U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

PhD

3.º
CICLO

FCUP
UA
UM
CMU
2015

U.PORTO

Network Aided Classification and Detection of Data

Vinay Uday Prabhu

FC

Carnegie
Mellon
University

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

universidade
de aveiro

Universidade do Minho

Network Aided Classification and Detection of Data

Vinay Uday Prabhu

Tese de Doutoramento apresentada à
Faculdade de Ciências da Universidade do Porto,
Carnegie Mellon University,
em Informática
2015

PhD

Network Aided Classification and Detection of Data

Vinay Uday Prabhu

June 3, 2015

Electrical and Computer
Engineering Department,
Carnegie Mellon University
Pittsburgh, PA, USA

Consortium of:
Universidades do Minho, Aveiro e Porto
Portugal

Thesis Committee

Rohit Negi, Carnegie Mellon University

Miguel Rodrigues, Faculdade de Ciencias da Universidade do Porto

Jose Moura, Carnegie Mellon University

Jaime dos Santos Cardoso, Faculdade de Engenharia da Universidade do Porto

Pulkit Grover, Carnegie Mellon University

Paulo Ferreira, University of Aveiro

Luis Torgo, Faculdade de Ciencias da Universidade do Porto

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Abstract

Two important technological aspects of the *Big data* paradigm have been the emergence of massive scale *Online Social Networks* (OSNs) (such as Facebook and Twitter), and the rise of the *open data movement* that has resulted in the creation of richly structured online datasets, such as Wikipedia, Citeseer and the US federal government's data.gov initiative. The examples of OSNs and online datasets cited above share the common feature that they can be thought of as Online Information Graphs, in the sense that the information embedded in them has a natural graph structure.

In this thesis, we consider using this underlying *Online Information Graph* as a statistical prior to enhance classification accuracy of some hard machine learning problems. Specifically, we look at instances where the graph is undirected and propose using the graph to define an Ising - Markov Random Field (MRF) prior. To begin with, we validate the Ising prior using a novel hypothesis testing framework based approach. Having validated the Ising prior, we demonstrate its utility by showcasing Network Aided Vector classification (NAC) of real world data from fields as varied as vote prediction in the US senate, movie earnings level classification (using IMDb dataset) and county crime-level classification (using the US census data). We then consider a special case of the classification problem which involves Network Aided Detection (NAD) of a global sentiment in an OSN. To this end, we consider *Latent Sentiment* (LS) detection as well as *Majority Sentiment* detection. We analyze the performance of the trivial sentiment detector for LS detection using a novel communications-oriented viewpoint, where we view the underlying network as providing a weak channel code that transmits one bit of information (the binary sentiment) and perform error exponent analysis for various underlying graph models. We also address the problem of optimal Maximum A posterior Probability (MAP) detection of majority sentiment in the highly noisy labels weak network effect (NW) scenario, deriving the High Temperature (HT) expansion formula for the partial partition function of the Ising model using the code-puncturing idea from coding theory and then proposing an approximate MAP detector that outperforms the Maximum Likelihood (ML) detector and the trivial detector.

Acknowledgement

To begin with, I would like to thank my advisors, Prof.Rohit Negi and Prof.Miguel Rodrigues for being a constant guiding force throughout my PhD.

Prof.Negi, simply put, is a researcher par excellence. His near absolute mastery over an incredibly large range of topics was instrumental in us bringing in ideas and techniques from social networks, statistical physics, communication theory and information theory into this thesis. He was always accessible right from day one of my arrival here and every single equation in this thesis first took birth on the white board of his office. His brutally honest assessment of the flaws in the idea, the math or the methodology meant that the stakes were always high while preparing for our update meetings. This doctoral study under him has been a baptism by fire which has forged me into becoming a better thinker and a better engineer, and for which I will remain eternally grateful.

Prof.Miguel has been a wonderfully supportive companion through this journey. Not once did he allow his rather challenging transition from University of Porto to University College London affect either the quality of our interactions or the frequency. It's hard to enumerate the number of occasions when he has gone out of his way to assist me with paper corrections, funding issues and research ideas in general. It was an absolute privilege working with him on Information theoretic secrecy ideas in the initial part of my PhD.

I'd also like to specially thank Prof.Jose Moura for having initiated me into Network Sciences via his graduate course (18-799), for having served on my PhD qualifier exam committee and later on my PhD defense committee. Defending my work at every stage of my PhD in front of an esteemed member of the US National Academy of Engineers with such a strong world-wide reputation as an expert in the field of Signal processing and Networks was one of the biggest ego boosts I received during my PhD. The key insights he shared during my PhD proposal exam were instrumental in shaping the structure of this thesis.

I'd now like to thank Prof. Jaime dos Santos Cardoso, for being such a wonderful teacher of Machine Learning and for serving on my PhD thesis committee. His comments and suggestions were invaluable in shaping the Network Aided Classification part of this work presented. I'd also like to thank the other members of my thesis committee, Prof. Pulkit Grover, Prof. Paulo

Ferreira and Prof.Luis Torgo, for agreeing at such a short notice to read and review my thesis and for their insightful comments which helped iron out some of the flaws in the initial draft. Special thanks are also due to Prof.Fernando Silva at University of Porto for his immense support in handling the administrative matters related to the dual-degree program.

I'd also like to thank Prof.Devendra Jalihal, my mentor at IIT Madras and Prof. Dimitrios Toumpakaris at University of Patras for initiating me into research and for providing recommendation letters without which I could not have begun this journey.

During the course of this PhD, I feel rather fortunate for having interacted with some incredibly smart and friendly researchers in the two academic eco-systems I was part of. On the Portuguese side, I owe a lot of gratitude to Tiago, Fab, the *Tres Pedros*, Rui, Joao, Joana, Ines, Maria, Teresa, Elizabete, Leandro, Ana and Hugo at University of Porto for initiating me to Portuguese fine dining, Fado music, Port Wine and for turning me into a quintessential *Portista* and a proud *Tripeiro*.

On the Pitt side of things, the *desi* quadrumvirate of Vishnu, Shintre, Randip and Aadi Ramdas were the guys who bore the brunt of my (anti) social presence. The memories of long hours of cricket net sessions in the Skibo gym, speed-gun tests, getting chucked out of pubs on Walnut street, (pseudo)philosophical discussions of the ills plaguing our zeitgeist and questionable trips to Niagara falls and white-water rafting will always remain etched in my memory. Randip, the prodigal Sardar's trials and tribulations as a fledgling *Green Beret* soldier constantly helped put my own challenges into perspective. I'd like to thank Joe the-Arnie-fan, Kevin-the-tank and Keith-the-cereal-killer for guiding me through the gym-rat phase of my life. Thanks to them, the 225 on the bench press is soon on the cards (Insert pinch of salt here).

On the administrative side, I'd like to acknowledge the help and guidance provided by the ICTI and IT-Porto leadership staff of Sara Brandao, Silvia Ribeiro, Lori Spears, Alexandra, Carolina Carvalho, Elimary Silva, Joao Barros and Joao Claros, who were all instrumental in making the trans-Atlantic hops between Porto and Pittsburgh seem trivial. I'd also like to thank the CMU ECE administrative staff, especially, Elaine Lawrence and Samantha Goldstein for helping me conquer the dreaded paper-work aspect of PhD and remain a legal-terrestrial-alien.

I'd like to thank Fundao para a Cincia e a Tecnologia -FCT (Fellowship grant: SFRH/BD/51847/2012) and the National Science Foundation -NSF- (NSF award CCF1422193) for the financial support provided during my PhD.

Last but not the least, I'd like to thank my family members for their unflinching support right throughout this venture. My dad to me has always been the center of my universe. The fact that his stellar achievements of being a double silver medalist in Asian Games (Athletics), a twice world cup attendee and the 2006 Dhyan Chand Award[1] winner never came in the way of his humility and down-to-earth demeanor speaks volumes of the character he bears. My mum (*Amma*), in her own right was an incredible poly-athlete herself, having excelled at the national level in soccer, volleyball, athletics and badminton. Growing up, I had the most wonderful parenting a kid can ever hope. *Ajji*, my grandma has raised me since the day I was born and her constant presence in my life has endowed her a very special place in my heart.
I was also blessed to have the best sibling cousins in *Ammu, Sant* and *Swats* and words cannot describe how much their support means to me.
Now, I'd like to acknowledge the role of Signe, my better half, whose crucial loving presence helped me deal with all the madness that comes with high intensity studies in a challenging environ such as CMU. She and her parents have been like a second family to me in Pittsburgh and I can't thank them enough for all the unconditional love and support rendered.

Lastly, I'd like to dedicate this thesis to Dr.K.N.Nagaraj, my spiritual Guru, my mentor from my pre-university days, who breathed his last on 04/03/2013.

Rest In Peace,

Sir.

Asatho Maa Sad Gamaya|| Thamaso Maa Jyothir Gamaya|| Mrithyur Maa Amritham Gamaya|| Om Shanti, Shanti, Shanti||
From untruth lead us to Truth. From darkness lead us to Light. From death lead us to Immortality. Om Peace, Peace, Peace.
- Brhadaranyaka Upanishad, I.iii.28

[1] which is India's highest award for lifetime achievement in sports and games, given by the Ministry of Youth Affairs and Sports, Government of India

Contents

1 Introduction **1**

1.1 Online Information Graphs 1

1.1.1 A real world example: Hashtag Hijacking 3

1.2 Markov Random Fields as label priors 5

1.2.1 Markov Random Fields: A brief introduction 6

1.2.2 The homogeneous Ising prior with constant external field 7

1.3 Thesis Outline 10

1.3.1 Mathematical notation in the thesis 12

1.4 Related work 13

1.4.1 Complex networks: Study of the underlying graph structure 13

1.4.2 Gossip and Epidemiology: Information processes on graphs 14

2 Model Validation **17**

2.1 Introduction 17

2.2 The homogeneous ferromagnetic Ising prior 18

2.2.1 Testing against the uniform (i.i.d) prior 18

2.3 Definitions 19

2.4 p-value 21

2.4.1 Approximating the p-value using sampling 22

2.4.2 Upper bounding the p-value 22

2.5 Results with real world data 23

2.6 Chapter Summary 26

3 Network Aided Classification 29

3.1 Introduction 29

3.1.1 Literature Survey of classification techniques for correlated data 30

3.2 The MAP-MRF model for Network aided classification 31

3.2.1 The MAP and MPM classifier definitions 31

3.2.2 Conditional independence in the likelihood 32

3.2.3 MAP classification under the i.i.d assumption 32

3.2.4 MAP classification with the Ising prior 32

3.3 ML estimation of θ 33

3.3.1 Maximum-Likelihood estimation of θ in the completely observed scenario . 34

3.3.2 ML Estimation of θ in a partially observed scenario 36

3.3.3 Constructing the conditional truncated MRF 37

3.4 Methodology 39

3.4.1 Training Phase 1: Extracting the training and testing datasets based on randomly sampled edge subsets 40

3.4.2 Training Phase 2: Maximum-Likelihood estimation of θ 40

3.4.3 Training Phase 3: Training the machine learning classifier. 41

3.4.4 Testing Phase 1: Potentiating the truncated conditional MRF 42

3.4.5 Testing Phase 2: Solving the MAP-MRF inference problem 42

3.5 Real world examples 43

3.5.1 NAC with generative models for likelihood 43

3.5.2 NAC with discriminative models for likelihood 44

3.6 SNA inspired community aided classification 49

3.7 Chapter Summary 52

4 Latent sentiment detection 67

4.1 Introduction 67

4.1.1 Sentiment detection in OSNs 68

4.1.2 Micro-blog /Tweet level sentiment classification 68

4.2 Latent sentiment detection problem 70

4.2.1 Model 70

4.2.2 Revisiting the `#iloveobamacare` Twitter network 71

4.2.3 Trivial sentiment detector 73

4.3 Communications-inspired analysis for the trivial detector 74

4.3.1 P_e upper bound 74

4.3.2 Computation of the upper bound 74

4.3.3 Error exponent 77

4.3.4 Computation of the error exponent 77

4.4 Numerical results: Error exponent of different networks 78

4.4.1 Variation of error exponent (α) with edge-potential θ for different network topologies 79

4.4.2 Variation of error exponent (α) with latent sentiment strength γ for different network topologies 80

4.4.3 Variation of error exponent (α) with BSC noise level ε for different network topologies 81

4.5 Chapter summary 83

5 Majority sentiment detection **84**

5.1 Introduction 84

5.1.1 Majority sentiment detection: Literature review 84

5.1.2 Problem setup 86

5.2 Model for majority vote detection 88

5.2.1 The non-homogeneous Ising prior 89

5.2.2 Revisiting the BSC model for the machine learning classifiers 90

5.2.3 The joint Bayesian modeling of $\mathbf{x}$ and $\mathbf{y}$ 90

5.2.4 Majority sentiment detectors 91

5.3 Positive part Partition function (PPF) and MAP detection 93

5.3.1 Motivating the HT framework 93

5.3.2 Main Result: The HT spanning expansion framework for the PPF 94

5.4 MAP detection in the Noisy data Weak network (NW) regime with homogeneous edge-potential 95

5.5 Probability of error of majority sentiment detection in the NW regime: Numerical results 97

5.5.1 Network prior with constant ε 99

5.5.2 Network prior with varying ε_i 100

5.6 Alternate HT expansion expression with appended pseudo-node model 101

5.6.1 Virtual node appending to absorb the node potentials 102

5.7 Applications of the HT expansion of the PPF 104

5.7.1 Application 1: Geometric interpretation of the approximate MAP detection . 104

5.7.2 Application 2: Extension of the HT framework for the generalized majority detection problem 109

5.7.3 Application 3: Relationship between the subgraph weights $w(S, U)$ and Super-Catalan numbers and Krawchuk polynomials 111

5.7.4 Application 4: Error probability of the trivial detector 112

5.7.5 Application 5: Lattice path sum interpretation of $w(S')$ 120

5.8 Chapter Summary 121

6 Thesis summary and Future Work 125

6.1 Main Contributions 125

6.2 Future work: OSN topology versus inference algorithm 126

6.2.1 Definitions 127

6.2.2 Barabasi-Albert (BA) model 128

6.2.3 Watts-Strogatz (WS) model 129

6.2.4 Empirical Results 132

Appendices 139

A Proof of Theorem 4.3.1 139

B Proof of Theorem 5.3.1 140

B.1 Stage 1: Using the high temperature character expansion identity 140

B.2 Stage 2: Computing the weights - $w(U,S)$: The binary code puncturing approach 141

B.3 Stage 3: Establishing the Krawchuk Polynomials/Super-catalan number connection. 147

B.4 Stage 4: Simplifications based on *oddity* of the cardinality of the vertex subset U 151

C Proof of Lemma 5.7.1 152

Bibliography 155

List of Tables

1 List of abbreviations used in this thesis x

2.1 Abbreviation of state acronyms 27

4.1 Table detailing the partition functions for various graph topologies 76

4.2 Table detailing the free entropy density for various graph topologies 78

List of Figures

1.1 Online Information Graph model using an MRF prior. 2

1.2 #iloveobamacare network 3

1.3 Twitter snapshot of the tweet announcing of the `#iloveobamacare` hashtag campaign 4

1.4 Tweet sentiment classification using just the textual features [1] 5

1.5 Online Information Graph model using an MRF prior. 8

1.6 Examples of Online Information Graphs. 15

1.7 Chapter Map 16

2.1 Definitions of type-I and type-II errors 19

2.2 The inter-county graph for Massachusetts -(Blue: Above median, Green:Below Median) 24

2.3 Brute force p-value calculations for 7 states of USA [AZ,CT,MA,ME,NH,NV and VT] 25

2.4 p-value comparisons for real world data sets involving counties of the states of USA 28

3.1 ML estimation of θ 35

3.2 Demonstration of the construction of the conditional (truncated) MRF 37

3.3 Demonstration of the NAC methodology on a 9 node grid graph 55

3.4 Movie returns classification using the IMDB dataset 56

3.5 Citeseer dataset. 57

3.6 The four statistical regions with nine divisions according to the United States Census Bureau. 58

3.7 Crime level classification in state counties 59

3.8 ML estimation the inverse temperature: Mean configuration energy with and without data as a function of the inverse temperature chosen 60

3.9 Variation of classification accuracy of the MPM and MAP solutions with respect to the inverse temperature chosen. 61

3.10 MAP showing the age adjusted Lung and Bronchus Cancer incidence rates in states 61

3.11 The spatial adjacency graph of the 48 contiguous states of USA 62

3.12 The community (regions) of the USA according to the US census bureau and the NG community detection algorithm 63

3.13 Histogram plot of smoking level features across different years 64

3.14 Estimated mean (μ_k) and variance (σ_k^2) across different years 64

3.15 Error rate comparisons for across different train-year test-year combinations 65

3.16 Error rate comparisons for the 4 models averaged across the different train-year test-year combinations 66

4.1 Model for latent sentiment detection. 70

4.2 The 'supportive' #iloveobamacare network ($t = +1$) scenario 72

4.3 The 'anti-supportive' #iloveobamacare network ($t = -1$) scenario 72

4.4 Variation of error exponent (α) with θ for different network topologies 79

4.5 Variation of the error exponent (α) with γ for different network topologies 81

4.6 Variation of the error exponent (α) with ε for different network topologies 82

5.1 The majority vote/opinion model 89

5.2 The Lollipop graph 99

5.3 Probability of error comparison between the ML (trivial) and the approximate MAP detectors for the 1001 node lollipop graph with $\theta = 0.1$ and fixed ε across the nodes. 100

5.4 Probability of error comparison between the Trivial, ML and the approximate MAP detectors for the 1001 node lollipop graph with $\theta = 0.1$ and $\varepsilon_i = \varepsilon_{avg} \times log(\Delta_i + 1)$. 101

5.5 Handling the external fields by using a clamped pseudo node 103

5.6 The spanning subsets of an appended 3-node chain graph showcasing the splitting of Z' into internal and external summations. 123

5.7 $P_{err|\mathbf{e}^T\mathbf{x}<0}$ vs q_{flip} for different topologies 124

6.1 Variation of classification error rate with varying power parameter of the BA model (with $n = 100$ and $m_{ba} = 2$) for the 4 inference algorithms considered. 130

6.2 Figure depicting graphs constructed by using the WS model with differing rewiring probabilities. 131

6.3 Variation of Clustering coefficient and average path length with varying rewiring probability of the WS model (with $n = 100$) 133

6.4 Variation of decoding time with varying power parameter of the BA model (with $n = 100$ and $m_{ba} = 2$) for the 4 inference algorithms considered. 136

6.5 Variation of classification error rate with varying rewiring probability of the WS model (with $n = 100$) for the 4 inference algorithms considered. 138

6 Possible range of pre-puncturing hamming weights possible for arriving at a post p-punctured codeword, $\tilde{\mathbf{b}}$, with hamming weight $d_H(\tilde{\mathbf{b}}) = i$. 145

7 Visualization of the codebook matrices in the code-puncturing approach 153

CW	Curie-Weiss	NG	Newman-Girvan
GIGO	Garbage In Garbage Out	NW	Noisy-Weak network effect
GPPF	Generalized Partial Partition Function	OH	Ohio
HT	High Temperature	OIG	Online Information Graphs
I.I.D	Independent and Identically Distributed	OSN	Online Social Network
ICM	Iterated Conditional Modes	P2P	Peer-to-Peer
IL	Illinois	PA	Pennsylvania
IMDb	Internet Movie Database	PF	Partition function
IN	Indiana	PPACA	Patient Protection and Affordable Care Act
LBP	Loopy Belief propagation	PPF	Positive part Partition function
LS	Latent Sentiment	PRN	Press Release Network
MAP	Maximum A Posteriori	RCV	Roll Call Vote
MF	Mean Field	SC	Super-Catalan
ML	Maximum Likelihood	SNA	Social Networks Analysis
MPM	Maximum Posterior Mean	SVM	Support Vector Machines
MRF	Markov Random Field	TRBP	Tree-Reweighted Belief Propagation
NAC	Network Aided Classification	UGM	Undirected Graphical Models
NAD	Network Aided Detection	US	United States

Table 1: List of abbreviations used in this thesis

Chapter 1

Introduction

1.1 Online Information Graphs

The emergence of the so called *Big data* paradigm and its disruptive potential has been has an extensively researched area of late [2–4]. This entails platforms of engagement where the scale of consumption and contribution of data is happening at a scale hitherto unseen in human history. One important technological artifact of this Big data paradigm has been the emergence of massive scale Online Social Networks (OSN) such as Facebook [5], Twitter [6] and Discus [7] with each having hundreds of millions of users engaging actively sharing, contributing and utilizing richly formatted data spanning textual forms, images, music and pictures. Parallel to this new mode of social engagement, we have has also been the rise of the *Open data* movement that has resulted in the creation of a plethora of online datasets, such as Wikipedia [8], Citeseer [9], Open Government sets (such as New York state's OPEN-NEWYORK data portal - [10] and the US federal government's DATA.GOV initiative [11]). This has already had far-reaching effects in the domains of fostering economic growth and innovation , championing of environmental issues, improving public health, sharing scientific research and preserving cultural heritage [12].

