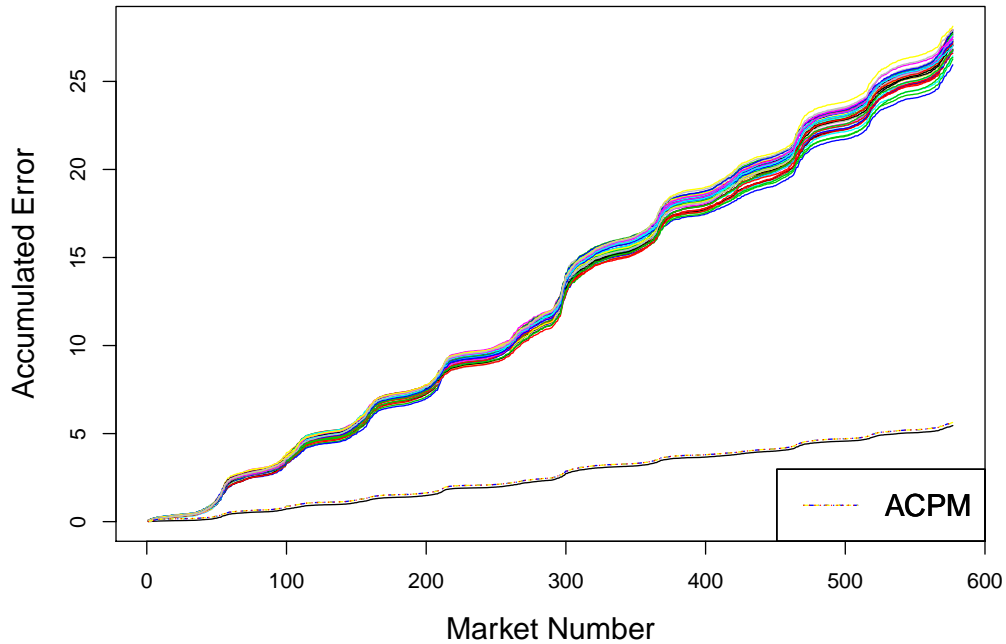


Figure 4-10: Experiment B2. Comparing accumulated error of ACPM and its 40 participants. One agent accesses data with 10% noise and 39 agents access data with 50% noise. Each solid line represents the accumulated error of a participating agent and the multi-coloured line represents the accumulated error of ACPM.

agents to a market which has only one high quality agent only increases the error by 0.02 (i.e changes market composition from type C with MAE=0.53 to type B with MAE=0.55).

4.5.10 Experiment B2

This experiment tests Hypothesis B2, which examines the performance of ACPM in a situation where one of the agents permanently outperforms the other agents.

Figure 4-10 compares the accumulated error of all participating agents and ACPM in this experiment. As can be seen from Figure 4-10, one agent (shown in black line) outperforms every other agent and ACPM (shown in multi-colour line) follows the best performing agent. The graph shows that ACPM accumulated error is slightly higher than the best agent accumulated error and this difference is unchanged until the end of experiment. In this experiment, all agents except the best performing agent have similar performance.

4.5.11 Experiment B3

This experiment tests Hypothesis B3. Hypothesis and Experiment B3 investigate the performance of ACPM when a group of agents with similar quality outperform the rest of agents permanently.

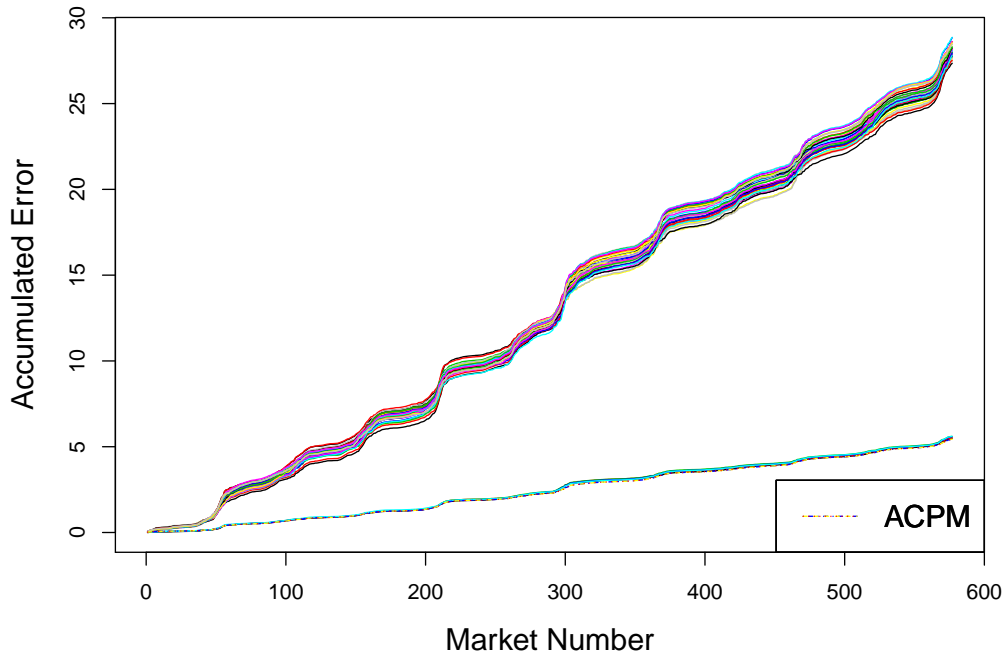


Figure 4-11: Experiment B3. Comparing accumulated error of ACPM and its 40 participants. 5 agents access data with 10% noise and 35 agents access data with 50% noise. Each solid line represents the accumulated error of a participating agent and the multi-coloured line represents the accumulated error of ACPM.

Figure 4-11 compares the accumulated error of all participating agents and ACPM in this experiment. The figure shows that the best performing agents obtains much less error compared to the rest of participants. Although ACPM performance is not equal or better than every best performing agents, but very close to them. In this graph, the accumulated error of ACPM and the best performing agents form almost a single line as their performance is very close.

4.5.12 Experiment B4

This experiment tests Hypothesis B4. Hypothesis and Experiment B4 examine the performance of ACPM when the system has only equally well performing participants.

Figure 4-12 compares the performance of ACPM and its participating agents. As the graph shows ACPM achieves performance better than every single agent in this situation. The figure also shows that, as the number of markets increases, the accumulated error of ACPM reduces in comparison to the error of each participant, i.e the gap between the accumulated error of ACPM and agents increases.

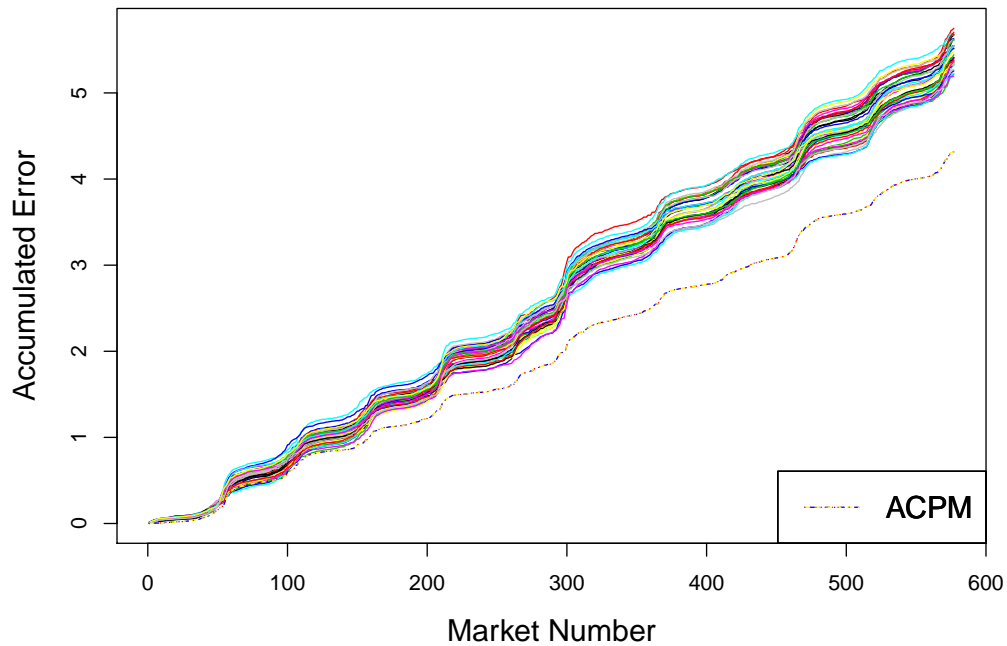


Figure 4-12: Experiment B4. Comparing accumulated error of ACPM and its 40 participants each accessing data with 10% noise. Each solid line represents the accumulated error of a participating agent and the multi-coloured line represents the accumulated error of ACPM.

4.5.13 Experiment B5

This experiment tests Hypothesis B5. Hypothesis and Experiment B5 examine the performance of ACPM when all agents access very poor quality data, i.e random data with uniform distribution.

Figure 4-13 compares the accumulated error of ACPM and its participating agents. As the figure demonstrates ACPM performs poorly when all of its participants are very low performing.

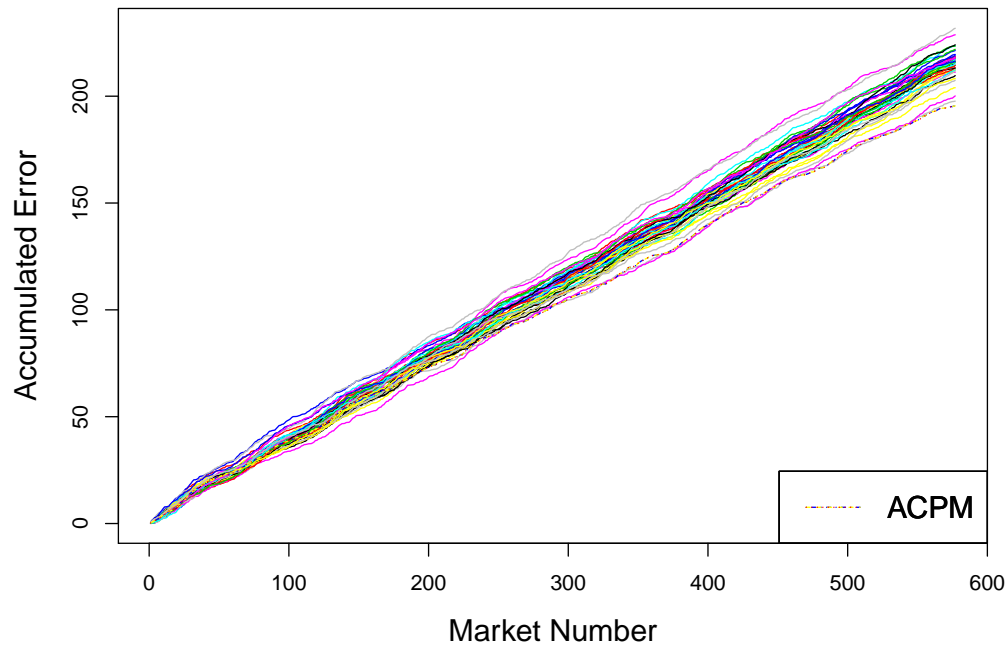


Figure 4-13: Experiment B5. Comparing accumulated error of ACPM and its 40 participants receiving random data with uniform distribution. Each solid line represents the accumulated error of a participating agent and the multi-coloured line represents the accumulated error of ACPM.

4.5.14 Experiment B6

This experiment tests Hypothesis B6. Hypothesis and Experiment B6 examine the performance of ACPM in situations where all agents have very poor quality data (random data) but for each 10 consecutive markets, one distinct agent makes perfect prediction, or in other words it attains zero error. The corresponding results are shown in Figure 4-14.

Figure 4-14 shows that ACPM can perform relatively well in such situations as, at each period, it can automatically adapt to the environment changes by distinguishing the high quality agent and follow that agent.

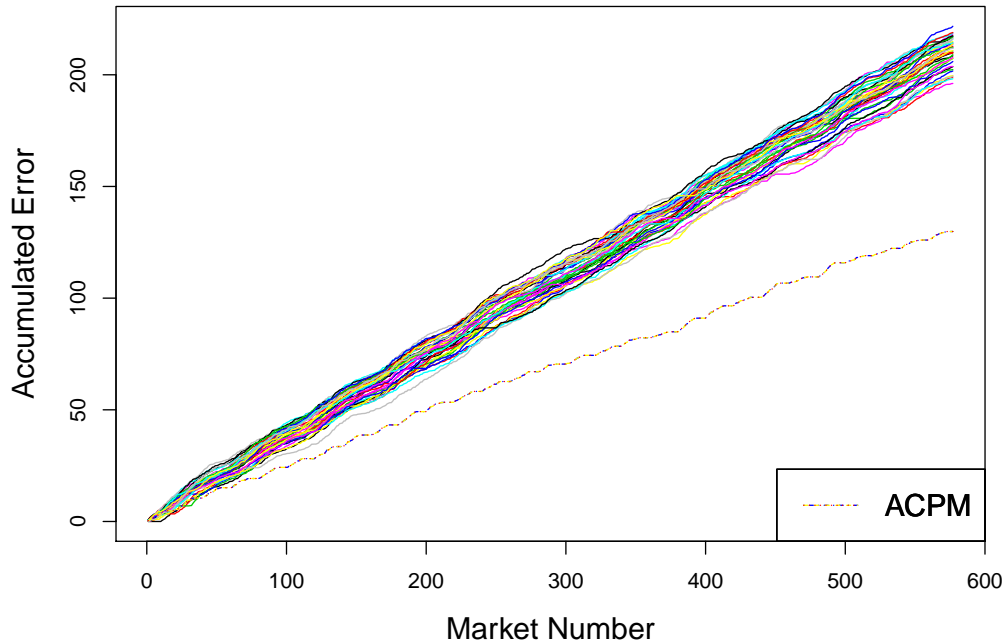


Figure 4-14: Experiment B6. Comparing accumulated error of ACPM and its 40 participants each accessing random data with uniform distribution. For each set of 10 markets, one of the agents has access to the correct answer. Each solid line represents the accumulated error of a participating agent and the multi-coloured line represents the accumulated error of ACPM.

4.5.15 Experiment B7

This experiment tests Hypothesis B7 and examines how ACPM responds in situations where the best performing agent suddenly loses quality or becomes unavailable. Figure 4-15 compares the error of ACPM and its participating agents.

As Figure 4-15 shows, ACPM (shown by a multi-coloured line) recognises the best performing agent (shown by black colour) after the first 10 markets. Then, in market number 100, the quality of the best performing agent (black line) suddenly drops and obtains error equal to 1. This causes ACPM to perform poorly for a few number of markets, however, ACPM recognises the next best performing agent (shown by red line) rapidly and follows that agent. As the figure shows, the same process happens for markets 200, 300..500, where the subsequent best performing agents lose quality¹⁸.

This graph demonstrates the performance of ACPM when the quality of the best performing agent drops, however, the same result is attained when the best performing agent becomes unavailable.

¹⁸We also experimented with other version of this setting. In particular, we created additional 95 agents with error equal to 1 in each record and added to this market. This required 12 and 10 markets for ACPM to learn and recover respectively.

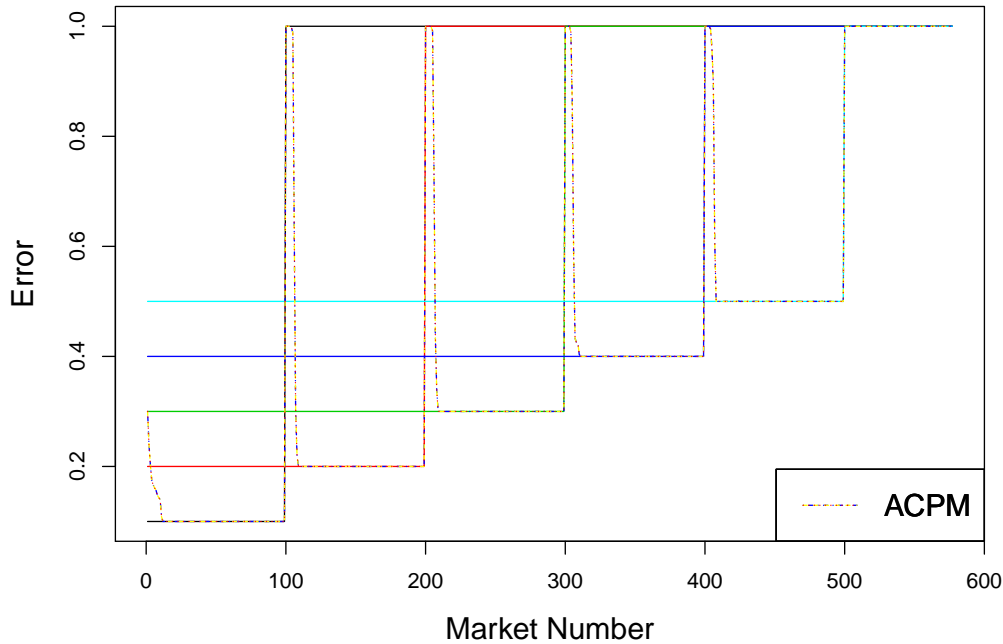


Figure 4-15: Experiment B7 - Comparing error of ACPM and its 5 participants with varying performance. Each solid line represents the accumulated error of a participating agent and the multi-coloured line represents the accumulated error of ACPM.

4.5.16 Experiment C1

This experiment tests Hypothesis C1, which states the performance of ACPM is domain independent. The experiment results are shown in Table 4.3 and 4.4. In particular, Table 4.3 presents the Mean Squared Error (MSE) of every model, listed in the table as benchmarks, and that for each of the two variations of ACPM i) when ACPM agents use the Q-learning trading strategy and ii) when ACPM agents use the constant trading strategy. Appendix E demonstrates these results using two evaluation metrics of MSE and MAE. Furthermore, for each data set, the table highlights the top three models; the best model is highlighted in red, the second in yellow, and the third in grey. Table 4.4 presents paired t-tests that compare the performance of ACPM with Q-learning against that of the alternatives (i.e., the individual models, as well as ACPM with the constant trading strategy). Here, the null hypothesis is that the two performances being compared are not significantly different. Therefore, within a tolerance $\alpha = 0.05$, when $p\text{-value} < 0.05$, ACPM is significantly better than the other models.

First we compare the two variations of ACPM. As can be seen in Table 4.3, ACPM with Q-learning is better than the constant trading strategy for all data sets, with the exception of the Istanbul Stock Exchange, where they both do equally well. This result is statistically significant for eight out of the ten data sets, as shown in Table 4.4. Next, we compare ACPM with Q-learning against the individual models. As can be seen in Table 4.3, ACPM ranks among the top two for all data sets. As shown in Table 4.4, the results are statistically significant in

111 out of 130 cases (i.e., p-value < 0.05).

ACPM achieves the best performance in four data sets (Bike Sharing, Servo, Housing, Computer Hardware), the second best performance in other data sets (Auto MPG, Yacht Hydrodynamics, Istanbul Stock Exchange, Forest Fires, Automobile, Airfoil Self Noise). Unlike ACPM, none of the benchmarks is placed in the top two positions for all data sets. Random Forest (rf), which is shown to be the best classifier in most data sets [Fernández-Delgado et al., 2014], achieves the best performance in three data sets (Yacht Hydrodynamics, Automobile, Airfoil Self Noise) , and the second best in one data set (Housing). ACPM with Q-learning trading strategy is only 9% worse than the best model, averaged over all these data sets, followed by ACPM with constant trading strategy, Random Forest (rf) and Boosted Tree Linear Regression (blackboost) which are 17%, 72% and 143%, respectively.

Part I

Forecasters	Bike Sharing	Auto MPG	Yacht Hydro-dynamics	Istanbul Stock Exchange	Servo
ACPM(Q-learn)	1.74E+06	9.61E+00	9.21E+00	1.47E-04	6.13E-01
ACPM(Constant)	1.76E+06(1.01)	9.77E+00(1.02)	1.14E+01(1.24)	1.47E-04(1.00)	6.16E-01(1.01)
treemap	1.21E+07(6.95)	1.28E+01(1.33)	2.41E+01(2.62)	1.71E-04(1.16)	9.25E-01(1.51)
cforest	1.30E+07(7.47)	1.17E+01(1.22)	2.31E+01(2.51)	1.64E-04(1.12)	9.71E-01(1.58)
rf	1.32E+07(7.59)	9.63E+00(1.00)	6.49E+00(0.71)	1.62E-04(1.10)	6.56E-01(1.07)
mlp	8.14E+06(4.68)	2.06E+01(2.14)	1.12E+02(12.16)	1.75E-04(1.19)	7.21E-01(1.18)
avNNet	2.38E+06(1.37)	5.72E+02(59.52)	3.24E+02(35.18)	2.02E-04(1.37)	2.41E+00(3.93)
glmboost	1.37E+07(7.87)	1.23E+01(1.28)	8.60E+01(9.34)	1.50E-04(1.02)	1.36E+00(2.22)
blackboost	1.34E+07(7.70)	1.16E+01(1.21)	2.30E+01(2.50)	1.66E-04(1.13)	1.06E+00(1.73)
lm	1.37E+07(7.87)	1.20E+01(1.25)	8.96E+01(9.73)	3.02E-04(2.05)	1.38E+00(2.25)
gaussprLinear	8.84E+06(5.08)	3.01E+01(3.13)	1.03E+02(11.18)	1.55E-04(1.05)	1.40E+00(2.28)
rpart	1.11E+07(6.38)	2.51E+01(2.61)	3.78E+01(4.10)	2.08E-04(1.41)	1.17E+00(1.91)
glm	1.37E+07(7.87)	1.20E+01(1.25)	8.96E+01(9.73)	3.02E-04(2.05)	1.38E+00(2.25)
knn	1.27E+07(7.3)	1.22E+01(1.27)	1.55E+02(16.83)	1.65E-04(1.12)	1.33E+00(2.17)
gaussprPoly	1.36E+07(7.82)	8.61E+00(0.90)	1.17E+02(12.70)	1.46E-04(0.99)	8.18E-01(1.33)

Part II

Forecasters	Forest Fires	Automobile	Housing	Airfoil Self Noise	Computer Hardware
ACPM(Q-learn)	4.12E+03	1.05E+07	1.56E+01	8.79E+00	5.08E+03
ACPM(Constant)	4.14E+03(1.01)	1.21E+07(1.15)	1.67E+01(1.07)	9.18E+00(1.04)	5.67E+03(1.12)
treemap	4.42E+03(1.07)	1.87E+07(1.78)	2.34E+01(1.50)	1.76E+01(20.00)	1.38E+04(2.72)
cforest	4.11E+03(1.00)	2.18E+07(2.08)	2.05E+01(1.31)	1.42E+01(1.62)	1.38E+04(2.72)
rf	4.43E+03(1.08)	9.42E+06(0.90)	1.59E+01(1.02)	6.96E+00(0.79)	6.07E+03(1.20)
mlp	4.64E+03(1.13)	1.02E+08(9.71)	5.31E+01(3.40)	3.68E+02(41.87)	2.62E+04(5.16)
avNNet	4.22E+03(1.02)	2.37E+08(22.57)	5.45E+02(34.94)	1.54E+04(1751.9)	3.78E+04(7.44)
glmboost	4.19E+03(1.02)	1.35E+07(1.29)	2.77E+01(1.78)	2.40E+01(2.73)	5.21E+03(1.03)
blackboost	4.13E+03(1.00)	1.75E+07(1.67)	1.98E+01(1.27)	1.31E+01(1.49)	1.44E+04(2.84)
lm	4.48E+03(1.09)	6.83E+09(650.48)	8.39E+01(5.38)	2.37E+01(2.7)	7.70E+03(1.52)
gaussprLinear	4.43E+03(1.08)	9.27E+07(8.83)	1.28E+02(8.21)	1.85E+03(210.4)	5.26E+03(1.04)
rpart	4.13E+03(1.00)	2.52E+07(2.4)	4.01E+01(2.57)	3.19E+01(3.63)	1.74E+04(3.43)
glm	4.48E+03(1.09)	6.83E+09(650.48)	8.39E+01(5.38)	2.37E+01(2.7)	7.70E+03(1.52)
knn	4.59E+03(1.11)	1.74E+07(1.66)	2.77E+01(1.78)	1.92E+01(2.18)	1.39E+04(2.74)
gaussprPoly	4.11E+03(1.00)	1.15E+07(1.1)	2.62E+01(1.68)	1.75E+01(1.99)	6.70E+03(1.32)

Table 4.3: Experiment C1 - MSE of ACPM and Benchmarks. ACPM outperforms most benchmarks for many data sets. The value in the parentheses gives the ratio by which ACPM performs better. The best, second best and third best models are highlighted by red, yellow and grey colour respectively.

Part I

Forecasters	Bike Sharing	Auto MPG	Yacht Hydro-dynamics	Istanbul Stock Exchange	Servo
ACPM(Constant)	7.75E-03	3.26E-02	1.81E-07	9.22E-01	3.22E-01
treebag	1.34E-200	1.60E-11	3.26E-21	8.71E-05	4.54E-04
cforest	2.89E-205	7.23E-08	1.57E-10	8.65E-03	5.71E-04
rf	1.12E-201	8.59E-01	9.96E-01	9.30E-03	8.13E-01
mlp	1.51E-90	7.17E-15	3.99E-28	1.16E-05	2.57E-01
avNNet	7.75E-21	2.48E-192	1.27E-24	2.52E-06	6.54E-05
glmboost	2.33E-196	2.61E-06	4.76E-61	6.54E-01	6.08E-17
blackboost	9.36E-200	4.48E-06	4.61E-12	3.60E-03	1.58E-07
lm	2.78E-196	1.28E-05	7.70E-66	2.77E-02	1.03E-16
gaussprLinear	1.09E-118	8.30E-11	6.87E-60	4.48E-01	9.00E-18
rpart	8.14E-201	1.41E-31	8.79E-42	3.47E-11	6.86E-05
glm	2.78E-196	1.28E-05	7.70E-66	2.77E-02	1.03E-16
knn	2.09E-212	1.74E-06	2.14E-25	2.27E-03	4.37E-05
gaussprPoly	1.93E-196	6.89E-01	2.57E-07	8.68E-01	2.97E-07

Part II

Forecasters	Forest Fires	Automobile	Housing	Airfoil Self Noise	Computer Hardware
ACPM(Constant)	1.06E-07	8.10E-03	1.96E-02	7.25E-08	5.84E-03
treebag	3.20E-12	6.76E-05	5.72E-14	3.79E-81	1.10E-05
cforest	3.16E-13	2.63E-08	3.78E-07	4.25E-55	4.16E-05
rf	8.93E-15	9.62E-01	5.68E-02	1.00E+00	3.65E-01
mlp	4.19E-03	1.64E-22	5.12E-39	1.83E-57	2.73E-15
avNNet	1.00E+00	8.80E-56	1.33E-185	0.00E+00	4.40E-15
glmboost	3.62E-15	5.24E-03	4.89E-11	8.07E-98	6.37E-02
blackboost	1.82E-27	2.55E-04	3.63E-07	9.22E-44	2.25E-06
lm	3.92E-30	2.39E-02	2.17E-05	1.41E-94	2.68E-05
gaussprLinear	4.08E-30	1.29E-33	2.60E-43	8.85E-55	7.06E-03
rpart	5.49E-26	7.76E-10	2.28E-28	5.34E-136	2.93E-09
glm	3.92E-30	2.39E-02	2.17E-05	1.41E-94	2.68E-05
knn	3.39E-05	2.06E-04	3.76E-09	1.46E-99	8.10E-04
gaussprPoly	2.74E-17	8.35E-02	1.35E-01	1.61E-42	7.84E-02

Table 4.4: Experiment C1 - The p-values of paired t-test for ACPM and benchmarks.

Part I

Forecasters	Bike Sharing	Auto MPG	Yacht Hydro-dynamics	Istanbul Stock Exchange	Servo
ACPM(Q-learn)	1.74E+06	9.61E+00	9.21E+00	1.47E-04	6.13E-01
EWAf	9.34E+06(5.38)	1.25E+01(1.3)	1.86E+01(2.02)	1.51E-04(1.03)	8.80E-01(1.44)
TBE	9.34E+06(5.37)	1.31E+01(1.37)	4.25E+01(4.62)	1.46E-04(0.99*)	8.25E-01(1.35)
FBE	2.38E+06(1.37)	8.75E+00(0.91*)	7.43E+00(0.81*)	1.50E-04(1.02)	7.42E-01(1.21)
EG	1.11E+05(0.06*)	9.77E+00(1.02)	1.48E+01(1.61)	1.49E-04(1.01)	6.43E-01(1.05)

Part II

Forecasters	Forest Fires	Automobile	Housing	Airfoil Self Noise	Computer Hardware
ACPM(Q-learn)	4.12E+03	1.05E+07	1.56E+01	8.79E+00	5.08E+03
EWAf	4.13E+03(1)	1.49E+08(14.23)	2.31E+01(1.48)	7.61E+00(0.87*)	8.25E-01(1.35)
TBE	4.14E+03(1)	1.49E+08(14.24)	2.00E+01(1.28)	8.97E+01(10.2)	8.87E+03(1.75)
FBE	4.19E+03(1.02)	1.12E+07(1.07)	2.07E+01(1.33)	7.00E+00(0.80*)	6.33E+03(1.25)
EG	4.35E+03(1.06)	5.52E+07(5.26)	1.78E+01(1.14)	1.45E+01(1.65)	6.74E+03(1.33)

Table 4.5: Experiment C2 - MSE of ACPM and Forecasters. ACPM outperforms most forecasters in each data set, except a few cases which are marked by *.

4.5.17 Experiment C2

This experiment tests Hypothesis C2, which states ACPM outperforms well-known forecasters from the literature of prediction with expert advice. Table 4.5 compares ACPM’s Mean Squared Error (MSE) with the MSE of four well-known forecasters from the literature of Prediction with Expert Advice (PEA), namely: (i) Exponentially Weighted Average Forecaster (EWAf), (ii) Tracking the Best Expert (TBE), (iii) Following the Best Expert (FBE), and (iv) Exponentiated Gradient algorithm (EG).

Table 4.5 presents the Mean Squared Error (MSE) of ACPM with Q-learning, as well as the MSE of each of these forecasters. As can be seen from the table, ACPM outperforms the majority of forecasters in every data set and ACPM is outperformed only in 6 cases out of 40 (marked *). In particular, ACPM is the best model in 5 data sets (Servo, Forest Fires, Automobile, Housing, Computer Hardware) and the second best in the 4 data sets (Bike Sharing, Auto MPG, Yacht Hydrodynamics, Istanbul Stock Exchange) and the third in Airfoil Self Noise data set. Table 4.6 presents the paired t-tests comparing ACPM against the PEA forecasters. As can be seen, the results are statistically significant in 25 out of 40 cases (i.e p-value<0.05).

4.6 Analysis

The main purpose of ACPM is to predict a continuous variable value, by integrating a range of data sources and aggregating the results of different analysis models. In ACPM, each agent plays the role of a market participant and represents an analysis model and a data source. Agents are initialised with an equal amount of artificial currency at the start of the experiment. ACPM

Part I

Forecasters	Bike Sharing	Auto MPG	Yacht Hydrodynamics	Istanbul Stock Exchange	Servo
EWAF	8.76E-169	1.35E-24	1.27E-01	3.84E-01	6.28E-06
TBE	8.76E-169	1.84E-20	2.42E-23	8.89E-01	4.33E-07
FBE	1.40E-21	5.64E-01	8.47E-01	6.70E-01	1.32E-02
EG	1.00E+00	9.67E-02	7.70E-08	5.86E-01	7.37E-01

Part II

Forecasters	Forest Fires	Automobile	Housing	airfoil self noise	Computer Hardware
EWAF	2.69E-08	3.67E-03	5.42E-01	1.00E+00	8.00E-03
TBE	2.86E-10	3.67E-03	1.20E-03	2.50E-281	8.00E-03
FBE	1.99E-13	5.59E-01	4.18E-01	1.00E+00	1.04E-03
EG	3.60E-07	1.21E-02	2.75E-01	6.16E-66	6.82E-02

Table 4.6: Experiment C2 - The p-values of paired t-test for ACPM and PEA models.

establishes a prediction market for each available record to predict the true value of the record by integrating the prediction of participating agents through market mechanism.

We designed a number of hypotheses and performed various experiments to test them. A number of our hypotheses are satisfied immediately from our experiments. Figure 4-1 demonstrates that ACPM performance is higher than the average performance of all participating agents in different market compositions and hence satisfies Hypothesis A1. This happens due to the ability of ACPM to identify the best performing agents in the market and relying on their predictions. Figure 4-2 shows that MAE of ACPM is lower than the best performing agent in different market types (Hypothesis A2). As can be seen from the figure, ACPM achieves higher performance than the best performing agent in market types 2 and 4, but not market types 1 and 3. The reason is that market types 2 and 4 have high quality agents which usually outperform the other agents, but market type 1 and 3 have only medium and/or low quality agents. To further test the validity of this hypothesis in different cases, Experiments B2 to B7 are performed.

Figure 4-10 shows that in situations where one agent permanently outperforms all other agents, ACPM accumulated error is slightly higher than that of the best agent (rejecting Hypothesis B2). In such situations, ACPM recognises the best agent and follows its prediction. Since ACPM requires a number of markets to distinguish that agent, ACPM performance is slightly worse than the best performing agent in such cases.

Experiment B3 investigates the performance of ACPM when a number of agents with similar quality permanently outperform the other agents. As Figure 4-11 shows, ACPM performance is not better than the best performing agents in such situations (rejecting Hypothesis B3), but very close to the best performing agents. As mentioned earlier, ACPM requires a number of markets to distinguish and follow these outstanding agents, therefore ACPM may obtain larger

error compared to the agents who permanently outperform the rest of population.

Figure 4-12 confirms that ACPM achieves performance better than every single agent when ACPM includes equally well performing participants (satisfying Hypothesis B4). Accordingly, this behaviour happens when there is diversity among the agents, similar to the success criteria of ensembles [Rosen, 1996] and real-world prediction markets [Ray, 2006].

According to the results of Experiment B5 which is shown by Figure 4-13, ACPM can not perform well when all its participating agents perform poorly (rejecting Hypothesis B5), as expected. In these types of markets, an agent performing well at one market does not necessarily perform well in the subsequent market, and hence ACPM keeps switching weights among the participants. Therefore, ACPM keeps following different agents in different markets and consequently ACPM performs poorly similar to its participating agents.

Experiment B6 extends Experiment B5 by choosing one of the participating agents randomly for a period of time to access the correct answer and hence the agent achieves zero error for that period. Figures 4-14 shows that in such situations, ACPM can perform better than every single agent (satisfying Hypothesis B6) as ACPM can automatically adapt to the new environment at each period and distinguish the high quality agent and follow that agent.

Figure 4-15 demonstrates that in situations where one agent permanently outperforms other agents, ACPM can recognise it in a few (ca. 10) markets and then follow that best agent. However, once the performance of this leading agent decreases or become unavailable, ACPM quickly recovers by singling out and following the next best agent (satisfying Hypothesis B7).

The system attains its high performance by dynamically weighting the prediction of each agent to form the system prediction. At the beginning of an experiment, all agents have an equal amount of budget and hence the same level of influence. Over time, high quality agents – either because of an effective analysis model or accessing a high quality data source– accumulate more budget than poorly performing agents, as the high quality agents gain revenue and low quality agents lose a proportion of their budget in each market as a result of their performance. The larger the budget of an agent, the larger investment it can make on its prediction. Subsequently the agents with larger budget have larger influence in the market prediction as the integration function weights each prediction by the amount of investment. The reward function rewards the market participants according to their prediction accuracy and the amount invested. Obviously, lower error and higher investment leads to higher revenue. In this way, agents are incentivised to make accurate predictions.