With such massive data at our disposal, it is only natural that a lot of research is currently underway to help make sense of this explosion of data in terms of both utilizing it for human good as well as understanding the explanatory science behind some of the interesting details that have emerged in these data repositories.

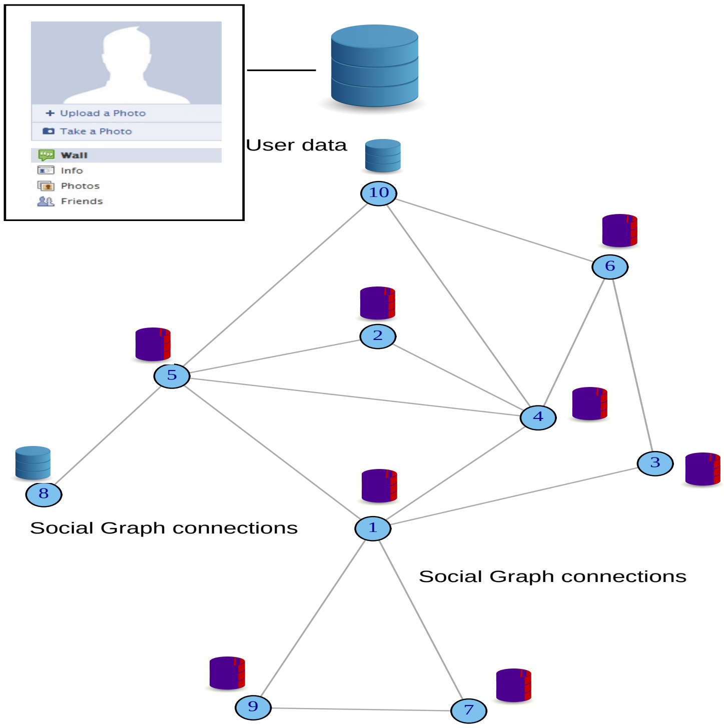


Figure 1.1: Online Information Graph model using an MRF prior.

This thesis is an attempt to contribute towards harnessing these massive repositories, such as OSNs, that have a graph structure associated with them.

The examples of OSNs and online datasets cited above all share the common feature that they can be thought of as *Online Information Graphs*, in the sense that the information embedded in them has a natural graph structure (See Figure 1.1). For example, in the case of Twitter, the graph vertices are the individuals and the graph edges are the follower/followee links. In Wikipedia, the graph vertices are the Topic pages while the graph edges are the hyperlinks relating the pages. In Open Government datasets, the graph can be obtained from the geographical locations of counties or states.

In this thesis , the focus is not on studying the science of these graphs from a *complex networks* viewpoint [13–15] or look at engineering problems such as efficient data retrieval. Rather, we focus on utilizing this underlying graph as an additional information source that helps solve some hard machine learning problems, specifically with regard to classification and detection of data. We begin by motivating a real world example from Twitter.

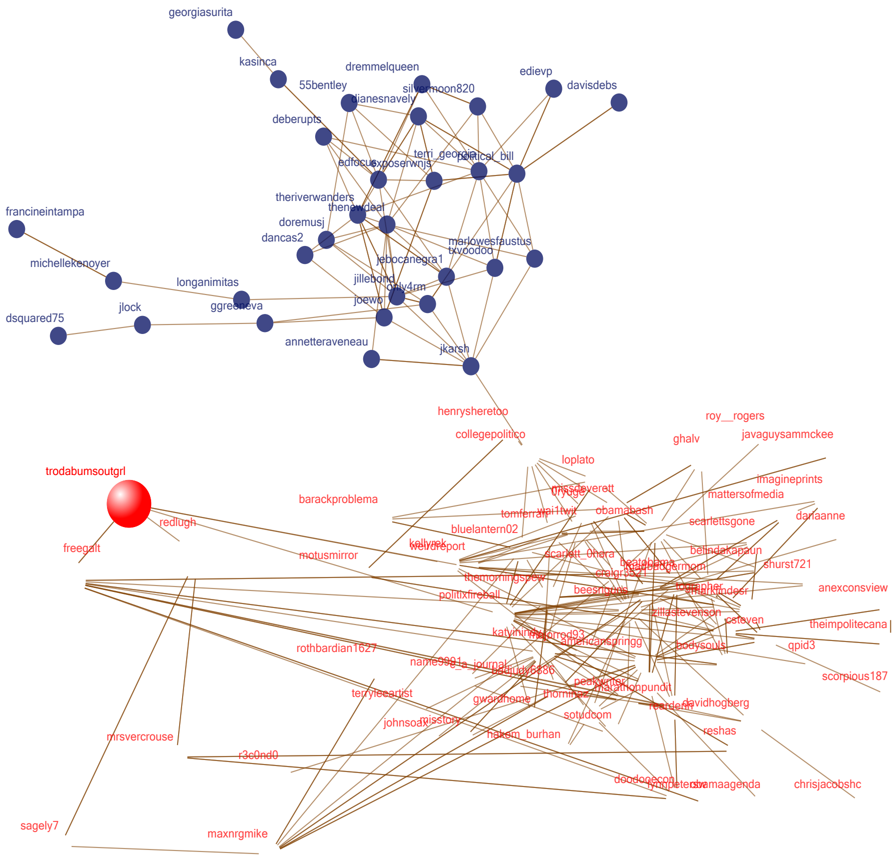
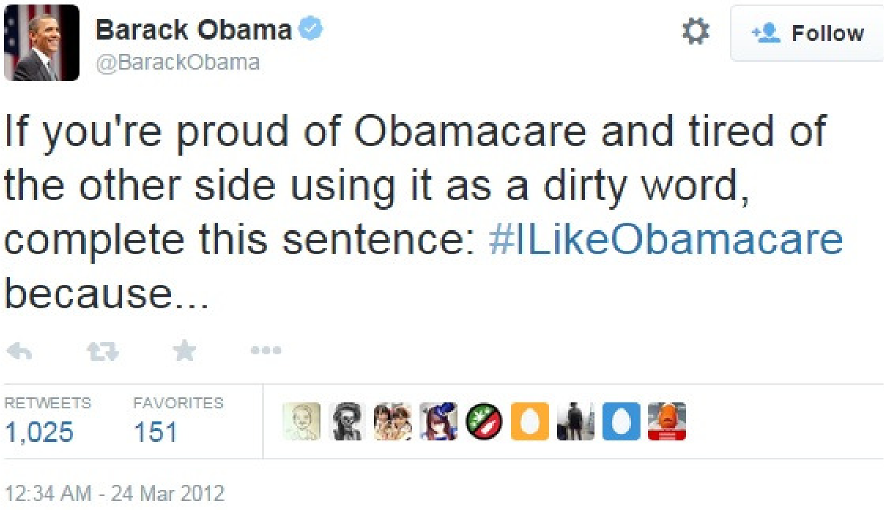


Figure 1.2: #iloveobamacare network

1.1.1 A real world example: Hashtag Hijacking

Seen in Figure 1.2 is the social graph of Twitter users who tweeted in response to the #iloveobamacare hashtag campaign initiated by the official Twitter account of the president of United States [16] seeking support towards the Patient Protection and Affordable Care Act (PPACA) [17], nick-named as *Obamacare* (See Figure 1.3). What followed was a deluge of Twitter users voicing their support of this campaign through positive tweets and also many Twitter users that attacked the #iloveobamacare hashtag with a series of sharp and sarcastic tweets resulting in what is called **Hashtag-Hijacking** [16]. With regard to Figure 1.4, we see that a certain Twitter user with the handle @trodadumsoutgrl used sarcasm to voice her opposition to the hashtag campaign but a machine learning tweet sentiment-algorithm such as *umigon* ([1]) which was trained using textual features failed to sense the sarcasm and wrongly classified it as a positive tweet. How-



Barack Obama
@BarackObama

If you're proud of Obamacare and tired of the other side using it as a dirty word, complete this sentence: #ILikeObamacare because...

RETWEETS 1,025
FAVORITES 151

12:34 AM - 24 Mar 2012

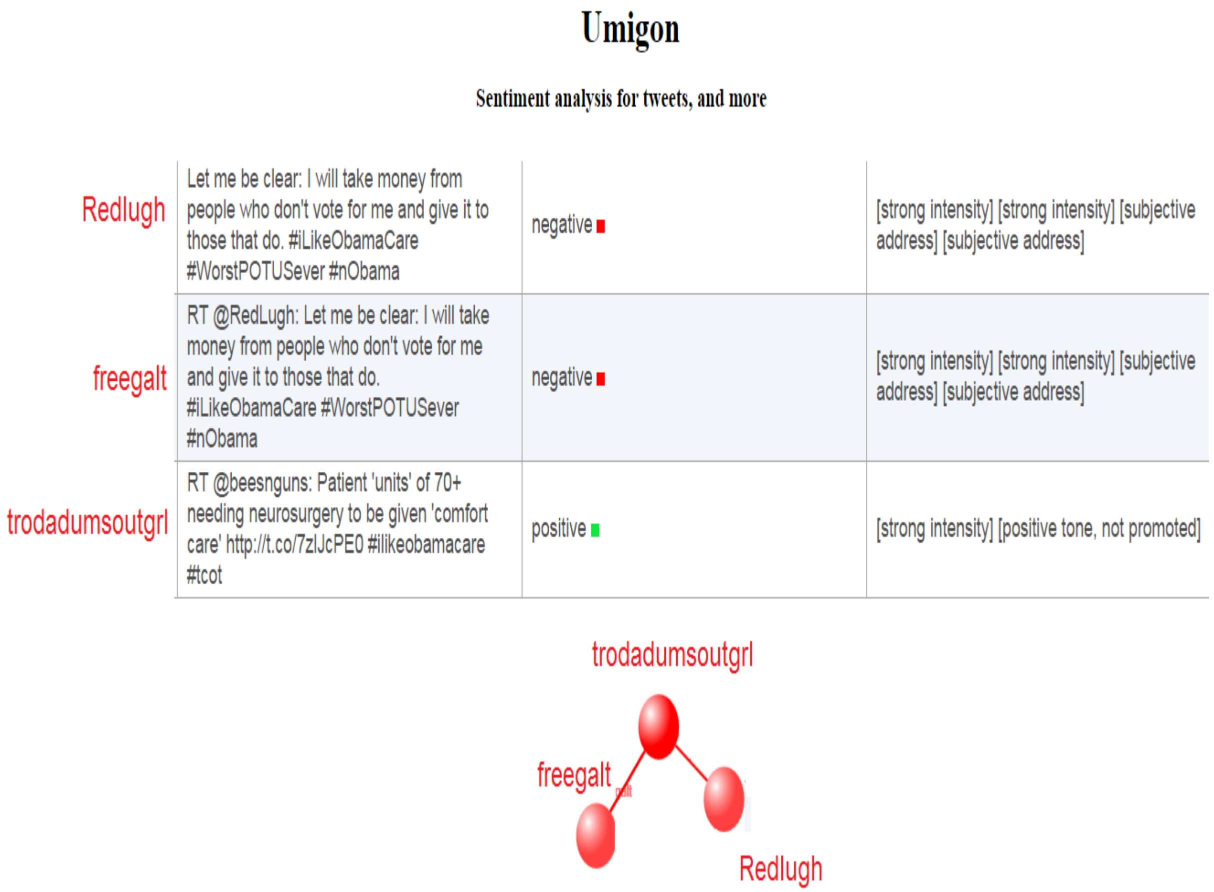
Figure 1.3: Twitter snapshot of the tweet announcing of the `#iloveobamacare` hashtag campaign

ever, a look at the underlying social graph reveals that two of her *neighbors* in the social graph had (re)tweeted tweets that were classified correctly as being negative. Now using the notion of *homophily*[1] ([18, 19]), one can argue that it is quite unlikely that the misclassified tweet was indeed positive. Thus, the network acts as an error correcting mechanism (or *code*) that can be harnessed as a statistical prior to help tackle hard classification and detection problems such as sarcasm detection as we just witnessed.

Extending this example, we posit that the classification performance of a purely features driven machine learning algorithm can be improved by noticing that the graph (of follower/followee links) between the users can provide a Bayes prior distribution to use for detection. If this prior can be learned (i.e., the model identified), it can be used to design more accurate detectors for many classification questions in Twitter (and other Online Information Graphs).

Keeping in mind the above Twitter example, we now go ahead and present the thesis set-up.

[1] Homophily or *love of the same* is defined as the tendency of individuals to associate and bond with similar others.



Umigon

Sentiment analysis for tweets, and more

Redlugh	Let me be clear: I will take money from people who don't vote for me and give it to those that do. #iLikeObamaCare #WorstPOTUSever #nObama	negative 🟥	[strong intensity] [strong intensity] [subjective address] [subjective address]
freegalt	RT @RedLugh: Let me be clear: I will take money from people who don't vote for me and give it to those that do. #iLikeObamaCare #WorstPOTUSever #nObama	negative 🟥	[strong intensity] [strong intensity] [subjective address] [subjective address]
trodadumsoutgrl	RT @beesnguns: Patient 'units' of 70+ needing neurosurgery to be given 'comfort care' http://t.co/7zlJcPE0 #ilikeobamacare #tcot	positive 🟩	[strong intensity] [positive tone, not promoted]

Figure 1.4: Tweet sentiment classification using just the textual features [1]

1.2 Markov Random Fields as label priors

We begin this section with a brief introduction in to Markov Random Fields, a member of the Undirected Graphical Models (UGM) family, which is used to define the network prior in this thesis.

In general, the formalism of Probabilistic Graphical Models (PGMs) serves as a unifying and rigorous framework for translating this loose notion of graph based correlation amongst data into a well defined probability distribution while elegantly capturing complex dependencies among random variables, and allowing building scalable large-scale multivariate statistical models [20, 21]. PGMs have found a lot of success in fields varying from bioinformatics, sensor networks [22], statistical physics, combinatorial optimization, signal and image processing, communication theory, information retrieval and statistical machine learning [20,21].

1.2.1 Markov Random Fields: A brief introduction

PGMs involve graphs, undirected or directed, in which nodes represent random variables, and the edge set of the graph captures the statistical dependence structure between random variables [23]. Simply put, they provide a compact representation of joint probability distributions. In our application, a node i in the representative graph $G(V,E)$, represents the related label discrete random variable x_i (taking values in its alphabet $\mathcal{X}$).

In scenarios where directionality of links holds no conceptual significance (such as spatial adjacency links), we propose to use a specific member of the graphical models family, namely the pair-wise Markov Random Fields (MRFs) [23]. The three Markov properties enumerated below help MRFs formalize the idea that the graph somehow encodes probabilistically, the statistical dependence between the related label random variables.

A given probability distribution $p(\mathbf{x})$ is said to be Markov with respect to a graph $G(V,E)$ if it satisfies the following three Markov properties.

1. **Global Markov Property:** Given two sets of nodes $A, B \subset V$ separated by $S \in V$, the associated random vectors, $\mathbf{x}_A$ and $\mathbf{x}_B$ are conditionally independent given $\mathbf{x}_S$.

$$p(\mathbf{x}_A, \mathbf{x}_B | \mathbf{x}_S) = p(\mathbf{x}_A|\mathbf{x}_S)p(\mathbf{x}_B|\mathbf{x}_S). \tag{1.1}$$

2. **Local Markov Property:** For any given node i with neighborhood, N_i, x_i is independent of the rest of the other variables $\mathbf{x}_{\setminus i}$, conditioned on the variables in N_i. Often, the neighborhood set of a node is termed as the node's *Markov blanket*.

$$p(x_i|\mathbf{x}_{\setminus i}) = p(x_i|\mathbf{x}_{N_i}). \tag{1.2}$$

3. **Pair-wise Markov property:** For any given 2 nodes i and j, we have x_i and x_j being independent conditioned on the rest of the variables, $\mathbf{x}_{\setminus\{i,j\}}$ if there is no edge connecting nodes i and j in the graph.

$$p(x_i, x_j|\mathbf{x}_{\setminus\{i,j\}}) = p(x_i|\mathbf{x}_{\setminus\{i,j\}})p(x_j|\mathbf{x}_{\setminus\{i,j\}}). \tag{1.3}$$

In our framework, we choose to use the pair-wise MRF [23] whose probability distribution can be written as,

$$p(x_1, x_2, ..., x_n) = p(\mathbf{x}) = \frac{1}{Z} \prod_{(i,j)\in E} \psi_{ij}(x_i, x_j), \tag{1.4}$$

where

$$Z = \sum_{\mathbf{x}\in\mathcal{X}^n} \prod_{(i,j)\in E} \psi_{ij}(x_i, x_j)$$

is the so-called *partition-function* which helps normalize the probability distribution and $\left\{\psi_{ij}(x_i, x_j)\right\}_{(i,j)\in E}$ are the *edge-potential functions*. Equation (1.4), essentially captures the side-information brought about by the network as a **probability prior**, which we will incorporate in our framework in the ensuing section.

1.2.2 The homogeneous Ising prior with constant external field

We begin this subsection by choosing the following specification of the edge-potential function $\psi_{ij}(x_i, x_j) = \exp\{\theta x_i x_j\}$ and a *signed* binary label set, $\mathcal{X} = \{-1, +1\}$. This results in a specific type of MRF called the *homogeneous Ising model* [24] , which is given by,

$$p(\mathbf{x}) = \frac{\exp\left\{\theta \sum\limits_{(i,j)\in E} x_i x_j\right\}}{Z(\theta)} = \frac{\exp\left\{\theta \mathbf{x}^T A \mathbf{x}\right\}}{Z(\theta)}. \tag{1.5}$$

Now consider this *Ising prior* on an undirected graph[2] with *n user* vertices V, as shown in Figure 1.5 with 6 user vertices. As an example, the user vertices could be Twitter individuals. The edge-set E of this sub-graph may be obtained using Twitter's follower/followee relationships, or in some cases, using the @-mentions in the tweets [25].

In the figure, $\mathbf{x} \in \{-1, +1\}^n$ is the binary vector of n user labels. In Twitter, these could be the sentiments of the individuals towards some event, with $x_i = +1$ (or $x_i = -1$) modeling the positive (or negative) sentiment of the i^{th} individual. Let $t \in \{-1, +1\}$ be a uniformly distributed binary latent variable which homogeneously influences every user of the network as a local field of strength γt. In the Twitter example, this could be the sentiment bias of the

[2] The graph G is undirected since it models correlation, rather than influence flows.

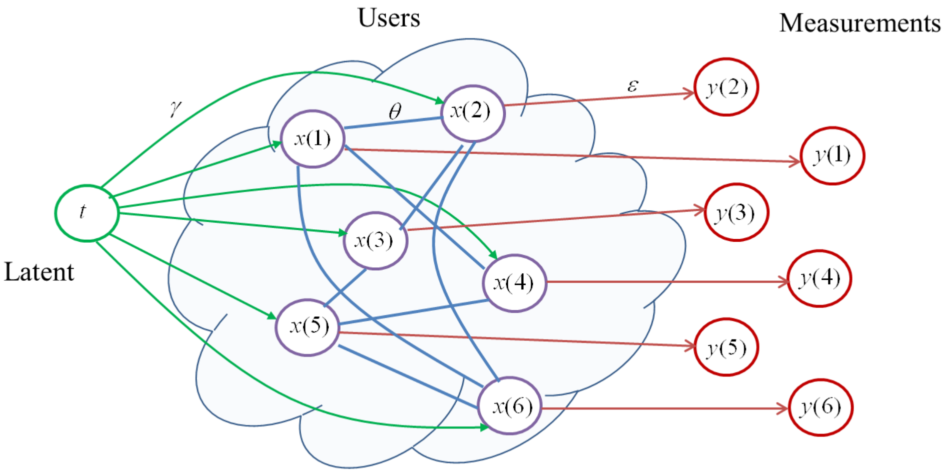


Figure 1.5: Online Information Graph model using an MRF prior.

population. (Setting $\gamma = 0$ eliminates t from consideration.) The user labels are assumed to be sampled from an underlying MRF prior. Thus, the conditional distribution of $\mathbf{x}$ given t is,

$$p(\mathbf{x}|t) \quad = \quad \frac{\exp\left\{\theta \mathbf{x}^T A \mathbf{x} + \gamma t \mathbf{e}^T \mathbf{x}\right\}}{\sum_{\mathbf{x}} \exp\left\{\theta \mathbf{x}^T A \mathbf{x} + \gamma t \mathbf{e}^T \mathbf{x}\right\}}, \tag{1.6}$$

where A is the upper triangular adjacency matrix of user sub-graph vertices ($A_{ij} = 1$ if $(i, j) \in E,\ i < j$, else $A_{ij} = 0$). $(\cdot)^T$ is vector transpose and $\mathbf{e}$ is the vector of all-ones. θ is called the *inverse temperature (Gibbs) parameter* or simply the *global edge-potential*. User label correlation increases when θ increases. Notice that from a communications perspective, $\mathbf{x}$ is a codeword randomly chosen in response to bit t. Let $\mathbf{y}$ be a noisy measurement of $\mathbf{x}$. In Twitter, it may be estimated from the features extracted from the user profiles or could even be the label vector estimated by a sophisticated classifier algorithm from user tweets, such as the ones in [26] and [27]. Since the above measurements are made separately for each vertex, the channel from $\mathbf{x}$ to $\mathbf{y}$ is memoryless, so that

$$p(\mathbf{y}|\mathbf{x}) \quad = \quad \prod_{i=1}^{n} p(y(i)|x(i)). \tag{1.7}$$

The measurements could be discrete or continuous. If the measurements are binary, or if (in the continuous case) the noise is Gaussian, the joint distribution may be written as an Ising model,

$$p(t, \mathbf{x}, \mathbf{y}) \quad = \quad \frac{1}{2\,Z(t)} \exp\left\{\theta \mathbf{x}^T A \mathbf{x} + \varepsilon \mathbf{y}^T \mathbf{x} + \gamma t \mathbf{e}^T \mathbf{x}\right\}, \tag{1.8}$$

$$Z(t) \quad = \quad (2\cosh(\varepsilon))^n \sum_{\mathbf{x}} \exp\left\{\theta \mathbf{x}^T A \mathbf{x} + \gamma t \mathbf{e}^T \mathbf{x}\right\}, \tag{1.9}$$

where ε depends inversely on the measurement noise variance.

To showcase the potential applications of the model, Figure 1.6 contains six diverse real-world scenarios that can be modeled by (1.8). (Only the sub-graph G of *user* vertices carrying the labels ($\mathbf{x}$) is shown for clarity. The latent variable and measurement vertices are not shown.) Figure 1.6(a) represents the #-Obamacare Twitter network dicussed earlier in this chapter.

Figure 1.6(b) shows a US Senator joint Press Release Network (PRN) of the 110th Congress being used to define the MRF prior. The PRN is constructed based on joint press releases by senators. Measurements include party affiliation, political leanings and pre-declared ideological positions [28]. This graph can be harnessed to predict the votes cast by the Senators during a Roll call vote on legislation.

Figure 1.6(c) shows an application using a graph constructed based on geographical adjacency of the 48 contiguous states of the United States. This can be used to predict the Lung-and-Bronchus cancer levels in the state (as being 'Hi' or 'Lo') [29] using state-level adult smoking rate measurements.

Figure 1.6(d) represents an application where the vertices represent movies obtained from the IMDB movie database [30]. An edge is drawn between 2 movies if they share a common production house. The inter-movie graph thus constructed is used to define the MRF prior. The measurements are the *lemmatized* textual features extracted from the script of the movies (via the IMDB database) using the Bag-of-words model [31].

Figure 1.6(e) shows a graph constructed based on geographical adjacency of the 102 contiguous counties of the Illinois state being used to define the MRF prior. Measurements are average income, voting tendencies and health statistics from openly available data sources such as census surveys [32]. In the world of currency finance, financial experts often analyze a basket of currencies with respect to a standard currency such as US dollar by constructing a Minimal Spanning Tree (MST) graph, as shown in Figure 1.6(f). This tree is constructed based on correlation of the daily time series of the exchange rates of the currencies [33], and thus, shows the financial dependence of countries on each other. Measurements include financial data (GDP, inflation, etc.) of the countries. This model can be used, for example, to detect weakening economies.

Having specified the graphical model of choice in (1.8) and looked at the myriad of Real world examples of the graphs, we now move on to present the thesis outline in the form of a chapter map which provides a visual diagrammatic representation of the work presented here.

1.3 Thesis Outline

The use of a prior distribution is a rather contentious one and invokes fierce debates in the statistics community [34,35]. However, in the machine learning community, as long as using the prior results in substantial improvement of classification accuracy, a conjecture based justification for the prior is tolerated[3]. Ostensibly, this approach comes with the innate risk that a misspecified prior whose validity is not ascertained a priori to its usage might result in classification accuracy being much worse than not using the prior at all. Hence, there is clearly a need for a *pre-processing* step where we subject the prior to some tests which will validate its utility.

In Chapter 2, we explore this very issue of coming up with the hypothesis testing based framework to justify the usage of the homogeneous ferromagnetic Ising prior of (1.5) (which we repeat here for clarity),

$$p(\mathbf{x}) = \frac{\exp\left\{\theta \sum_{(i,j)\in E} x_i x_j\right\}}{Z(\theta)}.$$

As seen Figure 1.7, if the hypothesis testing framework fails to validate the prior, then the practitioner is advised to explore other options of exploiting the suspected correlation amongst the labels or just stick to the prior assumption and use the likelihood maximization approaches for his machine learning problem.

Should the hypothesis testing framework validate the prior, we move on to Chapter 3 where we consider the Network Aided Classification (NAC) problem which is defined thus.

Network Aided Classification (Chapter 3): In this problem, the objective is to obtain an estimate $\hat{\mathbf{x}}$ of the true labels, given the measurement vector $\mathbf{y}$. An example is to use the IMDB information graph (Figure 1.6(d)) to predict which movies will be commercial hits, or to use the inter-county adjacency graph (Figure 1.6(e)) to predict, crime-levels in individual counties of the state. The estimate $\hat{\mathbf{x}}$ can be obtained as a MAP estimate or Maximum Posterior Marginal (MPM) [37] on the model (1.8),

[3] This cultural split in explored in depth in [36]

$$\hat{\mathbf{x}}_{map} = \arg \max_{\mathbf{x}\in\{-1,1\}^n} \sum_t p(t,\mathbf{x},\mathbf{y}), \quad (1.10)$$

$$\hat{x}_{i,mpm} = \arg \max_{x(i)\in\{-1,1\}} \sum_{t,\mathbf{x}_{\setminus i}} p(t,\mathbf{x},\mathbf{y}). \quad (1.11)$$

In chapters 4 and 5, we explore the Network Aided Detection (NAD) problems of Latent sentiment detection and Majority sentiment detection.

Latent sentiment detection (Chapter 4): This is the problem of classifying the latent variable t as ±1. The classifier $\hat{t} = \pm 1$ can be obtained from the a posteriori probability ratio, $l(\mathbf{y})$,

$$l(\mathbf{y}) = \frac{\sum_{\mathbf{x}} p(t=1,\mathbf{x},\mathbf{y})}{\sum_{\mathbf{x}} p(t=-1,\mathbf{x},\mathbf{y})}. \quad (1.12)$$

The model (1.8) is applicable to several real-world latent variable classification scenarios. We begin by assuming that there exists a latent sentiment ($t \in \{-1,+1\}$), which will cause a certain concrete event in the future. This event may be the passage (or defeat) of a bill in the Senate, or an up (or down) movement of the stock market, when $t = +1$ (or -1, respectively). The goal is to automatically predict this future event using the expressed sentiments gathered from Twitter tweets ($\mathbf{y}$ in our model), such as in Figure 1.6(a). Of course, in the Twitter example of Obamacare, one can do this knowing side-information such as the political stance of the individuals. Automatic detection aims to use $\mathbf{y}$ alone, without requiring human intervention through specialized side information. (It was later revealed in a national survey conducted by the `Pew Research Center` and `USA TODAY` [38], that 63% of liberally-minded voters supported Obamacare, i.e., $t = +1$.)

Majority vote or sentiment classification (Chapter 5): Another problem related to latent variable classification is that of majority vote or sentiment classification. Here, the goal is to estimate $m = sign(\mathbf{e}^T\mathbf{x})$. Practical scenarios relating to this model would be predicting the passing (or failing) of a certain bill in the Senate at the end of a Roll Call Vote (RCV) in Figure 1.6(b)(b), or evaluating the net-trend or the majority opinion of a group of Twitter-users tweeting about a certain trending topic of interest. In this case, the joint distribution of labels $\mathbf{x}$ and measurement vector $\mathbf{y}$ is the same as (1.8) with $\gamma = 0$. The optimal Maximum Aposteriori Probability

(MAP) [39] majority vote or sentiment detector $\hat{m} = \pm 1$ uses the ratio,

$$l_m(\mathbf{y}) \quad = \quad \frac{\sum\limits_{\mathbf{x}:\mathbf{e^T x \geq 0}} p(\mathbf{x}, \mathbf{y})}{\sum\limits_{\mathbf{x}:\mathbf{e^T x < 0}} p(\mathbf{x}, \mathbf{y})}.$$

The thesis summary and suggestions for future work is presented in Chapter 6.

1.3.1 Mathematical notation in the thesis

In this thesis, we use the following notation.

- $x,\ \theta$: (Lower case) a realization of a scalar random variable or a deterministic variable depending on the context .
- $\mathbf{x}, \boldsymbol{\theta}$: An n-dimensional vector of values.In this thesis vectors are considered to be column vectors, i.e., of dimension $n \times 1$ unless stated otherwise.
- x_i : the i^{th} component value of the vector $\mathbf{x}$.
- A or $\mathbf{C}$: (Upper case) A matrix of scalars.
- $\mathcal{X}$: (Upper case - calligraphic). Denotes a set. Typically, $\mathcal{X} = \{-1, +1\}$ otherwise mentioned.
- $\mathbf{x}^T,\ A^T$: The superscript T here indicates the transpose of the column vector $\mathbf{x}$ or the transpose of the matrix A .
- $\mathbf{x}^T\mathbf{y}$: The inner product of $\mathbf{x}$ and $\mathbf{y}$, i.e., $\sum\limits_{i=1}^{n} x_i y_i$
- $g(\mathbf{x})$: some scalar-valued function g of the vector $\mathbf{x}$.
- $\mathbf{y} = A\mathbf{x}$:(Matrix-vector multiplication) If A has dimension $n \times d$ and $\mathbf{x}$ has dimension $d \times 1$, then $\mathbf{x}$ has dimension $n \times 1$.
- $p(\mathbf{x})$: The probability mass function (pmf) of the random vector $\mathbf{x}$.
- $p(x|y),\ p(\mathbf{x}|\mathbf{y})$: The conditional distribution (or density) of the random variable X given that variable Y takes value y (generalizes to vector arguments $\mathbf{x}$ and/or $\mathbf{y}$ naturally).

- $E[g(\mathbf{x})]$: the expectation of the function $g(\mathbf{x})$ with respect to the probability distribution $p(\mathbf{x})$. That is, $E[g(\mathbf{x})] = \sum\limits_{\mathbf{x}} p(\mathbf{x})g(\mathbf{x})$.
- $\hat{m}$, $\hat{\mathbf{x}}$: An estimate of the variable m or the vector $\mathbf{x}$.
- $\prod\limits_{i=1}^{n}$: The product from $i = 1$ to $i = n$.
- $\sum\limits_{i=1}^{n}$: The sum from $i = 1$ to $i = n$.

For partial derivatives, we use the following notation, $\frac{\partial l(\theta,\boldsymbol{\varepsilon})}{\partial \theta} = l^1_{\theta}$, $\frac{\partial l(\theta,\boldsymbol{\varepsilon})}{\partial \varepsilon_i} = l^1_{\varepsilon_i}$, $\frac{\partial^2 l(\theta,\boldsymbol{\varepsilon})}{\partial \varepsilon_i^2} = l^2_{\varepsilon_i^2}$, $\frac{\partial^2 l(\theta,\boldsymbol{\varepsilon})}{\partial \theta^2} = l^2_{\theta^2}$ and $\frac{\partial^2 l(\theta\boldsymbol{\varepsilon})}{\partial \varepsilon_i \partial \theta} = l^2_{\varepsilon_i\theta}$.

1.4 Related work

Broadly speaking, this thesis is a contribution to the growing body of literature in the broad field of *Network Science*[4].

In this section, we review the distinct avenues under which research has been carried out which helps differentiate our contributions.

1.4.1 Complex networks: Study of the underlying graph structure

Many empirical studies conducted showed that real world networks, both man-made as well as naturally occurring ones, exhibited certain topological characteristics such as scale-free degree distribution, high transitivity and low graph diameter [13–15]. This lead to an explosion of interest under the banner of study of complex networks where a *complex network* was defined to be a graph with highly non-trivial topological features that are not expected to occur in simple networks such as lattices or Erdos-Renyi random graphs [40]. Many generative toy models such as the Watts-Strogatz Small World model [41], the Barabasi-Albert scale-free model [42] and the Exponential Random Graph Models [43] were proposed with a statistical physics narrative in order to explain the formulation of such networks and a plethora of modifications have been put

[4] The United States National Research Council defines network science as *"the study of network representations of physical, biological, and social phenomena leading to predictive models of these phenomena."*

forth for each of these models to fit certain idiosyncratic observations in specific sub-domains of interest (See [44–47]).

Some important applications of these studies include characterization of different notions of vertex-rankings (or centrality measures) [19, 48–50], understanding of different community formation models [51–53], link prediction models [54, 55], models pertaining to the temporal evolution of networks [56, 57], techniques to compress the graph adjacency matrix [58, 59] and innovative visualization paradigms [60–63] that would lead to meaningful abstraction of a complicated graph containing millions of vertices.