The other reason for high performance of ACPM is due to the Q-learning trading strategy. It is desirable that high quality agents preserve their original prediction (i.e choose the “PreservePr” action) and not alter it with what the crowd believes in, but low quality agents change their bids according to wisdom of the crowd (i.e choose the “ChangePr” action). In this way, high quality agents lead the market, and low quality agents follow them and hence the negative effect of low quality agents on market prediction is minimised. Figure 4-3 shows that while “PreservePr” action is the most popular action within high quality agents, “ChangePr” action is the most popular action for low quality agents (satisfying Hypothesis A3). The figure also shows that high quality agents choose the “ChangePr” action and low quality agents choose “PreservePr” action occasionally, because of change in their quality in different period of time.

Accordingly, agents using the Q-learning trading strategy learn to improve their prediction by considering the market prediction as another source of information. Figures 4-4 and 4-5 show that Q-learning does improve each agent's performance and consequently the system's performance (satisfying Hypothesis A4 and Hypothesis A5).

Since each agent in ACPM represents an analysis model and the performance of ACPM is always better or very close to its best performing agents, ACPM can easily outperform many analysis models including well-known regression and ensemble methods (satisfying Hypothesis A6). This finding is also verified based on Experiment A6 and further by the results of Experiment C1.

The other main capability of ACPM is its adaptation to the dynamic environment where the quality of an agent's data source can fluctuate over time. Once the quality of an agent prediction changes – either through loss or gain in data quality or the effectiveness of the analysis model for that data source, or both – its performance in the market is affected. Therefore, the agent's influence on the market prediction is tuned according to its quality, as discussed above.

Figure 4-7 demonstrates ACPM performance is close to the best performing agent in each period, hence we can infer ACPM shifts focus in response to changes in quality of individual predictions (satisfying Hypothesis A7). The graph shows that different models outperform the others in each period as the quality of their data source changes over time and accordingly the suitability of the models on the particular data source change as well. As expected, ACPM performance cannot always be equal to the best agent performance in each period as ACPM requires several markets to adapt to the environmental changes. However, ACPM performance is close to the best performing agent performance in each period. In conclusion, this result suggests that ACPM is a useful technique for integrating the result of different analysis models for data whose quality is not consistent over time.

Experiment A8 shows that ACPM approximately consumes $5.60E+06$ nano seconds for including the first agent to the system and $3.50E+05$ nano second for each additional agents, excluding the time consumed by the analysis models. Hence, ACPM consumes less than 2 minutes on top of the time required by the analysis models to have 50 agents (i.e 50 analysis models) and complete a data set with 5000 records (satisfying Hypothesis A8).

The results of Experiment C1 demonstrate that the performance of ACPM is either better or very close to the best performing agents in different application domains (satisfying Hypothesis C1). According to the No-Free-Lunch theorem [Wolpert, 1996], the best model will not be the same for all the data sets. Given that in online learning no information about the future is available, choosing the best model in advance is not possible. Furthermore, if some models outperform the other models for a particular number of records, it does not necessarily imply that the model will perform well for the entire unseen data, as at some point the nature of data may change and the model may not be able to maintain its high performance. In such cases, ACPM can dynamically distinguish and rely on current high performing agents.

Given that ACPM performance is either better or very close to its best performing agent performance, it is legitimate to accept that ACPM is resilient to addition of low quality agents (satisfying Hypothesis B1). This is also demonstrated by Figure 4-9 where we can see that addition of 100 low quality agents to the market containing one high quality participant only

increases the error by 0.02 while addition of one high quality agent to the market populated with 100 low quality agents decreases the error by 0.22.

Experiment C2 compares the performance of ACPM with that of well-known forecasters from the literature of prediction with expert advice. Looking at the results, one can see that certain forecasters may perform better than ACPM for certain data sets. However, the *overall* performance (taking all data sets into consideration) of ACPM with Q-learning is better than that of any forecaster (satisfying Hypothesis C2). This is particularly valuable in the context of online learning, where it is often not possible to choose the best model in advance, simply because we are dealing with previously-unseen data.

Chapter 5

Syndromic Surveillance

Appearance of highly virulent viruses warrant early detection of outbreaks to protect community health. The main goal of public health surveillance and more specifically ‘*syndromic surveillance systems*’ is early detection of an outbreak in a society using available data sources.

In this chapter, we discuss what are the challenges of syndromic surveillance systems and how ACPM can effectively be applied to the problem of syndromic surveillance. Section 5.1 provides an introduction and explains syndromic surveillance. Then, we discuss the syndromic surveillance data sources in Section 5.2 and present some syndromic surveillance systems in Section 5.3. The statement of the problem in this field is covered in Section 5.4. After that, we discuss Google Flu Trends (GFT) and GP model [Lampos et al., 2015], which is proposed by Google Flu Trends team to improve GFT engine performance, in Section 5.5 and 5.6 respectively. Also, in these sections, we evaluate the performance of ACPM as a syndromic surveillance system. Finally, Section 5.7 provides the conclusion of this chapter.

5.1 Introduction

According to the World Health Organisation (WHO) [World Health Organization, 2013], the United Nations directing and coordinating health authority, *public health surveillance* is:

The continuous, systematic collection, analysis and interpretation of health-related data needed for the planning, implementation, and evaluation of public health practice.

Public health surveillance practice has evolved over time. Although it was limited to pen and paper at the beginning of 20th century, it is now facilitated by huge advances in informatics. Information technology enhancements have changed the traditional approaches of capturing, storing, sharing and analysing of data and resulted efficient and reliable health surveillance techniques [Lombardo and Buckeridge, 2007]. The main objective and challenge of a health surveillance system is the earliest possible detection of a disease outbreak within a society for the purpose of protecting community health.

In the past, before the widespread deployment of computers, health surveillance was based

on reports received from medical care centres and laboratories. Although they are very specific¹, they decrease the timeliness and sensitivity² of a surveillance system [Lombardo and Buckeridge, 2007], while prevention of mortality of infected people for some diseases requires rapid identification and treatment. Clearly, the earlier a health threat within a population is detected, the lower the morbidity and the higher number of the saved lives. Consequently, syndromic surveillance systems have been created to monitor indirect signals of disease activity such as call volume to telephone triage advice lines and over-the-counter drug sales to provide faster detection [Ginsberg et al., 2008].

Syndromic Surveillance is an alternative to the traditional health surveillance system, which mainly depends on confirmed diagnoses, and aim to detect an outbreak as early as possible. Syndromic surveillance refers to techniques relying on population health indicators which are apparent before confirmatory diagnostic tests become available [Mandl et al., 2004]. Syndromic surveillance systems mostly concentrate on infectious diseases such as severe acute respiratory syndrome (SARS), anthrax and influenza. In order to decide whether an outbreak is evolving, syndromic surveillance systems monitor the quantity of patients with similar syndromes since indicators of a disease appear.

Syndromic surveillance aims to exploit information which is not primarily generated for the purpose of public health, but can be an indicator of an abnormal health event. Syndromic surveillance data sources include, but are not limited to, coding of diagnoses at admission to or discharge from emergency departments, confirmatory diagnostic cases, medical encounter pre-diagnostic data, absentee rates at schools and workplaces, over-the-counter pharmacy sales and posts on social media. Each of these data sources can generate a signal during disease development. Figure 5-1 shows the timeline of different data sources to detect an outbreak. The following section describes some of the syndromic surveillance data sources in more details.

5.2 Syndromic Surveillance Data Sources

Syndromic surveillance data sources should supply timely and pre-diagnosis health indicators. Most of this data is originally collected for other purposes and now serves a dual purpose [Chen et al., 2010a]. Syndromic surveillance data sources include:

1. Chief complaint record: These records include signs and symptoms of patient illness from emergency departments (ED) and ambulatory visits to hospitals. These records normally become available on the same day as the patient is seen.
2. Over the counter (OTC) sales: since some people may consider visiting a pharmacy rather than a physician in their early stage of sickness, these data might be more timely. They include detailed information and are available in near real time in electronic format. However, they might be affected by factors such as sales promotions, stockpiling of medicines during a season, and product placement changes in pharmacies.

¹Specificity: the proportion of people without the disease that a test finds negative

²Sensitivity: the proportion of people with the disease that a test finds positive

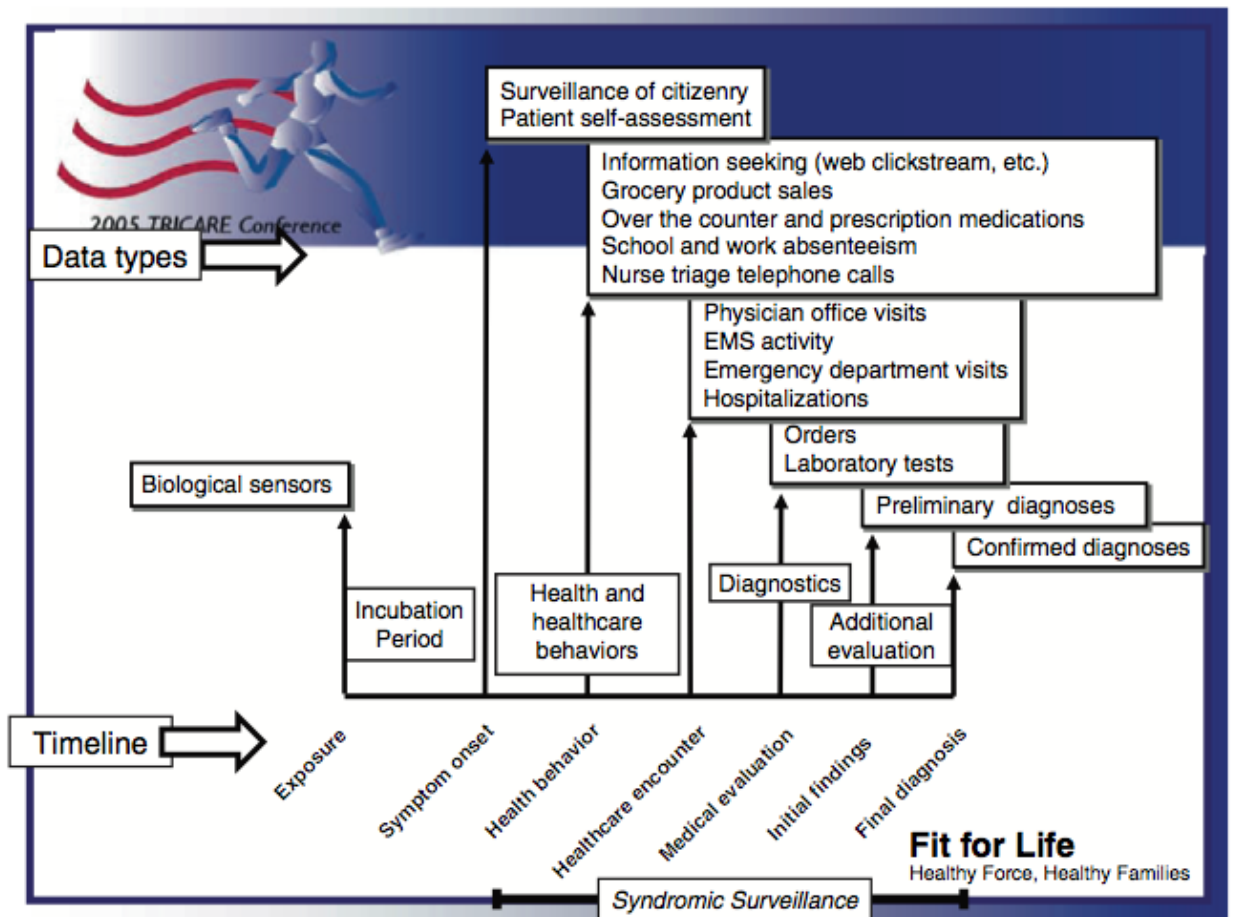


Figure 5-1: Conceptual timeline of pre-diagnosis data types and sources for syndromic surveillance [Chen et al., 2010a].

3. School or work absenteeism: Although absenteeism data seems to have good timeliness, their lack of medical detail complicates interpretation [Van den Wijngaard et al., 2008].
4. Hospital admission records: These data are not sufficiently timely as it might take several days from a patient's first visit until his/her hospitalisation.
5. Pre-diagnostic clinical data: These are indications by an illness before being confirmed via laboratory tests and include comments of health care practitioners, patient encounter information, triage nurse calls, 911 calls and ambulance dispatch calls. They are relatively timely.
6. International Classification of Disease 9th edition (ICD-9) and International Classification of Disease, 9th edition, Clinical Modification (ICD-9-CM): These are widely used in many syndromic surveillance systems due to their electronic format. They are usually generated for billing and insurance reimbursement purposes.
7. Laboratory test orders and results: Although laboratory test results are very reliable, they lack timeliness as they usually take a week to be completed.

8. Emergency Department (ED) diagnostic data: These are regularly available in electronic format but takes several days to be prepared.
9. Internet and open source information: These contain a huge source of health information and can be obtained via discussion forums, social media, government websites, news outlets, blogs, discussion sites, individual search queries, web crawling, use of click stream data, mass media and news report.

For example, some approaches have applied data mining techniques to

- Search engine logs
[Eysenbach, 2006],[Polgreen et al., 2008],[Eysenbach, 2009], [Ginsberg et al., 2009], [Lampos and Cristianini, 2010] and [Lampos et al., 2015]
- Twitter
[Culotta, 2010],[Achrekar et al., 2011],[Signorini et al., 2011],[Culotta, 2013] and [Paul et al., 2014]
- News articles
[Reilly et al., 1968], [Grishman et al., 2002],[Mawudeku and Blench, 2006],[Brownstein et al., 2008], [Collier et al., 2008] and [Linge et al., 2009]
- Web browsing patterns
[Johnson et al., 2004]) and blogs ([Corley et al., 2010])

Figure 5-2, graphs the popularity of various data sources in existing syndromic surveillance systems in the USA. As can be seen from the figure, while emergency department visit reports are widely used in such systems, work absenteeism is the least popular source.

5.3 Existing Syndromic Surveillance Systems

In recent years, a number of syndromic surveillance approaches have been proposed. Roughly 100 syndromic surveillance systems were deployed in the USA done by 2003 [Buehler et al., 2003]. Although they share similar goals, they are different in their system architecture, information processing, analysis algorithms, disease focus, and cover different geographic locations. Chen et al. [2010a] summarises the main international and USA local, state and national syndromic surveillance systems. In Europe, an inventory of syndromic surveillance systems is delivered through a new Public Health Action Programme called Triple-S³ (Syndromic Surveillance Survey, Assessment towards Guidelines for Europe).

The following two sections survey some of the major existing syndromic surveillance systems around the globe. Based on the utilised data sources, we divide the existing syndromic surveillance systems into two categories of i) traditional syndromic surveillance systems, described in Section 5.3.1 and ii) modern syndromic surveillance systems, described in Section 5.3.2.

³ <http://www.syndromicsurveillance.eu/> (Retrieved Oct 4, 2015).

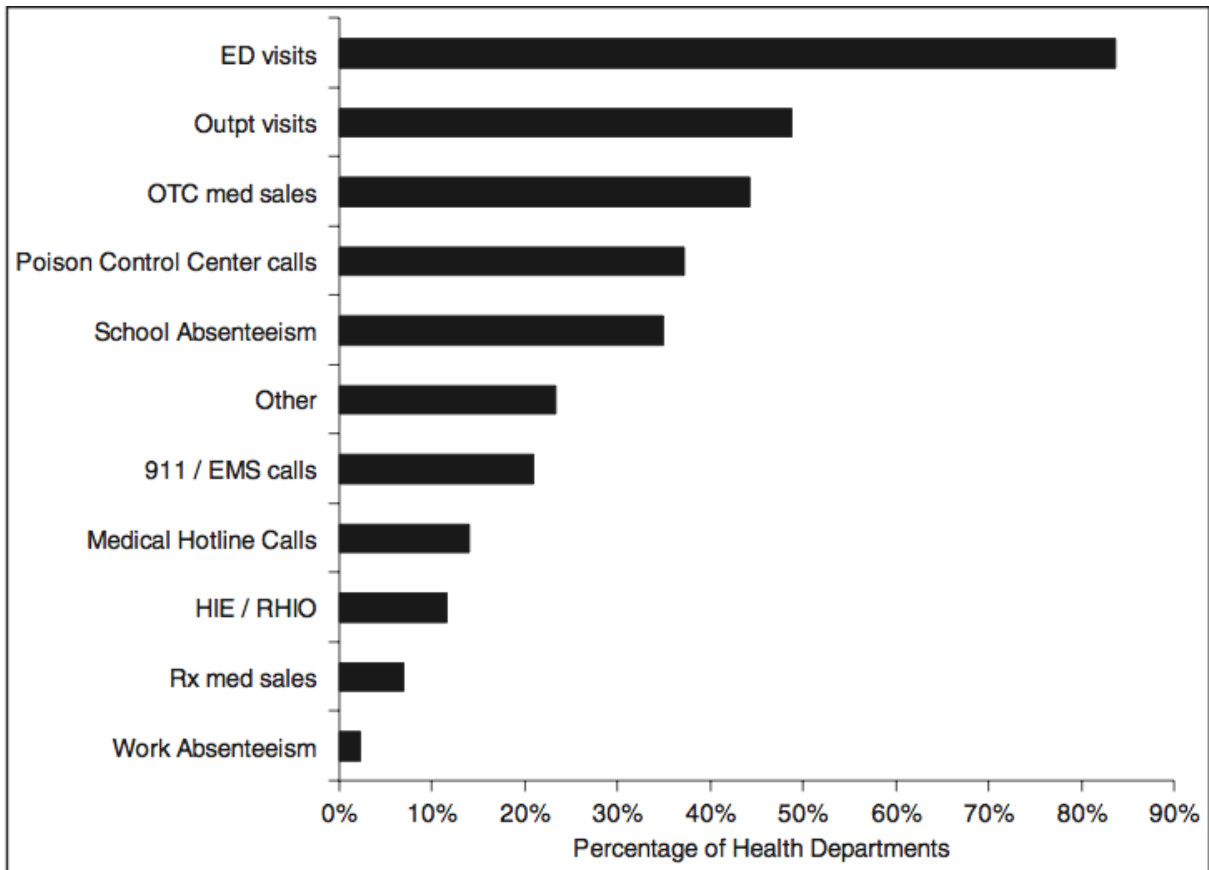


Figure 5-2: Distribution of different data source usage in existing syndromic surveillance systems in the USA [Buehler et al., 2008].

5.3.1 Traditional Syndromic Surveillance Systems

We refer to syndromic surveillance systems that do not utilise social media and internet based data as traditional syndromic surveillance. Some of them are listed below:

1. Early Notification of Community-based Epidemics (ESSENCE) [Lewis et al., 2002] is a syndromic surveillance system in the Washington D.C. area, undertaken by Department of Defense with the primary goal of early detection disease outbreak due to bioterrorism attacks.
2. Real time Outbreak and Disease Surveillance (RODS) [Tsui et al., 2003] is a public health surveillance system, in operation in western Pennsylvania since 1999, developed at the RODS laboratory of the Center for Biomedical Informatics at the University of Pittsburgh.
3. Composite Occupational Health and Operational Risk Tracking (COHORT) [Reichard et al., 2004] delivers real-time surveillance of the medical care of specified groups of military employees worldwide.
4. Syndromic Surveillance Information Collection (SSIC) has been developed by the association of the Clinical Information Research Group at the University of Washington and Public Health-Seattle and King County [Lober et al., 2003].

5. Infectious Disease Surveillance Information System (ISIS) [Widdowson et al., 2003] is an automated outbreak detection system for all types of pathogens in the Netherlands.
6. Early Aberration Reporting System (EARS) is developed by Center for Disease Control (CDC) [Hutwagner et al., 2003] and enables national, state and local health departments to analyse public health surveillance data using a collection of anomaly detection methods.
7. Japan National Institute of Infectious Diseases (NIID) [Ohkusa et al., 2005] has developed syndromic surveillance system to analyse over the counter sales data, outpatient visits, and ambulance transfer data in Tokyo.

We now provide a detailed description of two of the popular traditional syndromic surveillance system, namely BioSense and PHE ReSST.

BioSense

BioSense⁴ is a syndromic surveillance system in the United State which is part of CDC's Public Health Information Network framework. By monitoring the size, location and rate of spread of an outbreak, it detects an outbreak at the local, state and national levels. It monitors seasonal trends for influenza and other disease indicators. BioSense concentrates on syndrome categories including fever, respiratory, gastrointestinal illness (GI), hemorrhagic illness, localised cutaneous lesion, lymphadenitis, neurologic, rash, severe illness and death, specific infection, and botulism.

BioSense collects and shares information on emergency department visits, hospitalisations, clinical laboratory test orders, over-the-counter (OTC) drug sales and other health related data from multiple sources, including the Department of Veterans Affairs (VA), the Department of Defense (DoD), and civilian hospitals from around the USA. BioSense uses multiple analysing methods such as CUSUM [Page, 1954], EWMA [Roberts, 1959] and SMART [Kleinman et al., 2004].

PHE ReSST

The Public Health England (PHE)⁵ Real-time Syndromic Surveillance Team (ReSST) generates regular syndromic surveillance reports by collaborating with numerous national syndromic surveillance systems including the NHS Direct syndromic surveillance system. The NHS Direct syndromic surveillance system monitors the nurse-led telephone helpline data collected electronically by NHS Direct sites and generates alarms when call numbers are considerably higher than preceding years, after considering holiday and seasonal effects. It has the potential to detect large scale events, but is less likely to detect smaller and localised outbreaks [Doroshenko et al., 2005]. In addition, ReSST obtains data from GP In-Hours and GP Out-of-Hours syndromic surveillance systems which monitor daily consultations for a range of clinical syndromic indicators and community-based morbidity, recorded by GP practices inside and outside of routine surgery opening times, respectively.

⁴<http://www.cdc.gov/BioSense/> (Retrieved March 4, 2015).

⁵<http://www.hpa.org.uk/> (Retrieved March 4, 2015).

5.3.2 Internet-Based Syndromic Surveillance Systems

There are other real-time disease event detection systems which employ different approaches from the systems discussed in Section 5.3.1. They monitor online media from global sources, instead of monitoring disease cases reported by health related organisations such as hospitals and clinics. These “systems are built on top of open sources, exemplifying an idea of open development for public health informatics applications” [Chen et al., 2010a]. Though the modern systems are faster than traditional syndromic surveillance systems in detecting an anomaly in public health [Signorini et al., 2011, Ginsberg et al., 2008], they are vulnerable to a high rate of false positives in case of an unusual event within a population [Ginsberg et al., 2008]. This section describes some of the well known modern syndromic surveillance systems.

Google Flu Trends

Google Flu Trends⁶, established by Google, is a Web-based tool for near real-time detection of regional outbreaks of influenza [Ginsberg et al., 2008]. It monitors and analyses health-care seeking behaviour in the form of queries to its online search engine. According to Carneiro and Mylonakis [2009] “all the people searching for influenza-related topics are not ill, but trends emerge when all influenza-related searches are added together”; Consequently, there is a close relationship between the number of people searching for influenza-related topics and those who have influenza symptoms. Section 5.5 provides more information about Google Flu Trends.

Argus

The Argus system is a web-based global biosurveillance system designed to report and track the development of biological events threatening human, plant and animal health globally, excluding the USA [on Homeland Security. Subcommittee on Emerging Threats and Cybersecurity, 2009]. It is developed at Georgetown University and funded by the United States Government.

It automatically collects local and native language internet media reports including blogs and official sources such as World Health Organisation (WHO) and World Organisation for Animal Health (OIE) and infers their importance according to keywords appropriate to infectious disease surveillance [Nelson et al., 2010]. It relies on a human team of multilingual data analysts to assess the relations between the online media and presence of adverse health events [Chen et al., 2010a]. In particular, the data analysts monitor several thousand Internet sources daily. Then, six times in each day, they use Boolean keyword searching and Bayesian model tools [McCallum and Nigam, 1998] to select relevant media reports [Nelson et al., 2010]. Based on the selected media reports, they write their own report and post them on a secure Internet portal to be accessed with Argus users.

Since its operation in July 2000, “it has logged more than 30,000 biological events involving pathogens such as avian influenza, the Ebola virus, cholera, and other unusual pathogens that have caused varying states of social disruption throughout the world” [CDC, First Quarter 2008].

⁶<http://www.google.com/flutrends/> (Retrieved March 7, 2014).

GermTrax

GermTrax⁷ is a freely accessible website which gathers sickness and disease data from people worldwide and exhibits trends through an interactive map. More specifically, GermTrax is a collaborative disease tracking system which primarily relies on reports filled by ordinary people who are sick. This system collects information through user personal updates on social media websites such as Facebook and Twitter. Then, the system saves user geo-location data, while the users connect their social media accounts with the site. According to their website, GermTrax can help people by informing them of places where they might get sick and help health experts to discover large-scale sickness trends. Since it principally relies on disease reports from ordinary people, it is suitable for non-specific conditions such as colds and flu [Lan et al., 2012].

Health Map

Health Map⁸ is a multi stream real-time surveillance system and freely accessible. It monitors online information in order to obtain a comprehensive view of current infectious disease outbreaks globally. It observes, filters, visualises, and distributes online information about emerging infectious diseases for the benefit of diverse audience from public health officials to international tourists [Lemon et al., 2007]. Health Map gathers reports from 14 sources, which in turn embody information from over 20,000 web sites every hour. Information is obtained automatically through screen scraping, natural language interpretation, text mining, and parsing [Brownstein et al., 2008]. More specifically, Health Map use multiple web based data sources including online news sources, expert-curated discussion, and validated official reports from organisations such as the World Health Organisation (WHO⁹). Then, the alerts are classified by location and disease using automated text processing algorithms. Next, the system overlays the alerts on an interactive geographic map. According to Freifeld et al. [2008] “The filtering and visualization features of HealthMap thus serve to bring structure to an otherwise overwhelming amount of information, enabling the user to quickly and easily see those elements pertinent to her area of interest”.

5.4 Statement of the Problem

While traditional syndromic surveillance systems can detect an outbreak with high accuracy, they suffer from slow response. For example, Centers for Disease Control and Prevention (CDC) publishes USA national and regional data typically with a 1-2 week reporting lag using outpatient reporting and virological test results provided by laboratories nationally [Culotta, 2010, 2013, Ginsberg et al., 2008]. Therefore, such systems cannot predict an outbreak, but only can detect them after the onset.

On the other hand, modern syndromic surveillance systems monitor online media from global sources. Such modern syndromic surveillance systems resort to internet based data such as

⁷<http://www.germtrax.com/> (Retrieved Oct 4, 2015).

⁸<http://www.healthmap.org> (Retrieved Oct 4, 2015).

⁹<http://www.who.int/en/>(Retrieved Oct 4, 2015).

search engine queries, health news, and people posts on social networks to predict an outbreak earlier [Signorini et al., 2011, Carneiro and Mylonakis, 2009, Corley et al., 2010]. While some of them claim that they could achieve high accuracy, the rate of false alarms is unknown. Ginsberg et al. [2008] state, regarding Google Flu Trends, that “Despite strong historical correlations, our system remains susceptible to false alerts caused by a sudden increase in ILI-related queries. An unusual event, such as a drug recall for a popular cold or flu remedy, could cause such a false alert”. Therefore, an issue with internet based data sources is that their data quality fluctuates over time.

Moreover, most of these modern syndromic surveillance systems rely on one type of internet based data sources and disregard the advantage of other type of data sources, which are discussed in Section 5.2 (page 112). Consequently, they are only suitable for places where their source data is sufficiently available. For example, Twitter based systems cannot have a high accuracy for places where using twitter is not very common, if accessible. In addition, the quality and availability of data sources may change over time. For instance, Twitter may lose its popularity or become inaccessible in a place. Hence, integrating available data sources according to an adaptive weighting scheme over time seems necessary.

The other area that has received attention in the syndromic surveillance literature is the topic of alternative analysis algorithms for a given data sources. Given that the quality of data sources change over time, and the most suitable algorithm for a given data source is not known *a priori*, a reasonable response is to consider analysing each data source with a variety of algorithms and integrate their results.

Against this background, we believe, based on plentiful available data sources and analysis techniques, a state of the art syndromic surveillance mechanism should:

- Requirement 1. Perform as an ensemble to combine various analysis algorithms with the objective of increasing syndromic surveillance system performance. There are many different techniques with different strengths and weaknesses. An ensemble which utilises a combination of them seems likely to be able provide higher performance than systems which are depended on only one technique.
- Requirement 2. Extract information which resides in different data sources. In addition to obtaining information, it should be capable of integrating them according to their relevance and varying quality.
- Requirement 3. Be flexible to changes in composition of algorithms and data sources over time as any of them might be deleted, temporarily unavailable, or added to the system at any time.
- Requirement 4. Be able to adapt to its corresponding monitored population behaviour and habits. For example, if people of a particular region are more prone to tweet their feeling in social media such as Twitter than searching for a solution using online search engines, then a syndromic surveillance system should weight twitter results higher than a search engine queries in that particular region.

Requirement 5. Be able to adapt to the *changes* of its corresponding population behaviour. For example, if twitter become more popular in a place and people start tweeting their sickness symptoms earlier, rather than visiting a physician, the system must give more attention and weight to twitter than previously.

Requirement 6. Minimise the effect of misleading factors and noise such as advertisement, promotions, and holidays on different data sources and, consequently, diminish the rate false positives.

With this in mind, ACPM can be used a syndromic surveillance technique by fulfilling the aforementioned requirements as we discuss below:

- 1) ACPM can behave as an ensemble method by including numerous agents, each having different analysis algorithms.
- 2) Prediction markets are specially designed for the purpose of information aggregation [Perols et al., 2009]. ACPM adapt the prediction markets' concepts and incentives it participating agents to share their private information through market mechanism, hence make accurate prediction. In addition, ACPM dynamically weights the prediction of different agents according to their varying quality.
- 3) In ACPM, market and other agents operate independently and hence absence or presence of an agent does not impact the system considerably. Therefore, if one of the existing data sources becomes unavailable for any reasons, ACPM can simply respond to the issue. If a new data source or a model is discovered, ACPM can simply create an agent to access that data source or model to participate in the market and share its knowledge.
- 4) In ACPM, the agents can be trained in the market using historical data of that place and, consequently will be adapted to behaviour of people in that place.
- 5) ACPM can respond to the changes of its corresponding population behaviour since its agents keep learning and their weights keep changing according to their current performance in each market.
- 6) ACPM can minimise the effect of misleading factors and noise by fusing various data sources and models using an adoptable scheme.

In the following sections, we use two well-known models of (i) Google Flu Trends, and (ii) the latest improvement of Google Flu Trends model, named as GP [Lampos et al., 2015], as our case study and we show how ACPM can improve upon their performance.

5.5 Google Flu Trends Case Study

Google Flu Trends (GFT) was launched by Google in 2008 to alert health professionals to outbreaks early by indicating when and where influenza is striking in real time using aggregate web searches. GFT publishes flu predictions (ILI rate) for more than 25 countries. Google Flu

Trends is typically more immediate, up to 2 weeks ahead of traditional methods such as the CDC’s official reports. The basic idea behind GFT is that when people get sick, they turn to the Web for information.

Google Flu Trends algorithms recognise a small subgroup of the millions of search engine query terms that deliver the maximum correlation with the CDC published ILI rate. Then a subset of these queries which fit the historical CDC ILI rate data most accurately are chosen. Finally, univariate linear regression model is trained to be used in predicting future ILI rate using each day queries. According to Copeland et al. [2013] the challenge of their approach is the varying volumes of a particular query over time. For instance, during the holiday season, more people search for ‘gift’ than at any other period. Similarly, overall usage of Google search varies throughout the year and is growing over time. GFT used the official CDC data only in the initial training and did not use it to re-train its model regularly¹⁰.

The early Google paper indicated that the Google Flu Trends predictions were 97% accurate comparing with CDC data [Ginsberg et al., 2009]. However, in 2013, Olson et al. [2013] and Butler [2013] reported that GFT was predicting more than double that of CDC published. Later in 2014, Lazer et al. [2014] stated that GFT has been overestimating flu occurrence for most weeks after August 2011 and by a very large margin in the 2011-2012 flu season. He continued stating GFT can achieve better performance by combining its prediction with other near realtime health data such as lagged CDC data. Also, Google Flu Trend team announced¹¹

“We found that heightened media coverage on the severity of the flu season resulted in an extended period in which users were searching for terms we’ve identified as correlated with flu levels. In early 2013, we saw more flu-related searches in the US than ever before.”

GFT subsequently updated the model in response to concerns about accuracy. In 9th August 2015, GFT stopped publishing flu predictions without formally presenting any reasons. However, GFT historical prediction are still available for download.

5.5.1 Comparison of ACPM and GFT

In this section, we use ACPM as a syndromic surveillance system and compare the performance of ACPM and Google Flue Trend.

5.5.1.1 Experimental Setup

In these experiments, ACPM predicts the disease activity level of influenza-like illnesses (ILI) in a given week in the whole of the USA using publicly available data sources. The data used here contains more than 100 real data sources covering the period 4th January 2004 (when GFT provides data for most of USA states and cities) to 9th August 2015 (when GFT stopped publishing their results online), from the two data sources of Google Flu Trends (GFT) and Centers for Disease Control and Prevention (CDC).

¹⁰<http://googleresearch.blogspot.ae/2014/10/google-flu-trends-gets-brand-new-engine.html> (Retrieved Oct 4, 2015).

¹¹<http://blog.google.org/2013/10/flu-trends-updates-model-to-help.html> (Retrieved Oct 4, 2015).

Model Full Name	Mdoel Short Name
Bagged CART	treebag
Conditional Inference Random Forest	cforest
Random Forest	rf
Multi-Layer Perceptron	mlp
Model Averaged Neural Network	avNNet
Boosted Generalized Linear Model	glmboost
Boosted Tree Linear Regression	blackboost
Linear Regression	lm
Radial Basis Function Network	rbf
Gaussian Process	gaussprLinear
CART	rpart
Generalized Linear Model	glm
k-Nearest Neighbors	knn
Gaussian Process with Polynomial Kernel	gaussprPoly
Multivariate Adaptive Regression Spline	earth
Self-Organizing Map	bdk

Table 5.1: R’s caret package models. ACPM instantiates one participant for each of these models.

Data Sources

In these experiments, we use weekly Google Flu Prediction for different areas of the United States including states, cities and regions¹², for which GFT data is available since January 2004. The detailed list of places are covered in Appendix F. In here, we use the calendar definition of year where a year starts on 1st January and finishes on 31st December.

The CDC Influenza Division produces a weekly report on influenza-like illness¹³ activity in the USA¹⁴. We use CDC statistics including: i) ILI rate disaggregated by age groups (0-4 years, 5-24 years, 25-64 years, and older than 65 years), ii) USA national ILI rate, iii) total number of patients and iv) total number of outpatient healthcare providers in U.S. Outpatient Influenza-like Illness Surveillance Network (ILI network). Since CDC reports ILI rates with a two-week time lag, we use CDC data of two weeks earlier for each week of the experimentation period. In this way, we can align CDC data with the other data sources used in these experiments.

Models

We use different machine learning models in R’s caret package (version 6.0-37), which are capable of performing regression. Table 5.1 presents the models we use in this experiment. Model parameters are set to their default values.

¹² This data can be accessed from <https://www.google.org/flutrends/about>. (Retrieved Oct 4, 2015).