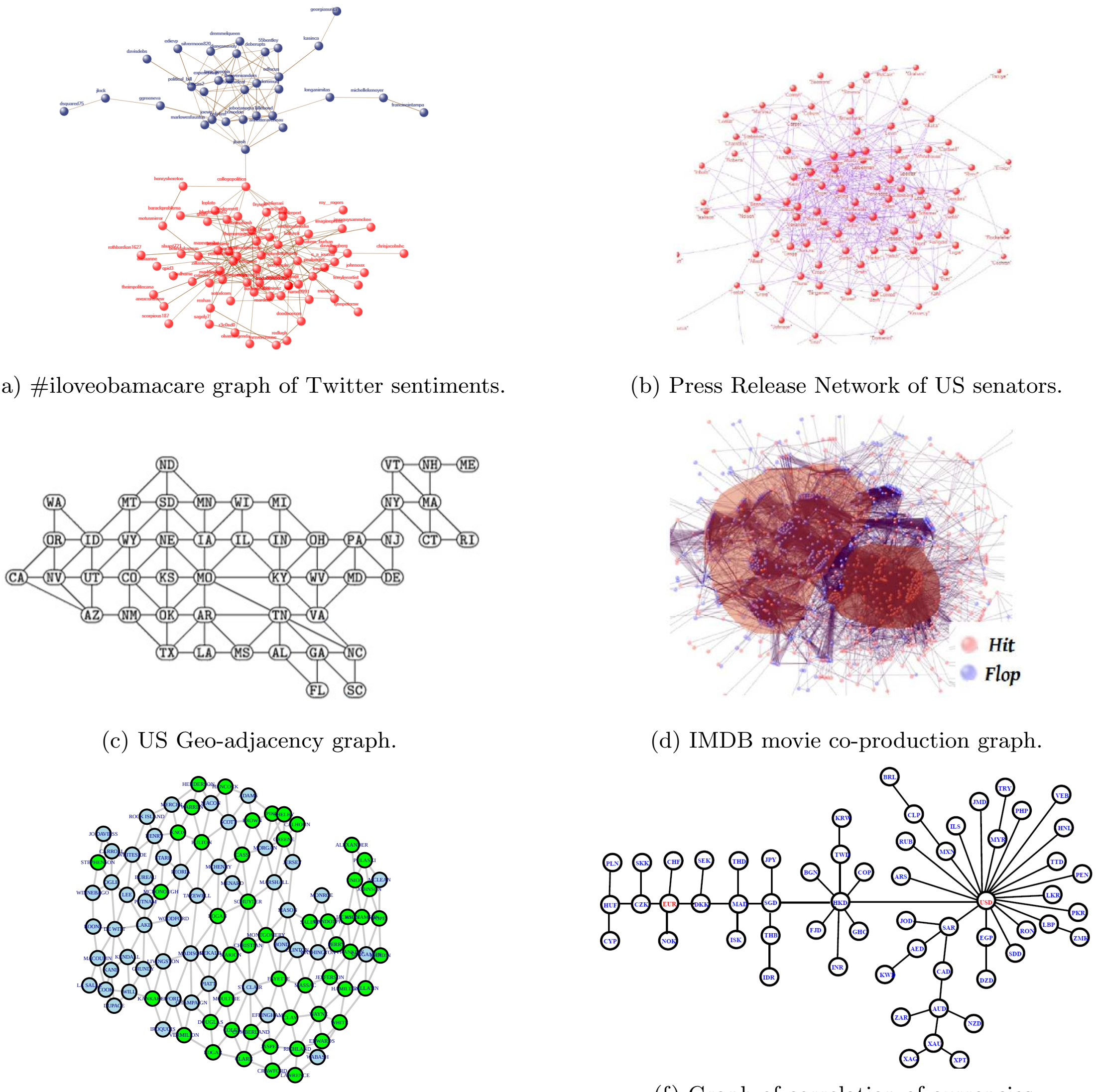
In contrast, this thesis does not study the graph topology in isolation, but rather as it relates to the problem of classification.

1.4.2 Gossip and Epidemiology: Information processes on graphs

In this line of research, human engineered networks such as sensor networks, peer-to-peer (P2P) networks and mobile ad-hoc networks, as well as human-to-human contact networks are analyzed. The basic onus lies on solving a global constrained optimization problem by allowing the nodes to communicate locally with each other (*gossip*) and performing distributed local computations. This line of research also entails studying the spreading of information processes (such as beliefs, rumor, traits and infections) on these graphs all while placing strong constraints on the order of neighborhood of communicability of the nodes, the computational resources available and *synchronizability* of the nodes [64–66].

The basic differentiating factor that sets this thesis apart is that in the studies listed above, the existence of an edge entails, literally, a physical link (such as in sensor networks, P2P networks, road networks, human-to-human contact networks etc), which places constraints on communicability between the nodes, whereas in our framework the presence of an edge entails capturing conditional dependence between the entities (nodes), an idea formalized by the Markov Random Field prior.

With this background in mind, we now delve in to our contributions, beginning with the hypothesis testing framework we use to validate the Ising prior specified in (1.5).



(a) #iloveobamacare graph of Twitter sentiments.

(b) Press Release Network of US senators.

(c) US Geo-adjacency graph.

(d) IMDB movie co-production graph.

(e) Illinois inter-county crimel level graph

(f) Graph of correlation of currencies.

Figure 1.6: Examples of Online Information Graphs.

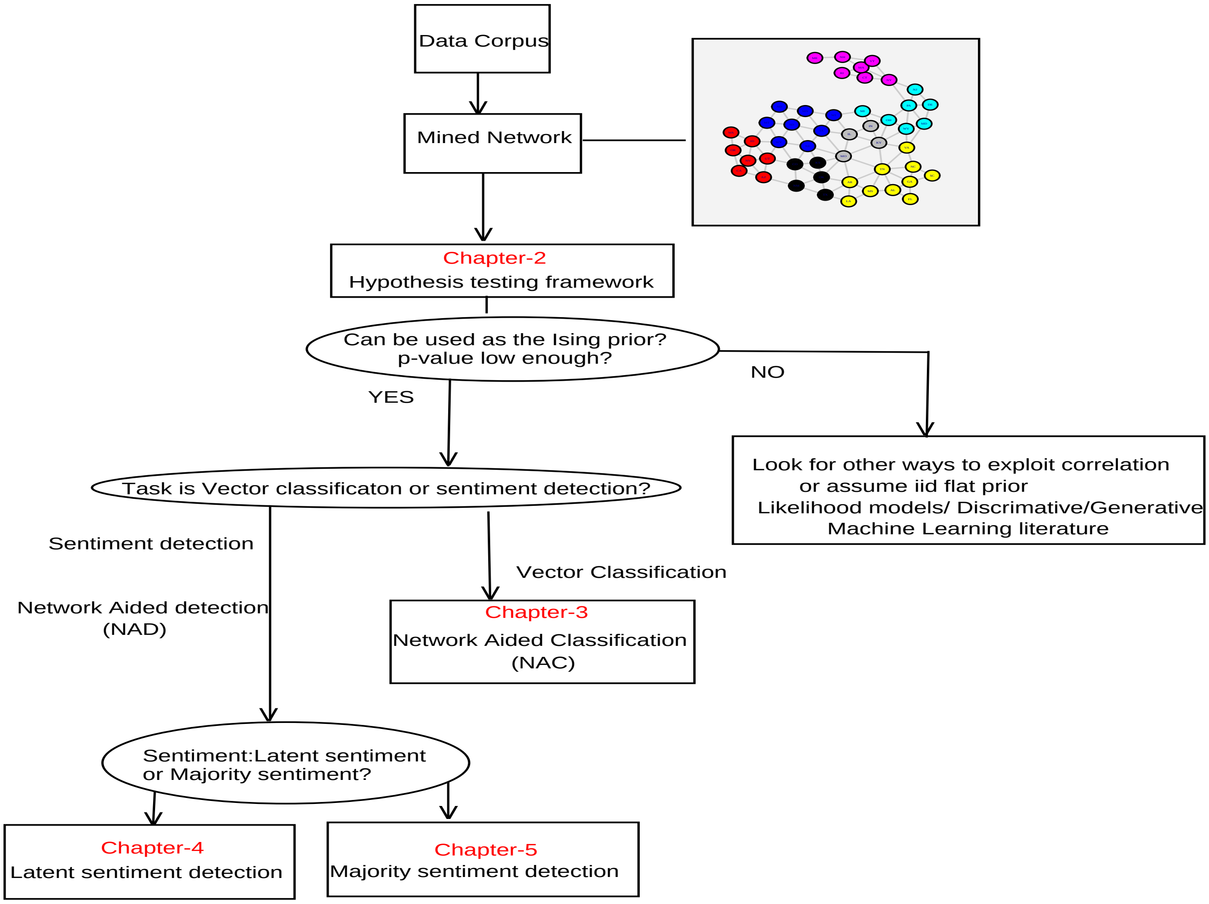

Data Corpus
Mined Network
Chapter-2
Hypothesis testing framework
Can be used as the Ising prior?
p-value low enough?
NO
YES
Look for other ways to exploit correlation
or assume iid flat prior
Likelihood models/ Discrimative/Generative
Machine Learning literature
Task is Vector classificaton or sentiment detection?
Sentiment detection
Vector Classification
Network Aided detection
(NAD)
Chapter-3
Network Aided Classification
(NAC)
Sentiment:Latent sentiment
or Majority sentiment?
Chapter-4
Latent sentiment detection
Chapter-5
Majority sentiment detection

Figure 1.7: Chapter Map

Chapter 2

Model Validation

2.1 Introduction

Recently, in the context of Gaussian Markov Random Fields (GMRF), the authors of [67] considered the problem of hypothesis testing against independence in the special case of the acyclic dependency graph, and derived the expression for the log-likelihood ratio of detection. This would serve as pre-inference model validation step before employing the GMRF as a prior for further inference.

However, in applications where discrete MRFs, such as the Ising model, are used as statistical priors [24,39,68,69], the justification for its usage is provided in the form of an *intuitive assumption* or a *hunch* which is eventually vindicated by the improvement in the classification error probability of the inference task at hand.

We have noticed during our research endeavors that modeling the underlying network as an Ising prior even when it *seems reasonable* may yield classification error rates worse than one obtains without the network on account of model mismatch. So now, the question is: *Is there a reasonably rigorous and quick way of ascertaining whether the underlying network can indeed be modeled as a (ferromagnetic) Ising prior*?

In order to answer this, we use tools from the hypothesis testing framework developed by the statisticians over the years [70,71].

Doing so will empower to use the (ferromagnetic) Ising prior for both classification and detection in Chapters 3-5.

2.2 The homogeneous ferromagnetic Ising prior

The aim of this chapter is to harness this hypothesis testing framework to vindicate the usage of the homogeneous ferromagnetic Ising prior which is used in the forthcoming chapters and is specified by (1.5) in the Chapter 1, which is,

$$p(\mathbf{x}) = \frac{\exp\left\{\theta \mathbf{x}^T A \mathbf{x}\right\}}{Z(\theta)}, \tag{2.1}$$

where θ is the common Ising edge potential, A is the upper triangular adjacency matrix and $Z(\theta)$ is the partition function defined as,

$$Z(\theta) = \sum_{\mathbf{x}} \exp\left\{\theta \mathbf{x}^T A \mathbf{x}\right\}. \tag{2.2}$$

We gather from (2.1) that this Ising prior belongs to the one-parameter exponential family, defined by,

$$p_\theta(\mathbf{x}) = c(\theta) exp\{Q(\theta) T(\mathbf{x})\} h(\mathbf{x}), \tag{2.3}$$

with $Q(\theta) = \theta$, the *sufficient statistic,* $T(\mathbf{x}) = \mathbf{x}^T A \mathbf{x}$, $c(\theta) = 1/Z(\theta)$ and $h(\mathbf{x}) = 1, \forall \mathbf{x}$. Note that $p_\theta(\mathbf{x})$ in (2.3) above reads as probability distribution of $\mathbf{x}$ *parameterized by* θ where θ is a deterministic variable ([72]).

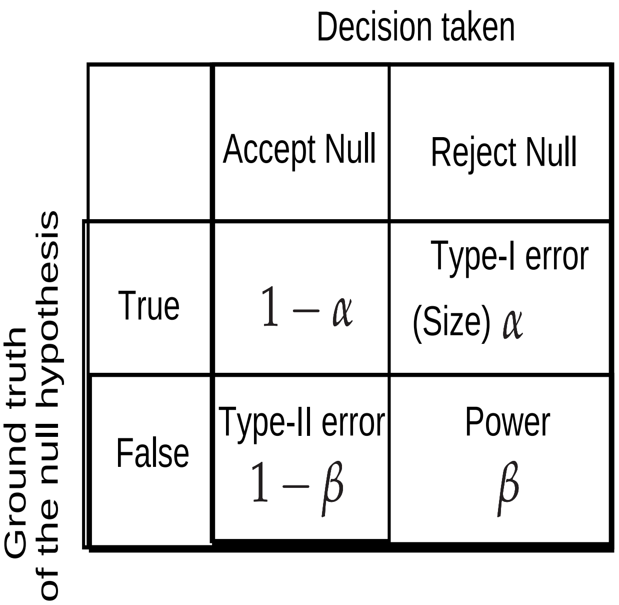
2.2.1 Testing against the uniform (i.i.d) prior

In this chapter, we look at the hypothesis testing problem with the Ising prior being the alternative hypothesis benchmarked against the null hypothesis which is the uniform ($\theta = 0$) prior.

$$\begin{aligned} H_0 &: \mathbf{x} \sim p_0(\mathbf{x}) = \left(\frac{1}{2}\right)^n \forall\ \mathbf{x} \in \{-1, +1\}^n \\ H_1 &: \mathbf{x} \sim p_1(\mathbf{x}) = \frac{\exp\left\{\theta \mathbf{x}^T A \mathbf{x}\right\}}{Z(\theta)}, \theta > 0. \end{aligned} \tag{2.4}$$

Another way of representing (2.4) is as follows ([71]),

$$H_0 : \theta = 0 \text{ vs } H_1 : \theta > 0. \tag{2.5}$$



		Decision taken	
		Accept Null	Reject Null
Ground truth of the null hypothesis	True	$1-\alpha$	Type-I error (Size) α
	False	Type-II error $1-\beta$	Power β

Figure 2.1: Definitions of type-I and type-II errors

2.3 Definitions

Consider the problem of testing

$$H_0 : \mathbf{x} \sim p_0(\mathbf{x}) \text{ vs } H_1 : \mathbf{x} \sim p_1(\mathbf{x}). \tag{2.6}$$

$p_0(\mathbf{x})$ is termed the *null hypothesis* distribution and $p_1(\mathbf{x})$ is termed the *alternative hypothesis distribution*.

If the null and the alternative hypothesis distributions $p_0(\mathbf{x})$ and $p_1(\mathbf{x})$ are parameterized by a single parameter, say, θ, the hypothesis testing problem in (2.6) is recast as

$$H_0 : \theta \in \Theta_0 \text{ vs } H_1 : \theta \in \Theta_1. \tag{2.7}$$

If $\Theta_0 = \{\theta_0\}$ and $\Theta_1 = \{\theta_1\}$ (where θ_0 and θ_1 are scalars), then the hypothesis test is said to be a *simple* hypothesis test. Else, the hypothesis test is said to be a *composite* hypothesis test.

Upon observing $\mathbf{x}$, we take a decision in favor of either the null hypothesis or the alternative hypothesis using $\phi(\mathbf{x})$, which is termed as the *decision rule*, *critical function* or simply, *test*, defined as,

$$\phi(\mathbf{x}) = \begin{cases} 1 & \text{if } H_1 \text{ chosen} \\ 0 & \text{otherwise.} \end{cases} \tag{2.8}$$

The *size* of a test ϕ is the *type-I* error rate *or error-rate of the first kind*, which is the probability of incorrect rejection of a true null hypothesis and is given by, (see Figure 2.1),

$$\alpha = \sup_{\theta \in \Theta_0} E_\theta[\phi(\mathbf{x})], \tag{2.9}$$

where $E_\theta[.]$ denotes the expectation taken with regard to $p_\theta(\mathbf{x})$.
The *power* of a test ϕ, denoted by β, is defined to be $(1 -$ type-II error rate), where *type-II error rate* is the probability of failure to reject a false null hypothesis (see Figure 2.1) . That is,

$$\beta(\theta) = E_\theta[\phi(\mathbf{x})],\ \forall\ \theta \in \Theta_1. \tag{2.10}$$

Definition If the family of densities $\{p_\theta : \theta \in [\theta_0, \theta_1] \subset \mathbb{R}\}$, with sufficient statistic, $T(\mathbf{x})$, is such that $p_{\theta'(\mathbf{x})}/p_\theta(\mathbf{x})$ is nondecreasing in $T(\mathbf{x})$ for each $\theta < \theta'$, then the family is said to have monotone likelihood ratio (MLR).

It is a well known result [70,71] that for the one-parameter exponential family in (2.3), if $Q(\theta)$ is nondecreasing, then this family has the MLR property.

Definition Let $\mathcal{C}_\alpha \equiv \{\phi : \phi \text{ is of size } \alpha\}$. A test ϕ^* is uniformly most powerful of size α (or UMP of size α) if it has size α (according to (2.9)) and if $E_\theta[\phi^*(\mathbf{x})] \geq E_\theta[\phi(\mathbf{x})]$ for all $\theta \in \Theta_1$ and all $\phi \in \mathcal{C}_\alpha$.

Theorem 2.3.1 (Karlin - Rubin [70]). *If $\mathbf{x}$ has density $p_\theta(\mathbf{x})$ with MLR in $T(\mathbf{x})$,*

1. *Then there exists a UMP level α test of $H_0 : \theta = \theta_0$ vs $H_1 : \theta > \theta_0$ which is of the form*

$$\phi_{T_{th}}(\mathbf{x}) = \begin{cases} 1 & \textit{if } T(\mathbf{x}) > T_{th} \\ \gamma & \textit{if } T(\mathbf{x}) = T_{th} \\ 0 & \textit{if } T(\mathbf{x}) < T_{th} \end{cases} \tag{2.11}$$

with $E_{\theta_0}[\phi_{T_{th}}(\mathbf{x})] = \alpha$.

2. *$\beta(\theta) = E_\theta[\phi_{T_{th}}(\mathbf{x})]$ is increasing in θ for $\beta < 1$.*

3. *For all θ', this same test is the UMP level $\alpha' \equiv \beta(\theta')$ test of $H_0 : \theta = \theta'$ vs $H_1 : \theta > \theta'$.*

Now, for the hypothesis test specified in (2.5), we design the Karlin-Rubin test of (2.11) as follows. Firstly, we set the *size* α, to an acceptably low value (typically, $\alpha = 0.05$). Then, we solve for the threshold test statistic T_{th} to be plugged in to (2.11) by solving, $E_{\theta_0}[\phi_{T_{th}}(\mathbf{x})] = \alpha$.
A more general approach advocated in statistics [70, 71] is to compute the *p-value* for the data observed and benchmark it with the fixed α and claim that the alternative hypothesis (of the Ising prior) can be favored with *strong* statistical significance. This leads us to the following section of defining this p-value and looking at methods to compute the same for our scenario.

2.4 p-value

In statistics [70, 71], the size α is used to set the bar for how *extreme* the data must be before we can confidently reject the null hypothesis. On the other hand, *p-value* is calculated to indicate how extreme the data *really is*. The *p-value* is defined as the probability, under the assumption of the null hypothesis H_0, of obtaining a test statistic equal to or more extreme than what was actually observed. If $\mathbf{x}_{obs}$ is the single observation, then, p-value is defined as,

$$p = P_0(T(\mathbf{x}) \geq T(\mathbf{x}_{obs})). \tag{2.12}$$

Here, $P_0(T(\mathbf{x}) \geq T(\mathbf{x}_{obs})$ denotes the probability with which the event $\mathbf{I}\left[\!\left[T(\mathbf{x}) \geqslant T(\mathbf{x}_{obs})\right]\!\right] = 1$ occurs when $\mathbf{x}$ is sampled from the null distribution ($\mathbf{x} \sim p_0(\mathbf{x})$).
So, with respect to the hypothesis test of (2.5),

$$\begin{aligned} p &= \mathrm{P}_0\left(\mathbf{x}^T A\mathbf{x} \geqslant \mathbf{x}_{obs}^T A\mathbf{x}_{obs}\right) \\ &= \frac{\sum\limits_{\mathbf{x}} \mathbf{I}\left[\!\left[\mathbf{x}^T A\mathbf{x} \geqslant \mathbf{x}_{obs}^T A\mathbf{x}_{obs}\right]\!\right]}{2^n}. \end{aligned} \tag{2.13}$$

For small n, we see that we can compute the p-value by brute force. For large n, the exponential sum over all $\mathbf{x} \in \mathcal{X}^n$ makes it infeasible to compute it exactly, in which case we will have to resort to either approximating the p-value by sampling or by upper-bounding it.
Now, we look at both of these strategies.

2.4.1 Approximating the p-value using sampling

One way to approximate p in (2.13) is by obtaining N_s samples from the uniform prior in (2.4) and computing,

$$p_{samp} = \frac{\sum_{s=1}^{N_s} \mathbf{I} \left[\left[\mathbf{x}_{(s)}^T A \mathbf{x}_{(s)} \geqslant \mathbf{x}_{obs}^T A \mathbf{x}_{obs}\right]\right]}{N_s}. \tag{2.14}$$

2.4.2 Upper bounding the p-value

In certain scenarios, we also demonstrate that the p-value upper-bound can be quickly calculated for graphs of certain topology (planar graphs). If the upper-bound thus evaluated is found to be sufficiently low, then we can be assured that the null hypothesis of the uniform prior is rejected in favor of the Ising prior as the true p-value can only be lower than the upper-bound estimate. Now, we describe the upper-bounding procedure. Given an observation, $\mathbf{x}_{obs}$, we see that using the Chernoff-bound in (2.13), we get,

$$p \leqslant \min_{c \geqslant 0} \left(\frac{E_0 \left(\exp\left\{c\mathbf{x}^T A \mathbf{x}\right\}\right)}{\exp\left\{c\mathbf{x}_{obs}^T A \mathbf{x}_{obs}\right\}} \right). \tag{2.15}$$

Here,

$$E_0 \left(\exp\left\{c\mathbf{x}^T A \mathbf{x}\right\}\right) = \sum_{\mathbf{x}} p_0\left(\mathbf{x}\right) \exp\left\{c\mathbf{x}^T A \mathbf{x}\right\} = \frac{Z(c)}{2^n}, \tag{2.16}$$

where $Z(c)$ is the Ising partition function defined in (2.2). Thus,

$$p \leqslant \frac{1}{2^n} \min_{c \geqslant 0} \left(\frac{Z(c)}{\exp\left\{c\mathbf{x}_{obs}^T A \mathbf{x}_{obs}\right\}} \right). \tag{2.17}$$

From (2.17), we see that computing the upper bound of the p-value entails computing the Ising partition function, a challenging computational problem which is tackled below.

p-value upper bounding for planar graphs using Kasteleyn construction based exact inference

In [73], Schraudolph and Kamenetsky presented a polynomial-time algorithm for the exact computation of the partition function of binary undirected graphical models defined on planar graphs using the *Kasteleyn's dimer covering* procedure [74].

Denoting this exact computation of the partition function under the graph planarity assumption to be, $Z_{planar}(c)$, and plugging this in (2.17), we have,

$$p_{planar} := \frac{1}{2^n} \min_{c \geqslant 0} \left(\frac{Z_{planar}(c)}{\exp\left\{c\mathbf{x}_{obs}^T A \mathbf{x}_{obs}\right\}} \right). \tag{2.18}$$

It is noteworthy to mention that the procedure detailed in [73] is a two-phase procedure. The first phase covers constructing the so called *Boolean half-Kasteleyn matrix*, which is based on the geometry of the underlying graph and independent of the edge-potential(s) overlaid on its edges. The second phase entails factoring in the edge-potentials in to what is termed as the *full Kasteleyn matrix*, and computing the partition function from its determinant.

In (2.18), we see that the partition function $Z(c)$ needs to be re-computed for multiple $c \geq 0$ but we only need to execute the second phase for each c chosen as the *Boolean half-Kasteleyn matrix* is invariant to changes in c.

p-value upper bounding using Tree-Reweighted Belief Propagation (TRBP)

In case the underlying graph is not planar, we will have to use another upper bounding procedure to bound the Ising partition function in (2.17).

In [75,76], Wainright and Jordan presented a class of upper bounds on the (log) partition function of an arbitrary undirected graphical model based on solving a convex variational problem (See Appendix). Denoting this upper bound by, $Z_{trbp}(c) \geqslant Z(c)$, and plugging this in (2.17), we have,

$$p_{trbp} := \frac{1}{2^n} \min_{c \geqslant 0} \left(\frac{Z_{trbp}(c)}{\exp\left\{c\mathbf{x}_{obs}^T A \mathbf{x}_{obs}\right\}} \right). \tag{2.19}$$

2.5 Results with real world data

In this subsection, we deal with real world data where the underlying graph is the spatial inter-county geographical adjacency graph which is planar by nature. All the counties whose average per-capita income is *higher* than the median are labeled $+1$ and -1 otherwise. The data regarding the per-capita incomes was mined from [77].

Now, we look at some states with $n < 20$ counties where brute-force computation of the p-value in (2.13) is possible.

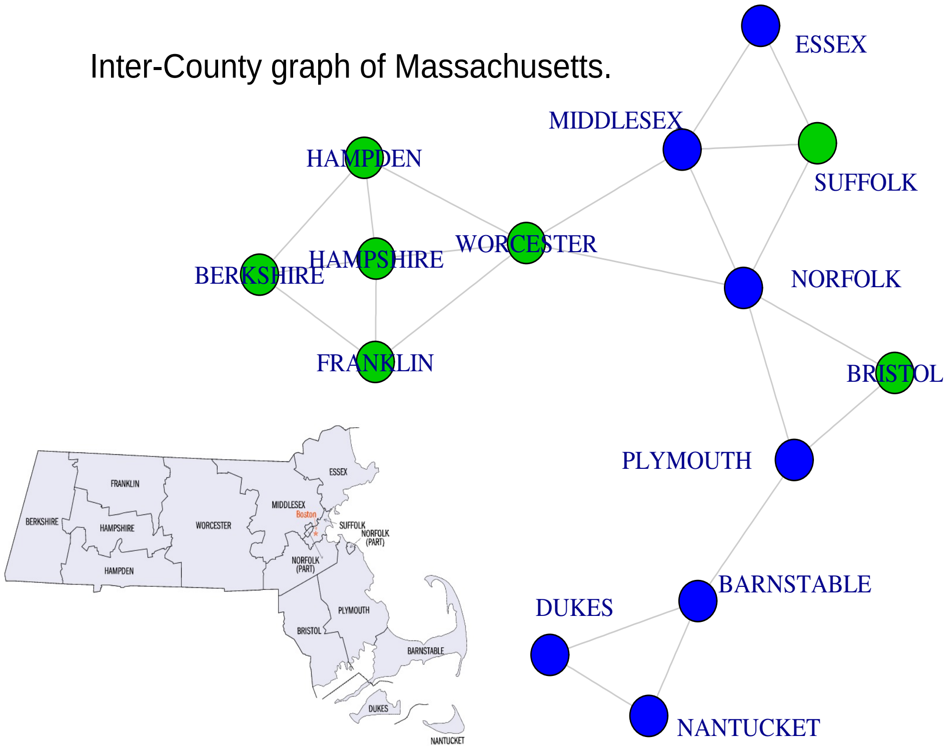


Figure 2.2: The inter-county graph for Massachusetts -(Blue: Above median, Green:Below Median)

Figure 2.3 contains the plot of brute force p-value calculations (denoted by $p_{brute-force}$) for 7 states of USA of AZ, CT, MA, ME, NH, NV and VT (For abbreviations used, the reader is referred to table 2.1) along with their observed sufficient statistics ($T(\mathbf{x}_{obs})$). The labeling used for the x-axis in the sub-plots details the state abbreviation with the number of counties in it.
As seen, of these 7 states, a low p-value of 0.07 was obtained in the case of MA (Figure 2.2).

Now, moving on to the case of $n > 20$, we evaluate p-values using the three procedures derived in (2.14),(2.18) and (2.19) respectively (Indicated by 'Samp', 'planar' and 'TRBP' in the tables in Figure 2.4). The results are as tabulated in Figure 2.4.
We categorize the states into 3 categories. The first category of states (included in TABLE-I on the left hand side of the figure (Figure 2.4)) are those states where the p-values obtained from all the 3 procedures were < 0.01, thereby providing strong evidence in support of usage of these priors as ferromagnetic Ising models. The second category of states (table-II) are those where

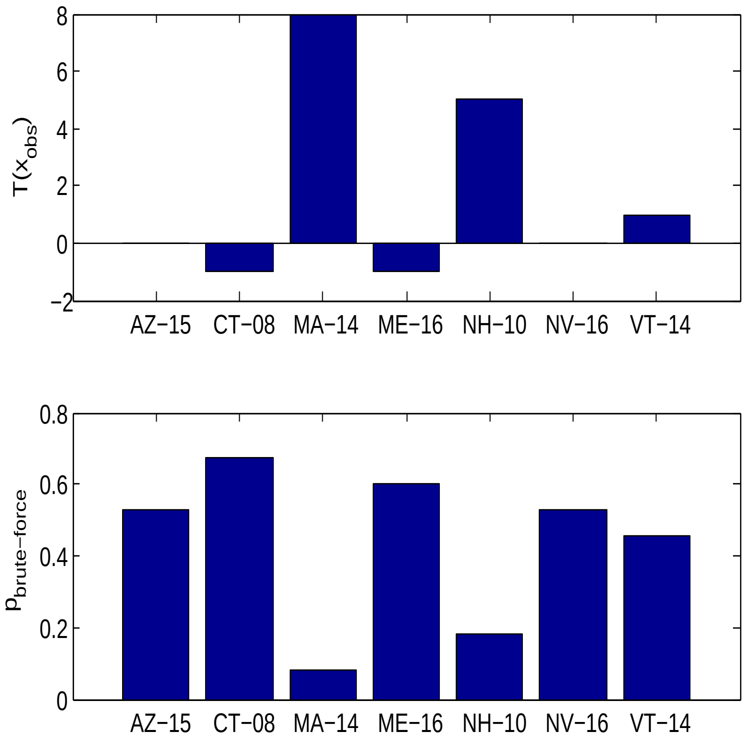


Figure 2.3: Brute force p-value calculations for 7 states of USA [AZ,CT,MA,ME,NH,NV and VT]

the p-values obtained were > 0.1 thereby providing an important reality check that one cannot simply assume the Ising prior based on a reasonable hunch. Specifically, looking at the state of WV, we see that it *visually appears* as if there seems to be some level of network effect in the sense that a lot of neighbors seem to share the same label, but when subjected to the p-value test reveals that the obtained p-value is too high to statistically justify using the Ising model.

Interestingly, we found a few states (listed in TABLE-III) where the p-value computed by using the Kamanetsky construction of (2.18) and the sampling approach of (2.14) reveals that the p-values are indeed low but the TRBP based upper bound of (2.19) is so high (> 0.1) that it might motivate the practitioner into concluding that the null model (of the uniform prior) cannot be rejected with confidence. This reveals an important lesson that one cannot really reject a dataset because it yielded a very high $p - value$ using the TRBP based upper bounding method. This is of very important significance to scenarios where the underlying graph is not planar and the Kasteleyn based exact inference method is not possible.

With reference to TABLE-I and TABLE-III, we see that there exists a large number of instances where the p-value is low enough (≤ 0.05) to reject the uniform prior null hypothesis in favor of the Ising prior, which now provides us with a strong footing to use this Ising prior in classification tasks in the upcoming chapter.

2.6 Chapter Summary

In this chapter, we have used the p-value based hypothesis testing framework to validate the homogeneous ferromagnetic Ising prior. We also showcased real world data where the null hypothesis of the data emanating from the uniform prior distribution is rejected with high statistical significance when benchmarked with the alternative hypothesis being the Ising prior. This paves the way for utilizing the ferromagnetic Ising prior in the forthcoming chapters for both network aided classification as well as network aided detection.

AL	Alabama	NE	Nebraska
AZ	Arizona	NV	Nevada
AR	Arkansas	NH	New Hampshire
CA	California	NJ	New Jersey
CO	Colorado	NM	New Mexico
CT	Connecticut	NY	New York
DE	Delaware	NC	North Carolina
FL	Florida	ND	North Dakota
GA	Georgia	OH	Ohio
ID	Idaho	OK	Oklahoma
IL	Illinois	OR	Oregon
IN	Indiana	PA	Pennsylvania
IA	Iowa	RI	Rhode Island
KS	Kansas	SC	South Carolina
KY	Kentucky	SD	South Dakota
LA	Louisiana	TN	Tennessee
ME	Maine	TX	Texas
MD	Maryland	UT	Utah
MA	Massachusetts	VT	Vermont
MI	Michigan	VA	Virginia
MN	Minnesota	WA	Washington
MS	Mississippi	WV	West Virginia
MO	Missouri	WI	Wisconsin
MT	Montana	WY	Wyoming

Table 2.1: Abbreviation of state acronyms

TABLE-I

	Planar	Samp	TRBP
GA	0.0068	0.0001	0.0001
OH	0.0020	0.0003	0.0029
PA	0.0063	0.0001	0.0664
TX	0.0043	0.0003	0.0031
OH	0.0001	0.0001	0.0029
PA	0.0004	0.0004	0.0664
KY	0.0122	0.0001	0.0037

TABLE-II

	Planar	Samp	TRBP
IA	0.4314	0.4345	1.0000
KA	0.1747	0.1615	0.8641
WV	0.1910	0.2039	0.9015

TABLE-III

	Planar	Samp	TRBP
CA	0.0041	0.0181	0.4363
IL	0.0020	0.0006	0.1700
IN	0.0020	0.0013	0.2132
MO	0.0124	0.0004	0.1564
NY	0.0904	0.0783	0.7123

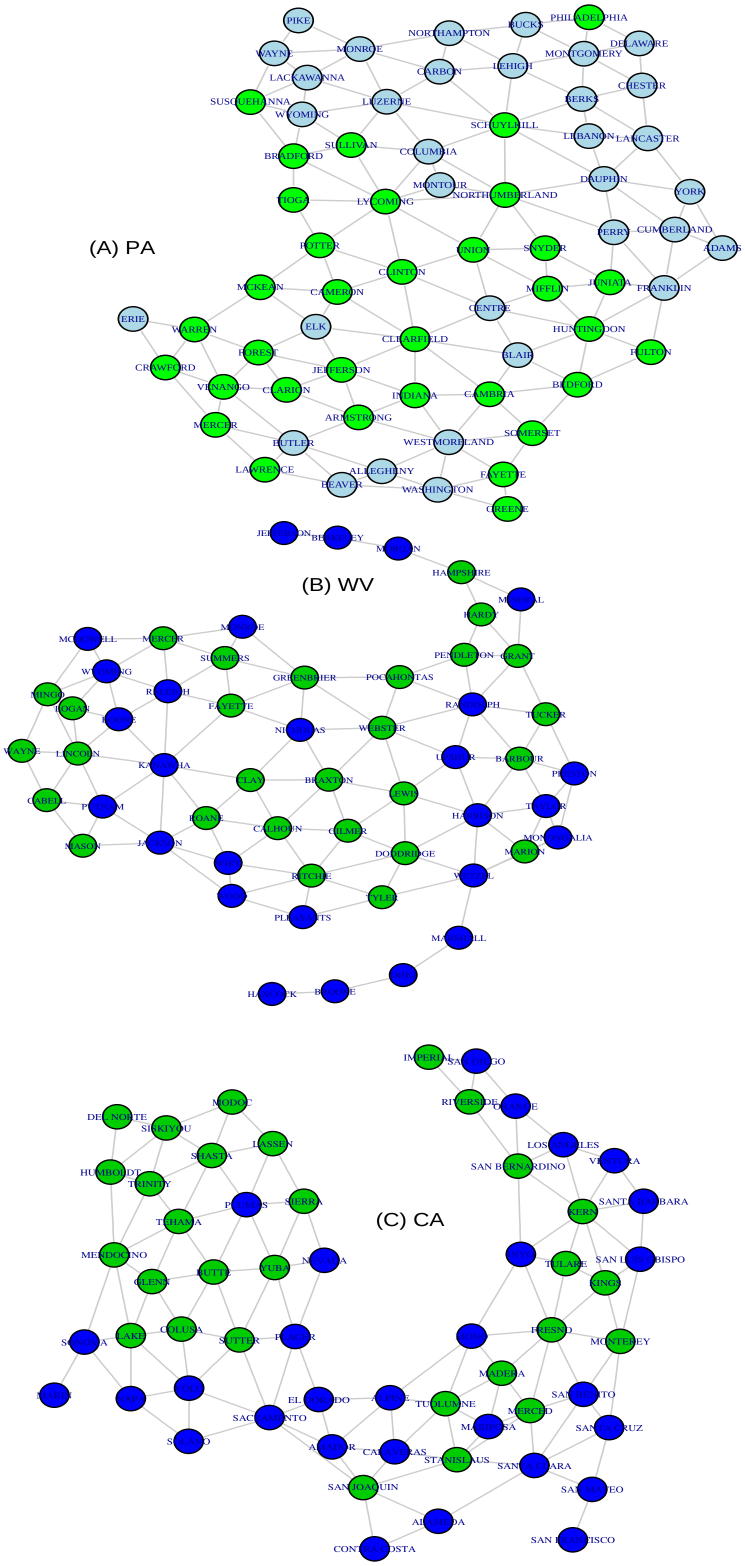


Figure 2.4: p-value comparisons for real world data sets involving counties of the states of USA

Chapter 3

Network Aided Classification

3.1 Introduction

In Chapter-2, we proposed a p-value based hypothesis testing framework to validate the homogeneous ferromagnetic Ising prior which is,

$$p(\mathbf{x}) = \frac{\exp\left\{\theta \mathbf{x}^T A \mathbf{x}\right\}}{Z(\theta)}. \tag{3.1}$$

We also showed real world examples where the null hypothesis of the data emanating from the uniform prior distribution is rejected with high statistical significance when benchmarked with the alternative hypothesis being the Ising prior. Now, the next logical step is to use this prior defined in (3.1) above and demonstrate its utility in terms of the classification accuracy improvement obtained on account of using it instead of the uniform prior. To this end, we dedicate this chapter for showcasing real-world applications that will employ the Network Aided Classification (NAC) framework.

To assist the reader, we begin this chapter by re-establishing the joint Bayesian model of $\mathbf{x} \in \mathcal{X}^n$, which is the label vector to be estimated and $\mathbf{Y} = [\mathbf{y}_1^T, ..., \mathbf{y}_n^T] \in \mathbb{R}^{n \times d}$, the $n \times d$ matrix[1] of *observations*, *features* or *evidence*,

$$p(\mathbf{x}, \mathbf{Y}) = p(\mathbf{x}) p(\mathbf{Y}|\mathbf{x}). \tag{3.2}$$

[1] (In the previous chapters, we have considered scalar features, $y_i; i = 1, ..., n$, and hence $\mathbf{y} = [y_1, ..., y_n]^T$ represented the $n \times 1$ vector of scalar features. Here, we allow the features to be $d \times 1$ vectors and hence have used $\mathbf{Y}$ (in capitals) to denote the resultant $n \times d$ feature matrix).