¹³ILI is defined as fever (temperature of 100°F [37.8°C] or greater) and a cough and/or a sore throat without a known cause other than influenza (<http://www.cdc.gov/flu/weekly/overview.htm>) (Retrieved Oct 4, 2015).

¹⁴ This data can be accessed from <http://gis.cdc.gov/grasp/fluview/fluportaldashboard.html>. (Retrieved Oct 4, 2015).

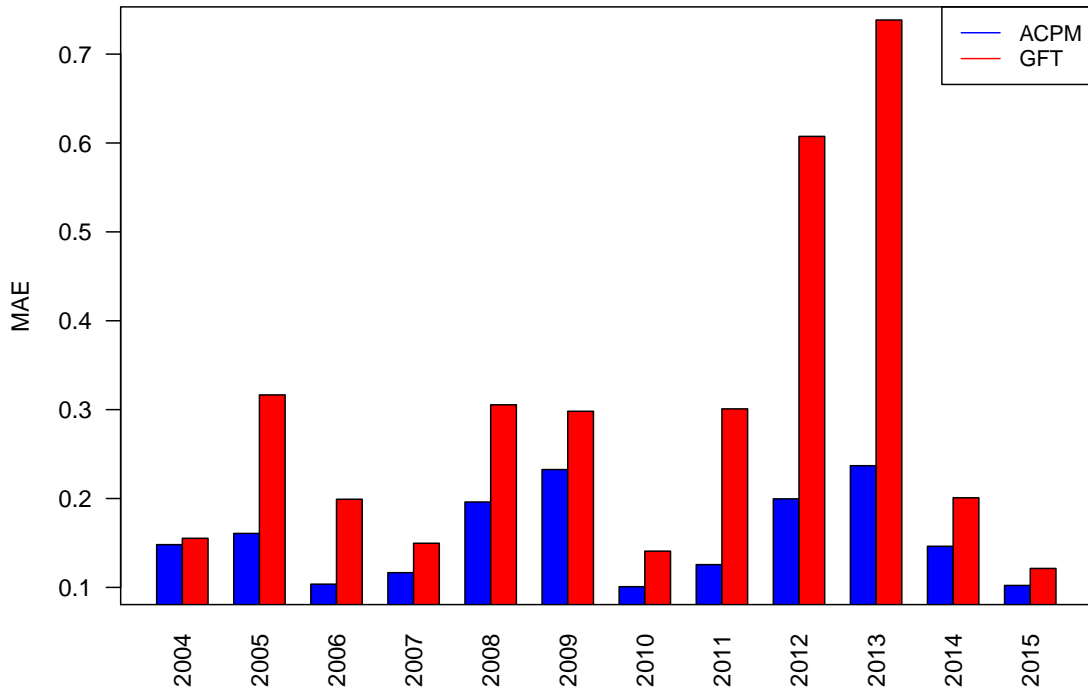


Figure 5-3: Comparing the Performance of ACPM and GFT for different periods using Mean Absolute Error (MAE).

Experiment Settings

We constructed an ACPM in which every agent has a unique analysis model corresponding to one of the models listed in Table 5.1. The data source for each agent is the entire data set. All agents use our proposed Q-learning trading strategy (see Section 3.3.2, page 59). The results are based on one run only, as they are deterministic. All ACPM parameters are set in accordance with discussions in Section 4.3 (page 80). Hence:

- i) The number of rounds is set to 2,
- ii) MaxRPT and MinRPT is set to 90%, in the first round, and
- iii) MinRPT and MaxRPT are set to 0.01% and 1% respectively, in the second round.

We measure the performance of ACPM by comparing the prediction of ACPM against the ground truth, which is the weekly ILI rate published by CDC. We use Mean Absolute Error (MAE), which is a common measure, in this literature.

5.5.1.2 Experimental Results

In this section, we compare the performance of ACPM and Google Flu Trends. Figure 5-3 and Figure 5-4 compare the error of ACPM and Google Flu Trend for the period between 2004 to

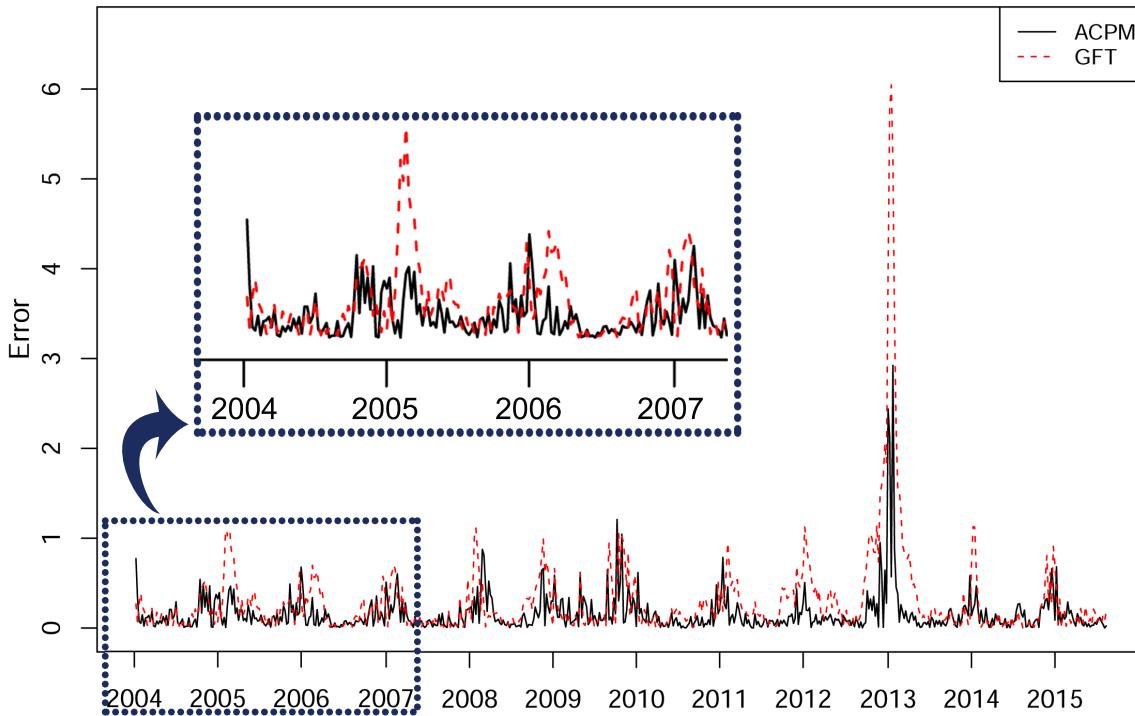


Figure 5-4: ACPM and GFT error in predicting ILI rate from 2004 to 2015

2015. Appendix G compares predictions and errors of ACPM and Google Flu Trend for each year of the experimented interval.

As Figure 5-3 shows, ACPM typically has a lower, sometimes much lower, MAE to that of Google Flu Trends in each year. Though this difference is relatively small in some years like 2004 and 2007, it is relatively large in most years and very large between 2011 and 2013. Table 5.2 shows the exact MAE value of ACPM and GFT in addition to t-test p-values. The null hypothesis is that the two accuracies compared are not significantly different. Therefore, within a tolerance $\alpha = 0.05$, when $p\text{-value} < 0.05$, ACPM is significantly better than GFT. As the table shows the results are highly significant in most years and also during the entire period of 2004-2015 ($p\text{-value} = 2.34E - 17$).

Figure 5-4 shows that ACPM performs poorly for the first few markets which we attribute to the learning period. However, after several markets, ACPM achieves higher performance than Google Flu Trend in most weeks. ACPM uses CDC data as one of its data sources, and since CDC report the data with two weeks time lags, ACPM uses the CDC data of the previous two weeks. This explains the existence of two weeks time lag between ACPM and GFT error in some periods such as early 2008 and late 2012.

5.6 GP Case Study

Lamos et al. [2015] published a paper in Nature Scientific Reports on 3rd August 2015 proposing a new model, called ‘GP’. Their model includes three improvements to the original Google Flu Trend. Firstly, they expand and re-weight the set of queries which are originally used by

Periods	ACPM MAE $\times 10^2$	GFT MAE $\times 10^2$	P-value
2004	0.148	0.155	3.73E-01
2005	0.161	0.317	5.82E-05
2006	0.104	0.199	7.89E-04
2007	0.117	0.150	4.42E-02
2008	0.196	0.305	4.15E-03
2009	0.233	0.298	5.66E-02
2010	0.101	0.141	2.18E-02
2011	0.126	0.301	2.33E-08
2012	0.200	0.607	3.12E-09
2013	0.237	0.738	4.64E-04
2014	0.146	0.201	4.85E-02
2015	0.102	0.121	1.57E-01
2004-2015	0.158	0.301	2.34E-17

Table 5.2: Performance of ACPM and GFT in predicting ILI rate using Mean Absolute Error (MAE) and p-values of paired t-test.

GFT. Then, they expand this improvement by using a nonlinear regression framework based on a Gaussian Process (GP) to investigate nonlinear relationship between query fractions and the ground truth (CDC ILI rate). Finally, they utilise time series structure. More specifically, they use ARMAX model [Hyndman and Khandakar, 2008] to find a relationship between previously available data and the current one.

They perform an evaluation using five consecutive influenza seasons, as defined by CDC, from 2008 to 2013. Based on their experiments, they conclude that GP approach performs better than GFT and a well established model, namely Elastic Net. They also mentioned that 2009-10 flu season is a unique flu period since during the peak of that flu season, GFT over-predicted the ILI rate, while GP and Elastic Net underestimated the ILI rate.

5.6.1 Comparison of ACPM and GP

This section compares the performance of ACPM and the model proposed by Lampos et al. [2015], known as the ‘GP’ model. We contacted the author and received their exact prediction for each experimented period to use in our experiments.

5.6.1.1 Experimental Setup

All settings are similar to the settings covered in Section 5.5.1.1 (page 121), except the part that ACPM includes an additional agent which uses GP prediction as its data source. The agent uses a simple algorithm which gives the prediction equal to the receiving data, hence no analysis is performed by the agent on that data.

In these experiments, we follow the same evaluation format as the work by Lampos et al. [2015], therefore we compare the performance of ACPM and GP in the flu seasons 2008 to 2013 as defined by CDC. These flu seasons include different numbers of weeks (see Table 5.3).

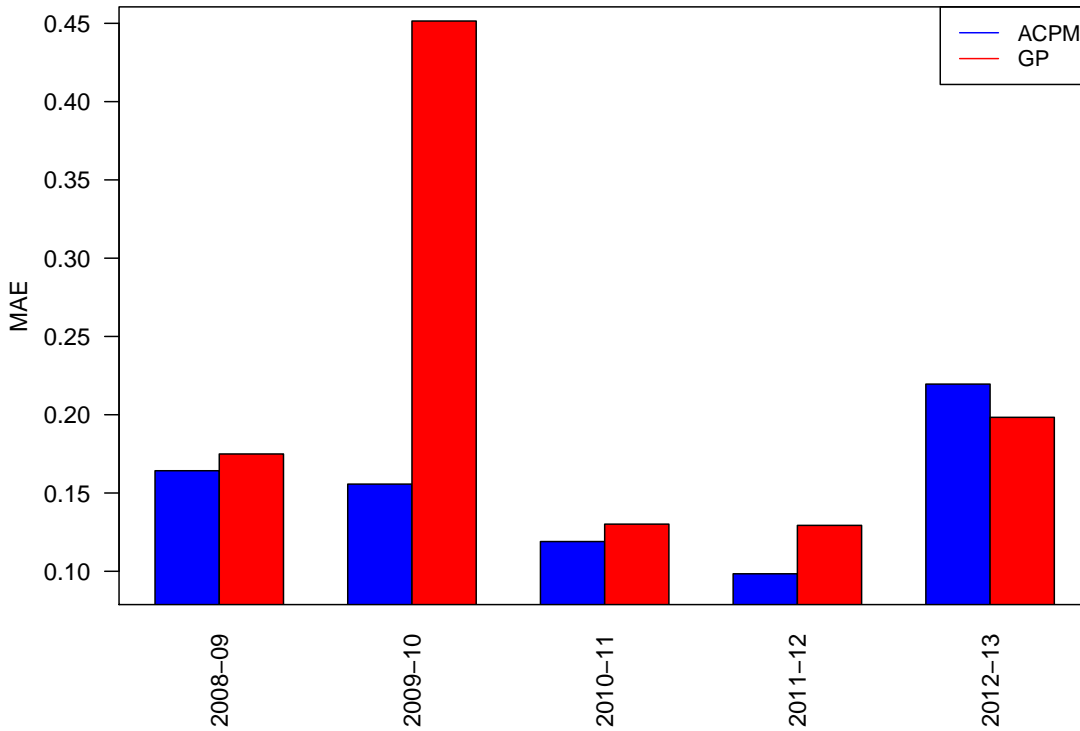


Figure 5-5: Comparing the Performance of ACPM and GP for different periods using Mean Absolute Error (MAE).

5.6.1.2 Experimental Results

Figures 5-5 and Table 5.3 compare the performance of ACPM and GP for different influenza seasons between 2008 and 2013. Figure 5-6 compares the error of ACPM and GP in each week of the entire period. In Table 5.3, the first column shows the experimented influenza seasons and the second column presents the number of weeks in each season. The third and the fourth columns show the Mean Absolute Error (MAE) of ACPM and GP respectively. The last column shows p-values for the paired t-tests comparing the error of ACPM and GP. See Appendix H for comparison of ACPM and GP predictions and errors for each flu season.

As Table 5.3 and Figure 5-5 show ACPM outperforms GP in most years except 2012-2013, where ACPM achieves MAE of 0.220 and GP achieves MAE of 0.198. As shown by Figure 5-6, this is mainly because of lower performance of ACPM compared to GP in early weeks of 2012-13 flu seasons. A few weeks earlier than that, in the late 2011-12 flu season, GP performs worse than ACPM which infers that the GP agent performs worse than the other ACPM agents. By reaching 2012-13 flu season, suddenly GP agent performance improves compared to other ACPM agents who mainly use GFT data. Given that GP agents had lower performance previously (in late 2011-12 flu season) compared to other ACPM agents, ACPM relies more on other participating agents than GP agent. Once the performance of GP improves (in early 2012-13 flu seasons), GP

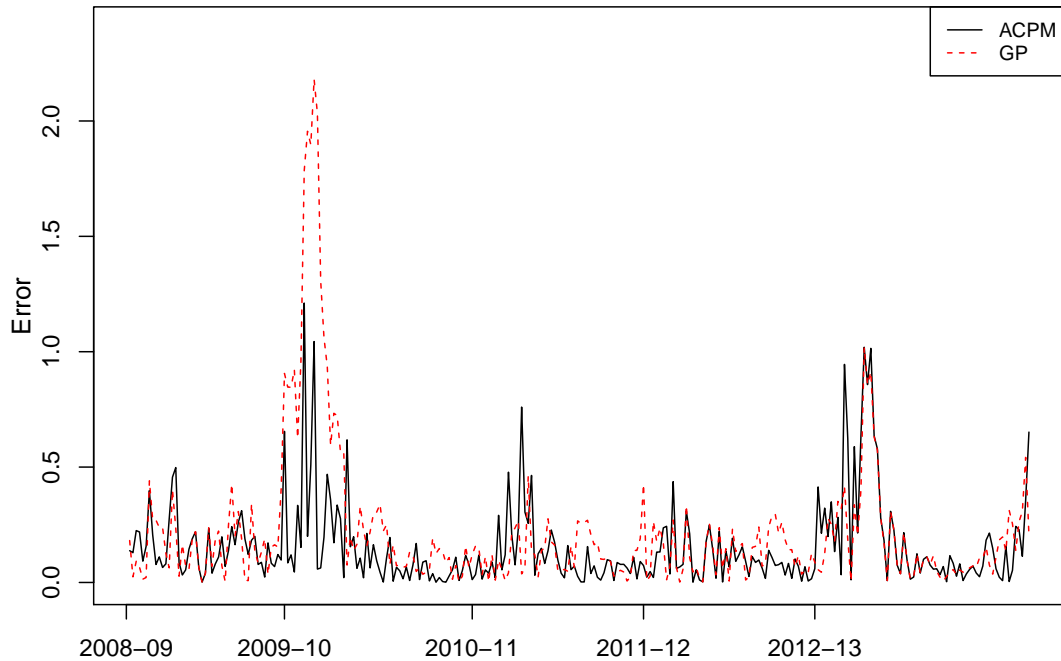


Figure 5-6: ACPM and GP error in predicting ILI rate from 2004 to 2015

outperforms ACPM for a several number of weeks. However, as shown by Figure 5-6, ACPM recovers rapidly and adapts to the new conditions of the markets (i.e changing quality of market participants) in a few weeks.

As mentioned earlier, Lamos et al. [2015] states that that 2009-10 flu season is a unique flu period since none of the models experimented by them nor GFT could make prediction close to the ground truth (CDC data). Interestingly, ACPM achieves much less error than GP in 2009-10 as shown by Figure 5-5.

5.7 Analysis

ACPM outperforms both the Google Flu Trend and GP models because:

- i) ACPM integrates different data sources such as CDC reports and Google Flu Trend prediction for different states and cities of USA.
- ii) ACPM analyses each data source with a variety of machine learning models and combines their results.
- iii) ACPM adjusts the influence of agents on market prediction automatically according to their quality. Over time, high quality agents – either because of their effective analysis

Period	Weeks	ACPM MAE $\times 10^2$	GP MAE $\times 10^2$	P-value
2008-09	48	0.164	0.175	2.75E-01
2009-10	57	0.156	0.451	7.40E-06
2010-11	52	0.119	0.130	3.22E-01
2011-12	52	0.098	0.129	6.31E-03
2012-13	65	0.220	0.198	8.73E-01
2008-2013	274	0.155	0.221	3.65E-05

Table 5.3: Performance of ACPM and GP in predicting ILI rate using Mean Absolute Error (MAE) and p-values of paired t-test.

model or accessing high quality data source – gain more revenue than low quality agents. Therefore, high quality agents accumulate more budget and they can make larger investment on their prediction than poor performing agents. Subsequently, high quality agents achieve larger influence in the market as the integration function weights each prediction by its corresponding investment amount.

- iv) ACPM adapts to the dynamic environment where the quality of data sources and the performance of a model on each data source fluctuates over time. Once the quality of an agent prediction changes, its performance in the market is affected and hence the influence of agents on the market prediction is tuned according to their current quality, as explained above.
- v) The Q-learning trading strategy causes high quality agents lead the market by preserving their original prediction and low quality agents follow them and hence minimise their negative effect on forming market prediction.
- vi) ACPM can minimise the effect of misleading factors and noise since ACPM integrates various data sources and combines the result of different machine learning models, while dynamically changing their weight according to their varying quality. For example, as shown by Figure 5-4, Google Flu Trend overestimated the Flu rate by large extent in 2013 due to a misleading factor which is a heightened media coverage on the severity of the flue [Stefansen, 2013]. Since Google Flu Trend is being used as one of ACPM data sources, ACPM also overestimates the Flu rate to some extent but much less than Google Flu Trend. In ACPM, as soon as an agent loses its quality, it either loses most of its budget or learns to improve its prediction using wisdom of the crowd as advised by its Q-learning trading strategy. Therefore, its original influence on the market prediction decreases and ACPM relies to other agents with higher current performance. In the similar way, once the quality of an agent improves, the influence of the agent on forming the market prediction increases.

Chapter 6

Conclusions and Further Work

In this thesis, we adapted the concept of prediction markets to propose a new machine learning technique, namely the Artificial Continuous Prediction Market (ACPM). ACPM instantiates a sequence of markets in which artificial agents play the role of market participants. Agents participate in the markets with the objective of increasing their own utility and hence indirectly cause the markets to aggregate their knowledge. Agents read a data source and use an analysis model to mine their data. Each market is run in a number of rounds in which agents have the opportunity to send their predictions and bets to the market. At the end of each round, the aggregated prediction is announced and agents can use this information to change their predictions and bets in the next round.

Each agent has a trading strategy which guides her on how to participate in the market. Using the Q-learning trading strategy, proposed in Chapter 3 (page 49), agents participate in a market in accordance with their past experience and performance. This intelligent trading strategy advises an agent which situations are more probable to bring her positive rewards and which actions take her into those situations. The Q-learning trading strategy uses wisdom of the crowd to measure the quality of an agent's prediction and decide the best action for the agent based on that measurement and the history of the agent's and the market's performance.

Once a market is over and the correct outcome is revealed, agents receive a reward based on their prediction quality and their bet. Agents with more accurate prediction receive greater rewards and accordingly accumulate more budget over time, which gives them the opportunity of investing larger amounts on their prediction and hence having higher influence on the market prediction in subsequent markets.

In ACPM, at the end of each round, the aggregated predictions of the crowd is announced to all agents, which provides a signal to agents about the private information of other traders so they can adjust their beliefs for the next round accordingly. This message passing between agents results in updating an agent's prediction and investment in response to other agents' trades in the markets and achieving a consensus between them. According to Dimitrov and Sami [2008] adjusting traders' beliefs in response to other traders' trade is a critical successful criterion in prediction markets and this process should result in a situation in which all traders reach a consensus belief that reflects all available information.

ACPM works as an online machine learning technique by creating a prediction market for

each record in the data set. At the end of each market, agents update their analysis models and trading strategy based on the utility the agents receive and the correct answer of the record with the objective of maximising their profit for future markets. This behaviour of agents (i.e re-training after each record) result in making ACPM an adaptive technique which is suitable for situations where the quality of data sources or the performance of the models varies over time. In cases where the quality of the best performing agents falls for any reason, ACPM can recover by relying on the prediction of other well performing agents.

In ACPM, there is no single agent responsible for the prediction and the prediction task is decentralised between agents. This decentralisation allows the system to be very robust in case any of the agents fails or is deleted from the system.

Furthermore, according to the No-Free-Lunch theorem [Wolpert, 1996], the best model will not be the same for all the data sets. Hence, a data analyser is required to try various models to choose the most appropriate ones, which makes the process of deciding the best analysis model for each data set a time consuming task. ACPM can automate the process of choosing the best analysis models and data sources, as it achieves a performance very close to the best agent, which represents a data sources and/or an analysis model.

ACPM is resilient to addition of low quality data sources and low performing analysis models. Hence, a data analyser can include a variety of models and data sources to the system without concern for their prediction qualities.

ACPM is not domain dependent. A data analyser can add any data source and analysis model which seems appropriate to a specific application problem. The agents automatically train their model and update their trading strategy without requirement for a handcrafted setting.

By instantiating one agent for each data source and/or analysis model, ACPM aggregates data from different data sources and/or works as a machine learning ensemble technique. In other words, each data source is analysed with various homogeneous or inhomogeneous base models and their results are aggregated. Moreover, ACPM agents can have different ensembles as their analysis model and in that case ACPM can be seen as *ensemble of ensembles*.

As mentioned above, ACPM can create an agent for each available analysis models including well-known machine learning models. In addition, the performance of ACPM is usually better or equal to the best performing agent. Consequently, ACPM can easily outperform other machine learning models since the best model is not the same for all data sets but ACPM can recognise the best agents (i.e best models) for each data set and follow their predictions.

6.1 Contributions

This thesis offers several contributions to the current literature as follows.

- I) We propose an online machine learning technique using the prediction market concept. Our proposed model works as an adaptive ensemble model which can also integrate data from various sources. ACPM analyses each data source with various models and combines

their results. Our work extends the literature of Artificial Prediction Markets (APM)¹ as follows:

- ACPM is an *online* technique, i.e creates a prediction for each record in the data set and agents update their analysis model and strategies after each market. Since ACPM is an online technique, it can easily handle the problems in which the quality of data sources vary over time. However, existing APMs are presented as an offline technique, i.e the participants only learn before entering the system and never revise their analysis model after that.
 - The agents in ACPM have an *adaptive trading strategy* similar to human traders in real prediction markets. This trading strategy uses wisdom of the crowd to advise agents how to participate in a market. This adaptive trading strategy is tuned by individual trading history and can dynamically identify the actions that maximise the agent’s rewards. However, existing APM participants are assumed to use fixed strategies, such as constant betting functions [Barbu and Lay, 2012], utility functions [Storkey, 2011, Storkey et al., 2012] or static risk measures [Hu and Storkey, 2014]. However, according to [Jian and Sami, 2012] market speed and effectiveness depends on proper strategy of traders.
 - ACPM agents are explicitly armed with the ability to revise their predictions in response to those of other agents. In contrast, the participants of existing APMs lack the ability to observe, and reflect upon, wisdom of the crowd during the market. According to Dimitrov and Sami [2008], the aggregated information in prediction markets provides signals to each trader about the private information of other traders so the trader can adjust its prediction accordingly and the successful aggregation of information in prediction markets relies ‘critically’ on this process.
- II) We propose a *trading strategy for traders in prediction markets using reinforcement learning*. Our experiments show that adopting this trading strategy improves both agents’ and markets’ performance. Reinforcement learning has been used as a trading strategy for traders in financial markets (see Section 2.1.4.6, page 33), but not for prediction markets.
- III) We study the *syndromic surveillance* literature and highlight an existing problem in this field. We believe a syndromic surveillance system is required to integrate data sources from different sources, analyse each data source with various models while dynamically adapting to the non-stationary quality of data. The system needs to be flexible to the addition and deletion of data sources. After investigating this literature, we suggest ACPM as a solution. We experimentally demonstrate the success of our model by comparing its performance with two well known models of Google Flu Trends (GFT) and GP model [Lampos et al., 2015] (see Chapter 5, page 111).
- IV) Some works [Surowiecki, 2005] indicate that wisdom of the crowd holds if participants present their opinions independently, however, some other works [Miller and Steyvers, 2011]

¹See Section 2.3.2 (page 43) for a review of existing APM techniques.

suggest that allowing a limited communication between subjects can improve wisdom of the crowd aggregation. In addition, according to Dimitrov and Sami [2008] wisdom of the crowd in real prediction markets comes from aggregated information which provide signals to traders about the private information of other participants so they can adjust their beliefs in response to other traders [Dimitrov and Sami, 2008].

Against this background, this thesis shows that ACPM performance increases when its agents use our proposed Q-learning trading strategy (i.e there is a limited communication between participants). Using this trading strategy, ACPM agents participate in the first round of a market independently. In the subsequent rounds, they use the aggregated wisdom of the crowd – obtained in the previous round – to adjust their beliefs in response to the signal of other participants only if this approach has shown to improve their performance in past.

- V) Using a number of experiments, we study the *performance and capability* of ACPM in various situations. In particular, we demonstrate ACPM performance: i) is more than the average of all participants' performance, ii) is either better or very close to the best performing agents in different application domains, iii) is resilient to addition of agents with low performance, which can be resulted because of their poor quality data source and/or their low performing analysis model, iv) is higher than many well-known classifiers and ensemble models in different data sets.

We also show that our proposed Q-learning trading strategy encourages high quality agents to ignore market prediction as another source of information and mainly rely on their own predictions. In contrast, this trading strategy encourages low quality agents to change their prediction based on the aggregated prediction of other agents. Hence, Q-learning trading strategy, compared to the constant strategy, improves ACPM and agents' performance.

Using a number of experiments we studied the performance of ACPM in situation where the agents' performance quality change. These experiments showed that ACPM shifts focus in response to changes in quality of agents predictions. In other words, the influence of agents on the overall prediction is changed as their performance change. The experiments also confirm that ACPM requires relatively a small number of records to recover in situations where the quality of the best performing agents suddenly drop. Finally, the experiments exhibited that excluding the time required by the analysis models, ACPM requires relatively small time consumption.

6.2 ACPM Limitations

Our ACPM has some limitations that needs to be considered.

Time Consumption: In Chapter 4 (page 77), we investigated the amount of time required by ACPM for including different number of agents. The results showed that ACPM requires a relatively small amount of time to include each additional agents excluding the time required by their analysis model. However, some analysis models may require a large

amount of time to predict or re-train. Since ACPM needs to wait for all agents to predict before moving to the next round of the market and wait for all agents to re-train before moving to the next market (i.e record), this time is added on top of the time consumption required by ACPM. Hence, if the time consumption is an important factor by the data analyst, then she may need to consider the trade off between including an analysis model and the the time it adds to the time consumption required by ACPM.

Non-volatile Environment: In Chapter 4 (page 77), we showed that in non-volatile environment in which a particular agent outperforms all other agents constantly, ACPM performance is equal to the performance of that particular agent. This behaviour happens as the best performing agent keeps accumulating budget and hence the difference of its budget and other agents become large. The agent also learns to not incorporate wisdom of the crowd to update its prediction as its prediction is always better than the crowd. In cases where the data analyst has the knowledge of such agent beforehand – which represents a combination of a data source and an analysis model – the data analyst will not gain from using ACPM and can easily use that combination of data source and analysis model directly. However, if such knowledge is not known a priori, then ACPM is an effective technique to distinguish such agent and achieve a performance as good as the best performing agent.

Volatile Environment: In Chapter 4 (page 77), we demonstrated that ACPM can rapidly recover in cases where the best performing agent suddenly loses quality or becomes unavailable. However, in cases where the environment is too dynamic and this behaviour happens very frequently in the sense that the best performing agents in one market are the worst performing agents in the next market, ACPM cannot perform well. In such situations, ACPM may put high weights on poor quality predictions and low weights on high quality predictions and keeps switching weight between agents. To date, this limitations is seen as an open question to us.

6.3 Future Works

This research line can be extended in various dimensions as discussed below.

Incorporating Human Participants

A very interesting future works is collaborating human traders in an experiment to compare the performance of human traders and ACPM agents in terms of the accumulated wealth during a sequence of markets. In other words, this experiment helps us to investigate how our proposing Q-learning trading strategy can be useful for human traders in real prediction markets.

Analogously, by asking human traders to use our proposed Q-learning trading strategy, we can study how utilising such trading strategy by human participants can affect the performance of a real prediction market in terms of prediction accuracy.

The other interesting research line, in this context, is investigating how collaboration of agents and human participants can affect the performance of ACPM. For example, Polgreen et al. [2007] forecast infectious disease activity using a real prediction market with health care expert participants. In Chapter 5 (page 111), we used our artificial prediction market to achieve the same objective. We can set up a simulation which contains both human participants and artificial agents and study how this collaboration can increase the accuracy of prediction.

Variations of ACPM Mechanisms

The other interesting extension to ACPM is studying various trading strategies, market trading protocols and attitudes towards risk, and to quantify the effect of each variation on the overall aggregation process.

- **Participant Trading Strategy**

One possibility is a trading strategy which guides agents participation based on their obtained confidence over the previous markets. For example, agents who have gained a successful history are more confident on their prediction and are more prone to invest in their prediction than agents who lost most of the time. The other option is surveying human traders in real prediction markets to see what trading strategy they use in practice. Based on their reports, we can design new trading strategies for our agents.

The other interesting alternative is considering how non-myopic agents can affect ACPM performance. In this thesis, we assume all agents are myopic and do not trade in a market in way to deceive other agents and manipulate their beliefs. To achieve this, we can be inspired by [Chen et al., 2010b] who investigated non-myopic strategies in real prediction markets.

- **Market Trading Protocols**

In Section 2.2.3 (page 38), we described different prediction market protocols. Our proposed model in this thesis is similar to the pari-mutuel market. We can investigate how other prediction market mechanisms such as continuous double auction, call auction, dynamic pari-mutuel markets and market scoring rules can affect the performance of ACPM. However, most of these mechanisms are only suitable for classification rather than regression tasks, which is the focus of this thesis, hence they need to be adapted for this purpose.

- **Attitudes Towards Risk**

The other interesting research avenue is incorporating agent risk measures in addition to their trading strategy. When there is an element of uncertainty in a financial market, traders may exhibit different risk attitude. While ‘risk averse’ traders prefer to invest in assets with guaranteed payoff, ‘risk-seeking’ traders are more willing to invest in assets with higher payoffs even if they are associated with higher risk. A ‘risk neutral’ trader is indifferent between assets with equal expected payoffs even if one of them is associated

with more risk than the other one and hence the trader's decision is not affected by the degree of the uncertainty.

In this thesis, ACPM agents do not take any risk measure into consideration when participating in the market, but participate in such a way as to maximise their expected utility. As a matter of future work, we are interested to investigate how incorporating risk measure by agents can affect the performance of agents and the market in term of obtained accuracy.

Considering Different Timeliness of Data Sources

In this thesis, we assume that all agents' information with regards to a specific event arrive at the same time, i.e. all data sources have the same timeliness. However, ACPM can be enhanced to incorporate new information and provide a reflected prediction as soon as new information arrives. For example, in the domain of syndromic surveillance systems, open source information such a Google Flu Trend (GFT) data arrives two weeks earlier than the reports from CDC. In this case, ACPM can be advanced to calculate and present market prediction as soon as agents who own GFT data participate in the market. However, a market prediction should reflect the arrival of other information once other agents receive their data and participate in the market. Given that agents have the opportunity to participate until the deadline of the market, agents can update their prediction and bet according to the updated market prediction (i.e wisdom of the crowd).

Other Machine Learning Tasks

One future aim is facilitating ACPM to perform other forms of machine learning tasks. In this thesis, we focus on regression tasks, which is the prediction of a continuous variable. However, ACPM can be adjusted to do classification which refers to prediction of one outcome out of several discrete classes.

To achieve this objective, we need to make some changes to the current mechanism of ACPM. To perform classification, we should adopt other prediction market mechanisms such as continuous double auction, call auction, dynamic pari-mutuel markets and market scoring rules which are fundamentally designed for classification. One promising option is the popular logarithmic market scoring rule (LMSR) proposed by Hanson [2003].

Other Real World Applications

The other alternative future work is applying ACPM on different real world application domains. For example, ACPM can be applied to the application domain of weather forecast. Different weather forecast techniques such as persistence methods, climatology methods, analog method, barometric based methods, statistical methods and time series models are being used in this literature. In addition, various data sources including different forecast websites or forecasts made by various organisations which are made for their own specific purpose exist.

ACPM can instantiate one agent for each combination of data sources and techniques in order to integrate their predictions, while dynamically adjusting their weights according to their varying quality.

Appendices

Appendix A

An Alternative Reward Function

Once the market is over, agents are notified of the correct answer and receive revenue for each of their bids, as determined by a reward function. In this section, we discuss an alternative reward function to be used by the market maker at the end of each market.

Similar to real prediction markets, agents' revenues should depend both on their prediction accuracy and their bet. In this way agents are motivated to make accurate prediction and invest according to their belief about the quality of their prediction. Therefore, we calculate the agent's revenue:

$$\text{Revenue} = \text{Reward} \times \text{bet}, \tag{A.1}$$

where $\text{Reward} \in \mathbb{R}^+$ and determines a reward based on the accuracy of the prediction.

A reward function can impact the influence of agents in the subsequent markets by the amount of revenue it returns to them. If the reward function returns nothing to an agent for the bid it submitted to the market maker, then the agent lose a portion of its budget for the investment it made, therefore its budget and consequently its investment and influence on upcoming markets will be decreased. On the other hand, if an agent receives a large amount of reward in a market, then the influence of agent for future markets is increased as the agent's budget increases and consequently it can invest more on its predictions.

Intuitively, to incentivise agents to make accurate predictions, the agents' rewards should depend on the agent's prediction error. The more error a prediction includes, the less reward it should be assigned to. Therefore, we use a reward function which is inversely proportional to the agent's prediction error.

In the simplest form, the reward function has a linear format. The linear function differentiates equally between agents' rewards in proportion to their prediction errors. However, in some situations, it is preferred to not differentiate between agents' rewards by large extent if the difference between their prediction is less than some degree. For example, imagine a population of agents which have similar performing quality. Thought in one market, some of them may outperform the other ones, we prefer not to differentiate their influences in the next markets and hence we want them to retain almost equal amount of budget (i.e influence). In such situation, an exponential reward function is suitable which decreases the reward with less steep than a

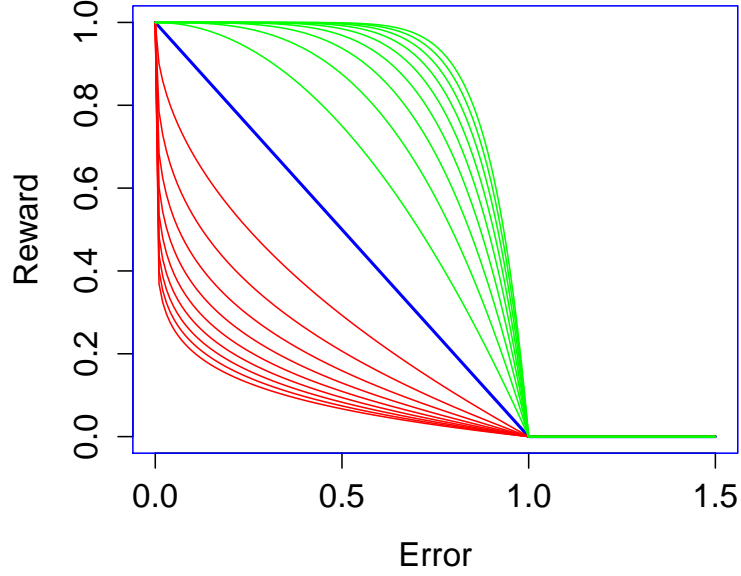


Figure A-1: The reward functions for $C = 1$, $\beta = 1$ and $P \in (\frac{1}{10}, \frac{1}{9}, \dots, \frac{1}{2}, 1, 2 \dots 10)$. Note that errors greater than the cut-off ($C = 1$) receive zero rewards.

linear function.

Figure A-1 shows a family of piecewise exponential and linear reward functions. As the figure shows, the reward linearly decreases by increasing error in the linear function (blue line). But, in convex lines (green lines), the reward slightly decreases by increasing error from zero up to a certain point and rapidly after that and the opposite behaviour exists for concave lines (red lines). Therefore, the curvature of the reward function (P) differentiates between agents' reward in term of the proportion of their rewards to their prediction errors.

In addition, a reward function should handle the situations where large number of agents have low quality in the market and only minority of them have high quality. In such cases, we would prefer that low quality agents lose majority of their budgets rapidly. For this reason, the reward function should include a cut-off parameter (C) to assigns zero reward to any prediction which contains error above it. Note that we want the low quality agents lose the majority of their budget but not all, since they may gain quality at any point of time and loss of an agent means a loss of a data source and/or an analysis model. For this reason, we define rate per transaction parameters which are discussed in Chapter 3 (page 49).

According to discussed points, we propose the following reward function.

$$\text{Reward} = \max((1 - (E/C)^P)\beta, 0), \quad (\text{A.2})$$

where $E = \text{abs}(\text{TrueValue} - \text{AgentPrediction})$ is the prediction *error*, $C \in \mathbb{R}^+$ is the reward cut-off, above which agents receive zero revenue (see Figure A-1, in which, for example $C = 1$), $P \in \mathbb{R}^+$ is the (non-)linearity of the reward function (where $P > 1$ is the family of functions above the diagonal (green), $0 < P < 1$ are those below (red) and $P = 1$ is the linear function) and $\beta \in \mathbb{R}^+$ is the reward for an accurate prediction (if $\beta = 2$ then a prediction with $E = 0$ gets

twice its investment).

Formula (A.2) expresses that an agent’s reward is inversely proportional to the error in its prediction and the agent is incentivised to make a prediction as accurate as it can. This makes the reward function *incentive compatible*. The incentive is further affected by parameter C which expresses that $\forall E \geq C : \text{Reward} = 0$. Hence, agents with the error $E < C$ receive revenue and the rest do not.

The parameter P changes the curvature of the reward function. By increasing P to more than 1, the curvature of the reward function increases toward the convex side, which means the proportion of reward to E decreases for smaller E and increases for larger E. Setting $P < 1$ (red lines) means that a few high quality agents can dominate the market, while the predictions of all others, including (perhaps) good quality agents, are under-weighted.

The parameter β affects the magnitude of the reward: $\beta \leq 1$ means an agent gets back only its investment in case of a perfect prediction ($E=0$) or less otherwise. Therefore, $\beta \leq 1$ disincentives participation, but $\beta > 1$ means reward is greater than investment, in the case of a good ($E < C$) prediction.

Thus, by rewarding the participants according to their performance in a given market and updating their budgets accordingly, ACPM can keep track of the overall performance of each of the participants according to their accumulated budgets. Furthermore, since better performing participants gain more revenue and accumulate bigger budgets, they can invest more in subsequent markets and hence have more influence on those markets’ predictions (see Formula 3.4).

Choosing Parameters Automatically

ACPM reward function includes three parameters of β , C and P. The main objective of parameter β is motivating agents to participate and any value greater than 1 meets this objective. Therefore, Parameter β can simply be set to 2. But, the market maker tunes the other reward function parameters, namely C and P from market to market so that the best performing agents, called **merit agents**, have the greatest influence on the market prediction (i.e merit agents hold larger budgets compared to other agents). The market maker first identifies the merit agents by keeping track of the agents with the least error in the previous markets.

The merit agents should have the greatest influence on the market prediction. In particular, we do not want them to lose the majority of their budget by making a single poor prediction. Therefore, the market maker sets C (the cut-off parameter) to the error of the worst merit agent. More formally, let E_1 be the error of the best prediction, E_2 the next best and so on, such that E_i is the error of the i^{th} best prediction in a given round and let MA be the set of merit agents, then $C = \max_{i \in MA} E_i$.

As mentioned earlier, the curvature (steep) of the reward function differentiates between participating agents by affecting the amount of reward they are given with regards to their predictions’ errors. For example, if the proportion of two agents’ errors is equal to 2, the proportion of their rewards is equal to 2 with a simple linear reward function, but the proportion of their rewards can be close to 1 with a reward function with low steep. The latter case is more suitable for situations where it is not preferred to differentiate between agents influence

significantly even if the quality of their predictions are not close.

The behaviour of reward function be can adapted to different situations (i.e markets with various population in term of their performance quality) by setting the steep of the reward function based on the quality of participating agents (i.e their prediction error). In other words, the market maker can reward agents in way so that the proportionality of agents' rewards with regards to their errors (**reward function steep**) be a reflection of the magnitude of their errors (**error proportions**).

Formally, denoting error proportions by (**EP**), and reward function steep by (**RS**), we seek to set the reward function steep equal to error proportions, i.e $\min(|ED - RD|)$, where:

$$EP = \frac{1}{n} \sum_{i=2}^n \frac{E_1}{E_i} \tag{A.3}$$

$$RS = \frac{1}{n} \sum_{i=2}^n \left| \frac{R_i - R_1}{E_i - E_1} \right| \tag{A.4}$$

where R is the reward and calculated by formula (A.2) and n is the number of merit agents. We concentrate on merit agents since each market prediction is most influenced by its merit agents and hence ACPM parameters should be tuned based on these agents. Using this formula, the steep of reward function is low (P is high) when the market include agents with roughly similar quality (i.e small differences between their prediction error) and inversely the steep of reward function is high (P is low) when the participating agent quality varies by large extent.

Given that the other parameter are now set, the best value for parameter P can be achieved by calculating agents' rewards for different values of P . Then, the best value of P is the one which results in the least difference between EP and RS . In other words, the parameter P can be calculated by solving $P = \operatorname{argmin}_P(|EP - RS|)$.

Finally, since we want to set these parameters according to the real quality of agents and not their revised predictions informed by wisdom of the crowd, these parameters are set in each market based on the first round bids.

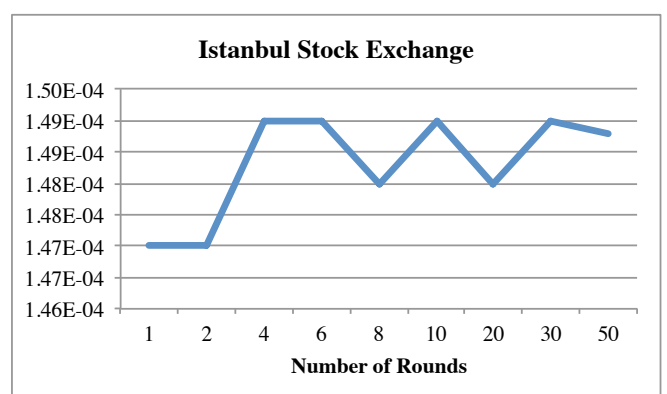
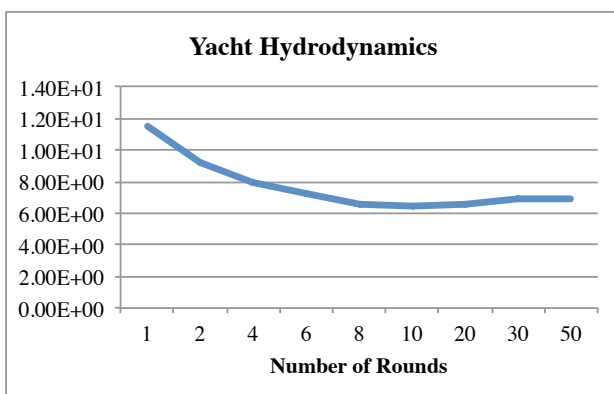
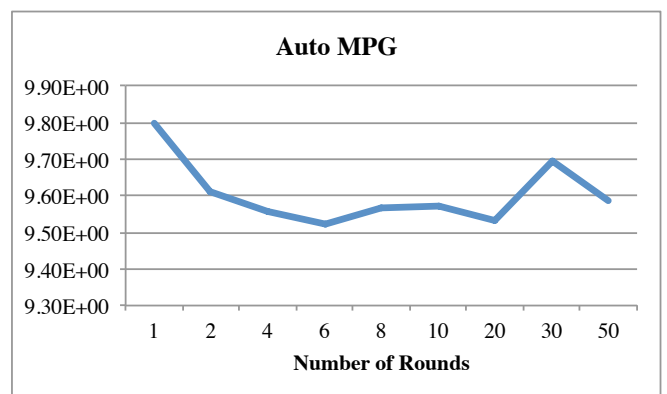
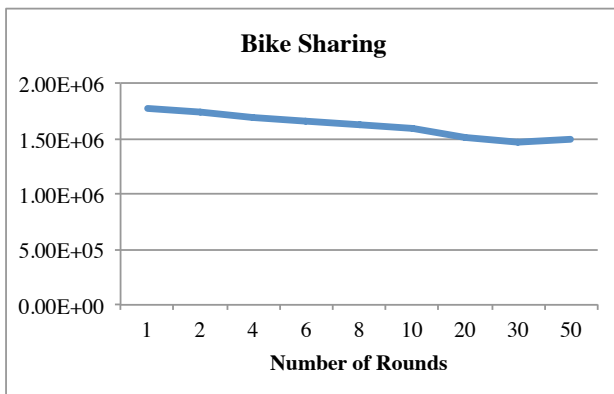
Appendix B

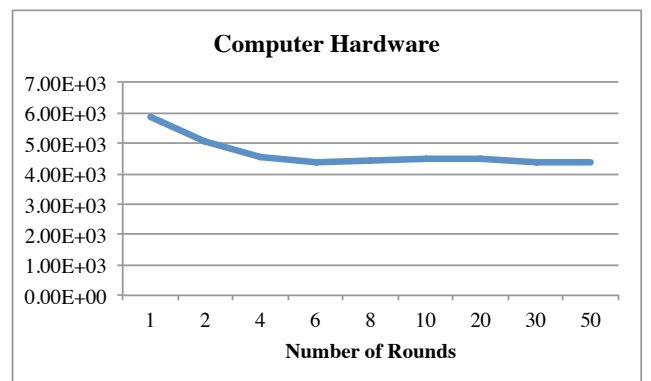
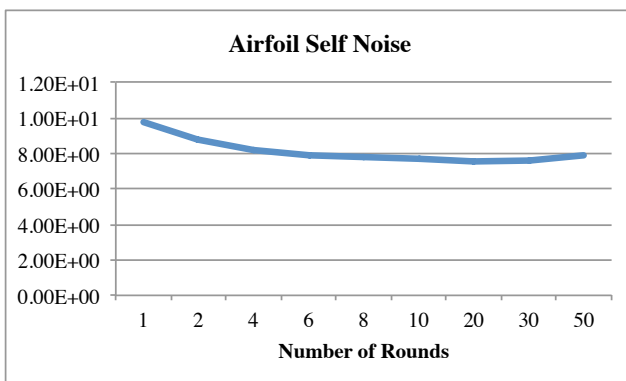
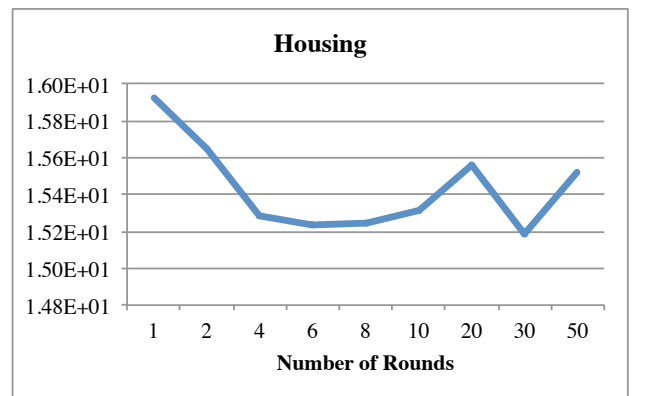
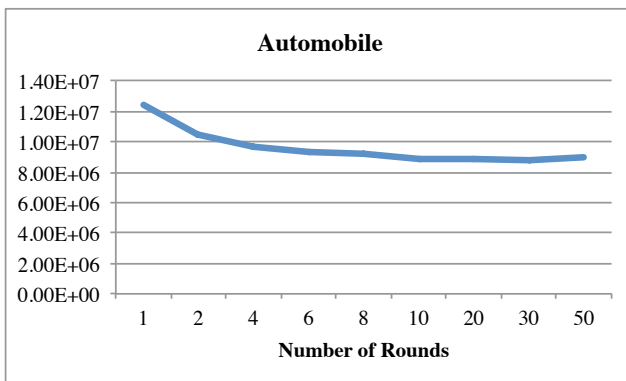
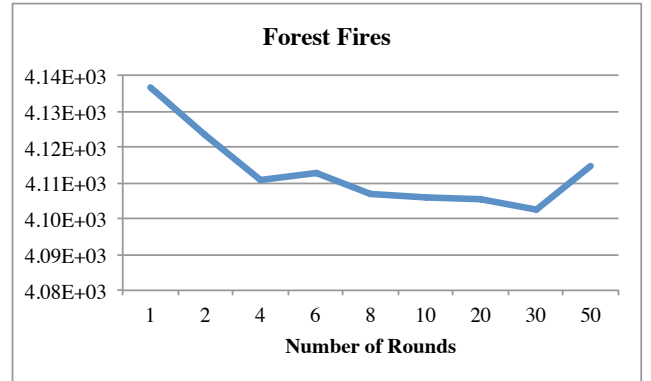
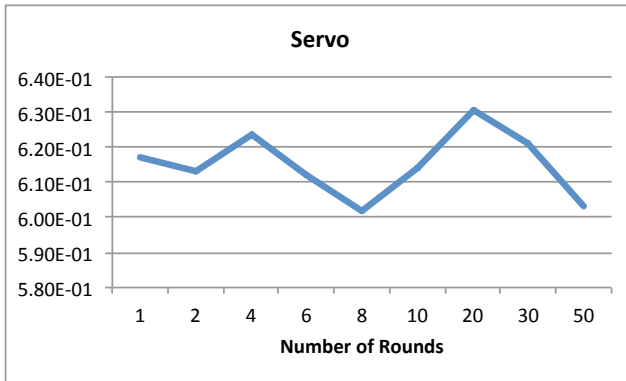
ACPM Parameter Setting

Chapter 4 investigates the performance of ACPM using a number of experiments on UCI data sets. In here, we present the performance of ACPM for different number of rounds and MaxRPT values using the experimental setting 5 (see page 86).

B.1 Number of Round Parameter

This subsection presents ACPM performance for different number of rounds using various UCI data sets.

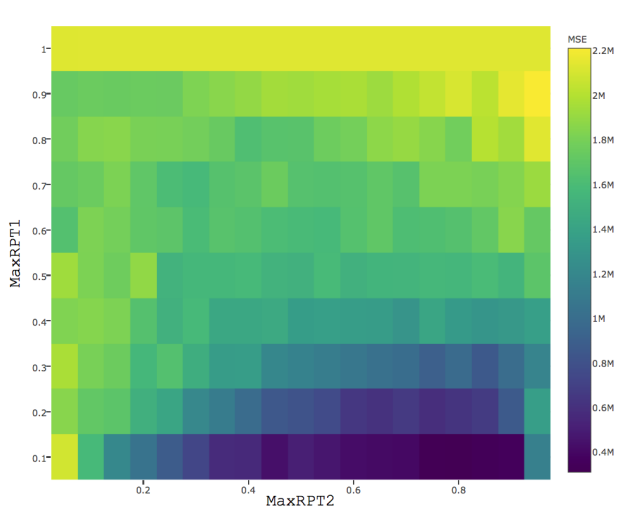




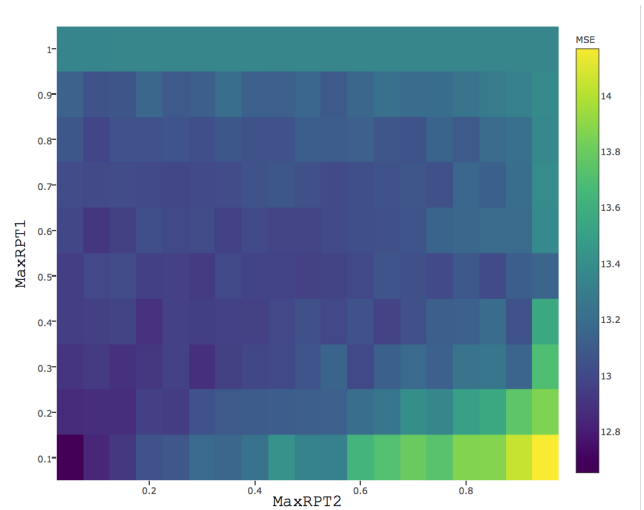
B.2 MaxRPT Parameter

In this subsection, we present the performance of ACPM for different values of MaxRPT for markets with two rounds using various UCI data sets. We refer to the first round MaxRPT as ‘MaxRPT1’ and second round MaxRPT as ‘MaxRPT2’.

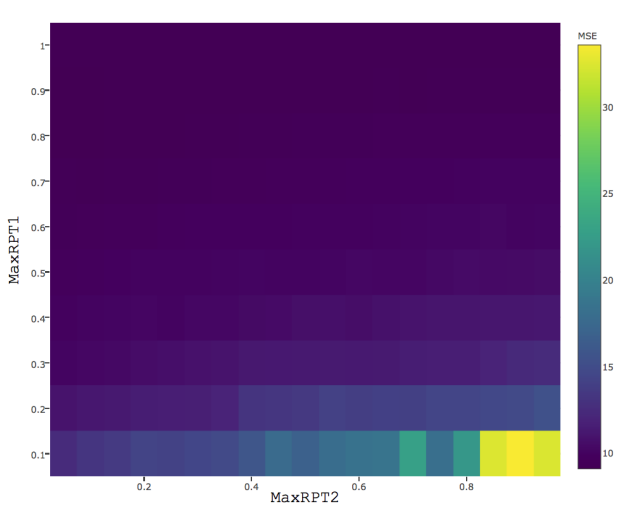
Bike Sharing



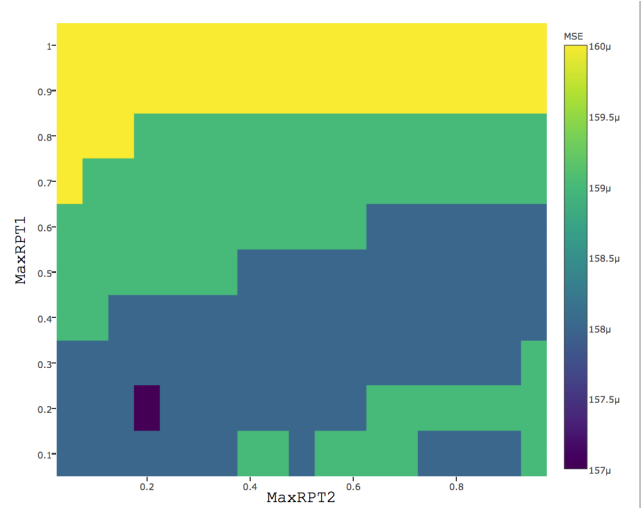
Auto MPG



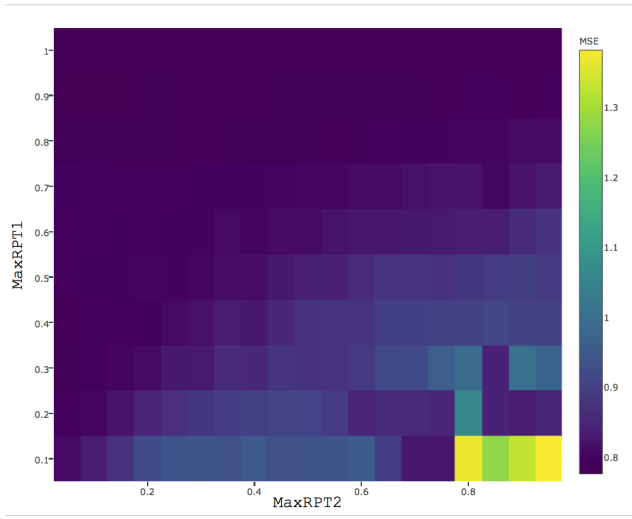
Yacht Hydrodynamics



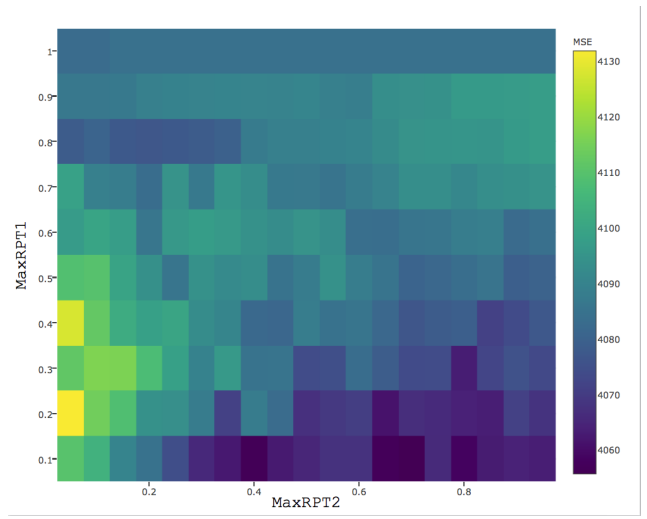
Istanbul Stock Exchange



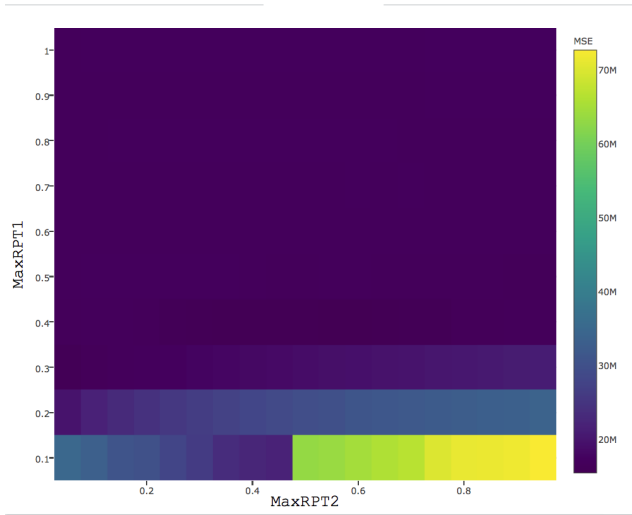
Servo



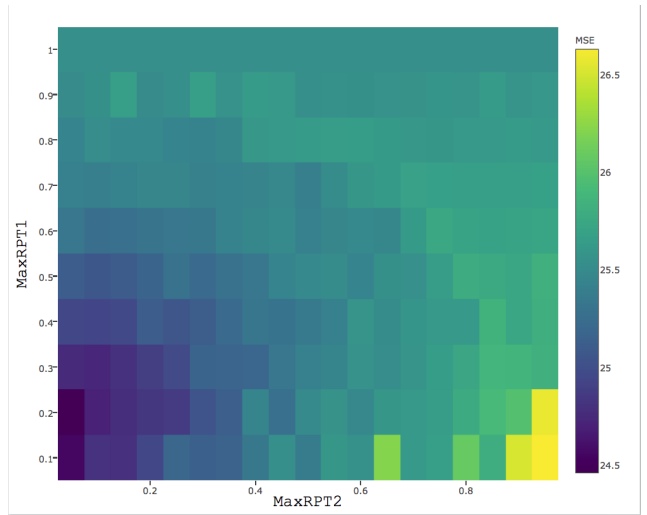
Forest Fires



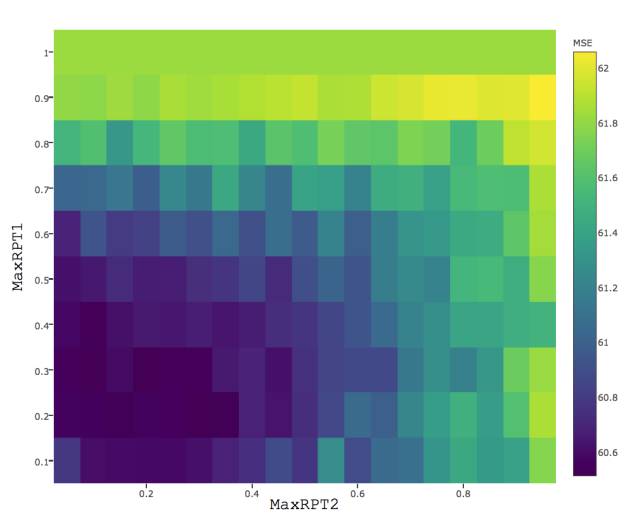
Automobile



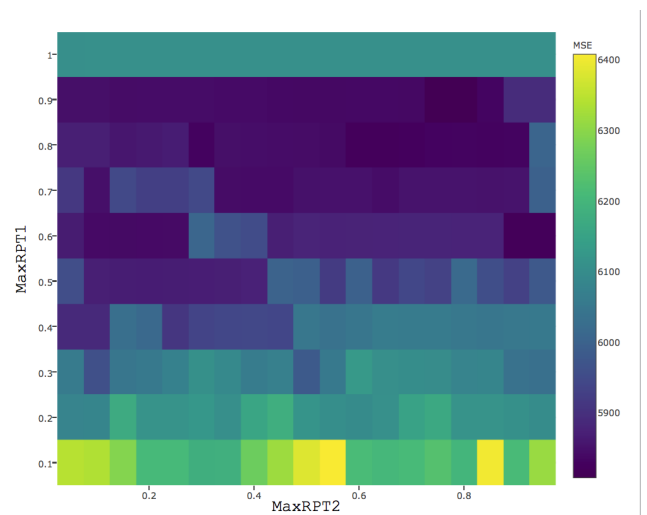
Housing



Airfoil Self Noise



Computer Hardware



Appendix C

Evaluation of ACPM Using Syndromic Surveillance Data Set

Chapter 4 (page 77) investigates the performance of ACPM with different market compositions. Therefore, four market types with different proportions of participants with regard to their data source quality are defined. Each market has 100 agents each accessing a distinct data source. The data sources used for each market are as follows.

Market Type 1 Information

- **Data From CDC:**

Unweighted ILI , Age 0-4, AGE 5-24, Age 65

- **Data From Google Flu Trend:**

Alabama, Arizona, California, Colorado, Connecticut, District.of.Columbia, Georgia, Indiana, Iowa, Kansas, Kentucky, Louisiana, Maryland, Massachusetts, Michigan, Minnesota, Missouri, Nebraska, Nevada, New.Jersey, Ohio, Oregon, South.Carolina, Tennessee, Utah, Washington, Wisconsin, HHS.Region.1 CT ME MA NH RI VT., HHS.Region.8 CO MT ND SD UT WY., HHS.Region.9 AZ CA HI NV., HHS.Region.10 AK ID OR WA., Birmingham AL, Phoenix AZ, Tempe AZ, Tucson AZ, Fresno CA, Los.Angeles CA, Oakland CA, Sacramento CA, San.Diego CA, San.Francisco CA, San.Jose CA, Santa.Clara CA, Sunnyvale CA, Denver CO, Gainesville FL, Miami FL, Orlando FL, Atlanta GA, Roswell GA, Chicago IL, Indianapolis IN, New.Orleans LA, Boston MA, Baltimore MD, St.Paul MN, Kansas.City MO, St.Louis MO, Charlotte NC, Durham NC, Raleigh NC, Omaha NE, Albuquerque NM, Las.Vegas NV, Albany NY, Buffalo NY, New.York NY, Rochester NY, Cincinnati OH, Cleveland OH, Columbus OH, Dayton OH, Oklahoma.City OK, Tulsa OK, Beaverton OR, Portland OR, Philadelphia PA, Nashville TN, Austin TX, Dallas TX, Houston TX, Irving TX, San.Antonio TX, Salt.Lake.City UT, Norfolk VA, Richmond VA, Seattle WA, Madison WI, Milwaukee WI, Illinois, HHS.Region.4 AL FL GA KY MS NC

SC TN., HHS.Region.6 AR LA NM OK TX., Washington DC, HHS.Region.3 DE DC MD PA VA WV.

Market Type 2 Information

- **Data From CDC:**

Unweighted ILI, Age 0-4, Age 5-24, Age 65

- **Data From Google Flu Trend:**

United.States, Alabama, Arizona, California, Colorado, Connecticut, District.of.Columbia, Florida, Illinois, Indiana, Kansas, Kentucky, Louisiana, Maryland, Massachusetts, Minnesota, Missouri, Nebraska, Nevada, New.Jersey, Ohio, Oregon, South.Carolina, Tennessee, Utah, Washington, Wisconsin, HHS.Region.1 CT ME MA NH RI VT., HHS.Region.3 DE DC MD PA VA WV., HHS.Region.4 AL FL GA KY MS NC SC TN., HHS.Region.6 AR LA NM OK TX., HHS.Region.8 CO MT ND SD UT WY., HHS.Region.9 AZ CA HI NV., HHS.Region.10 AK ID OR WA., Birmingham AL, Phoenix AZ, Tempe AZ, Tucson AZ, Fresno CA, Los.Angeles CA, Oakland CA, San.Diego CA, San.Francisco CA, San.Jose CA, Santa.Clara CA, Sunnyvale CA, Denver CO, Washington DC, Gainesville FL, Miami FL, Orlando FL, Tampa FL, Atlanta GA, Roswell GA, New.Orleans LA, Boston MA, Baltimore MD, St.Paul MN, St.Louis MO, Charlotte NC, Durham NC, Raleigh NC, Omaha NE, Albuquerque NM, Las.Vegas NV, Albany NY, Buffalo NY, New.York NY, Rochester NY, Cincinnati OH, Cleveland OH, Columbus OH, Dayton OH, Oklahoma.City OK, Tulsa OK, Beaverton OR, Portland OR, Philadelphia PA, Pittsburgh PA, Nashville TN, Austin TX, Dallas TX, Irving TX, San.Antonio TX, Arlington VA, Norfolk VA, Richmond VA, Madison WI, Milwaukee WI, Michigan, Chicago IL, Houston TX, Seattle WA, Iowa

Market Type 3 Information

- **Data From CDC:**

ILI Total, Total Patients, Number of Providers, Weighted ILI, Unweighted ILI, Age 0-4, Age 5-24, Age 65

- **Data From Google Trend:**

Sore Throat, Cough, Fever, Flu, Flu Symptoms

- **Data From Google Flu Trend:**

Alabama, Arizona, California, Colorado, Connecticut, District.of.Columbia, Hawaii, Indiana, Kansas, Kentucky, Louisiana, Maryland, Massachusetts, Minnesota, Missouri, Nebraska, Nevada, New.Hampshire, New.Jersey, Ohio, Oregon, South.Carolina, Tennessee, Utah, Washington, Wisconsin, HHS.Region.1 CT ME MA NH RI VT., HHS.Region.8 CO MT ND SD UT WY., HHS.Region.9 AZ CA HI NV., HHS.Region.10 AK ID OR WA., Birmingham AL, Phoenix AZ, Tempe AZ, Tucson AZ, Fresno CA, Los.Angeles CA, Oakland CA, San.Diego CA, San.Francisco CA, San.Jose CA, Santa.Clara CA, Sunnyvale CA, Denver CO, Gainesville FL, Miami FL, Orlando FL, Atlanta GA, Roswell GA, Honolulu HI, Indianapolis IN, New.Orleans LA, Boston MA, Somerville MA, Baltimore MD, St.Paul MN, Kansas.City MO, St.Louis MO, Charlotte NC, Durham NC, Raleigh NC, Omaha NE, Newark NJ, Albuquerque NM, Las.Vegas NV, Albany NY, Buffalo NY, New.York NY, Rochester NY, Cincinnati OH, Cleveland OH, Columbus OH, Dayton OH, Oklahoma.City OK, Tulsa OK, Beaverton OR, Portland OR, Philadelphia PA, Nashville TN, Austin TX, Dallas TX, Irving TX, San.Antonio TX, Norfolk VA, Richmond VA, Bellevue WA, Madison WI, Milwaukee WI

Market Type 4 Information

- **Data From CDC:**

ILI Total, Total Patients, Number of Providers, Weighted ILI, Unweighted ILI, Age 0-4, Age 5-24, Age 65, Week Number

- **Data From Google Trend:**

Sore Throat, Cough, Fever, Flu, Flu Symptoms

- **Data From Google Flu Trend:** United.States, Alabama, California, Colorado, Connecticut, District.of.Columbia, Florida, Hawaii, Illinois, Indiana, Kansas, Kentucky, Louisiana, Maryland, Minnesota, Missouri, Nebraska, Nevada, New.Hampshire, New.Jersey, Ohio, Oregon, South.Carolina, Tennessee, Utah, Wisconsin, HHS.Region.1, CT, ME, MA, NH, RI, VT., HHS.Region.3, DE, DC, MD, PA, VA, WV., HHS.Region.4, AL, FL, GA, KY, MS, NC, SC, TN., HHS.Region.6, AR, LA, NM, OK, TX., HHS.Region.8, CO, MT, ND, SD, UT, WY., HHS.Region.9, AZ, CA, HI, NV., HHS.Region.10, AK, ID, OR, WA., Birmingham, AL, Phoenix, AZ, Tempe, AZ, Tucson, AZ, Fresno, CA, Los.Angeles, CA, Oakland, CA, San.Diego, CA, San.Francisco, CA, Sunnyvale, CA, Denver, CO, Washington, DC, Gainesville, FL, Miami, FL, Orlando, FL, Tampa, FL, Atlanta, GA, Roswell, GA, Honolulu, HI, New.Orleans, LA, Boston, MA, Somerville, MA, Baltimore, MD, St.Paul, MN, St.Louis, MO, Charlotte, NC, Durham, NC, Raleigh, NC, Newark, NJ, Albuquerque, NM, Las.Vegas, NV, Albany, NY, Buffalo, NY, New.York, NY, Rochester, NY, Cincinnati,

OH, Cleveland, OH, Columbus, OH, Dayton, OH, Oklahoma.City, OK, Beaverton, OR,
Portland, OR, Philadelphia, PA, Pittsburgh, PA, Austin, TX, Dallas, TX, Irving, TX,
San.Antonio, TX, Arlington, VA, Norfolk, VA, Bellevue, WA, Madison, WI, Milwaukee,
WI

Appendix D

Evaluation of ACPM Using Artificial Data

In section 4.5 (page 88), for sake of clarity, the results of experiments of B2 to B6 are presented based on one run. In here, we present result of same experiments averaged over 100 runs, as the process of generating noise in these experiments is not deterministic. In these figures, each coloured solid line represents the accumulated error of a participating agent. The dotted line shows the accumulated error of ACPM where the width of dots shows the standard error. The participants' accumulated errors (and hence the lines) are very close to each other, almost forming a single line.