Driven by the assumption that the labels are independent and identically distributed (i.i.d), the modeling and optimization of the likelihood $p(\mathbf{Y}|\mathbf{x})$ in (3.2) above is focused strongly in the machine learning community (See [78,79]).

3.1.1 Literature Survey of classification techniques for correlated data

Within machine learning, Statistical Relational Learning (SRL) tools have been developed [80] to addresses the problem of performing probabilistic inference on correlated data. Collective classification (CC) has emerged as an important SRL sub-category, where related data instances are jointly classified as opposed to sample-wise classification performed with the i.i.d data assumption.

CC algorithm examples include Relaxation labeling [81], Jensen's Gibbs sampler based Collective classifier [82], iterative classification techniques in relational data [83], Taskar's Discriminative probabilistic models based collective classifier [84], Getoor's link based classifier [85] and the weighted-vote relational neighbor algorithm (wvRN) [86]. The NAC techniques explored in this chapter can be considered to be in this general collective classification framework.

The NAC framework itself was inspired by two important instances in literature where the underlying graph is used to define a graphical model [23] and used as a statistical prior for the labels. The first is in the area of image segmentation in computer vision [87] where the nodes are the pixels and the graph is the 2-D image pixel grid. Secondly, there has also been the use of web-hyperlink based graph, to classify documents and webpages alike [86,88].

We would like to emphasize that our goal here is not to claim that the NAC framework being proposed betters all of the above mentioned instances of collective classification in terms of classification accuracy or speed of execution. Instead, the onus here is to showcase a wide array of real world examples where NAC is used in conjunction with a standard off-the-shelf non-specialized discriminative or generative classifier such as SVM or Bag-of-words text classifier thereby motivating the operational value of employing the MRF or Ising model based prior. These real world instances also help contextualize the theoretical explorations made in Chapters 4 and 5, involving latent and majority sentiment classification.

Towards this end, we motivate the idea that network aided enhanced classification opportunities are indeed ubiquitous and span several domains such as political science, sociology, finance and health policy.

Now, we move on to formulate the NAC model as a MAP-MRF problem in the coming section.

3.2 The MAP-MRF model for Network aided classification

3.2.1 The MAP and MPM classifier definitions

The posterior distribution of the labels given the evidence, **Y** is,

$$p(\mathbf{x}|\mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{Y})}. \tag{3.3}$$

The labels **x** are now estimated as the Maximum A posteriori Probability (MAP) [39] configuration (mode of the posterior distribution),

$$\hat{\mathbf{x}}_{map} = \arg \max_{\mathbf{x}\in\{-1,1\}^n} \{p(\mathbf{x}|\mathbf{Y})\} \tag{3.4}$$

which minimizes the hit-loss cost function,

$$C_{map}(\mathbf{x}, \hat{\mathbf{x}}) = 1 - \mathbf{I}[[\mathbf{x} = \hat{\mathbf{x}}]]. \tag{3.5}$$

There exists some literature [37] that evaluating the Maximum Posterior Marginal (MPM) [37] configuration, defined as,

$$\hat{x}_{i,mpm} = \arg \max_{x_i\in\{-1,1\}} \{p(x_i|\mathbf{Y})\} \tag{3.6}$$

and which minimizes the MPM cost function which is

$$C_{mpm}(\mathbf{x}, \hat{\mathbf{x}}) = \sum_{i=1}^{n} \{1 - \mathbf{I}[[x_i = \hat{x}_i]]\}. \tag{3.7}$$

provides better results compared to the MAP classifier. In this chapter, we use both and highlight when there is an appreciable difference seen in the classification error rate.

3.2.2 Conditional independence in the likelihood

As in [39, 89, 90], conditional independence amongst measurements $\mathbf{y}_i$ given the true label, x_i is assumed. That is,

$$p(\mathbf{Y}|\mathbf{x}) = \prod_{i=1}^{n} p(\mathbf{y}_i|x_i). \tag{3.8}$$

Many popular classification algorithms including Support Vector Machines (SVM) ([91]) and Back-propagation for Neural Networks [92] implicitly make the assumption that the samples in the data corpus are a collection of independent and identically distributed (i.i.d.) samples. This assumption leads to solving the problem of label assignment on a sample-by-sample basis.

3.2.3 MAP classification under the i.i.d assumption

Given that the labels themselves are independent of each other, we have the vector MAP assignment problem in (3.4) reducing to,

$$\begin{aligned} \hat{x}_i &= \arg\max_{k \in \mathcal{X}} p(x_i = k|\mathbf{y}_i) \\ &= \arg\max_{k \in \mathcal{X}} \{p(\mathbf{y}_i|x_i = k)p(x_i = k)\} \, ; i = 1, ..., n. \end{aligned} \tag{3.9}$$

In many instances in literature [78, 79], the focus is to try and model the likelihood as a discriminative joint function of the label and feature as $p(\mathbf{y}_i|x_i = k) = g(k, \mathbf{y}_i)$, to finally solve the label assignment problem as the solution of the following maximization problem [78, 79],

$$\hat{x}_i = \arg\max_{k \in \mathcal{X}} \{g(k, \mathbf{y}_i)\} \, ; i = 1, ..., n. \tag{3.10}$$

Depending on the nature of data being classified different joint label-feature discriminative functions ($g(k, \mathbf{y}_i)$) are proposed and convex relaxation tricks and other heuristics are used to solve the final optimization problem.

3.2.4 MAP classification with the Ising prior

Let the underlying graph associated with the dataset $\{\mathbf{x}, \mathbf{Y}\}$ be $G(V, E)$. Here, V is the vertex set ($|V| = n$) and E denotes the edge set. We now bring in Ising prior for the labels $x_i \in \{-1, +1\}$

defined in (3.1) in the Introduction section, which is,

$$p(\mathbf{x}) = \frac{\exp\left(\sum\limits_{(i,j)\in E} \theta x_i x_j\right)}{Z(\theta)}. \tag{3.11}$$

Here, $Z(\theta)$ is the normalizing partition function, which is, $Z(\theta) = \sum\limits_{\mathbf{x}} \exp\left(\sum\limits_{(i,j)\in E} \theta x_i x_j\right)$. Now, plugging in this Ising prior of (3.11) in to (3.4), we have, the MAP classification problem with the Ising prior to be,

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}\in\mathcal{X}^n} \left\{ \prod_{i=1}^{n} p(\mathbf{y}_i|x_i).\exp\left(\sum_{(i,j)\in E} \theta(x_i x_j)\right)\right\} \tag{3.12}$$

Now, defining the node potential functions to be

$$\phi_i(x_i) = p(\mathbf{y}_i|x_i); \forall i \in V, \tag{3.13}$$

and edge-potential functions to be

$$\psi_{ij}(x_i, x_j) = exp(\theta x_i x_j); \forall (i,j) \in E, \tag{3.14}$$

we see that (3.12) reduces into a *max-inference* problem for an MRF whose potential functions are as defined in (3.13) and (3.14) respectively. (That is, $p(\mathbf{x}) = \frac{1}{Z} \prod\limits_{i\in V} \phi_i\left(x_i\right) \prod\limits_{(i,j)\in E} \psi_{ij}(x_i, x_j)$). Now, this is a well researched problem in the machine-learning community and has been shown to be NP-Hard in [93]. Hence, a slew of heuristics have been proposed for perform approximate inference such as Loopy Belief propagation (LBP), Tree-Reweighted Belief Propagation (TRBP), Mean Field (MF) max-inference algorithm and Iterative Conditional Modes (ICM) algorithm (See [20, 79] for the surveys). In our experiments, we employed the one which yielded the best accuracy and which is duly noted in the experiment details that follow the real-world examples to be demonstrated in the upcoming sections.

3.3 ML estimation of θ

As seen, (3.12) requires the knowledge of the Ising edge-potential θ, which is also termed as the Gibb's inverse temperature parameter and the global hyper-parameter [89, 90] in literature. As seen, θ tunes the balance of influence between the Ising/MRF prior and that of the feature driven likelihood. This section is dedicated towards estimating the same from the training data based on the Maximum Likelihood Estimation (MLE) ideas presented in [79].

In this chapter, we have two scenarios. The first is the fully observed scenario, where we learn θ from a single complete observation $\mathbf{x}_{obs} \in \{-1, +1\}^n$ pertaining to a year or a certain state and then use the learned $\hat{\theta}_{ml}$ to perform NAC on data pertaining to a different year (or a different state).

In the second scenario, we have a single snapshot network based dataset whose vertex set is split into training and testing subsets. The vertices pertaining to the training subset constitutes the observed labels, $\mathbf{x}_o \in \{-1, +1\}^{n_{train}}$ and the vertices pertaining to the testing subset constitute the hidden labels, $\mathbf{x}_h \in \{-1, +1\}^{n_{test}}$.

In the subsection, we describe ML estimation of the edge-potential θ covering both the scenarios.

3.3.1 Maximum-Likelihood estimation of θ in the completely observed scenario

The likelihood of obtaining an observation $\mathbf{x}_{obs} \in \{-1, +1\}^n$ being sampled from a homogeneous Ising prior parameterized by the global edge-potential θ is,

$$p(\mathbf{x}_{obs}; \theta) = \frac{\exp\left\{\theta \mathbf{x}_{obs}^T A \mathbf{x}_{obs}\right\}}{Z(\theta)}, \tag{3.15}$$

and that the log-likelihood function is,

$$l(\theta) = \theta \mathbf{x}_{obs}^T A \mathbf{x}_{obs} - \log\left(Z(\theta)\right) \tag{3.16}$$

Given this observation, the maximum likelihood estimated of θ is,

$$\hat{\theta}_{ml} = \arg\max_{\theta} \left[\theta \mathbf{x}_{obs}^T A \mathbf{x}_{obs} - \log\left(Z(\theta)\right)\right]. \tag{3.17}$$

Given that Ising models are in the exponential family, we know that likelihood function is convex in θ [79], so it has a unique global maxima which we can be estimated using gradient-based optimizers.

Taking the derivative with respect to θ and setting it to zero, we arrive at the so called *moment matching condition*, which is,

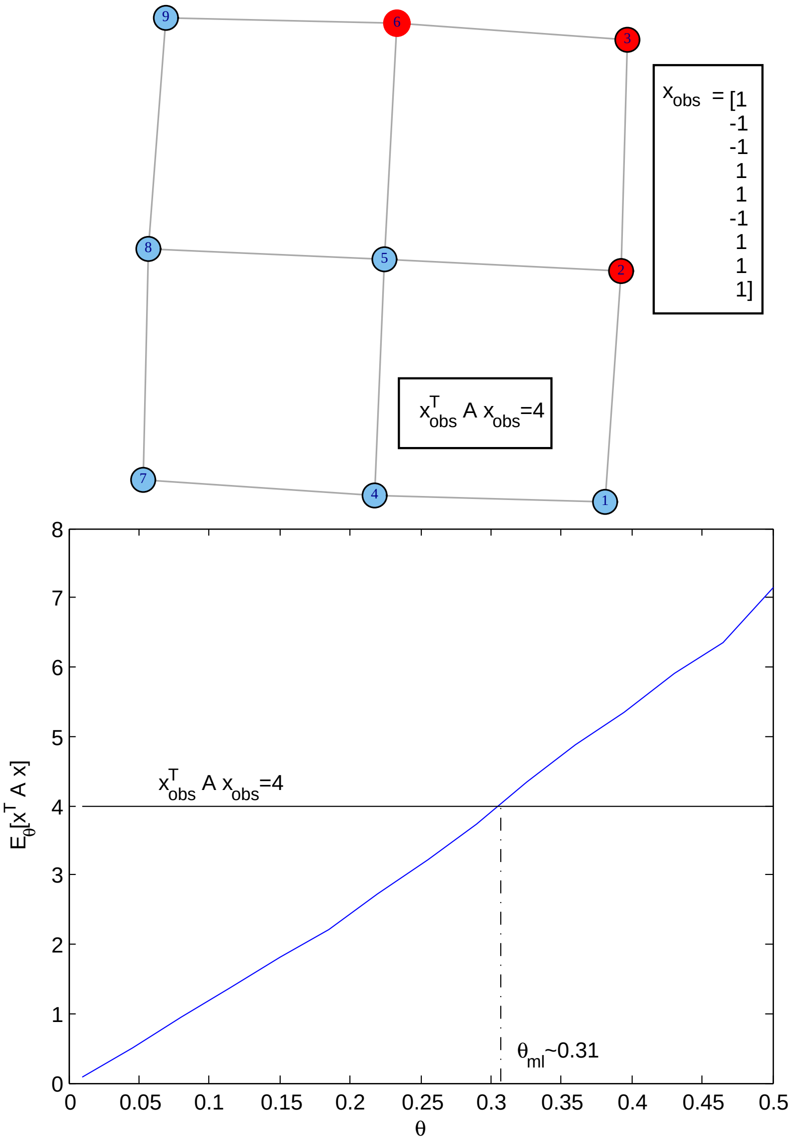


Figure 3.1: ML estimation of θ

$$\mathbf{x}_{obs}^{T} A \mathbf{x}_{obs} = \mathbf{E}_{\hat{\theta}_{ml}} \left[\mathbf{x}^{T} A \mathbf{x} \right], \tag{3.18}$$

where, the expectation $\mathbf{E}_{\hat{\theta}_{ml}}[]$ is defined as,

$$\mathbf{E}_{\hat{\theta}_{ml}} \left[\mathbf{x}^{T} A \mathbf{x} \right] = \sum_{\mathbf{x}} \left[p(\mathbf{x}; \hat{\theta}_{ml}) \left\{ \mathbf{x}^{T} A \mathbf{x} \right\} \right]. \tag{3.19}$$

Thus $\hat{\theta}_{ml}$ can be estimated by computing the expectation over a certain range of θ and finding that θ at which the expectation will be meeting the actual observed statistic which is $\mathbf{x}_{obs}^T A\mathbf{x}_{obs}$. Note that when n is small, the expectation can be computed by brute-force. However, when n is large, we need to resort to computing the expectation via approximation techniques such as Gibbs sampling based methods.

Figure 3.1 shows the ML estimation of $\hat{\theta}_{ml}$ for a 9-node grid-graph with the observation $\mathbf{x}_{obs} = [+1, -1, -1, +1, +1, -1, +1, +1, +1]^T$.

3.3.2 ML Estimation of θ in a partially observed scenario

Let $\mathbf{x}_o \in \{-1, +1\}^{n_{train}}$ denote the observed labels and $\mathbf{x}_h \in \{-1, +1\}^{n_{test}}$ denote the hidden labels. Now, for a particular realization of the hidden variables $\mathbf{x}_h$, let us define the concatenated vector of the observed values and the hidden values as, $\mathbf{x}_{oh} = [\mathbf{x}_o; \mathbf{x}_h] \in \{-1, +1\}^n$.

We see that the likelihood of the observation $\mathbf{x}_o$ is obtained by marginalizing over the hidden variables, leading to,

$$p(\mathbf{x}_o; \theta) = \frac{\sum\limits_{\mathbf{x}_h} [\tilde{p}(\mathbf{x}_{oh}; \theta)]}{Z(\theta)}, \tag{3.20}$$

where $\tilde{p}(\mathbf{x}_{oh}; \theta) = \exp\left\{\theta \mathbf{x}_{oh}^T A\mathbf{x}_{oh}\right\}$ is the un-normalized distribution of $\mathbf{x}_{oh}$. This implies that the log-likelihood function is,

$$l(\theta) = \log\left(\sum_{\mathbf{x}_h} [\tilde{p}(\mathbf{x}_{oh}; \theta)]\right) - \log\left(Z(\theta)\right). \tag{3.21}$$

As in the fully observed case, the ML estimate of θ is,

$$\hat{\theta}_{ml} = \arg\max_{\theta} [l(\theta)] = \arg\max_{\theta} \left[\log\left(\sum_{\mathbf{x}_h} [\tilde{p}(\mathbf{x}_{oh}; \theta)]\right) - \log\left(Z(\theta)\right)\right]. \tag{3.22}$$

Now, taking the derivative of the log-likelihood and setting it to zero,and using $\frac{\partial}{\partial\theta} \log\left(\sum\limits_{\mathbf{x}_h} [\tilde{p}(\mathbf{x}_{oh}; \theta)]\right) = \mathbf{E}_{\hat{\theta}_{ml}}\left[\mathbf{x}_{oh}^T A\mathbf{x}_{oh}\right]$, we get,

$$\hat{\theta}_{ml} : \underbrace{\mathbf{E}_{\hat{\theta}_{ml}}\left[\mathbf{x}_{oh}^T A\mathbf{x}_{oh}\right]}_{\text{Clamped - expectation}} = \underbrace{\mathbf{E}_{\hat{\theta}_{ml}}\left[\mathbf{x}^T A\mathbf{x}\right]}_{\text{Unclamped - expectation}} . \tag{3.23}$$

In (3.23) above, the expectation on the LHS (the *clamped expectation*) is computed by *clamping* the visible nodes to their observed values, and the *unclamped expectation* on the RHS is computed by

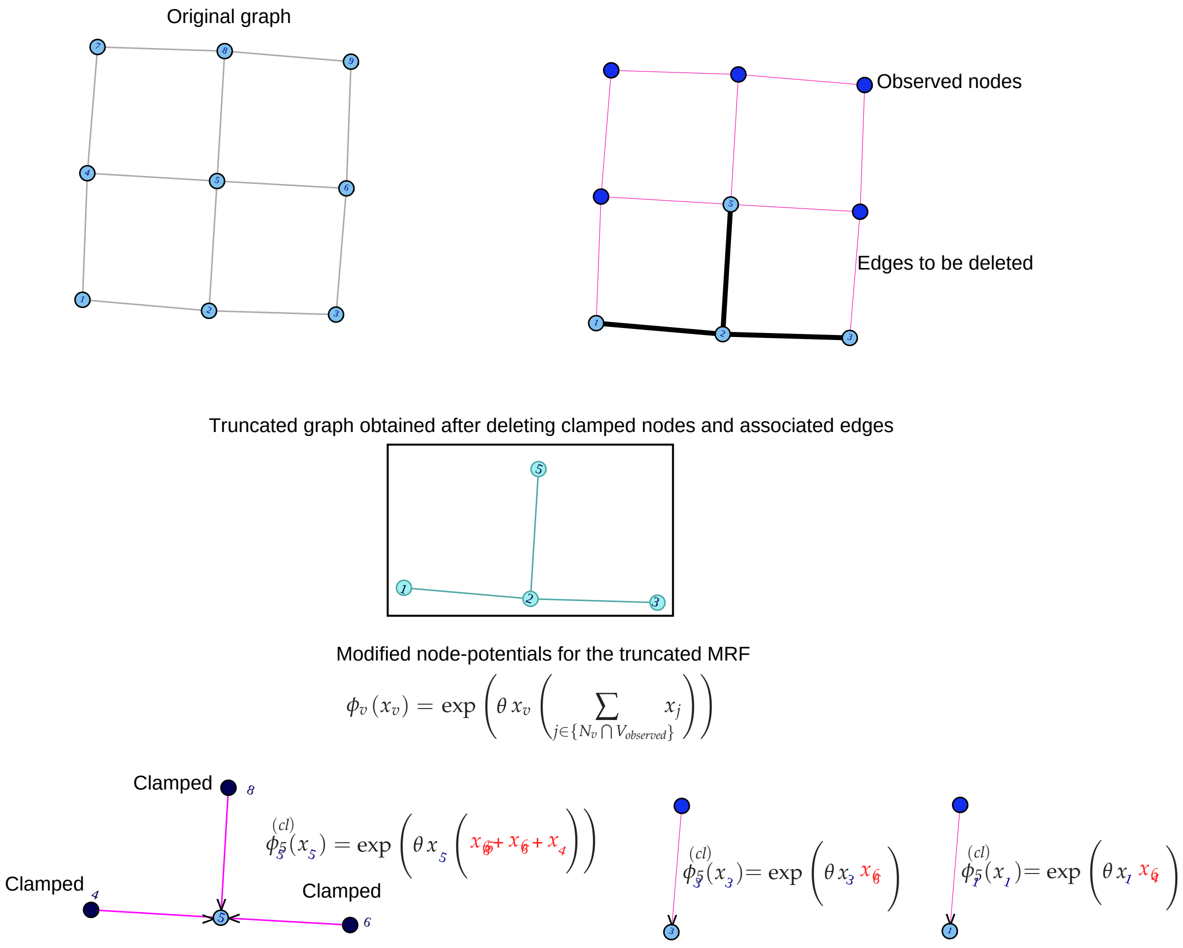

Original graph
Observed nodes
Edges to be deleted
Truncated graph obtained after deleting clamped nodes and associated edges
Modified node-potentials for the truncated MRF
$\phi_v(x_v) = \exp\left(\theta x_v \left(\sum_{j \in \{N_v \cap V_{observed}\}} x_j\right)\right)$
Clamped
Clamped
Clamped
$\phi^{(cl)}_5(x_5) = \exp\left(\theta x_5 \left(x_8 + x_6 + x_4\right)\right)$
$\phi^{(cl)}_3(x_3) = \exp\left(\theta x_3\, x_6\right)$
$\phi^{(cl)}_1(x_1) = \exp\left(\theta x_1\, x_4\right)$

Figure 3.2: Demonstration of the construction of the conditional (truncated) MRF

letting the visible nodes be free.