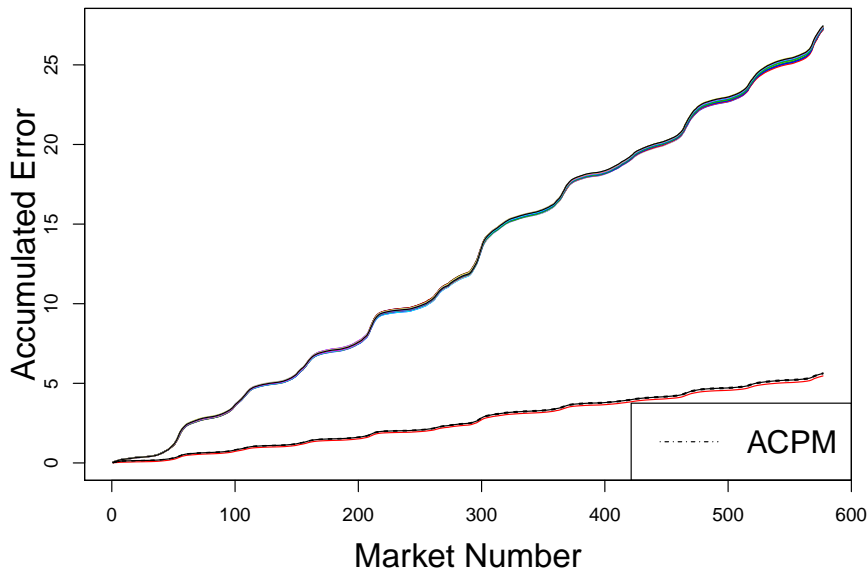


Figure D-1: Scenario B2 - Comparing accumulated error of ACPM and its 40 participants. 39 agents access data with 50% noise and one agents access data with 10% noise.

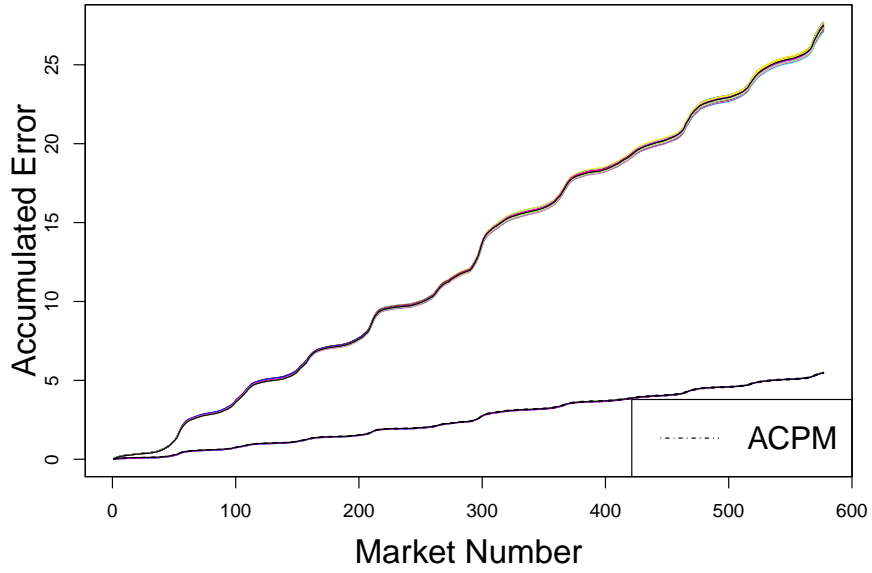


Figure D-2: Scenario B3- Comparing accumulated error of ACPM and its 40 participants. 35 agents access data with 50% noise and 5 agents access data with 10% noise.

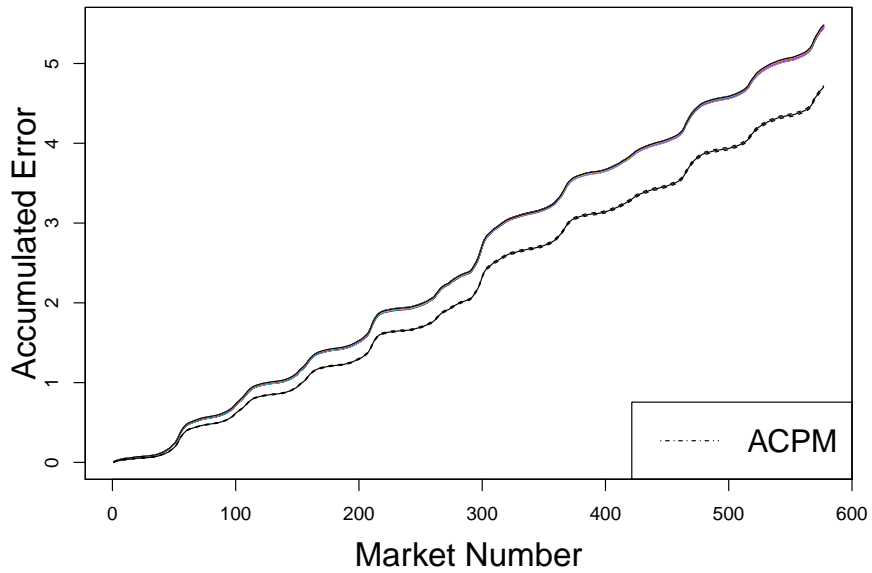


Figure D-3: Scenario B4- Comparing accumulated error of ACPM and its 40 participants each accessing data with 10% noise.

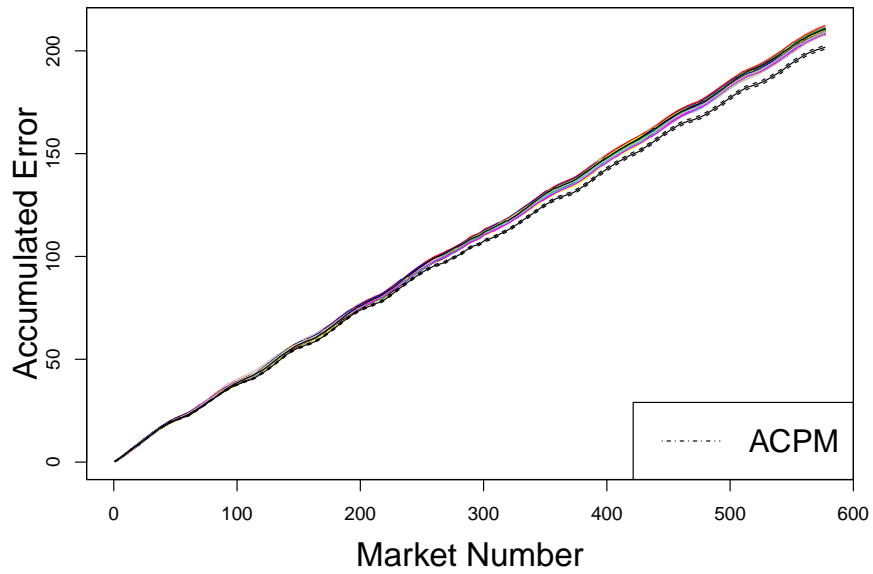


Figure D-4: Scenario B5- Comparing accumulated error of ACPM and its 40 participants each accessing random data with uniform distribution.

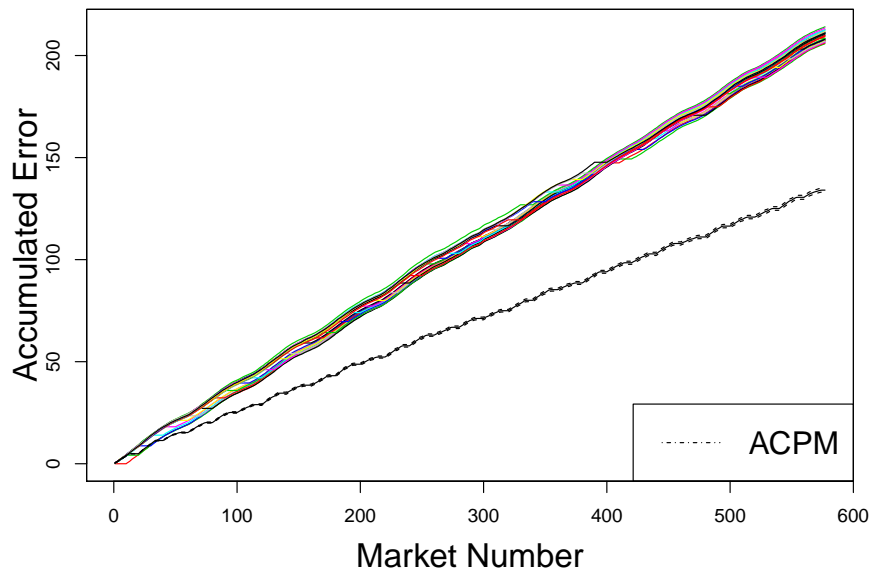


Figure D-5: Scenario B6 -Comparing accumulated error of ACPM and its 40 participants each accessing random data with uniform distribution. For each 10 consecutive markets one of the agents accesses the correct answer.

Appendix E

Evaluation of ACPM on UCI Data Sets

Chapter 4 (page 77) demonstrates the performance of ACPM and several benchmarks using a number of experiments on UCI data sets in Table 4 (page 77). In that table we use MSE as metric. In this appendix, we graph the performance of ACPM and the benchmarks using both MSE and MAE.

Figure E-1: Part I. MSE of ACPM and the Benchmarks

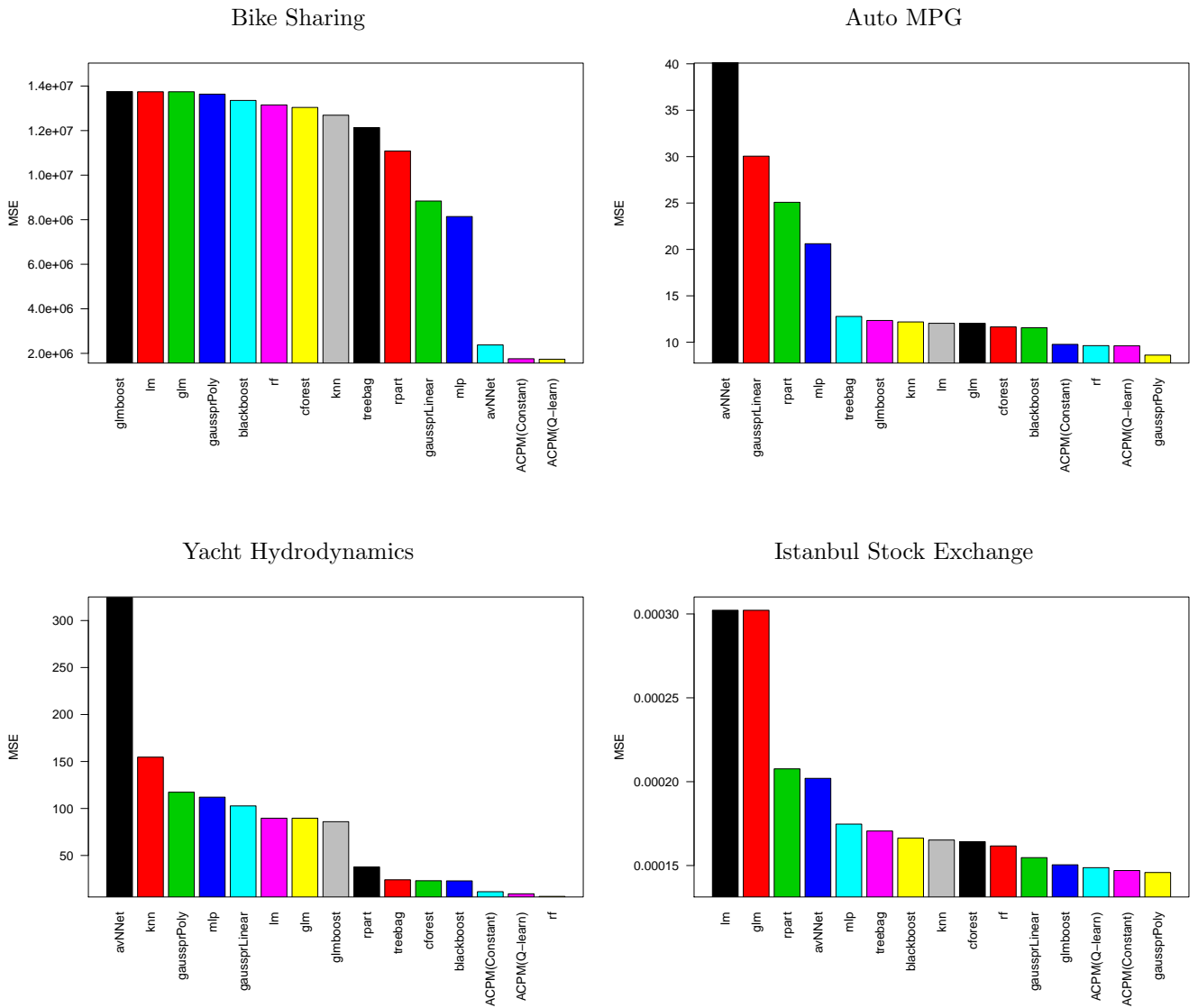


Figure E-2: Part II. MSE of ACPM and the Benchmarks

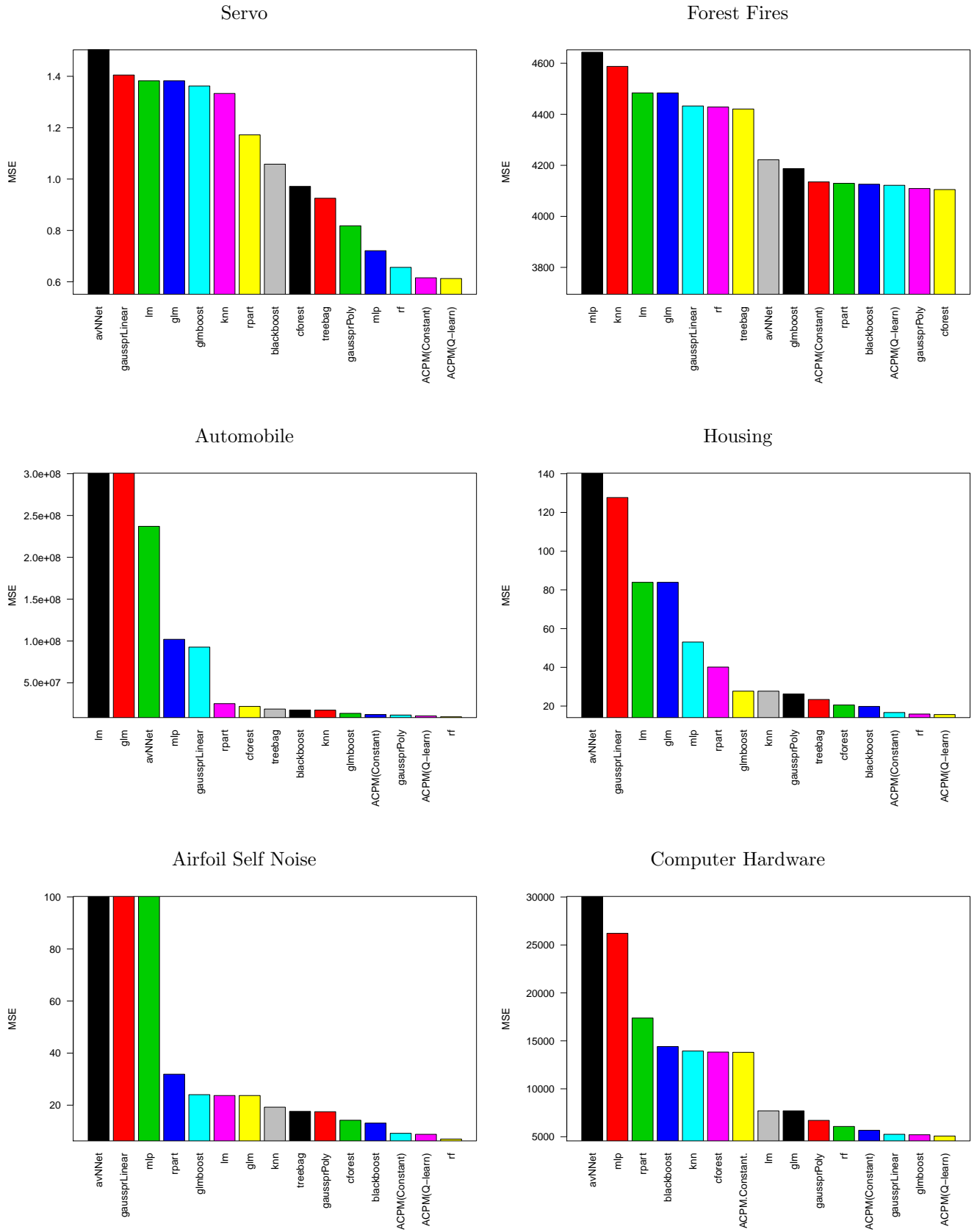


Figure E-3: Part I. MAE of ACPM and the Benchmarks

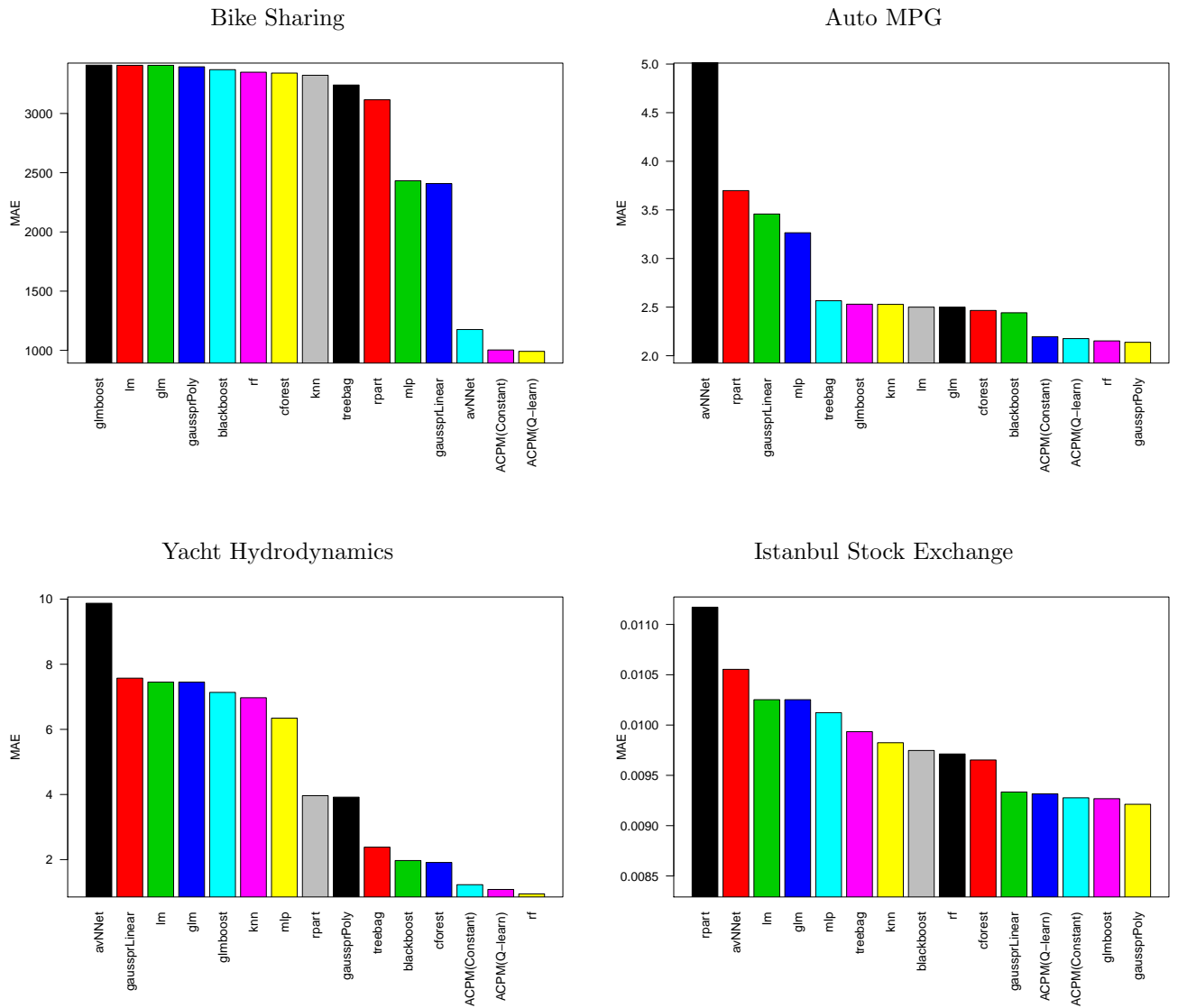
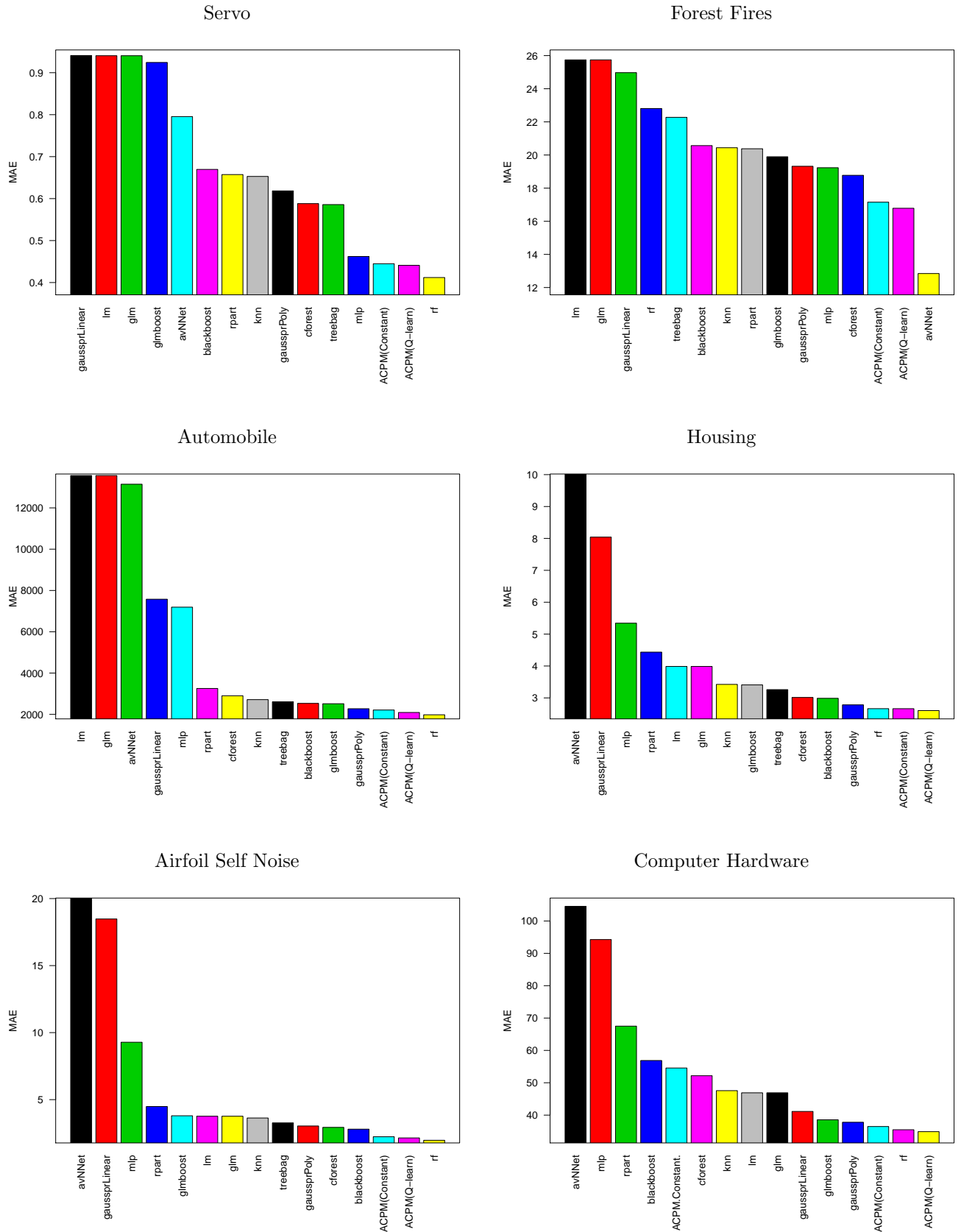


Figure E-4: Part II. MAE of ACPM and the Benchmarks



Appendix F

Evaluation of ACPM Using Syndromic Surveillance Data Set

Chapter 5 (page 111) evaluates ACPM using syndromic surveillance application domain. In our experiments, agents have access to Google Flu Trend prediction for different places mentioned below.

United States, Alabama, Alaska, Arizona, Arkansas, California, Colorado, Connecticut, Delaware, District of Columbia, Florida, Georgia, Hawaii, Idaho, Illinois, Indiana, Iowa, Kansas, Kentucky, Louisiana, Maine, Maryland, Massachusetts, Michigan, Minnesota, Mississippi, Missouri, Montana, Nebraska, Nevada, New Hampshire, New Jersey, New Mexico, New York, North Carolina, North Dakota, Ohio, Oklahoma, Oregon, Pennsylvania, Rhode Island, South Carolina, South Dakota, Tennessee, Texas, Utah, Vermont, Virginia, Washington, West Virginia, Wisconsin, Wyoming, HHS Region 1, HHS Region 2, HHS Region 3, HHS Region 4, HHS Region 5, HHS Region 6, HHS Region 7, HHS Region 8, HHS Region 9, HHS Region 10, Anchorage, AK, Birmingham, AL, Little Rock, AR, Mesa, AZ, Phoenix, AZ, Scottsdale, AZ, Tempe, AZ, Tucson, AZ, Berkeley, CA, Fresno, CA, Irvine, CA, Los Angeles, CA, Oakland, CA, Sacramento, CA, San Diego, CA, San Francisco, CA, San Jose, CA, Santa Clara, CA, Sunnyvale, CA, Colorado Springs, CO, Denver, CO, Washington, DC, Gainesville, FL, Jacksonville, FL, Miami, FL, Orlando, FL, Tampa, FL, Atlanta, GA, Roswell, GA, Honolulu, HI, Des Moines, IA, Boise, ID, Chicago, IL, Indianapolis, IN, Wichita, KS, Lexington, KY, Baton Rouge, LA, New Orleans, LA, Boston, MA, Somerville, MA, Baltimore, MD, Grand Rapids, MI, St Paul, MN, Kansas City, MO, Springfield, MO, St Louis, MO, Jackson, MS, Cary, NC, Charlotte, NC, Durham, NC, Greensboro, NC, Raleigh, NC, Lincoln, NE, Omaha, NE, Newark, NJ, Albuquerque, NM, Las Vegas, NV, Reno, NV, Albany, NY, Buffalo, NY, New York, NY, Rochester, NY, Cleveland, OH, Columbus, OH, Dayton, OH, Oklahoma City, OK, Tulsa, OK, Beaverton, OR, Eugene, OR, Portland, OR, Philadelphia, PA, Pittsburgh, PA, State College, PA, Providence, RI, Columbia, SC, Greenville, SC, Knoxville, TN, Memphis, TN, Nashville, TN, Austin, TX, Dallas, TX, Ft Worth, TX, Houston, TX, Irving, TX, Lubbock, TX, Plano, TX, San Antonio, TX, Salt Lake City, UT, Arlington, VA, Norfolk, VA, Reston, VA, Richmond, VA, Bellevue, WA, Seattle, WA, Spokane, WA, Madison, WI, Milwaukee, WI

Appendix G

Comparing ACPM and Google Flu Trend

Chapter 5 (page 111) compares the performance of ACPM and Google Flu Trend. This appendix illustrates the prediction and errors of ACPM and Google Flu Trend for each year of the experimented interval (2004 to 2015).