The clamped expectation can be computed by sampling the hidden variables defined on a conditional (truncated) MRF which can be constructed as follows.

3.3.3 Constructing the conditional truncated MRF

MRFs are closed under conditioning. That is, if we condition on the values of the observed variables ($\mathbf{x}_o$), the resulting conditional distribution, $p(\mathbf{x}_h|\mathbf{x}_o)$ will still remain an MRF [21,79].

We convert an unconditional MRF into a conditional (truncated) MRF by using the following three step truncation procedure.

1. **Removal of observed nodes**: We remove the observed nodes (and corresponding node potentials) from the unconditional model.

2. **Removal of edges between observed nodes**: We remove the edges (and corresponding edge potentials) that exist between the observed node variables from the unconditional model.

3. **Removal of edges between observed nodes and hidden nodes and reweighing the node-potentials of the hidden nodes**: For every edge between an observed node and a hidden node, we multiply the node potential of the regular node by the relevant clamped edge potential, and then remove the edge (and corresponding edge potential) from the model. Denoting V_o to be the subset of nodes associated with the observed variables, V_h denote the vertex set pertaining to the hidden unobserved nodes and N_v to be the neighboring nodes of a given variable v, we have, the conditioned (clamped) node potentials of the hidden node variables ($\phi_v^{(cl)}$) to be,

   $$\phi_v^{(cl)}(x_v) = \prod_{j \in \{N_v \bigcap V_o\}} \psi_{vj}(x_v, x_j); \forall v \in V_h. \tag{3.24}$$

   Now let $E_{truncated}$ denote the edge subset containing the edges between the hidden variable nodes. Thus, we see that after this step, we are left with a truncated graph $G(V_h, E_{truncated})$.

At the end of this procedure, we make three observations. Firstly, we will have no terms left depending on the observed nodes. Secondly, the edge-potentials between the hidden nodes remain unchanged. Thirdly, $G(V_h, E_{truncated})$ is bound to be sparser compared to the original graph, $G(V, E)$ and may have a *simpler* topology (example, a tree), which might allow for exact inference or sampling, an idea that might be exploited for computing the clamped expectation in the previous subsection. In Figure 3.2, we have described this truncation procedure for the 9 node grid graph. In this particular case, we see that the conditional MRF defined over (x_1, x_2, x_3, x_5) is indeed a tree.

To conclude, we now present the final form of the truncated (or conditional) MRF obtained after the three step procedure described above.

$$
\begin{aligned}
p\left(\mathbf{x}_h|\mathbf{x}_o\right) = \frac{p\left(\mathbf{x}_h, \mathbf{x}_o\right)}{p\left(\mathbf{x}_o\right)} &= \frac{p\left(\mathbf{x}_h, \mathbf{x}_o\right)}{\sum\limits_{\mathbf{x}_h}\left[p\left(\mathbf{x}_h, \mathbf{x}_o\right)\right]} \\
&= \frac{\prod\limits_{v \in V_h} \phi_v^{(cl)}\left(x_v\right) \prod\limits_{(i,j)\in E_{truncated}} \left\{\exp(\theta x_i x_j)\right\}}{\sum\limits_{\mathbf{x}} \prod\limits_{v \in V_h} \phi_v^{(cl)}\left(x_v\right) \prod\limits_{(i,j)\in E_{truncated}} \left\{\exp(\theta x_i x_j)\right\}}.
\end{aligned} \tag{3.25}
$$

Note that while the above procedure was for the homogeneous Ising prior, defined by,

$$
\begin{aligned}
p(\mathbf{x}) &= \frac{1}{Z} \prod_{(i,j)\in E} \psi_{ij}(x_i, x_j) \\
&= \frac{1}{Z(\theta)} \prod_{(i,j)\in E} \exp(\theta x_i x_j),
\end{aligned} \tag{3.26}
$$

it also holds for a general MRF with varying node and edge potentials defined by,

$$
p(\mathbf{x}) = \frac{1}{Z} \prod_{v \in V_h} \phi_v\left(x_v\right) \prod_{(i,j)\in E} \psi_{ij}(x_i, x_j), \tag{3.27}
$$

with

$$
p\left(\mathbf{x}_h|\mathbf{x}_o\right) = \frac{\prod\limits_{v \in V_h} \phi_v^{(cl)}\left(x_v\right) \prod\limits_{(i,j)\in E_{truncated}} \psi_{ij}(x_i, x_j)}{\sum\limits_{\mathbf{x_h}} \prod\limits_{v \in V_h} \phi_v^{(cl)}\left(x_v\right) \prod\limits_{(i,j)\in E_{truncated}} \psi_{ij}(x_i, x_j)}, \tag{3.28}
$$

where,

$$
\phi_v^{(cl)}\left(x_v\right) = \phi_v\left(x_v\right) \prod_{j \in \{N_v \bigcap V_o\}} \psi_{vj}(x_v, x_j); \forall v \in V_h. \tag{3.29}
$$

This result is used in the following section when incorporating the feature vector information into the NAC framework via the node potentials.

Now, we describe the NAC methodology for both the discriminative and generative approaches.

3.4 Methodology

Given the inherent graph-structure in the datasets to follow, the procedure we used to showcase the utility of NAC deviates from the standard methodology used in classical machine learning literature. Hence, before showcasing NAC with the real world datasets, in this section, we explain the 4-fold cross validation methodology used for the single snapshot network based datasets in detail.(Refer to Pseudo-code 1).

To begin with, we have the graph $G(V, E)$, with n nodes and M edges. We employ random edge-sampling to split Edge-set E into 4 random equal-sized subsets: $E = \bigcup_{k=1}^{4} E_k : |E_k| \approx M/4$.

3.4.1 Training Phase 1: Extracting the training and testing datasets based on randomly sampled edge subsets

For the k^{th} fold, we obtain the training data and testing data as follows. Firstly, for $k \in \{1, 2, 3, 4\}$, we set $E_{test} = E_k$. Now, we obtain $E_{train} = E \setminus E_{test}$[2], using which we extract testing and training vertex sets as, $V_{test} = \mathcal{V}(E_k)$, $V_{train} = V \setminus V_{test}$. The function $\mathcal{V}(E_k)$ here returns the subset of the vertices that are connected by the edges in the edge-subset E_k and is defined as,

$$\mathcal{V}(E_k) = \{u \in V : \exists v \in V, (u, v) \in E_k || (v, u) \in E_k\}. \tag{3.30}$$

Finally, the training and testing datasets consisting of the feature vectors and labels are constructed by,

$$D_{test} = \{\mathbf{Y}_{test}, \mathbf{x}_{test}\},\ D_{train} = \{\mathbf{Y}_{train}, \mathbf{x}_{train}\}, \tag{3.31}$$

where,

$$\begin{aligned} \mathbf{Y}_{train} &= \{\mathbf{y}_v : v \in V_{train}\}, \mathbf{x}_{train} = \{x_v : v \in V_{train}\} \\ \mathbf{Y}_{test} &= \{\mathbf{y}_v : v \in V_{test}\}, \mathbf{x}_{test} = \{x_v : v \in V_{test}\}. \end{aligned} \tag{3.32}$$

3.4.2 Training Phase 2: Maximum-Likelihood estimation of θ

We use the ML estimation technique outlined in subsection 3.3.2 for the partially observed case thus. The training labels $\mathbf{x}_{train}$ constitute the observed variables while the variables pertaining to the testing part are the hidden variables. Using (3.22) and (3.23), we compute $\hat{\theta}_{ml}$.

[2] Here $B \setminus A = \{x \in B \,|\, x \notin A\}$ denotes the standard set-difference operation

3.4.3 Training Phase 3: Training the machine learning classifier.

Generative model based approach

This scenario entails using a generative model for the likelihood parameterized as $p(\mathbf{y}_i|x_i = k; \mathbf{\Omega}_k)$, where $\mathbf{\Omega}_k$ is the class conditional parameter set. In case the feature vectors are scalars and we use a 2-class conditional Gaussian model, we have,

$$p(y_i|x_i = k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left[-\frac{(y_i - \mu_k)^2}{2\pi\sigma_k^2}\right]; k = -1, +1, \tag{3.33}$$

with $\mathbf{\Omega}_k = \{\mu_k, \sigma_k^2\}$.

In the case of text classification, $p(\mathbf{y}_i|x_i = k; \mathbf{\Omega}_k)$ is parameterized using the standard bag-of-words model [31], which represents the document as a word count feature vector pertaining to a dictionary $\mathbf{W} = [w_1, ..., w_d]^T$. That is, $\mathbf{y}_i = [c_1^{(i)}, ..., c_d^{(i)}]$, where $c_w^{(i)}$ is the count of the w^{th} word in document i. The model parameter would be the word-probability vector where $\rho_w^{(k)}$ is the probability of the word w appearing in the k^{th} class/label. Given that a document $\mathbf{y}_i$ has a label k, its word counts are modeled using the standard multinomial distribution model, that is,

$$p(\mathbf{y}_i|x_i = k) = \prod_{w=1}^{d} \left(\rho_w^{(k)}\right)^{c_w^{(i)}}. \tag{3.34}$$

As seen, the generative model parameter set in this case is $\mathbf{\Omega}_k = \{\rho_w^{(k)}; w = 1, ..., d\} \in \mathbb{R}^d$.

The parameters pertaining to (3.33) and (3.34) above can be estimated using the standard maximum likelihood techniques (as shown in standard machine learning textbooks [78,79]) using the training dataset $D_{train} = \{\mathbf{Y}_{train}, \mathbf{x}_{train}\}$.

Discriminative model based approach

To begin with, we split the training data D_{train} in to 2 subsets, $D_{train-1}$ and $D_{train-2}$, typically in the 4:1 ratio.

We then use the first part of the training dataset,$D_{train-1} = \{\mathbf{Y}_{train-1}, \mathbf{x}_{train-1}\} \in \mathbb{R}^{n_{train-1} \times (d+1)}$, to learn the discriminative function $\hbar(.)$ of the chosen discriminative classifier such as the Support Vector Machine (SVM) [91].

Then, we use this trained discriminative classifier to predict the labels in the second part of the training dataset. That is,

$$\hat{x}_{train-2,i} = \hbar(\mathbf{y}_{train-2,i}); i = 1, .., n_{train-2}. \tag{3.35}$$

Now, the predicted labels $\hat{x}_{train-2,i}$ and the true labels $x_{train-2,i}$ are used to estimate the entries of the confusion matrix, $p(\hat{x}_i|x_i = k)$, by

$$p\left(\hat{x}_i = k|x_i = l\right) = \frac{\sum\limits_{i=1}^{n_{train-2}} \{I[[\hat{x}_i = k]]I[[x_i = L]]\}}{n_{train-2}}, \tag{3.36}$$

which are then used to replace the likelihood $p(\mathbf{y}_i|x_i = k)$ used in the generative model case by $p(\hat{x}_i|x_i = k)$.

3.4.4 Testing Phase 1: Potentiating the truncated conditional MRF

Using the procedure explained in subsection 3.3.3, we firstly construct $G(V_{test}, E_{test})$. The node potentials of this truncated conditional MRF are then computed using the testing data for the generative case ($\phi_v^{(gen)}\left(x_v\right)$) and the discriminative case ($\phi_v^{(dis)}\left(x_v\right)$) by,

$$\begin{aligned}
\phi_v^{(gen)}\left(x_v\right) &= p\left(y_v|x_v;\hat{\Omega}\right) \times \exp\left(\hat{\theta}_{ml}x_v\left(\sum_{j\in\{N_v\bigcap V_{train}\}} x_j\right)\right);\forall v \in V_{test} \\
\phi_v^{(dis)}\left(x_v\right) &= p\left(\hat{x}_v|x_v\right) \times \exp\left(\hat{\theta}_{ml}x_v\left(\sum_{j\in\{N_v\bigcap V_{train}\}} x_j\right)\right);\forall v \in V_{test}.
\end{aligned} \tag{3.37}$$

The edge-potentials of the truncated conditional MRF are set to,

$$\psi_{ij}(x_i, x_j) = \exp\left(\hat{\theta}_{ml}x_ix_j\right), \forall (i, j) \in E_{test} \tag{3.38}$$

3.4.5 Testing Phase 2: Solving the MAP-MRF inference problem

Now, finally we estimate the test data labels by solving the MAP-MRF inference problem defined on the truncated conditional MRF defined by the graph $G(V_{test}, E_{test})$ and node and edge-potentials defined by (3.37) and (3.38) respectively. That is,

$$\begin{aligned}\hat{\mathbf{x}}_{test,nac-gen} &= \max_{\mathbf{x}\in\{-1,+1\}^{n_{test}}} \left[\prod_{v\in V_{test}} \phi_v^{(gen)}(x_v) \prod_{(i,j)\in E_{test}} \psi_{ij}(x_i,x_j)\right] \\ \hat{\mathbf{x}}_{test,nac-dis} &= \max_{\mathbf{x}\in\{-1,+1\}^{n_{test}}} \left[\prod_{v\in V_{test}} \phi_v^{(dis)}(x_v) \prod_{(i,j)\in E_{test}} \exp\left(\hat{\theta}_{ml} x_i x_j\right)\right]\end{aligned} \tag{3.39}$$

For this, we employ approximate inference algorithms such as LBP, TRBP, ICM or Mean-field inference which are as detailed in [20]. The estimated and the true test data labels are used to compute the classification accuracy for the k^{th} cross validation fold as,

$$acc(k) = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \{\mathbf{I}[[\hat{x}_{test,i} = \hat{x}_{test,i}]]\} \tag{3.40}$$

Now, this procedure is repeated for each of the 4-folds and the final average classification accuracy is computed by,

$$Accuracy = \frac{1}{4}\sum_{k=1}^{4} acc(k). \tag{3.41}$$

Figure 3.3 details the explained NAC (generative model based) procedure for the 9-node grid graph.

Now, we move on to employ this procedure for real world datasets spanning disparate domains.

3.5 Real world examples

In this section, we will showcase examples involving real world data where the NAC framework yields superior classification accuracy compared to the case where the i.i.d. assumption amongst the labels prevails. To demonstrate the ubiquity of this idea, we showcase examples from disparate areas such as health policy, political science and finance.

3.5.1 NAC with generative models for likelihood

Now, we describe a real world dataset where MAP classification done according to (3.39) with a generative model based likelihood that results in superior classification performance compared to the i.i.d. samples assumption based classification.

The IMDB project

The aim of this project was to classify movies as *hit* or *flop*. A movie was declared to be a hit if it grossed more than \$2 million or as a flop otherwise. The network which is built by linking movies if they shared a production company was first used in [94] to perform classification using a node-centric framework utilizing only the graph structure and is as shown in Figure 3.4. The dataset has 572 hit movies and 597 flops ($n = 1169$) yielding a baseline accuracy 51.07%. For, the textual features, we used the movie plots mined from IMDB *data dump* [30] and a 9457 size dictionary was derived after *stemming*, *lemmatizing* and *stop word removal* [95]. The generative likelihood model used was the Bag of words model described in (3.34).

After a 4-fold cross validation procedure described in Section 3.39, the obtained classification results are as tabulated in Figure 3.4. We used LBP as the approximate inference algorithm to solve (3.39).As seen, the mean classification accuracy increased from $\sim$ 60% to about 83%, thereby indicating that the textual features extracted from the movie plots were weak predictors of whether a movie made money or not while also vindicating our claim of Network Aided Classification with the *network effect* emanating from the co-production network being the more dominant predictive factor.

3.5.2 NAC with discriminative models for likelihood

The Citeseer project

The CiteSeer dataset, which was extracted from the CiteSeer database [96] had 3312 papers split into 6 classes labeled *Agents, AI, DB, IR, ML* and *HCI*. We bunched together papers labeled *'Agents,'AI, and 'ML'* together as they fell under Class 'I' (I.2;I.2.7;I.2.11) of the ACM Computing Classification System [97], and then combined the *'DB','IR' and 'HCI'* papers together as they fell under Class 'H' (H.2 ; H.3 and H.5) of the ACM Computing Classification System. This resulted in a two class dataset with a 1435/1877 split between class 'I' papers and class 'H' papers resulting in a baseline accuracy of 56.7% After *lemmatizing* and stop word removal, the dictionary size was 3718 words.

After a 4-fold cross validation procedure described in Section 3.39, the obtained classification results are as tabulated in Figure 3.5. We used LBP as the approximate inference algorithm to solve

(3.39). As seen, the mean classification accuracy increased from $\sim$ 66% to about 81%, thereby vindicating our claim of Network Aided Classification even with discriminative classification.