Figure G-1: Comparing Prediction of ACPM and Google Flu Trend in 2004

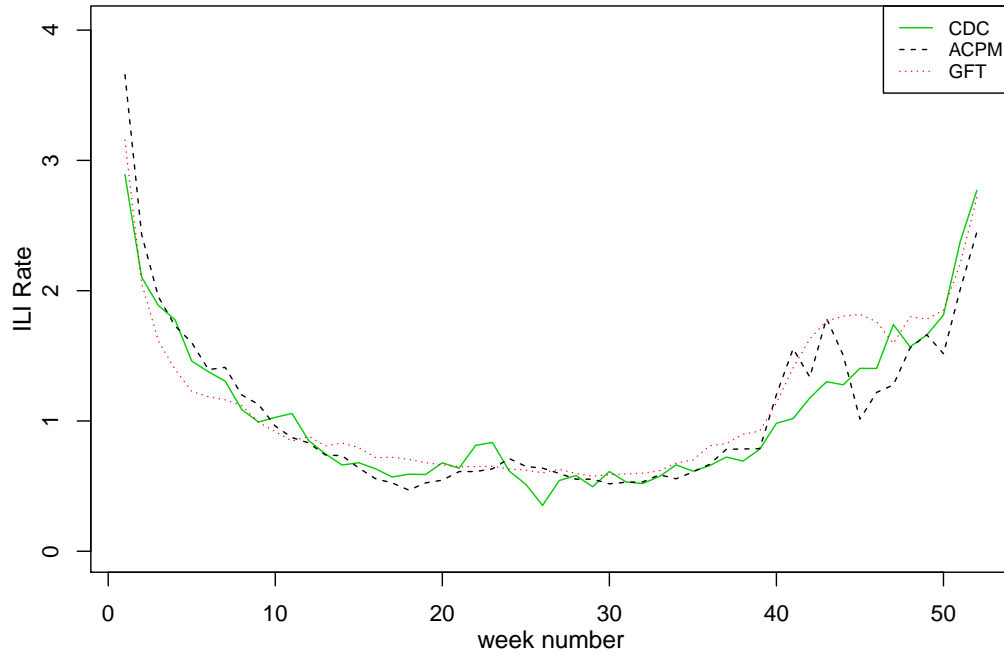


Figure G-2: Comparing ACPM Error and Google Flu Trend Error in 2004

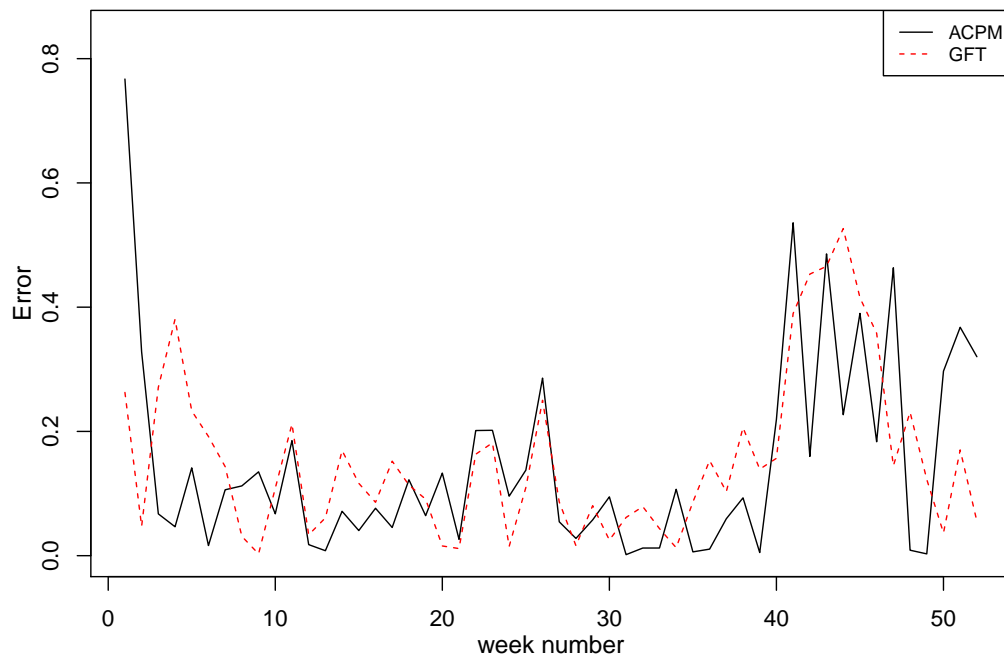


Figure G-3: Comparing Prediction of ACPM and Google Flu Trend in 2005

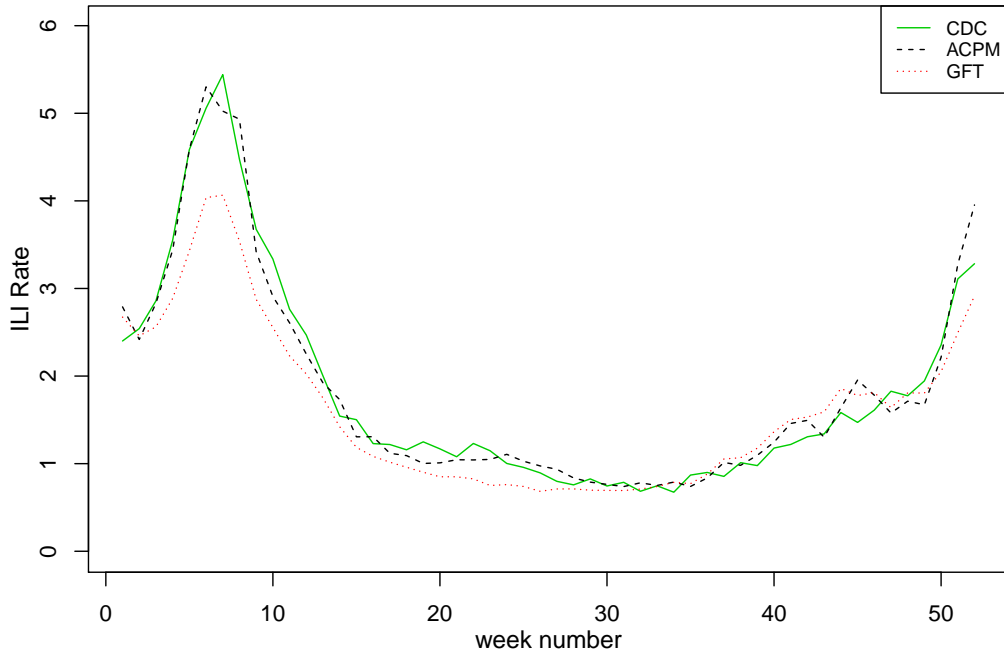


Figure G-4: Comparing ACPM Error and Google Flu Trend Error in 2005

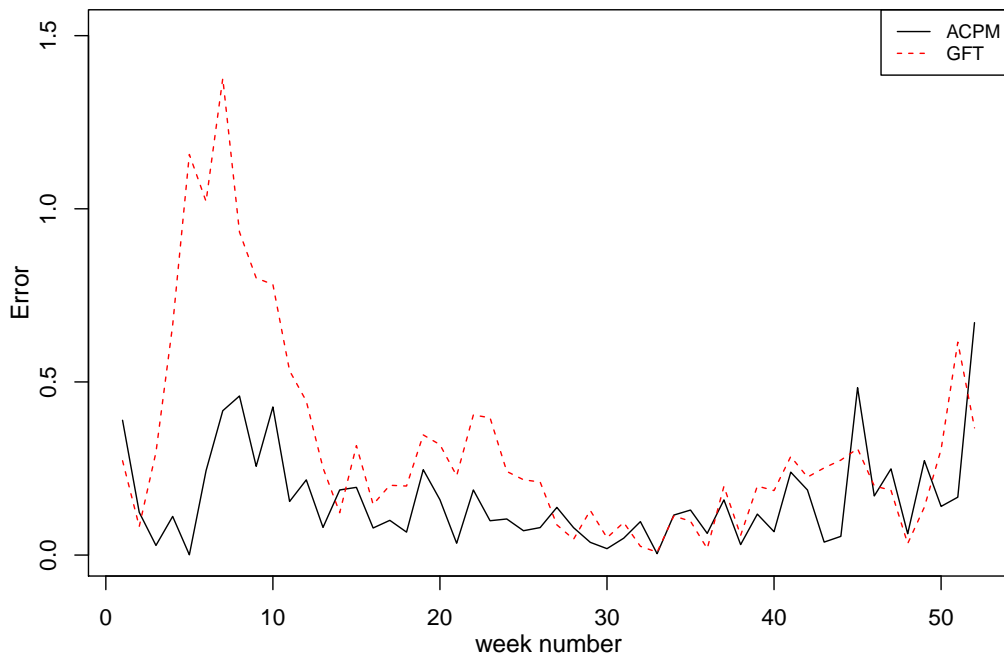


Figure G-5: Comparing Prediction of ACPM and Google Flu Trend in 2006

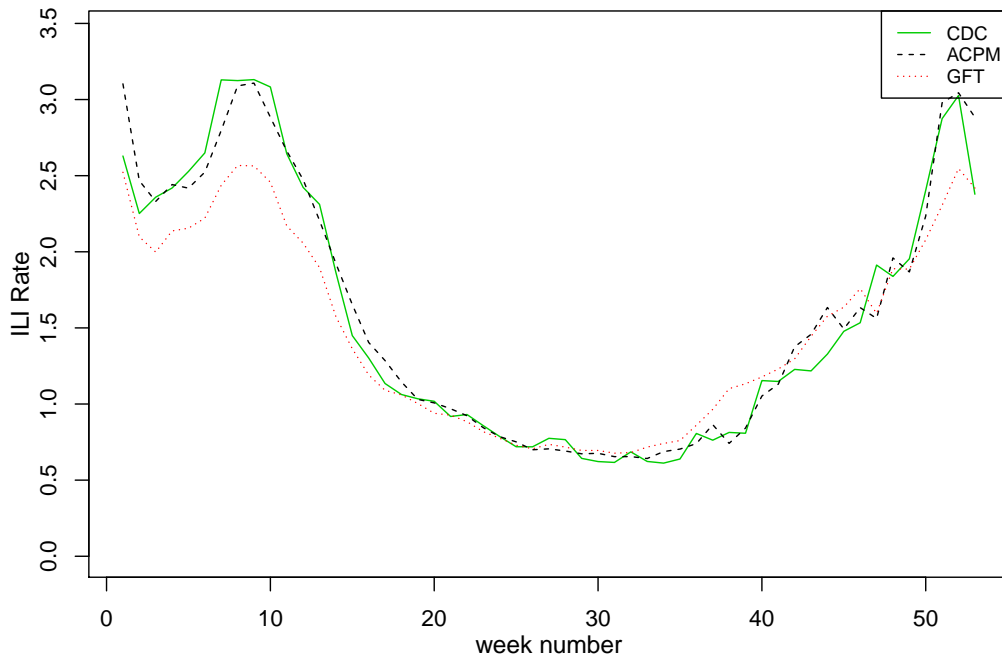


Figure G-6: Comparing ACPM Error and Google Flu Trend Error in 2006

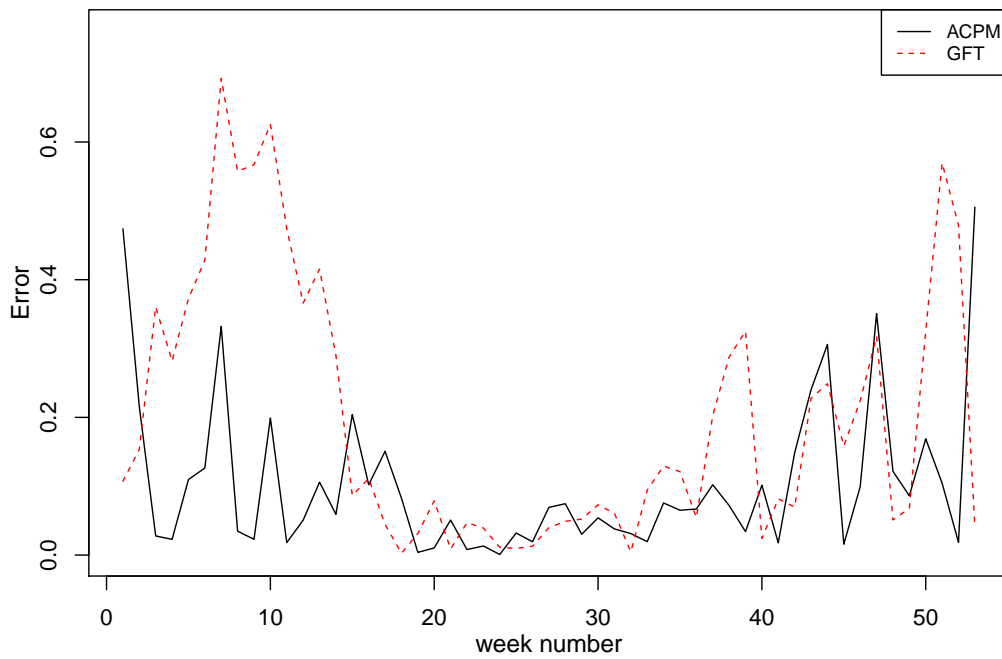


Figure G-7: Comparing Prediction of ACPM and Google Flu Trend in 2007

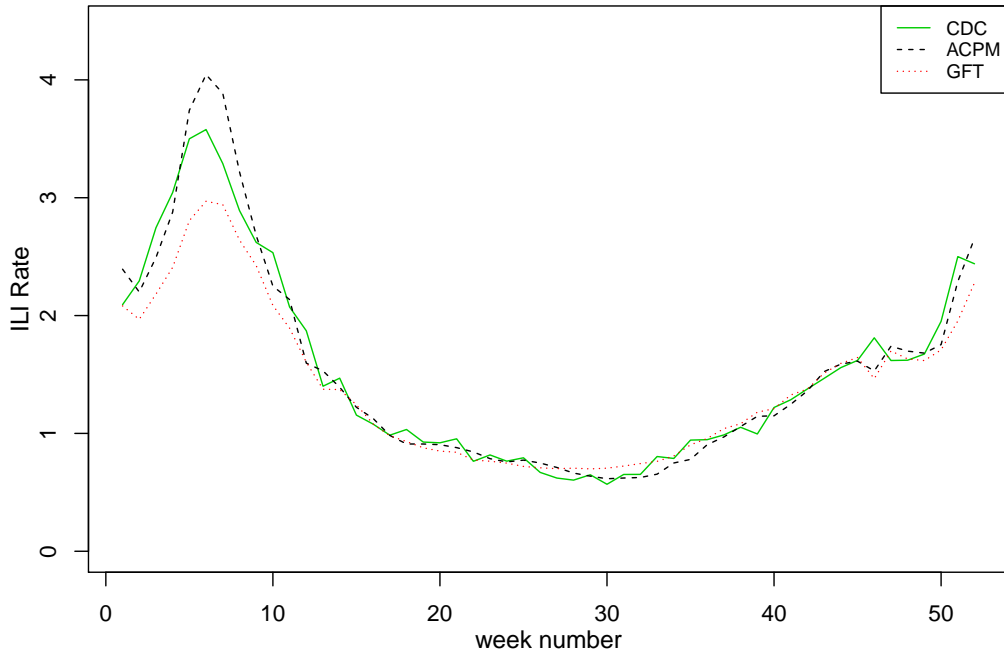


Figure G-8: Comparing ACPM error and Google Flu Trend Error in 2007

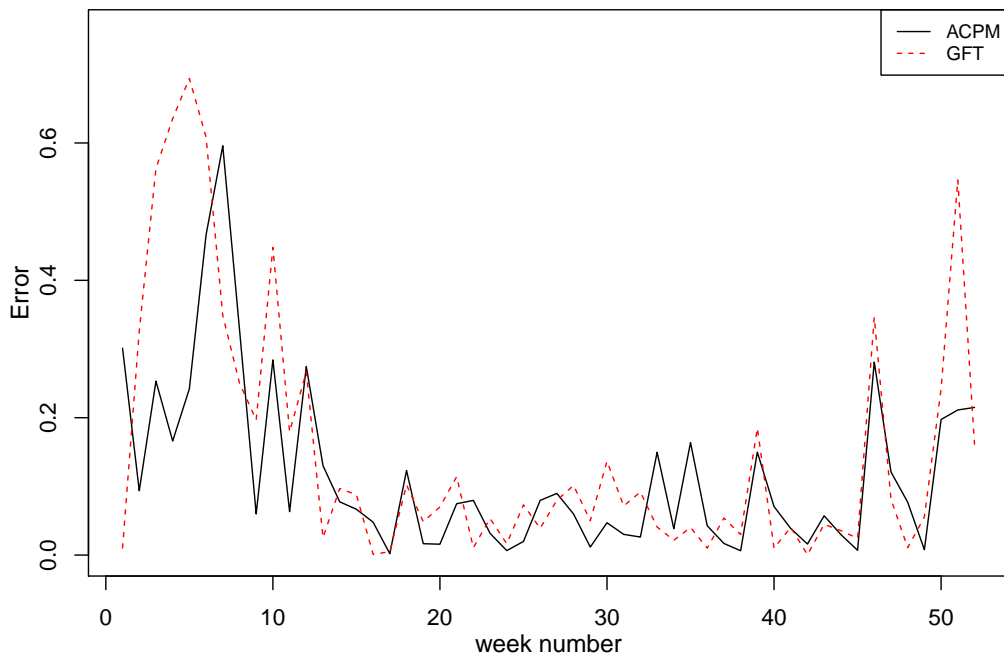


Figure G-9: Comparing Prediction of ACPM and Google Flu Trend in 2008

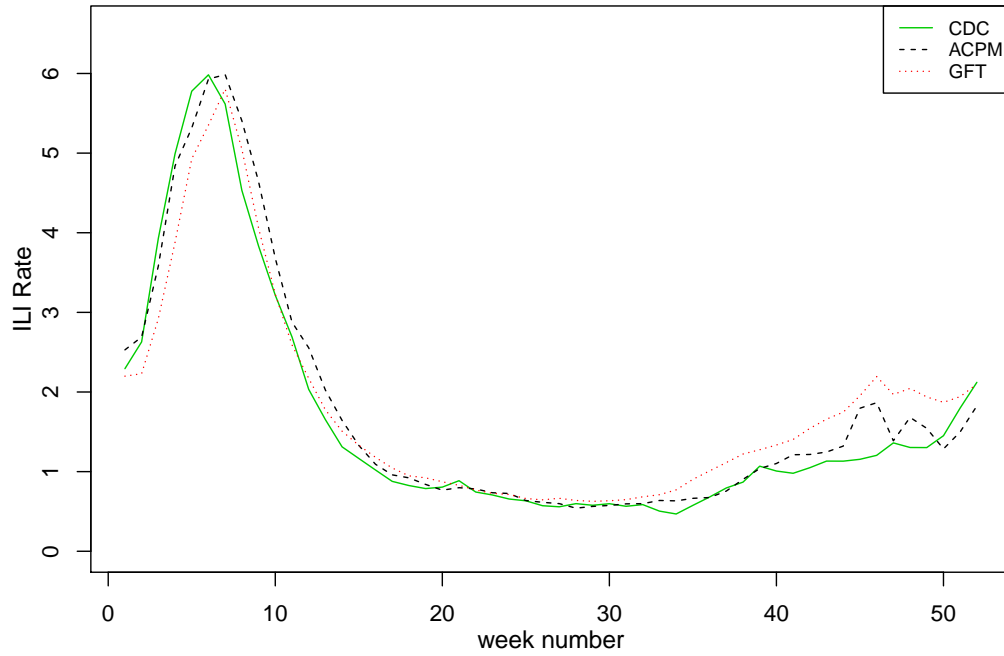


Figure G-10: Comparing ACPM Error and Google Flu Trend Error in 2008

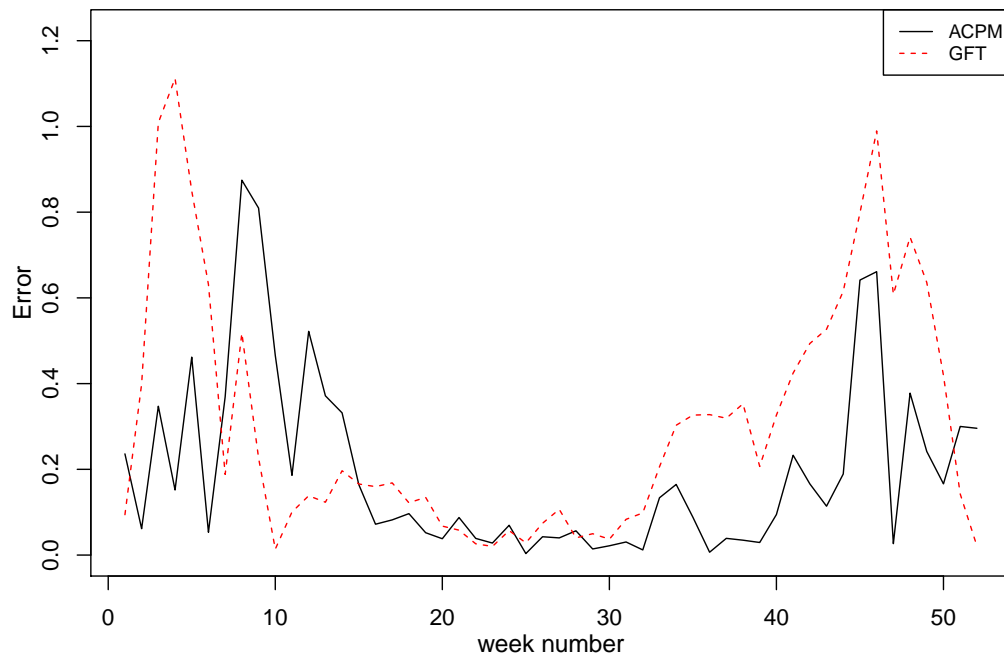


Figure G-11: Comparing Prediction of ACPM and Google Flu Trend in 2009

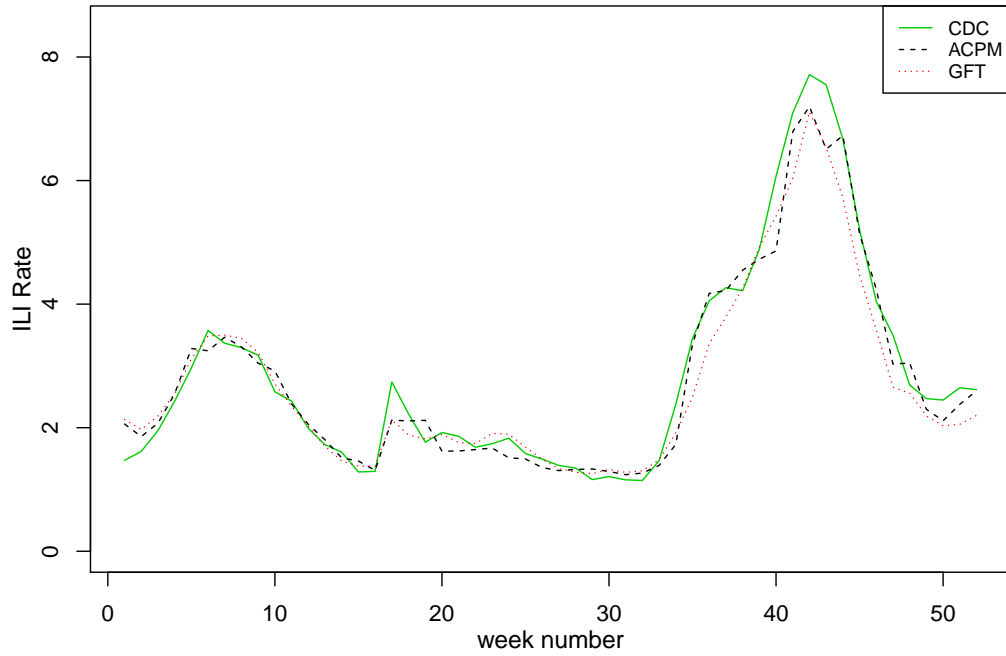


Figure G-12: Comparing ACPM Error and Google Flu Trend Error in 2009

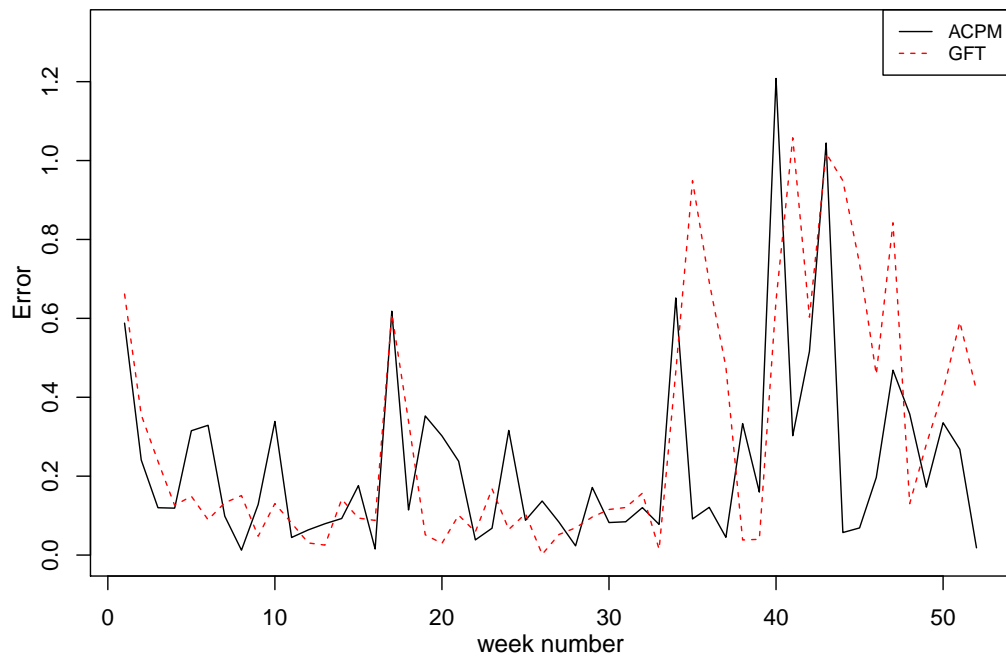


Figure G-13: Comparing Prediction of ACPM and Google Flu Trend in 2010

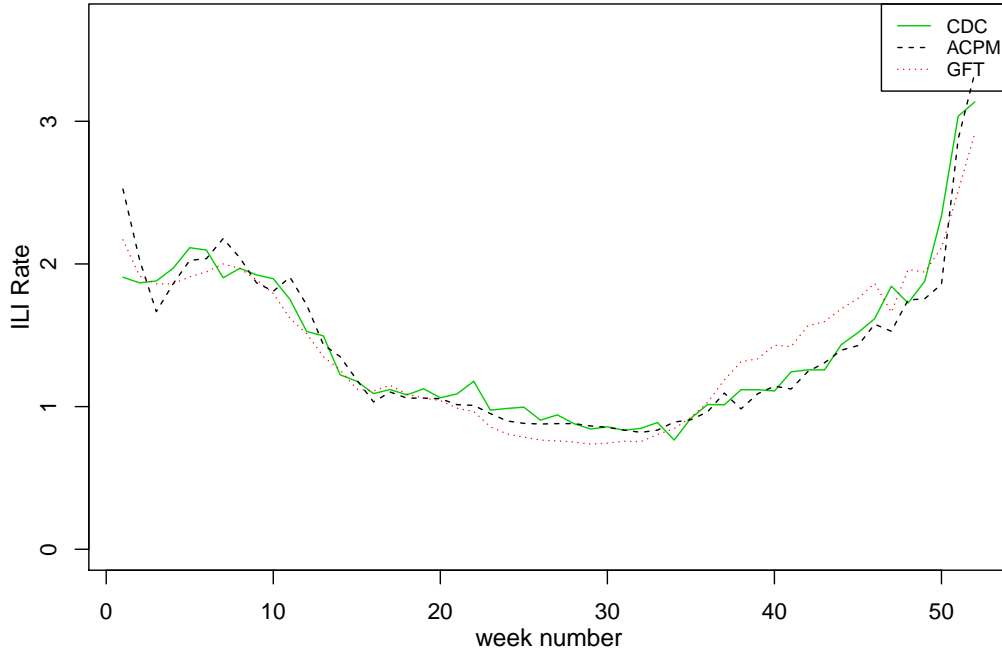


Figure G-14: Comparing ACPM Error and Google Flu Trend Error in 2010

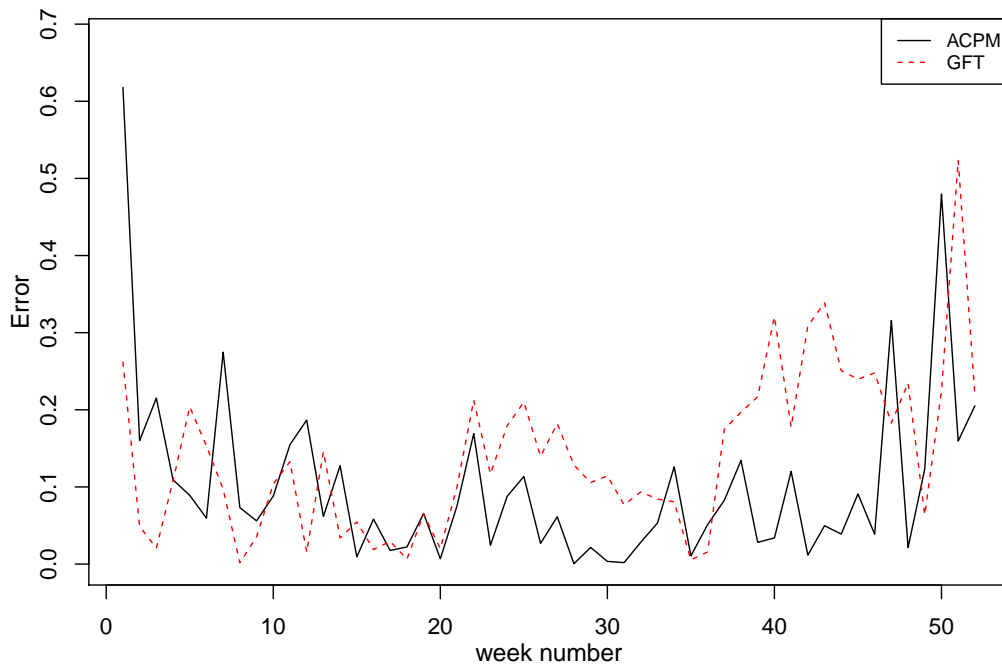


Figure G-15: Comparing Prediction of ACPM and Google Flu Trend in 2011

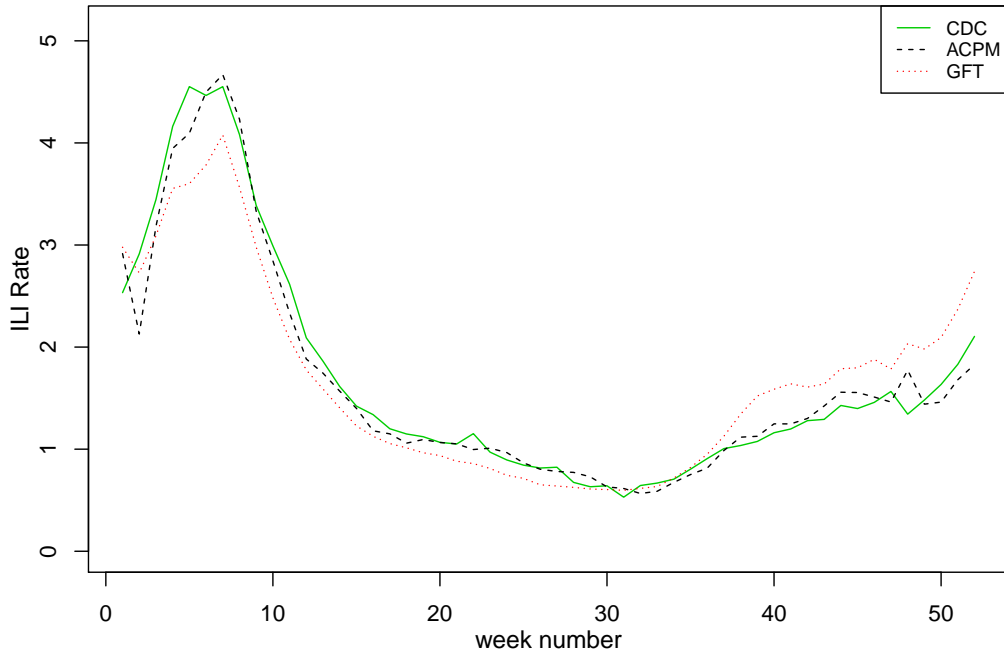


Figure G-16: Comparing ACPM Error and Google Flu Trend Error in 2011

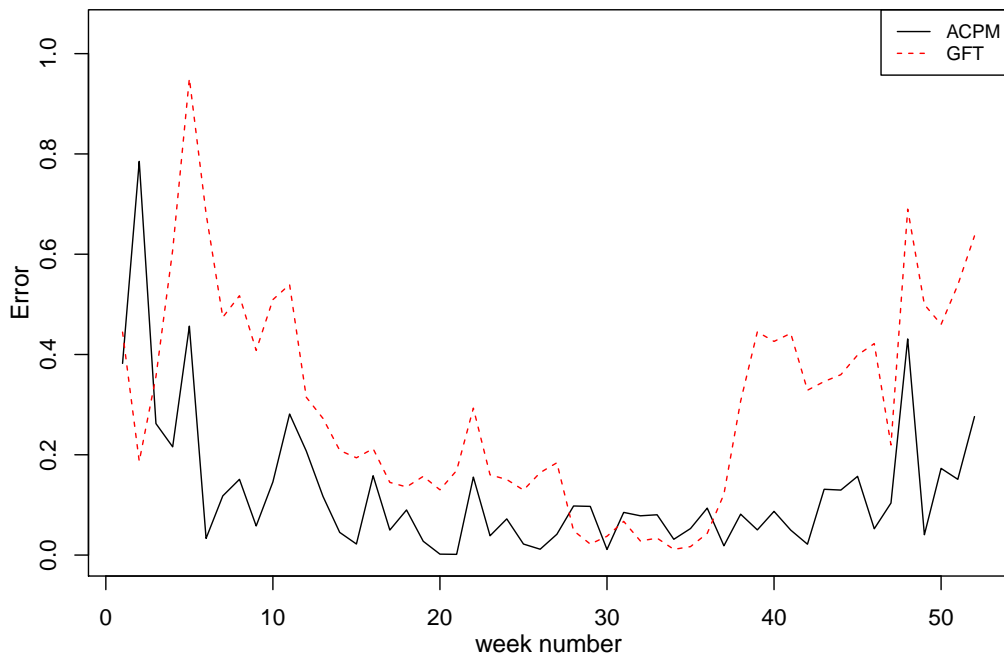


Figure G-17: Comparing Prediction of ACPM and Google Flu Trend in 2012

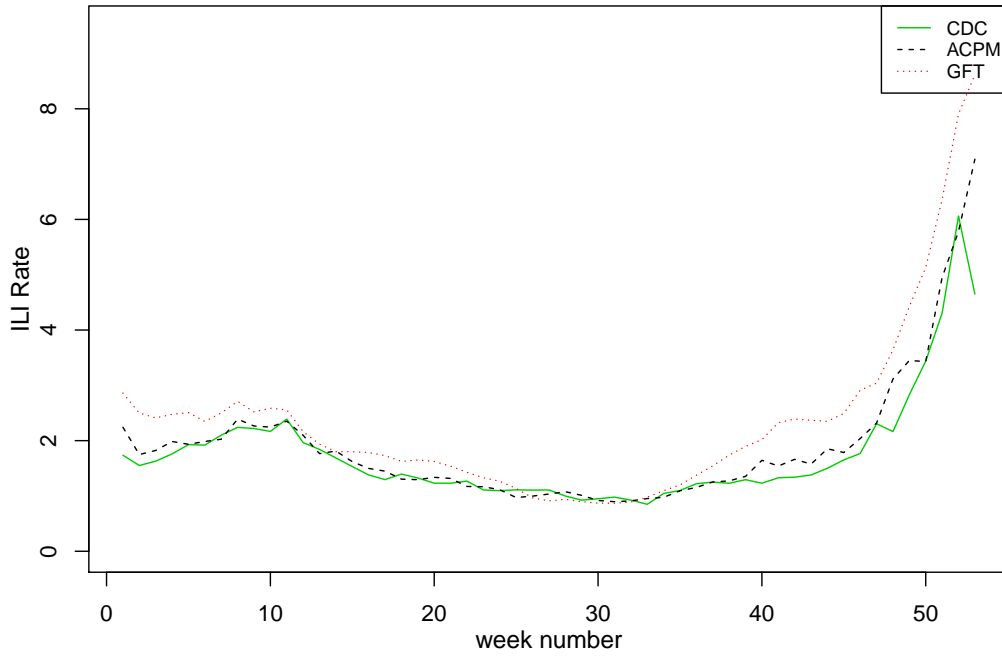


Figure G-18: Comparing ACPM Error and Google Flu Trend Error in 2012

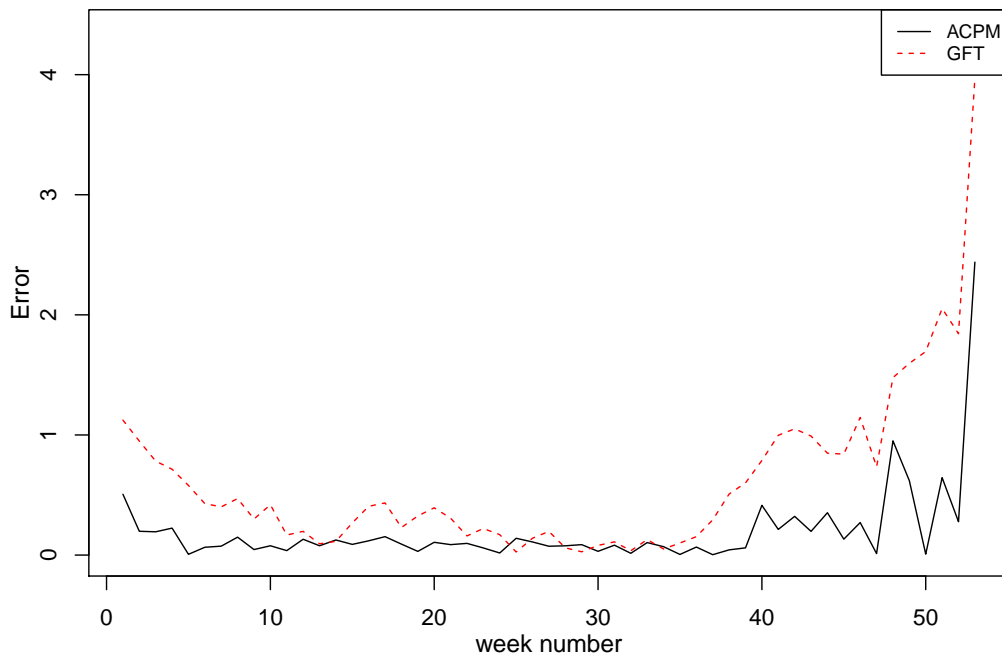


Figure G-19: Comparing Prediction of ACPM and Google Flu Trend in 2013

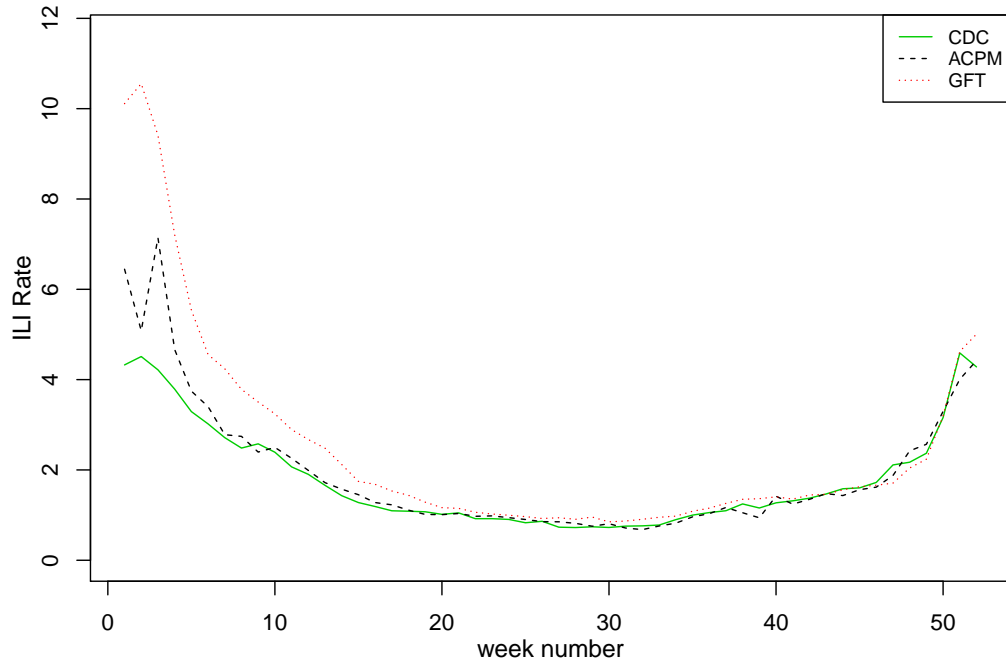


Figure G-20: Comparing ACPM Error and Google Flu Trend Error in 2013

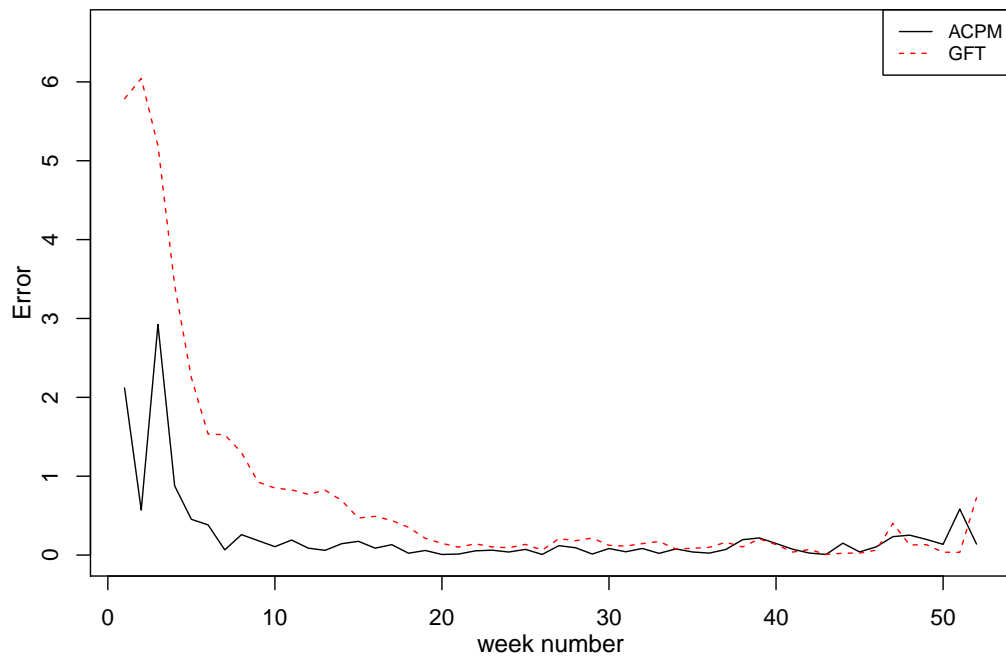


Figure G-21: Comparing Prediction of ACPM and Google Flu Trend in 2014

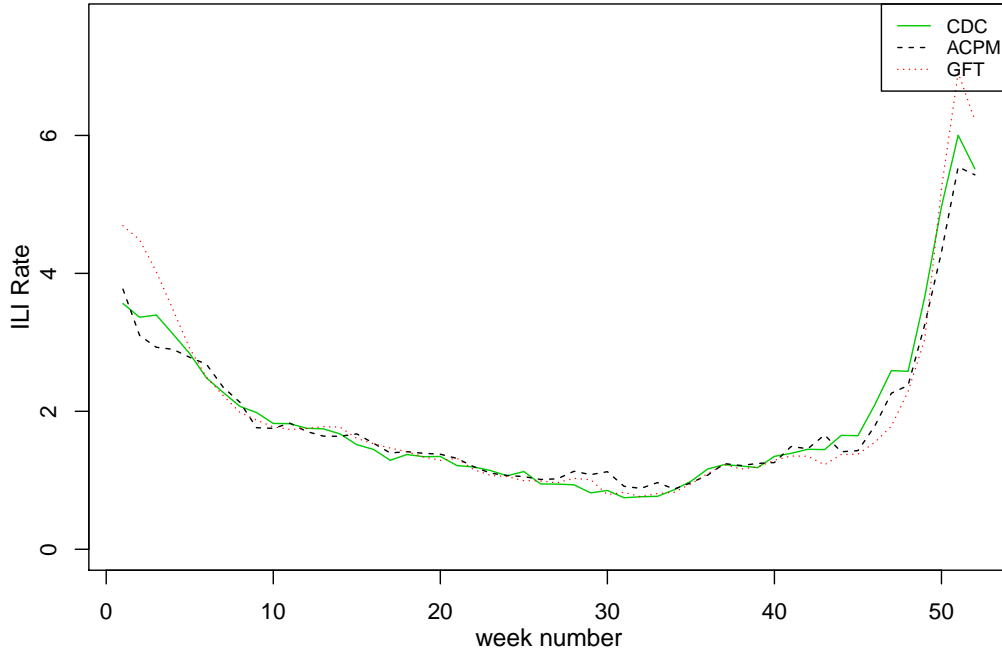


Figure G-22: Comparing ACPM Error and Google Flu Trend Error in 2014

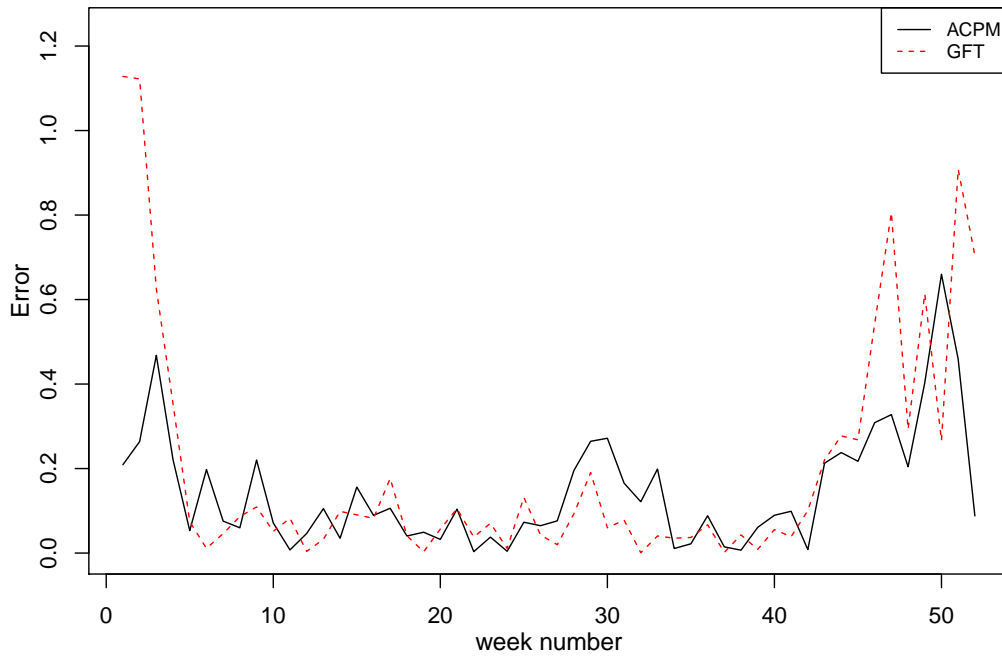


Figure G-23: Comparing Prediction of ACPM and Google Flu Trend in 2015

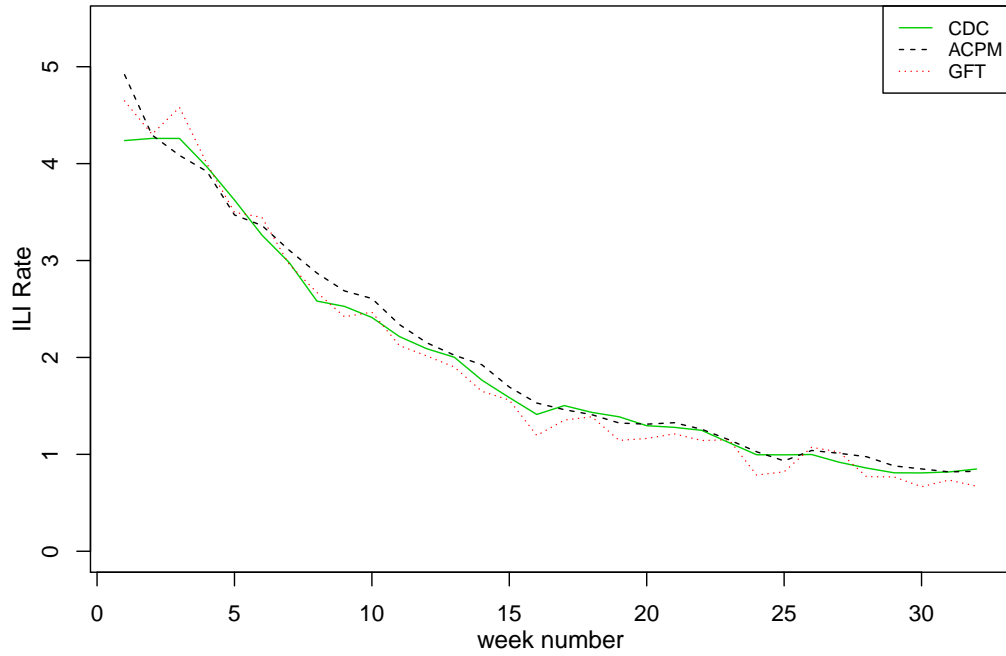
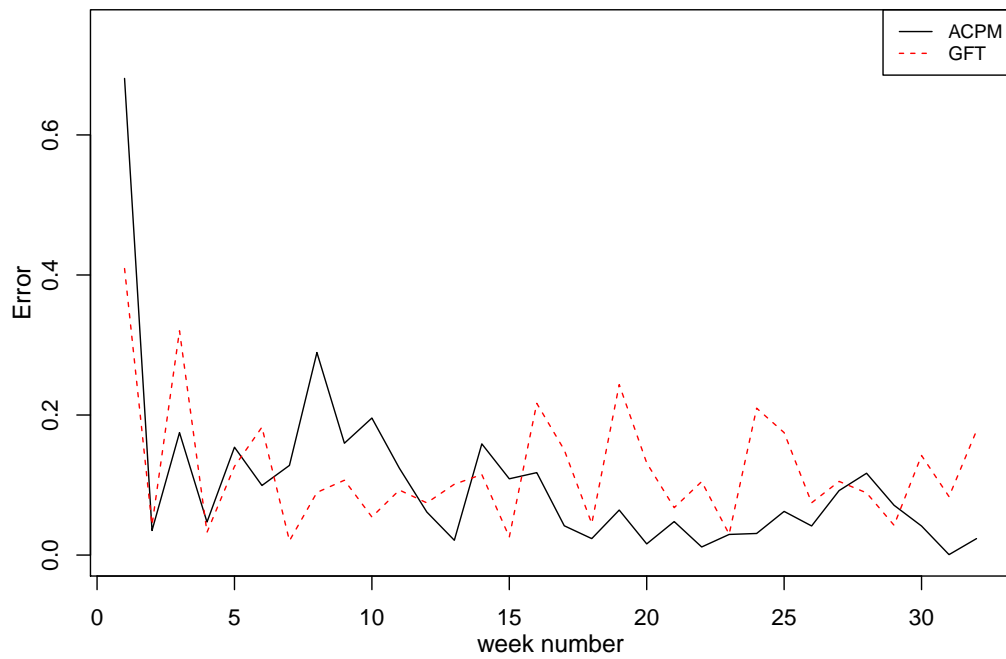


Figure G-24: Comparing ACPM Error and Google Flu Trend Error in 2015



Appendix H

Comparing ACPM and GP

Chapter 5 (page 111) compares the performance of ACPM and GP model [Lampos et al., 2015]. This appendix illustrate comparison of their predictions and errors for each Flu Season, between 2008 and 2013.

Figure H-1: Comparing Prediction of ACPM and GP in Flu Season 2008-9

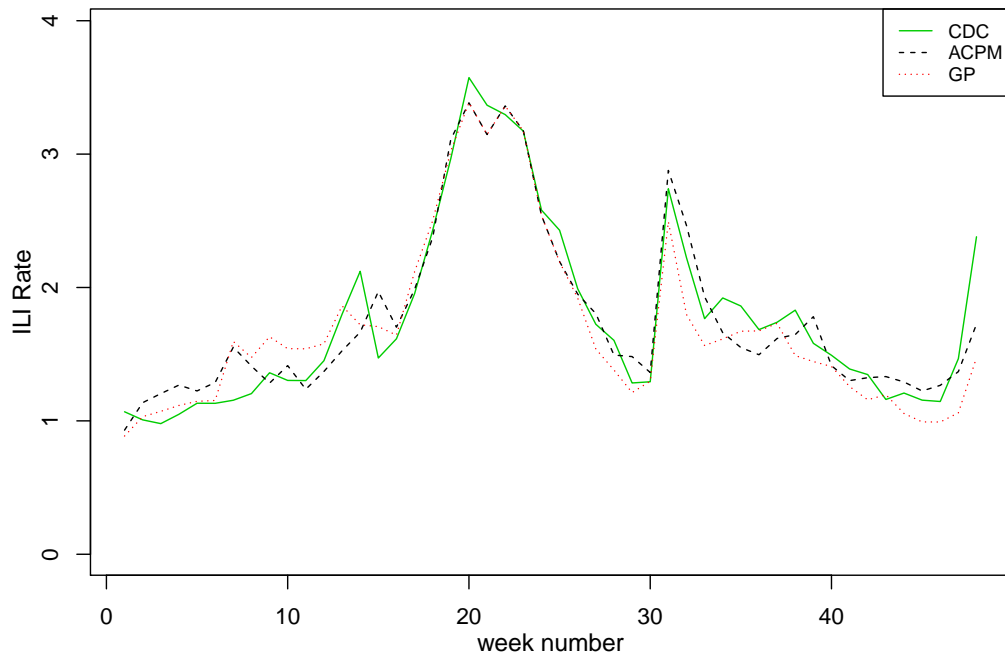


Figure H-2: Comparing ACPM and GP error in Flu Season 2008-9

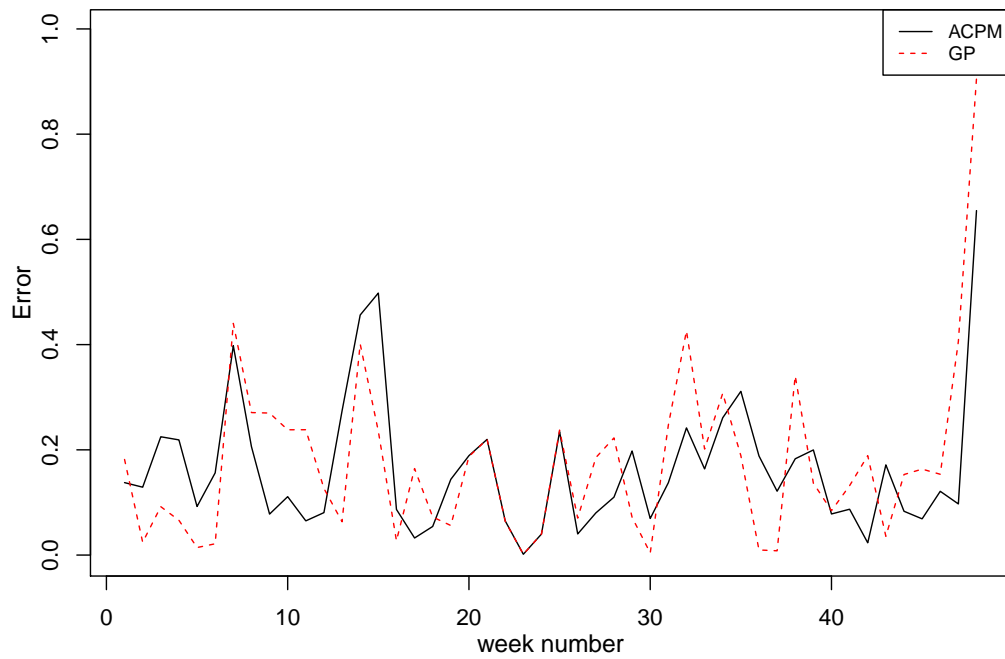


Figure H-3: Comparing Prediction of ACPM and GP in Flu Season 2009-10

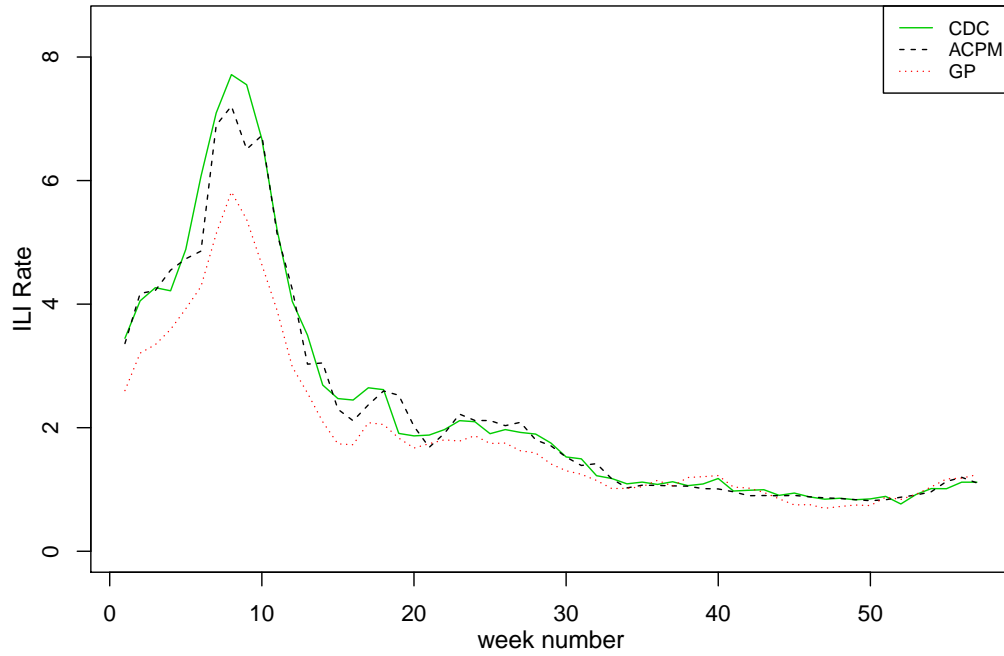


Figure H-4: Comparing ACPM and GP error in Flu Season 2009-10

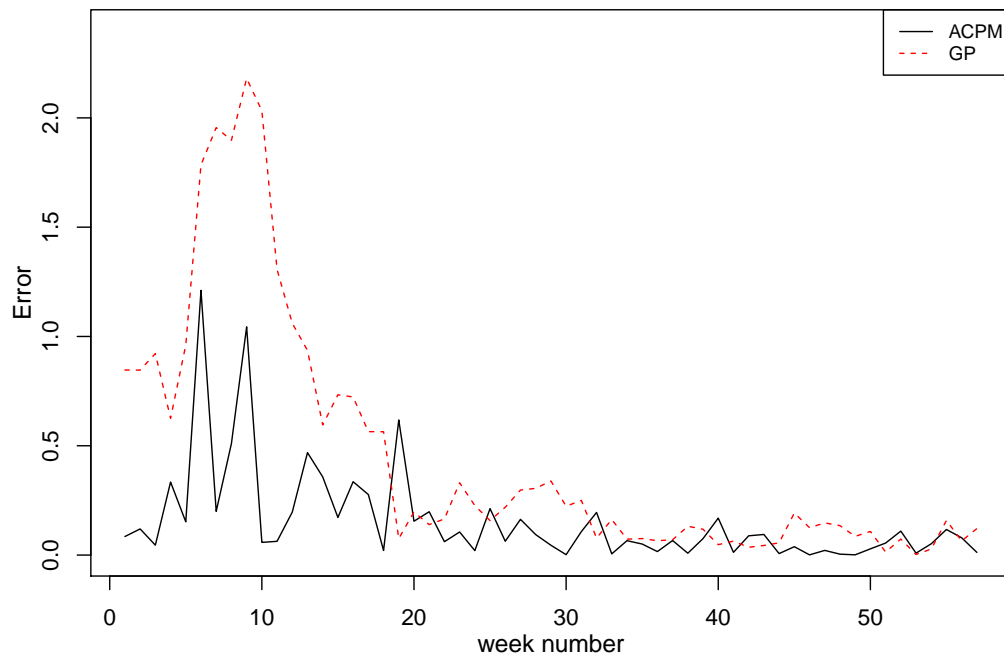


Figure H-5: Comparing Prediction of ACPM and GP in Flu Season 2010-11

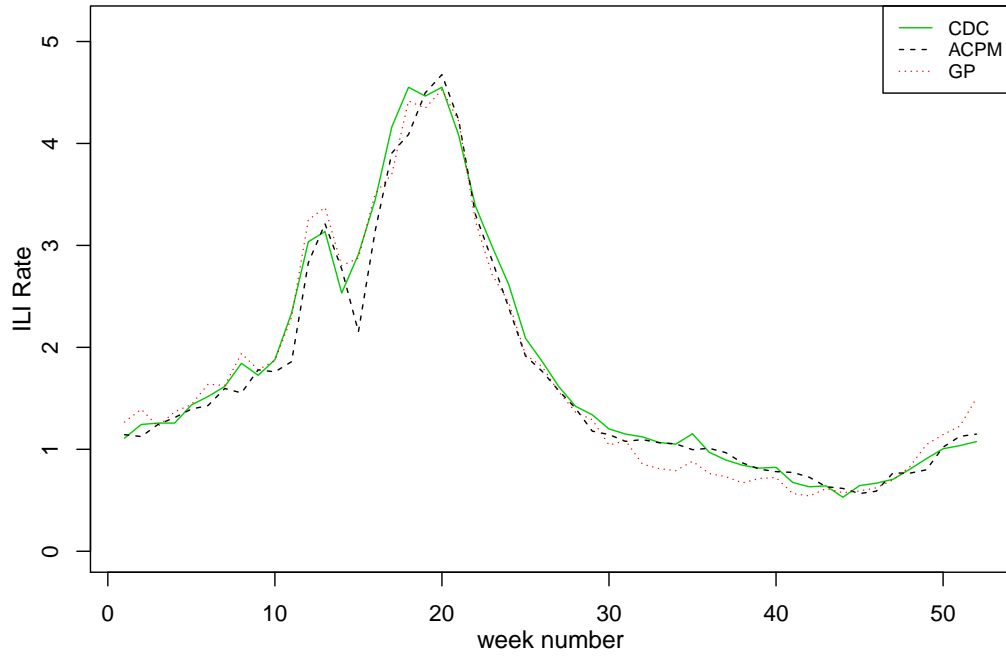


Figure H-6: Comparing ACPM and GP error in Flu Season 2010-11

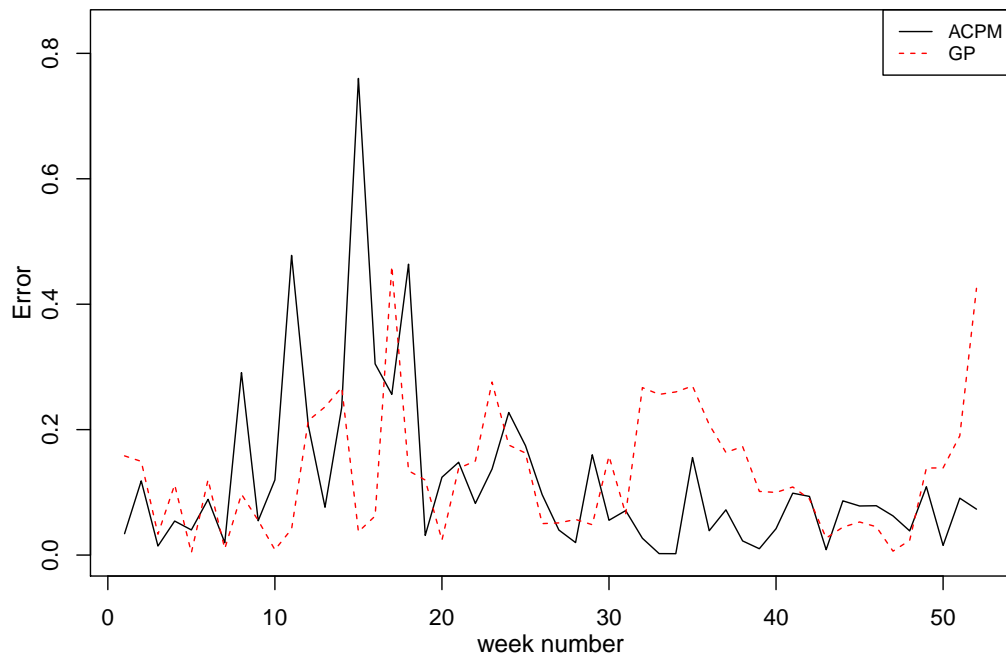


Figure H-7: Comparing Prediction of ACPM and GP in Flu Season 2011-12

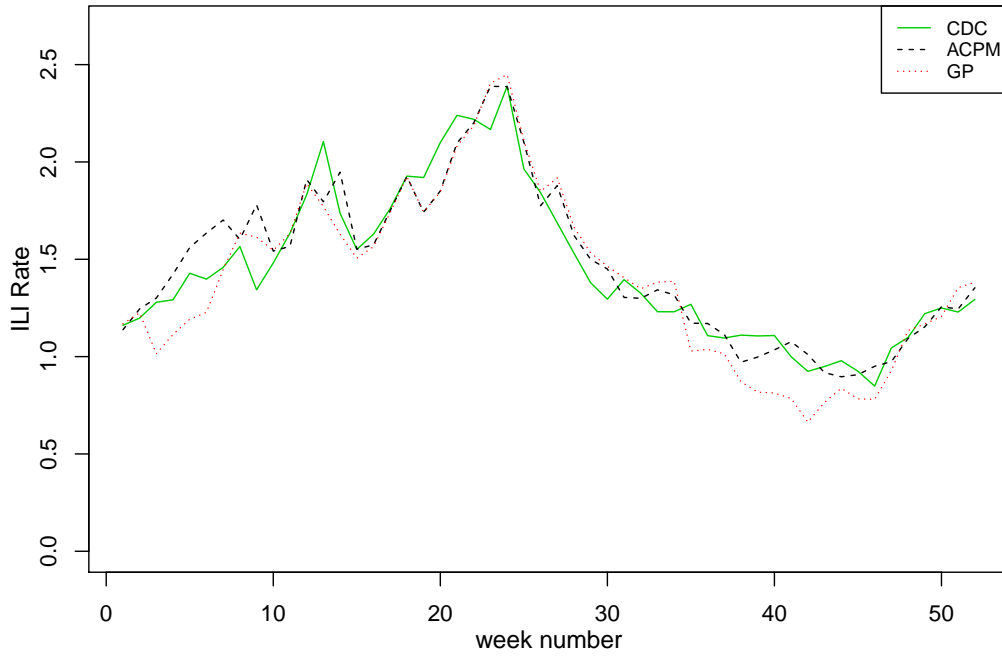


Figure H-8: Comparing ACPM and GP Error in Flu Season 2011-12

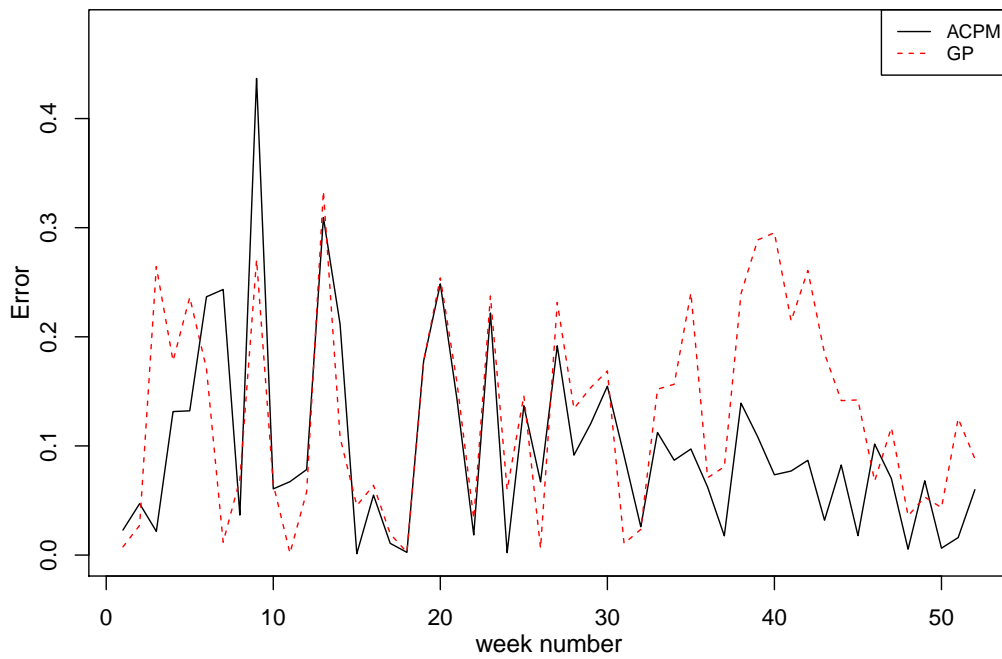


Figure H-9: Comparing Prediction of ACPM and GP in Flu Season 2012-13

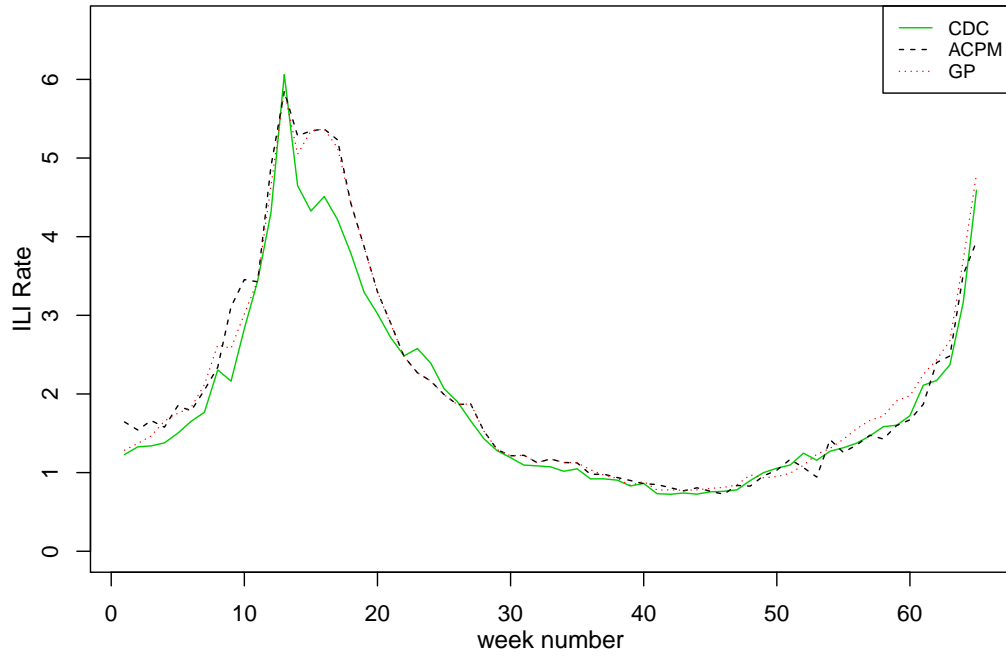
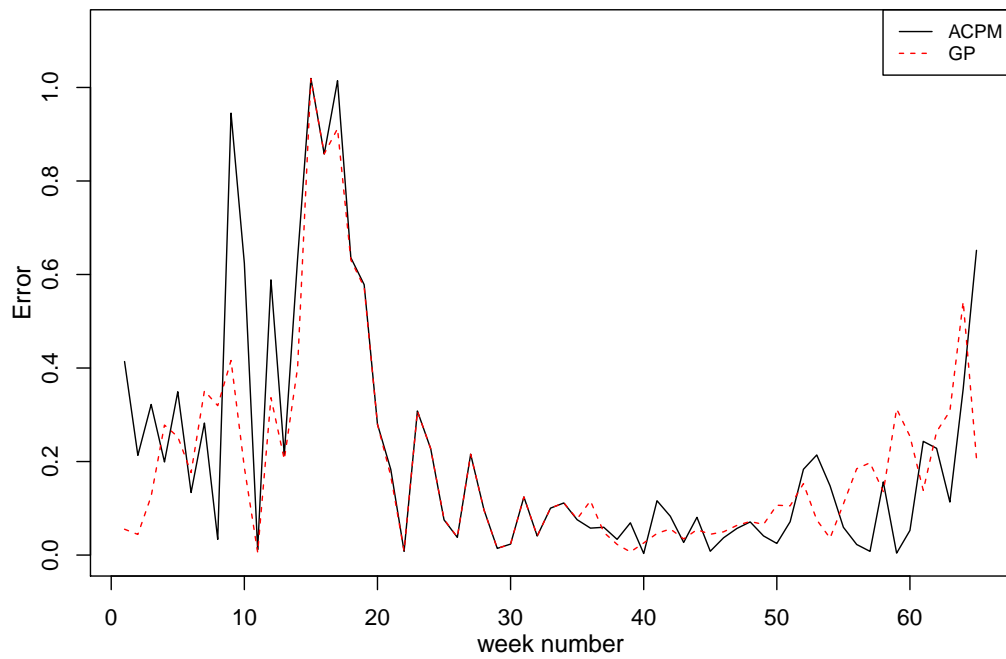


Figure H-10: Comparing ACPM and GP Error in Flu Season 2012-13



Bibliography

- Jacob Abernethy, Yiling Chen, and Jennifer Wortman Vaughan. An optimization-based framework for automated market-making. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 297–306. ACM, 2011. (Cited on page 36)
- Jacob Abernethy, Yiling Chen, and Jennifer Wortman Vaughan. Efficient market making via convex optimization, and a connection to online learning. *ACM Transactions on Economics and Computation*, 1(2):12, 2013. (Cited on page 43)
- Jacob D Abernethy and Rafael M Frongillo. A collaborative mechanism for crowdsourcing prediction problems. In *Advances in Neural Information Processing Systems*, pages 2600–2608, 2011. (Cited on page 42)
- Harshavardhan Achrekar, Avinash Gandhe, Ross Lazarus, Ssu-Hsin Yu, and Benyuan Liu. Predicting flu trends using twitter data. In *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pages 702–707. IEEE, 2011. (Cited on page 114)
- Shipra Agrawal, Erick Delage, Mark Peters, Zizhuo Wang, and Yinyu Ye. A unified framework for dynamic prediction market design. *Operations Research*, 59(3):550–568, 2011. (Cited on page 40)
- Kenneth J. Arrow, Robert Forsythe, Michael Gorham, Robert Hahn, Robin Hanson, John O. Ledyard, Saul Levmore, Robert Litan, Paul Milgrom, Forrest D. Nelson, George R. Neumann, Marco Ottaviani, Thomas C. Schelling, Robert J. Shiller, Vernon L. Smith, Erik Snowberg, Cass R. Sunstein, Paul C. Tetlock, Philip E. Tetlock, Hal R. Varian, Justin Wolfers, and Eric Zitzewitz. The promise of prediction markets. *Science*, 320(5878):877–878, 2008. (Cited on page 38)
- Adrian Barbu and Nathan Lay. An introduction to artificial prediction markets for classification. *The Journal of Machine Learning Research*, 13(1):2177–2204, 2012. (Cited on pages 13, 14, 46, 74, and 131)
- A Barto and RH Crites. Improving elevator performance using reinforcement learning. *Advances in neural information processing systems*, 8:1017–1023, 1996. (Cited on page 28)
- Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1-2):105–139, 1999. (Cited on page 21)

- Eric B Baum and Igor Durdanovic. Evolution of cooperative problem solving in an artificial economy. *Neural Computation*, 12(12):2743–2775, 2000. (Cited on page 76)
- Dalia Baziukaite. Investigation of q-learning in the context of a virtual learning environment. *Informatics in Education*, 6(2):255–268, 2007. (Cited on page 28)
- Tom Bell. Private prediction markets and the law. *Journal of Prediction Markets*, 3(1):89–110, 2009. (Cited on page 38)
- Joyce Berg and Thomas A Rietz. Prediction markets as decision support systems. *Information systems frontiers*, 5(1):79–93, 2003. (Cited on pages 34, 35, and 41)
- Joyce Berg, Robert Forsythe, Forrest Nelson, and Thomas Rietz. Results from a dozen years of election futures markets research. *Handbook of experimental economics results*, 1:742–751, 2008. (Cited on pages 34 and 36)
- Alina Beygelzimer, John Langford, and David M Pennock. Learning performance of prediction markets with kelly bettors. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1317–1318. International Foundation for Autonomous Agents and Multiagent Systems, 2012. (Cited on page 36)
- David Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, 1956. (Cited on page 24)
- David Bray, Karen Croxson, and William H Dutton. Information markets: Feasibility and performance. *papers.ssrn.com*, 2008. (Cited on pages 12, 13, 36, 38, and 41)
- Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996. (Cited on page 21)
- Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. (Cited on pages 11, 21, and 22)
- John S Brownstein, Clark C Freifeld, Ben Y Reis, and Kenneth D Mandl. Surveillance sans frontieres: Internet-based emerging infectious disease intelligence and the healthmap project. *PLoS medicine*, 5(7):e151, 2008. (Cited on pages 114 and 118)
- James W Buehler, Ruth L Berkelman, David M Hartley, and Clarence J Peters. Syndromic surveillance and bioterrorism-related epidemics. *Emerging infectious diseases*, 9(10):1197, 2003. (Cited on page 114)
- James W Buehler, Amy Sonricker, Marc Paladini, Paula Soper, and Farzad Mostashari. Syndromic surveillance practice in the united states: findings from a survey of state, territorial, and selected local health departments. *Advances in Disease Surveillance*, 6(3):1–20, 2008. (Cited on pages 6, 77, and 115)
- Declan Butler. When google got flu wrong. *Nature*, 494(7436):155, 2013. (Cited on page 121)

- Anne MP Canuto, Marjory CC Abreu, Lucas de Melo Oliveira, João C Xavier, and Araken de M Santos. Investigating the influence of the choice of the ensemble members in accuracy and diversity of selection-based and fusion-based methods for ensembles. *Pattern recognition letters*, 28(4):472–486, 2007. (Cited on page 21)
- Herman Anthony Carneiro and Eleftherios Mylonakis. Google trends: a web-based tool for real-time surveillance of disease outbreaks. *Clinical infectious diseases*, 49(10):1557–1564, 2009. (Cited on pages 117 and 119)
- CDC. CDC global health E-Brief, building usg interagency collaboration through global health engagement, First Quarter 2008. (Cited on page 117)
- Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006. (Cited on pages 23, 24, 25, 26, 28, 75, and 87)
- Nicolo Cesa-Bianchi, Yoav Freund, David Haussler, David P Helmbold, Robert E Schapire, and Manfred K Warmuth. How to use expert advice. *Journal of the ACM (JACM)*, 44(3):427–485, 1997. (Cited on page 23)
- Tung Chan. *Artificial markets and intelligent agents*. PhD thesis, Massachusetts Institute of Technology, June 2001. (Cited on page 36)
- Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien. *Semi-Supervised Learning*. The MIT Press, 1st edition, 2010. (Cited on page 20)
- H. Chen, D. Zeng, P. Yan, and P. Yan. *Infectious Disease Informatics: Syndromic Surveillance for Public Health and Biodefense*. Integrated series in information systems. Springer Science + Business Media, 2010a. (Cited on pages 6, 77, 112, 113, 114, and 117)
- Yiling Chen and David M. Pennock. A utility framework for bounded-loss market makers. *CoRR*, abs/1206.5252, 2012. (Cited on page 40)
- Yiling Chen and Jennifer Wortman Vaughan. A new understanding of prediction markets via no-regret learning. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 189–198. ACM, 2010. (Cited on pages 13, 42, and 74)
- Yiling Chen, Chao-Hsien Chu, Tracy Mullen, and David M Pennock. Information markets vs. opinion pools: An empirical comparison. In *Proceedings of the 6th ACM conference on Electronic commerce*, pages 58–67. ACM, 2005. (Cited on page 41)
- Yiling Chen, Lance Fortnow, Nicolas Lambert, David M Pennock, and Jennifer Wortman. Complexity of combinatorial market makers. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 190–199. ACM, 2008a. (Cited on pages 36 and 42)
- Yiling Chen, Sharad Goel, and David M Pennock. Pricing combinatorial markets for tournaments. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 305–314. ACM, 2008b. (Cited on page 36)

- Yiling Chen, Stanko Dimitrov, Rahul Sami, Daniel M Reeves, David M Pennock, Robin D Hanson, Lance Fortnow, and Rica Gonen. Gaming prediction markets: Equilibrium strategies with a market maker. *Algorithmica*, 58(4):930–969, 2010b. (Cited on page 134)
- Yiling Chen, Mike Ruberry, and Jenn Wortman Vaughan. Cost function market makers for measurable spaces. In *Proceedings of the fourteenth ACM conference on Electronic commerce*, pages 785–802. ACM, 2013. (Cited on page 36)
- Nigel Collier, Son Doan, Ai Kawazoe, Reiko Matsuda Goodwin, Mike Conway, Yoshio Tateno, Quoc-Hung Ngo, Dinh Dien, Asanee Kawtrakul, Koichi Takeuchi, et al. Biocaster: detecting public health rumors with a web-based text mining system. *Bioinformatics*, 24(24):2940–2941, 2008. (Cited on page 114)
- Vincent Conitzer. Prediction markets, mechanism design, and cooperative game theory. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 101–108. AUAI Press, 2009. (Cited on pages 35 and 38)
- Patrick Copeland, Raquel Romano, Tom Zhang, Greg Hecht, Dan Zigmund, and Christian Steffensen. Google disease trends: an update. *Nature*, 457:1012–1014, 2013. (Cited on page 121)
- Courtney D Corley, Diane J Cook, Armin R Mikler, and Karan P Singh. Text and structural data mining of influenza mentions in web and social media. *International journal of environmental research and public health*, 7(2):596–615, 2010. (Cited on pages 114 and 119)
- Aron Culotta. Detecting influenza outbreaks by analyzing twitter messages. *CoRR*, abs/1007.4748, 2010. (Cited on pages 114 and 118)
- Aron Culotta. Lightweight methods to estimate influenza rates and alcohol sales volume from twitter messages. *Language Resources and Evaluation*, pages 1–22, 2013. (Cited on pages 114 and 118)
- Nigel Cutland, Ekkehard Kopp, and Walter Willinger. Universal portfolios. *Mathematical finance*, 1(1):1–29, 1991. (Cited on page 24)
- Ely Dahan, Arina Soukhoroukova, and Martin Spann. New product development 2.0: Preference markets?how scalable securities markets identify winning product concepts and attributes. *Journal of Product Innovation Management*, 27(7):937–954, 2010. (Cited on page 41)
- Koen W De Bock, Kristof Coussement, and Dirk Van den Poel. Ensemble classification based on generalized additive models. *Computational Statistics & Data Analysis*, 54(6):1535–1546, 2010. (Cited on page 21)
- M. A. H. Dempster and V. Leemans. An automated FX trading system using adaptive reinforcement learning. *Expert Syst. Appl.*, 30(3):543–552, 2006. (Cited on page 33)
- M. A. H. Dempster and Yazann S. Romahi. Intraday FX trading: An evolutionary reinforcement learning approach. In Hujun Yin, Nigel M. Allinson, Richard T. Freeman, John A. Keane, and

- Simon J. Hubbard, editors, *Intelligent Data Engineering and Automated Learning - IDEAL 2002, Third International Conference, Manchester, UK, August 12-14, Proceedings*, volume 2412 of *Lecture Notes in Computer Science*, pages 347–358. Springer, 2002. (Cited on page 33)
- Stanko Dimitrov and Rahul Sami. Non-myopic strategies in prediction markets. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 200–209. ACM, 2008. (Cited on pages 13, 14, 37, 49, 129, 131, and 132)
- Alexander Doroshenko, D Cooper, G Smith, E Gerard, F Chinemana, N Verlander, A Nicoll, et al. Evaluation of syndromic surveillance based on national health service direct derived data: England and wales. *MMWR Morb Mortal Wkly Rep*, 54(Suppl):117–122, 2005. (Cited on page 116)
- Gunther Eysenbach. Infodemiology: tracking flu-related searches on the web for syndromic surveillance. In *AMIA Annual Symposium Proceedings*, volume 2006, page 244. American Medical Informatics Association, 2006. (Cited on pages 79 and 114)
- Gunther Eysenbach. Infodemiology and infoveillance: framework for an emerging set of public health informatics methods to analyze search, communication and publication behavior on the internet. *Journal of medical Internet research*, 11(1), 2009. (Cited on page 114)
- Eugene F Fama. Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2):383–417, 1970. (Cited on page 35)
- Meir Feder, Neri Merhav, and Michael Gutman. Universal prediction of individual sequences. *Information Theory, IEEE Transactions on*, 38(4):1258–1270, 1992. (Cited on page 24)
- Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, 15(1):3133–3181, 2014. (Cited on pages 11, 22, and 103)
- Robert Forsythe, Forrest Nelson, George R Neumann, and Jack Wright. Anatomy of an experimental political stock market. *The American Economic Review*, pages 1142–1161, 1992. (Cited on page 34)
- Robert Forsythe, Thomas A Rietz, and Thomas W Ross. Wishes, expectations and actions: a survey on price formation in election stock markets. *Journal of Economic Behavior & Organization*, 39(1):83–110, 1999. (Cited on pages 34 and 38)
- Clark C Freifeld, Kenneth D Mandl, Ben Y Reis, and John S Brownstein. Healthmap: global infectious disease monitoring through automated classification and visualization of internet media reports. *Journal of the American Medical Informatics Association*, 15(2):150–157, 2008. (Cited on page 118)
- Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997. (Cited on pages 22 and 23)

- Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *ICML*, volume 96, pages 148–156, 1996. (Cited on page 22)
- Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. (Cited on page 22)
- Jerome H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367 – 378, 2002. Nonlinear Methods and Data Mining. (Cited on page 22)
- Rafael M Frongillo, Nicolás Della Penna, and Mark D Reid. Interpreting prediction markets: a stochastic approach. In *Advances in Neural Information Processing Systems*, pages 3266–3274, 2012. (Cited on page 43)
- Francis Galton. Vox populi (the wisdom of crowds). *Nature*, 75:450–451, 1907. (Cited on page 34)
- John M Gandar, William H Dare, Craig R Brown, and Richard A Zuber. Informed traders and price variations in the betting market for professional basketball games. *Journal of Finance*, pages 385–401, 1998. (Cited on page 34)
- Xi Gao, Yiling Chen, and David M Pennock. Betting on the real line. In *Internet and Network Economics*, pages 553–560. Springer, 2009. (Cited on page 36)
- Xiu Gao and Laiwan Chan. An algorithm for trading and portfolio management using q-learning and sharpe ratio maximization. In *Proceedings of the international conference on neural information processing*, pages 832–837. Proc. ICONIP, 2000. (Cited on page 33)
- Jeremy Ginsberg, Matthew H Mohebbi, Rajan S Patel, Lynnette Brammer, Mark S Smolinski, and Larry Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–1014, 2008. (Cited on pages 112, 117, 118, and 119)
- Jeremy Ginsberg, Matthew H Mohebbi, Rajan S Patel, Lynnette Brammer, Mark S Smolinski, and Larry Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–1014, 2009. (Cited on pages 79, 114, and 121)
- Dhananjay K Gode and Shyam Sunder. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of political economy*, pages 119–137, 1993. (Cited on page 36)
- Irving John Good. Rational decisions. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 107–114, 1952. (Cited on page 55)
- Denise Gorse. Application of stochastic recurrent reinforcement learning to index trading. In *ESANN 2011, 19th European Symposium on Artificial Neural Networks, Bruges, Belgium, April 27-29, 2011, Proceedings*, 2011. (Cited on pages 32 and 34)
- Andreas Graefe and J Scott Armstrong. Comparing face-to-face meetings, nominal groups, delphi and prediction markets on an estimation task. *International Journal of Forecasting*, 27(1):183–195, 2011. (Cited on page 35)

- Andreas Graefe et al. Prediction markets for forecasting drug development. *Foresight: The International Journal of Applied Forecasting*, pages 8–12, 2010. (Cited on page 42)
- Ralph Grishman, Silja Huttunen, and Roman Yangarber. Information extraction for enhanced access to disease outbreak reports. *Journal of biomedical informatics*, 35(4):236–246, 2002. (Cited on page 114)
- Thomas S Gruca, Joyce Berg, and Michael Cipriano. The effect of electronic markets on forecasts of new product success. *Information Systems Frontiers*, 5(1):95–105, 2003. (Cited on page 38)
- James Hannan. Approximation to Bayes risk in repeated play. *Contributions to the Theory of Games*, 3(97-139):2, 1957. (Cited on page 25)
- Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, pages 993–1001, 1990. (Cited on page 21)
- Robin Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1):107–119, 2003. (Cited on pages 40, 42, 55, and 135)
- Robin Hanson. Logarithmic market scoring rules for modular combinatorial information aggregation. *Journal of Prediction Markets*, 1:3–15, 2007. (Cited on pages 40 and 55)
- Robin Hanson, Ryan Oprea, and David Porter. Information aggregation and manipulation in an experimental market. *Journal of Economic Behavior & Organization*, 60(4):449–459, 2006. (Cited on pages 12, 34, and 49)
- David Haussler, Jyrki Kivinen, and Manfred K Warmuth. Tight worst-case loss bounds for predicting with expert advice. In *Computational Learning Theory*, pages 69–83. Springer, 1995. (Cited on page 23)
- Friedrich August Hayek. The use of knowledge in society. *American Economic Review*, 35(4):519–530, 1945. (Cited on page 35)
- David P Helmbold and Manfred K Warmuth. Learning permutations with exponential weights. *The Journal of Machine Learning Research*, 10:1705–1736, 2009. (Cited on page 42)
- Mark Herbster and Manfred K Warmuth. Tracking the best expert. *Machine Learning*, 32(2):151–178, 1998. (Cited on pages 26 and 87)
- Geoffrey Holmes, Mark Hall, and Eibe Frank. Generating rule sets from model trees. In *Proceedings of the 12th Australian Joint Conference on Artificial Intelligence: Advanced Topics in Artificial Intelligence*, pages 1–12. Springer-Verlag, 1999. (Cited on page 22)
- Lu Hong and Scott E. Page. Groups of diverse problem solvers can outperform groups of high-ability problem solvers. *Proceedings of the National Academy of Sciences of the United States of America*, 101(46):16385–16389, 2004. (Cited on page 34)
- Torsten Hothorn and Berthold Lausen. Bundling classifiers by bagging trees. *Computational Statistics & Data Analysis*, 49(4):1068–1078, 2005. (Cited on page 21)

- Jinli Hu and Amos J. Storkey. Multi-period trading prediction markets with connections to machine learning. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Proceedings*, pages 1773–1781. JMLR.org, 2014. (Cited on pages 13, 14, 45, 46, 47, 75, and 131)
- Ms Lori Hutwagner, Mr William Thompson, G Matthew Seeman, and Tracee Treadwell. The bioterrorism preparedness and response early aberration reporting system (ears). *Journal of Urban Health*, 80(1):i89–i96, 2003. (Cited on page 116)
- Rob Hyndman and Yeasmin Khandakar. Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software*, 27(1):1–22, 2008. (Cited on page 125)
- Lian Jian and Rahul Sami. Aggregation and manipulation in prediction markets: effects of trading mechanism and information distribution. *Management Science*, 58(1):123–140, 2012. (Cited on pages 14, 36, and 131)
- Heather A Johnson, Michael M Wagner, William R Hogan, Wendy Chapman, Robert T Olaszewski, John Dowling, Gary Barnas, et al. Analysis of web access logs for surveillance of influenza. *Stud Health Technol Inform*, 107(Pt 2):1202–1206, 2004. (Cited on page 114)
- Janyl Jumadinova. *Foretell: Aggregating Distributed, Heterogeneous Information from Diverse Sources Using Market-based Techniques*. PhD thesis, UNIVERSITY OF NEBRASKA AT OMAHA, 2013. (Cited on page 36)
- Janyl Jumadinova and Prithviraj Dasgupta. A multi-agent system for analyzing the effect of information on prediction markets. *International Journal of Intelligent Systems*, 26(5):383–409, 2011. (Cited on page 36)
- Janyl Jumadinova and Prithviraj Dasgupta. Prediction market-based information aggregation for multi-sensor information processing. In *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*, pages 75–89. Springer, 2013. (Cited on page 36)
- John L Kelly Jr. A new interpretation of information rate. *Information Theory, IRE Transactions on*, 2(3):185–189, 1956. (Cited on pages 36 and 43)
- Cheng Wai Khuen, Chan Huah Yong, and Fazilah Haron. A framework for multi-agent negotiation system using adaptive fuzzy logic in resource allocation. *International Journal of Information Technology*, 11(4):p35–49, 2005. (Cited on page 28)
- Jyrki Kivinen and Manfred K Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997. (Cited on pages 27 and 87)
- Ken Kleinman, Ross Lazarus, and Richard Platt. A generalized linear mixed models approach for detecting incident clusters of disease in small areas, with an application to biological terrorism. *American Journal of Epidemiology*, 159(3):217–224, 2004. (Cited on page 116)

- Brian Kulis and Peter L Bartlett. Implicit online learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 575–582, 2010. (Cited on page 45)
- Vasileios Lampos and Nello Cristianini. Tracking the flu pandemic by monitoring the social web. In *Cognitive Information Processing (CIP), 2010 2nd International Workshop on*, pages 411–416. IEEE, 2010. (Cited on page 114)
- Vasileios Lampos, Andrew C Miller, Steve Crossan, and Christian Stefansen. Advances in nowcasting influenza-like illness rates using search query logs. *Scientific reports*, 5, 2015. (Cited on pages 12, 18, 111, 114, 120, 124, 125, 127, 131, and 172)
- Rongjian Lan, Michael D Lieberman, and Hanan Samet. The picture of health: map-based, collaborative spatio-temporal disease tracking. In *Proceedings of the First ACM SIGSPATIAL International Workshop on Use of GIS in Public Health*, pages 27–35. ACM, 2012. (Cited on page 118)
- Nathan Lay and Adrian Barbu. Supervised aggregation of classifiers using artificial prediction markets. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 591–598, 2010. (Cited on pages 9, 13, 43, 44, 47, and 74)
- Nathan Lay and Adrian Barbu. The artificial regression market. *arXiv preprint arXiv:1204.4154*, 2012. (Cited on pages 45, 46, and 47)
- David Lazer, Ryan Kennedy, Gary King, and Alessandro Vespignani. The parable of google flu: traps in big data analysis. *Science*, 343(14 March), 2014. (Cited on page 121)
- Jae Won Lee and Jangmin O. A multi-agent q-learning framework for optimizing stock trading systems. In Abdelkader Hameurlain, Rosine Cicchetti, and Roland Traunmüller, editors, *Database and Expert Systems Applications, 13th International Conference, DEXA 2002, Aix-en-Provence, France, September 2-6, 2002, Proceedings*, volume 2453 of *Lecture Notes in Computer Science*, pages 153–162. Springer, 2002. (Cited on page 33)
- Jae Won Lee, Jonghun Park, Jongwoo Lee, Euyseok Hong, et al. A multiagent approach to q-learning for daily stock trading. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 37(6):864–877, 2007. (Cited on page 34)
- Stanley M Lemon, Margaret A Hamburg, P Frederick Sparling, Eileen R Choffnes, Alison Mack, et al. *Global Infectious Disease Surveillance and Detection: Assessing the Challenges—Finding Solutions, Workshop Summary*. National Academies Press, 2007. (Cited on page 118)
- Michael D Lewis, Julie A Pavlin, Jay L Mansfield, Sheilah O’Brien, Louis G Boomsma, Yevgeniy Elbert, and Patrick W Kelley. Disease outbreak detection system using syndromic data in the greater Washington DC area. *American journal of preventive medicine*, 23(3):180–186, 2002. (Cited on page 115)

- Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R news*, 2(3): 18–22, 2002. (Cited on page 22)
- Jens P Linge, Ralf Steinberger, TP Weber, Roman Yangarber, Erik van der Goot, DH Al Khudhairy, and NI Stilianakis. Internet surveillance systems for early alerting of health threats. *Euro surveillance*, 14(AVRJUIN):200–201, 2009. (Cited on page 114)
- Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994. (Cited on pages 23 and 42)
- William B Lober, Ms Lisa J Trigg, Bryant T Karras, Mr David Bliss, Jack Ciliberti, Ms Laurie Stewart, and Jeffrey S Duchin. Syndromic surveillance using automated collection of computerized discharge diagnoses. *Journal of Urban Health*, 80(1):i97–i106, 2003. (Cited on page 115)
- J.S. Lombardo and D.L. Buckeridge. *Disease Surveillance: A Public Health Informatics Approach*. Wiley, 2007. (Cited on pages 111 and 112)
- Philip M Long and Rocco A Servedio. Random classification noise defeats all convex potential boosters. *Machine Learning*, 78(3):287–304, 2010. (Cited on page 22)
- Stefan Luckner. Prediction markets: Fundamentals, key design elements, and applications. *BLED 2008 Proceedings*, page 27, 2008. (Cited on pages 35, 38, and 39)
- James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967. (Cited on page 62)
- Ananth Madhavan. Trading mechanisms in securities markets. *the Journal of Finance*, 47(2): 607–641, 1992. (Cited on page 39)
- Kenneth D Mandl, J Marc Overhage, Michael M Wagner, William B Lober, Paola Sebastiani, Farzad Mostashari, Julie A Pavlin, Per H Gesteland, Tracee Treadwell, Eileen Koski, et al. Implementing syndromic surveillance: a practical guide informed by the early experience. *Journal of the American Medical Informatics Association*, 11(2):141–150, 2004. (Cited on page 112)
- Bernard Mangold, Mike Dooley, Gary W Flake, Havi Hoffman, Tejaswi Kasturi, David M Pennock, and Rael Dornfest. The tech buzz game [stock market prediction]. *Computer*, 38(7): 94–97, 2005. (Cited on page 39)
- W Robert Mann. Mean value methods in iteration. *Proceedings of the American Mathematical Society*, 4(3):506–510, 1953. (Cited on page 44)
- Charles F. Manski. Interpreting the predictions of prediction markets. *Economics Letters*, 91(3):425 – 429, 2006. (Cited on page 36)
- Abla Mawudeku and Michael Blench. Global public health intelligence network (gphin). In *7th Conference of the Association for Machine Translation in the Americas*, pages 8–12, 2006. (Cited on page 114)

- Andrew McCallum and Kamal Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48, 1998. (Cited on page 117)
- Oliver Mihatsch and Ralph Neuneier. Risk-sensitive reinforcement learning. *Machine learning*, 49(2-3):267–290, 2002. (Cited on page 33)
- Paul Milgrom and Nancy Stokey. Information, trade and common knowledge. *Journal of Economic Theory*, 26(1):17–27, 1982. (Cited on page 38)
- Brent J Miller and Mark Steyvers. The wisdom of crowds with communication. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society, at Austin, TX*, 2011. (Cited on page 131)
- Tom M. Mitchell. *Machine learning*. McGraw Hill series in computer science. McGraw-Hill, 1997. (Cited on page 19)
- Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012. (Cited on page 20)
- John E. Moody and Matthew Saffell. Reinforcement learning for trading systems and portfolios. In Rakesh Agrawal, Paul E. Stolorz, and Gregory Piatetsky-Shapiro, editors, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), New York City, New York, USA, August 27-31, 1998*, pages 279–283. AAAI Press, 1998. (Cited on page 33)
- John E. Moody and Lizhong Wu. Optimization of trading systems and portfolios. In *Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering, CIFE 1997, New York City, USA, March 24-25, 1997*, pages 300–307. IEEE, 1997. (Cited on page 33)
- K.P. Murphy. *Machine Learning: A Probabilistic Perspective*. Adaptive computation and machine learning. MIT Press, 2012. (Cited on page 20)
- Y Narahari. *Game Theory and Mechanism Design*, volume 4. World Scientific, 2014. (Not cited)
- Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. *Front. Neurobot.*, 2013, 2013. (Cited on page 22)
- NP Nelson, JS Brownstein, DM Hartley, et al. Event-based biosurveillance of respiratory disease in mexico, 2007-2009: connection to the 2009 influenza a (h1n1) pandemic. *Euro Surveill*, 15(30), 2010. (Cited on page 117)
- Ralph Neuneier. Optimal asset allocation using adaptive dynamic programming. *Advances in Neural Information Processing Systems*, pages 952–958, 1996. (Cited on pages 28 and 33)
- Ralph Neuneier. Enhancing q-learning for optimal asset allocation. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems 10, [NIPS Conference, Denver, Colorado, USA, 1997]*, pages 936–942. The MIT Press, 1997. (Cited on page 33)

- Yuriy Nevmyvaka, Yi Feng, and Michael Kearns. Reinforcement learning for optimized trade execution. In *Proceedings of the 23rd international conference on Machine learning*, pages 673–680. ACM, 2006. (Cited on page 34)
- Andrew Ng. Reinforcement learning and control. CS229 Lecture notes, 2011. (Cited on page 62)
- Evdokia Nikolova and Rahul Sami. A strategic model for information markets. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 316–325. ACM, 2007. (Cited on pages , 12, 34, 40, and 49)
- Yasushi Ohkusa, M Shigematsu, K Taniguchi, and N Okabe. Experimental surveillance using data on sales of over-the-counter medications-japan, November 2003–April 2004. *MMWR Morb Mortal Wkly Rep*, 54:47–52, 2005. (Cited on page 116)
- Kenneth Oliven and Thomas A. Rietz. Suckers are born but markets are made: Individual rationality, arbitrage, and market efficiency on an electronic futures market. *Management Science*, 50(3):336–351, 2004. (Cited on page 34)
- Donald R Olson, Kevin J Konty, Marc Paladini, Cecile Viboud, and Lone Simonsen. Reassessing google flu trends data for detection of seasonal and pandemic influenza: a comparative epidemiological study at three geographic scales. *PLoS Comput Biol*, 9(10):e1003256, 2013. (Cited on page 121)
- United States. Congress. House. Committee on Homeland Security. Subcommittee on Emerging Threats and Cybersecurity. *One year later: implementing the biosurveillance requirements of the 9/11 Act: hearing before the Subcommittee on Emerging Threats, Cybersecurity, and Science and Technology of the Committee on Homeland Security, House of Representatives, One Hundred Tenth Congress, second session, July 16, 2008*, volume 4. Government Printing Office, 2009. (Cited on page 117)
- Abraham Othman and Tuomas Sandholm. When do markets with simple agents fail? In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 865–872. International Foundation for Autonomous Agents and Multiagent Systems, 2010. (Cited on page 40)
- Abraham Othman and Tuomas Sandholm. Automated market makers that enable new settings: Extending constant-utility cost functions. In *Auctions, Market Mechanisms, and Their Applications*, pages 19–30. Springer, 2012. (Cited on page 36)
- ES Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954. (Cited on page 116)
- Scott E Page. Making the difference: Applying a logic of diversity. *The Academy of Management Perspectives*, 21(4):6–20, 2007. (Cited on page 37)
- Michael J Paul, Mark Dredze, and David Broniatowski. Twitter improves influenza forecasting. *PLoS currents*, 6, 2014. (Cited on page 114)

- David M Pennock. A dynamic pari-mutuel market for hedging, wagering, and information aggregation. In *Proceedings of the 5th ACM conference on Electronic commerce*, pages 170–179. ACM, 2004. (Cited on page 39)
- David M Pennock and Rahul Sami. Computational aspects of prediction markets. *Algorithmic Game Theory (N. Nisan, T. Roughgarden, E. Tardos, V. Vazirani, editors)*, pages 651–674, 2007. (Cited on page 36)
- Johan Perols, Kaushal Chari, and Manish Agrawal. Information market-based decision fusion. *Management Science*, 55(5):827–842, 2009. (Cited on pages 43, 47, and 120)
- Mark Peters, Anthony Man-Cho So, and Yinyu Ye. Pari-mutuel markets: Mechanisms and performance. In *Internet and Network Economics*, pages 82–95. Springer, 2007. (Cited on pages 39 and 52)
- Charles R Plott and Kay-Yut Chen. Information aggregation mechanisms: Concept, design and implementation for a sales forecasting problem. *Working paper No. 1131*, 2002. (Cited on pages 34 and 41)
- Philip M. Polgreen, Forrest D. Nelson, and George R. Neumann. Use of prediction markets to forecast infectious disease activity. *Clinical Infectious Diseases*, 44(2):pp. 272–279, 2007. (Cited on pages 42 and 134)
- Philip M Polgreen, Yiling Chen, David M Pennock, Forrest D Nelson, and Robert A Weinstein. Using internet searches for influenza surveillance. *Clinical infectious diseases*, 47(11):1443–1448, 2008. (Cited on pages 79 and 114)
- Walter A Powell, Robin Hanson, Kathryn B Laskey, and Charles Twardy. Combinatorial prediction markets: An experimental study. In *Scalable Uncertainty Management*, pages 283–296. Springer, 2013. (Cited on page 13)
- David Rajakovich and Vladimir Vladimirov. Prediction markets as a medical forecasting tool: Demand for hospital services. *Journal of Prediction Markets*, 3(2):78–106, 2009. (Cited on page 42)
- Russ Ray. Prediction markets and the financial “wisdom of crowds”. *Journal of Behavioral Finance*, 7(1):2–4, 2006. (Cited on pages 12, 13, 34, 49, and 108)
- GD Reichard, Peter Demitry, and Joseph Catalino. Cohort: An integrated information approach to decision support for military subpopulation health care. Technical report, Air Force Medical Operations Agency, AFMOA/SGZI,5201 Leesburg Pike Sky 3, Ste 1400,Falls Church,VA,22041-3203, 2004. (Cited on page 115)
- Aimee R Reilly, Emily A Iarocci, Carrienne M Jung, David M Hartley, and Noele P Nelson. Indications and warning of pandemic influenza compared to seasonal influenza. *Indicator*, 1967(128), 1968. (Cited on page 114)

- Herbert Robbins. Asymptotically subminimax solutions of compound statistical decision problems. In *Herbert Robbins Selected Papers*, pages 7–24. Springer, 1985. (Cited on pages 24, 28, and 87)
- S. W. Roberts. Control chart tests based on geometric moving averages. *Technometrics*, 1(3): 239–250, 1959. (Cited on page 116)
- Bruce E Rosen. Ensemble learning using decorrelated neural networks. *Connection Science*, 8 (3-4):373–384, 1996. (Cited on page 108)
- Earl S Rosenbloom and William Notz. Statistical tests of real-money versus play-money prediction markets. *Electronic Markets*, 16(1):63–69, 2006. (Cited on page 38)
- Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990. (Cited on page 22)
- Emile Servan-Schreiber, Justin Wolfers, David M Pennock, and Brian Galebach. Prediction markets: Does money matter? *Electronic Markets*, 14(3):243–251, 2004. (Cited on page 41)
- Yurii Mikhailovich Shtar'kov. Universal sequential coding of single messages. *Problemy Peredachi Informatsii*, 23(3):3–17, 1987. (Cited on page 24)
- Alessio Signorini, Alberto Maria Segre, and Philip M. Polgreen. The use of twitter to track levels of disease activity and public concern in the U.S. during the Influenza A H1N1 Pandemic. *PLoS ONE*, 6:e19467, 05 2011. (Cited on pages 114, 117, and 119)
- Erik Snowberg, Justin Wolfers, and Eric Zitzewitz. Prediction markets for economic forecasting. Working Paper 18222, National Bureau of Economic Research, July 2012. (Cited on page 12)
- Arina Soukhoroukova, Martin Spann, and Bernd Skiera. Sourcing, filtering, and evaluating new product ideas: An empirical exploration of the performance of idea markets. *Journal of Product Innovation Management*, 29(1):100–112, 2012. (Cited on page 41)
- AA Spanjersberg. Combination of different systems for the recognition of handwritten digits. In *Proc. 2nd Int. Joint Conf. on Pattern Recognition, Copenhagen*, pages 208–209, 1974. (Cited on page 21)
- Martin Spann and Bernd Skiera. Internet-based virtual stock markets for business forecasting. *Management Science*, 49(10):1310–1326, 2003. (Cited on pages 38 and 39)
- Martin Spann, Holger Ernst, Bernd Skiera, and Jan Henrik Soll. Identification of lead users for consumer products via virtual stock markets. *Journal of Product Innovation Management*, 26 (3):322–335, 2009. (Cited on page 41)
- Brian Spears, Christina LaComb, John Interrante, Janet Barnett, and Deniz Senturk-Dogonaksoy. Examining trader behavior in idea markets: an implementation of GE's imagination markets. *The Journal of Prediction Markets*, 3(1):17, 2009. (Cited on page 42)

- Christian Stefansen. Flu trends updates model to help estimate flu levels in the us, 10 2013. (Cited on page 128)
- Amos J. Storkey. Machine learning markets. In Geoffrey J. Gordon, David B. Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pages 716–724. JMLR.org, 2011. (Cited on pages 13, 14, 45, 46, 47, 75, and 131)
- Amos J. Storkey, Jono Millin, and Krzysztof Geras. Isoelastic agents and wealth updates in machine learning markets. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012. (Cited on pages 13, 14, 45, 46, 47, 75, and 131)
- James Surowiecki. *The wisdom of crowds*. Anchor, 2005. (Cited on page 131)
- Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998. (Cited on pages 20, 28, 29, and 31)
- Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction, 2011. (Cited on pages 5, 28, 29, 30, and 31)
- Gerald Tesauro. Neurogammon wins computer olympiad. *Neural Computation*, 1(3):321–323, 1989. (Cited on page 28)
- Gerald Tesauro. TD-Gammon, a self-teaching backgammon program, achieves master-level play. *Neural computation*, 6(2):215–219, 1994. (Cited on page 21)
- Fu-Chiang Tsui, Jeremy U Espino, Virginia M Dato, Per H Gesteland, Judith Hutman, and Michael M Wagner. Technical description of rods: a real-time public health surveillance system. *Journal of the American Medical Informatics Association*, 10(5):399–408, 2003. (Cited on page 115)
- Georgios Tziralis and Ilias Tatsiopoulos. Prediction markets: an information aggregation perspective to the forecasting problem. *World Review of Entrepreneurship, Management and Sustainable Development*, 3(3-4):251–259, 2007. (Cited on page 37)
- G. Upton and I. Cook. *Understanding Statistics*. OUP Oxford, 1996. (Cited on page 55)
- Cees Van den Wijngaard, Liselotte Van Asten, Wilfrid Van Pelt, Nico JD Nagelkerke, Robert Verheij, Albert J De Neeling, Arnold Dekkers, Marianne AB Van der Sande, Hans Van Vliet, and Marion PG Koopmans. Validation of syndromic surveillance for respiratory pathogen activity. *Emerging infectious diseases*, 14(6):917, 2008. (Cited on page 113)
- Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards*. PhD thesis, University of Cambridge England, 1989. (Cited on pages , 15, 31, and 59)

- Christof Weinhardt, Dirk Neumann, and Carsten Holtmann. Germany: computer-aided market engineering. *Commun. ACM*, 49(7):79, 2006. (Cited on page 37)
- Marc-Alain Widdowson, Arnold Bosman, Edward van Straten, Mark Tinga, Sandra Chaves, Liesbeth van Eerden, and Wilfred van Pelt. Automated, laboratory-based system using the internet for disease outbreak detection, the netherlands. *Emerging infectious diseases*, 9(9):1046, 2003. (Cited on page 116)
- Leighton Vaughan Williams. *Prediction markets: Theory and applications*, volume 66. Routledge, 2011. (Cited on pages 37 and 38)
- Justin Wolfers and Eric Zitzewitz. Prediction markets. Working Paper 10504, National Bureau of Economic Research, May 2004. (Cited on pages 36 and 37)
- David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992. (Cited on page 22)
- David H Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390, 1996. (Cited on pages 11, 109, and 130)
- World Health Organization. Public health surveillance, Februray 2013. (Cited on page 111)
- World Health Organization. Influenza (seasonal). Fact sheet N211, March 2014. (Cited on page 12)
- Wei Zhang and Thomas G. Dietterich. High-performance job-shop scheduling with A time-delay TD-lambda network. In David S. Touretzky, Michael Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, November 27-30, 1995*, pages 1024–1030. MIT Press, 1995. (Cited on page 28)
- Zhaohui Zheng, Hongyuan Zha, Tong Zhang, Olivier Chapelle, Keke Chen, and Gordon Sun. A general boosting method and its application to learning ranking functions for web search. In *Advances in neural information processing systems*, pages 1697–1704, 2008. (Cited on page 22)