County crime-level estimation project

Besides the Citeseer project, we showcase another scenario where the SVM-based discriminative classifier's classification accuracy was improved using the underlying network albeit in a slightly different setting.

The United States Census Bureau defines four statistical regions (North-East, Mid-West, South and West), with nine divisions [98] with regard to the 48 contiguous states. This division is as shown in Figure 3.6. Now, focusing on *Division 3: East North Central*, we pick two states Illinois (IL) and Indiana (IA) for our analysis.

Using census data [77], we mined the county level crime rates for counties across the two states and labeled the county 'Hi' or 'Lo' based on whether the county had a crime rate higher than the population median or not. Now, in order to predict these levels, numerical features such as Voting rate, Voting tendency, Median household income and Mean Tax rates were used. The underlying graph used in this NAC framework was the spatial inter-county geographical adjacency graph.

Unlike, the setting with the citeseer dataset, we trained the SVM with the RBF kernel and also learnt the Ising global edge potential θ using one state's data and tested on the neighboring states' data. For ML of θ, we used the procedure outlined in section 3.3.1 which caters to the fully observed snapshot scenario. The results are as shown in Figure 3.7. The SVM alone had made 13 erroneous classifications out of 92 counties for Indiana (IN) when trained on data coming from the 102 counties of Illinois (IL), which was reduced to 7 errors upon using the Ising prior. Similarly, the SVM had made 19 erroneous classifications out of 102 counties for Illinois(IL) when trained on data coming from the 92 counties of IN, which was then reduced to 6 erroneous classifications upon using the Ising priors. As seen, the θ for both the states nearly matched ($\sim$ 0.23). We would like to assert that this was not coincidental and these couple of states were chosen using the a priori knowledge of *similarity* between the two states, an idea has applications like lessening the cost of census surveys by harnessing the data of one region and re-using it in a similar neighboring region using the NAC framework.

Bipartisan cloture roll call vote prediction project

The application here is that of predicting roll call votes during bipartisan cloture votes in the United States senate, which is deemed challenging in political science owing to the fact that senators tend to exude lesser allegiance towards their party and state affiliations during these votes . We harness the joint press release network (PRN) of these senators (first introduced in Figure 1.6(b)) to define the Ising prior and perform network aided classification to predict roll call votes.

Background: In the Unites States senate, a senator is allowed to 'filibuster' [99] which involves speaking for an indefinite period of time on any subject whatsoever in order to prevent action on bills that would otherwise pass with a simple majority. Cloture is the sole counter-procedure by which the Senate can vote to place a strict fixed time limit on consideration of a bill or other matter, and thereby overcome a filibuster. Under the cloture rule [100], the Senate may limit consideration of a pending matter to 30 additional hours, but only by vote of three-fifths of the full Senate, normally 60 votes. While filibusters appear to be a choice mainly exercised by the minority party, there have been plenty of instances where the majority party senators have taken up filibustering, indicating the rather bipartisan nature of the senate itself [101].

Dataset description: In this project, we take up a rather famous cloture vote instance titled the "Immigration Reform cloture - Senate Vote on the cloture vote for S. 1348 -Secure Borders,Economic Opportunity and Immigration Reform Act of 2007", [102] which was rejected on June 07, 2007. It was a motion to invoke cloture on a bill to provide for comprehensive immigration reform, including an expansion of the visa waiver program. 11 senators of the majority (Democratic party) and 38 senators of the minority party (Republican party) voted against the passing while 38 Democrat senators and 6 Republican senators voted for it. As is clearly evident, there was severance of party allegiance. Further, it is known that a Senator who votes in favor of cloture does not necessarily vote in favor of the bill. Similarly, a senator who voted against the cloture might end up voting in favor of the bill. These issues render the problem of vote prediction particularly challenging to solve [99]. In this project, we harness the undirected inter-senator PRN as an Ising prior [103] capturing the influence structure that might exist between senators.

The nodes in this network correspond to the senators and there exists an edge between two senators if they have organized and addressed a press release together, thereby indicating a strong level of intellectual compatibility and hence the presence of a strong influential tie between the two. This PRN for the senate in 2007 (110^{th} congress) has 92 nodes (senators) and 477 edges with a average degree of 5.1848.

Re-parameterization of the likelihood: We begin by firstly showing the re-parameterization of the max-inference problem in MRFs in to one of finding the lowest energy state of a Random Field Ising Model (RFIM) in case the discriminative classifier $\hbar(.)$ is such that the probability of misclassification is symmetric with respect to $x_i \in \{-1, +1\}$.

This is a model which will be extensively used in the upcoming chapters of network aided detection.

The simple idea is that the symmetric errors allow for two model simplifications. Firstly, in case of binary labels, they facilitate conceptualizing the discriminative classifier output as a bit at the output of a Binary Symmetric Channel (BSC) whose input is the true label, which in turn allows bringing in a classical communications theoretic framework of analysis in to Network Aided Detection.

Secondly, it also facilitates an Ising-styled parameterization of the node-potentials which can be used to *absorb* the observed label(s) as the external field of an Ising model. This is as shown below.

Specifically, defining $p(y_i = +1|x_i = +1) = p(y_i = -1|x_i = -1) = q_i$, we can write the parameterized node-wise likelihoods as,

$$p(y_i|x_i) = \frac{1}{2}\sqrt{q_i(1-q_i)}\left(\frac{q_i}{1-q_i}\right)^{\frac{x_i y_i}{2}}. \tag{3.42}$$

Combining (3.42) and (3.1), we can re-write the posterior probability as a Random Field Ising Model (RFIM) [68],

$$p(\mathbf{x}|\mathbf{y};\theta) = \frac{\exp\left(\sum_{i=1}^{n} h_i x_i + \theta \sum_{(i,j)\in E} x_i x_j\right)}{Z_{\mathbf{y}}(\theta)}. \tag{3.43}$$

The external fields, h_i are simply,

$$h_i = \log(\frac{q_i}{1-q_i})\frac{y_i}{2} \tag{3.44}$$

and the *posterior partition function* $Z_{\mathbf{y}}(\theta)$ would be,

$$Z_{\mathbf{y}}(\theta) = \sum_{\mathbf{x}} \exp\left(\sum_{i=1}^{n} h_i x_i + \theta \sum_{(i,j)\in E} x_i x_j\right). \tag{3.45}$$

ML estimation of the inverse temperature (θ): In section 3.3, we dealt with MLE of θ when we have the true labels ($\mathbf{x}$) as the training samples. In this subsection, we describe MLE of θ when we have an instance of the noisy labels $\mathbf{y}$ rather than the true labels ($\mathbf{x}$).
To begin with, let us define the energy of a configuration $\mathbf{x}$ to be [103],

$$\varepsilon(\mathbf{x}) = \sum_{(i,j)\in E} x_i x_j. \tag{3.46}$$

Now,we can define mean energy under the prior distribution ($\bar{\varepsilon}_0(\theta)$) and mean energy under the posterior distribution ($\bar{\varepsilon}_1(\mathbf{y};\theta)$) for a given inverse temperature θ, to be,

$$\begin{aligned} \bar{\varepsilon}_0(\theta) &= \sum_{\mathbf{x}} \varepsilon(\mathbf{x}) p(\mathbf{x};\theta) \text{ and} \\ \bar{\varepsilon}_1(\mathbf{y};\theta) &= \sum_{\mathbf{x}} \varepsilon(\mathbf{x}) p(\mathbf{x}|\mathbf{y};\theta) \end{aligned} \tag{3.47}$$

respectively.As shown in [89], [90], the Maximum Likelihood (ML) estimate of θ that maximizes the model evidence, that is, $\hat{\theta}_{ml} = \arg\max_{\theta}\{p(\mathbf{y};\theta)\}$, coincides with the θ at which the mean energy under the prior (Mean energy with no data) is equal to the mean energy under the posterior (Mean energy with no data), or,

$$\bar{\varepsilon}_0(\hat{\theta}_{ml}) = \bar{\varepsilon}_1(\mathbf{y};\hat{\theta}_{ml}). \tag{3.48}$$

In Fig.(3.8), we have plotted the Mean configuration energy curves under the prior (no data) and posterior (with data) for varying θ for the PRN-Ising model . As seen, the $\hat{\theta}_{ml,PRN} \approx 0.0474$.

Results Now, we presents results that showcase the improvement in terms of classification performance[3] brought about by using the Ising prior.

[3] which translates to vote prediction in our model

Essentially, we would like to showcase the network-effect induced improvement in classification accuracy over the baseline performance which relates to the scenario where we use just the node-wise features (h_i) and not the network. This baseline is provided by the feature-only maximum likelihood classifier, which in our case would simply be [90],

$$\hat{x}_{i,ml} = sign(h_i); i = 1, ..., n. \tag{3.49}$$

For finding the MAP state of the RFIM, we used the LBP as the approximate inference algorithm. We tried out other approximate inference algorithms such as Mean Field, Tree Reweighted Belief Propagation, Graph-cuts based inference and Simulated annealing but did not notice any improvement over the performance of LBP. In Fig.(3.9), we have plotted the variation of classification accuracy of the ML, MPM and MAP solutions obtained on running LBP for varying θ for the PRN-Ising model. We see that using the PRN prior results in improved performance over the ML classifier at $\hat{\theta}_{ml}$.
The MPM solution is seen to perform better than the MAP solution over a wide range of θ for the PRN Ising model. This conveys that the MPM solution here is more robust to the variations in the use of $\hat{\theta}$. At $\hat{\theta}_{ml}$, the improvement in classification accuracy achieved by the use of the PRN prior over the ML classifier is to the tune of 5.2%. $\hat{\theta}_{best}$ at which best classification accuracy of 84% is achieved turns out to be 0.06. The figure also clearly demonstrates the negative effect of choosing too large a θ for the PRN Ising model which results in over-smoothed solutions having worse accuracy than the ML solution. The standard UGM toolbox (2011 version) [104] has been used for LBP inference and ML estimation of θ.

3.6 SNA inspired community aided classification

One of the important focus areas in the field of Social Networks Analysis (SNA) is the study of prevalence of community structure in social networks which is basically the division of network nodes into groups within which the *intra-group* edge connections are dense, but with the *inter-group* edge density being substantially lower. Nodes that are part of a community tend to share similar properties compared to nodes that do not belong to the community. The definition of a community is rather loose and has been studied under various stylized frameworks as listed in [13].

In this section, we will show how this idea of the underlying graph being split into dense communities can be harnessed in our NAC framework to provide better classification performance. To begin with, let us assume that the underlying graph $G(V, E)$ is split in to N_c communities. . That is, the vertex set V can be written as the union of N_c vertex subsets,

$$V = \bigcup_{c=1}^{N_c} V_c. \tag{3.50}$$

Similarly, we can partition the edge set E into

$$E = \bigcup_{c=1}^{N_c+1} E_c, \tag{3.51}$$

where E_c is the edge subset whose each member edge has both the nodes at its ends in the same c^{th} community and E_{c+1} is the edge subset containing all the inter-community edges.

Now, using the idea that all the relationships between the nodes of a given community are *similarly weighed*, we make the model assumption that the edge potentials associated with the edges in a given community can be tied to a common value. Thus, this procedure yields a natural parametrization of the edge potentials and paves a way for finer characterization of the Ising prior as,

$$p_{comm}(\mathbf{x}) = \frac{\exp\left\{\sum_{(i,j)\in E}\left[\sum_{c=1}^{N_c+1} \theta_c \mathbf{I}\,[\![(i,j) \in E_c]\!]\right] x_i x_j\right\}}{Z(\boldsymbol{\theta_{comm}})}, \tag{3.52}$$

where $\boldsymbol{\theta_{comm}} = [\theta_1, ..., \theta_{N_c+1}] \in \mathbb{R}^{N_c+1}$. Now, this is especially useful in scenarios where tieing all the edge potentials to a common global edge potential will lead to natural model mismatch. Just for the reader's clarity, we present here the homogeneous Ising model with the global common edge-potential, θ, to be,

$$p_{global}(\mathbf{x}) = \frac{\exp\left\{\theta \sum_{(i,j)\in E} x_i x_j\right\}}{Z(\theta)}. \tag{3.53}$$

In the following subsection, we will present results from a real world dataset where we showcase a scenario where this community structure inspired parameterization of the Ising edge potentials results in enhanced classification accuracy.

State level Lung and Bronchus Cancer level classification

From [105], we extracted the age-adjusted lung and bronchus cancer rates in each of the 48 states of the USA for years 2003 to 2008. The demographic chosen was men belonging to all races. We converted these rates into binary labels based on whether the state was above or below the population median. The labels mined thus are as shown in Figure 3.10.

For features, we downloaded the cigarette consumption levels in these states from [105] for these very years and used as 1-D features. The generative model we used for the cigarette consumption level y_i was 2-class Gaussian mixture, defined as,

$$p(y_i|x_i = k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left[-\frac{(y_i - \mu_k)^2}{2\pi\sigma_k^2}\right]; k = -1, +1. \tag{3.54}$$

Figure 3.13 shows the Histogram plot of smoking level features across different years and Figure 3.14 shows the estimated class conditional means (μ_k) and variances σ_k^2 for the above mentioned years.

Now, using the underlying spatial inter-state adjacency matrix as the graph for the Ising prior (See Figure 3.11), we firstly estimated 7 communities in this graph using the Newman-Girvan(NG) community detection algorithm ([106]). The resultant graph is as shown in Figure 3.12(b).

We then used this community information to train the resultant 8 edge potentials using one year's data. We repeated the same procedure for community split defined in an alternate way.

As in the crime-level classification application, the United States Census Bureau [98] splits the 48 contiguous states into nine divisions for survey purposes. This split is as shown in Figure 3.12(a). We used these 9 divisions to provide an alternative community specification which was then used to train another Ising model with 10 different edge potentials.

Now, a single training year's features data was used to estimate the class conditional means and variances, and testing was carried out on another year's data.

The results are as tabulated in Figure 3.15. The label 'Global' in the plot refers to the case where the network prior in (3.53) was used. 'C-9' and 'C-7' refer to the cases where the prior used was from (3.52) with the edge potential vectors being parameterized according to community definitions being derived from the 9 region US census bureau categorization or from the 7 community

Newman-Girvan approach [106]. The 'ML' refers to the Maximum Likelihood classifier which is,

$$\hat{x}_i^{(ML)} = \arg \max_{k \in \{-1,+1\}} \left[p(y_i | x_i = k) \right]; i = 1, .., n. \tag{3.55}$$

From Figure 3.15, we can clearly see that both the community definition inspired Ising prior based NAC classifiers obtained improvements in classification accuracy compared to the feature only classifier (denoted by ML in the plots). In train-year/test-year combinations such as (2005/2007), (2006/2007) and (2007/2008), the 'C-9' based NAC outperformed the 'C-7' NAC variant. However, when the results were averaged across the 15 train-year/test-year combinations, the 'C-7' NAC variant had the best average classification rate. Figure 3.16 provides the classification accuracies after averaging over all the 15 train-year/test-year combinations.

3.7 Chapter Summary

We envisage a machine learning practitioner using our NAC framework as a *plug-in* addendum to the classification paradigm he is already using, which might be either discriminative model based or generative model based.

To this end, we demonstrate real world examples from disparate domains such as crime-level classification, cloture vote prediction and movie income classification where the NAC framework was plugged-in seamlessly with both discriminative as well as generative model based classifiers to bring about improvement in classification accuracy.

We also demonstrated using the lung-cancer level detection project as to how the community-structure idea emanating from the SNA research could be neatly incorporated to achieve richer specification of the Ising prior that will result in lessening the model mismatch that might occur on account of under-parameterization of the Ising prior and hence result in improved classification accuracy.

Finally, in the PRN based cloture vote prediction, we had seen that in case the discriminative classifier $\hbar(.)$ has the probability of misclassification being symmetric with respect to $x_i \in \{-1, +1\}$, we model the discriminative classifier output as a bit at the output of a Binary Symmetric Channel (BSC) whose input is the true label.

Now, we move on to Network Aided Detection problems in the forthcoming chapters where we use this BSC model idea and propose a classical communications theoretic framework of analysis in to our Network Aided Classification paradigm.

Pseudo-code 1: Methodology for NAC

```
Initialize: Split Edge-set into 4 random equal-sized subsets: $E = \bigcup_{k=1}^{4} E_k : |E_k| \approx M/4$;
for $k$ = 1:4 do
    1: $E_{test} = E_k,\ E_{train} = E \setminus E_{test},\ V_{test} = \mathcal{V}(E_k),\ V_{train} = V \setminus V_{test},\ D_{test} = \{\mathbf{Y}_{test}, \mathbf{x}_{test}\},\ D_{train} = \{\mathbf{Y}_{train}, \mathbf{x}_{train}\}$;
    2: Use $\mathbf{x}_{train}$ to estimate $\hat{\theta}_{ml}$;
    if Discriminative Model based Classification then
        1: Split $D_{train}$ into 2 parts; $D_{train-1}$ and $D_{train-2}$ in 4:1 ratio ;
        2: Use $D_{train-1}$ to learn the Discriminative Classifier: $\hbar : \mathbb{R}^d \rightarrow \mathcal{X}$;
        3: for $i$=1: $n_{train-2}$ do
            $\hat{x}_{train-2,i} = \hbar(\mathbf{y}_{train-2,i})$;
        end
        3: Use $\hat{\mathbf{x}}_{train-2}$ and $\mathbf{x}_{train-2}$ to estimate the confusion-matrix probabilities: $p(\hat{x}_i|x_i)$ ;
        (4a) : for $i$=1: $n_{test}$ do
            $\hat{x}_{test,i,no-nac} = \hbar(\mathbf{y}_{test,i})$;
        end
        (4b) : $\hat{\mathbf{x}}_{test,nac} = \max_{\mathbf{x}\in\{-1,+1\}^{n_{test}}} \left[ \prod_{v\in V_{test}} \phi_v^{(dis)}(x_v) \prod_{(i,j)\in E_{test}} \psi_{ij}(x_i, x_j) \right]$ ;
    end
    else if Generative Model based Classification then
        1: Use $D_{train}$ to learn the generative model parameters $\hat{\Omega}_k$;
        2: for $i$=1: $n_{test}$ do
            $\hat{x}_{test,i,no-nac} = \arg\max_{k\in\mathcal{X}} p(\mathbf{y}_{test,i}|x_{test,i} = k; \hat{\Omega}_k)$;
        end
        3: $\hat{\mathbf{x}}_{test,nac} = \max_{\mathbf{x}\in\{-1,+1\}^{n_{test}}} \left[ \prod_{v\in V_{test}} \phi_v^{(gen)}(x_v) \prod_{(i,j)\in E_{test}} \exp\left(\hat{\theta}_{ml} x_i x_j\right) \right]$;
    end
    $acc(k) = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \{\mathbf{I}[[\hat{x}_{test,i} = \hat{x}_{test,i}]]\}$;
end
$Accuracy = \frac{1}{4} \sum_{k=1}^{4} acc(k)$;
```

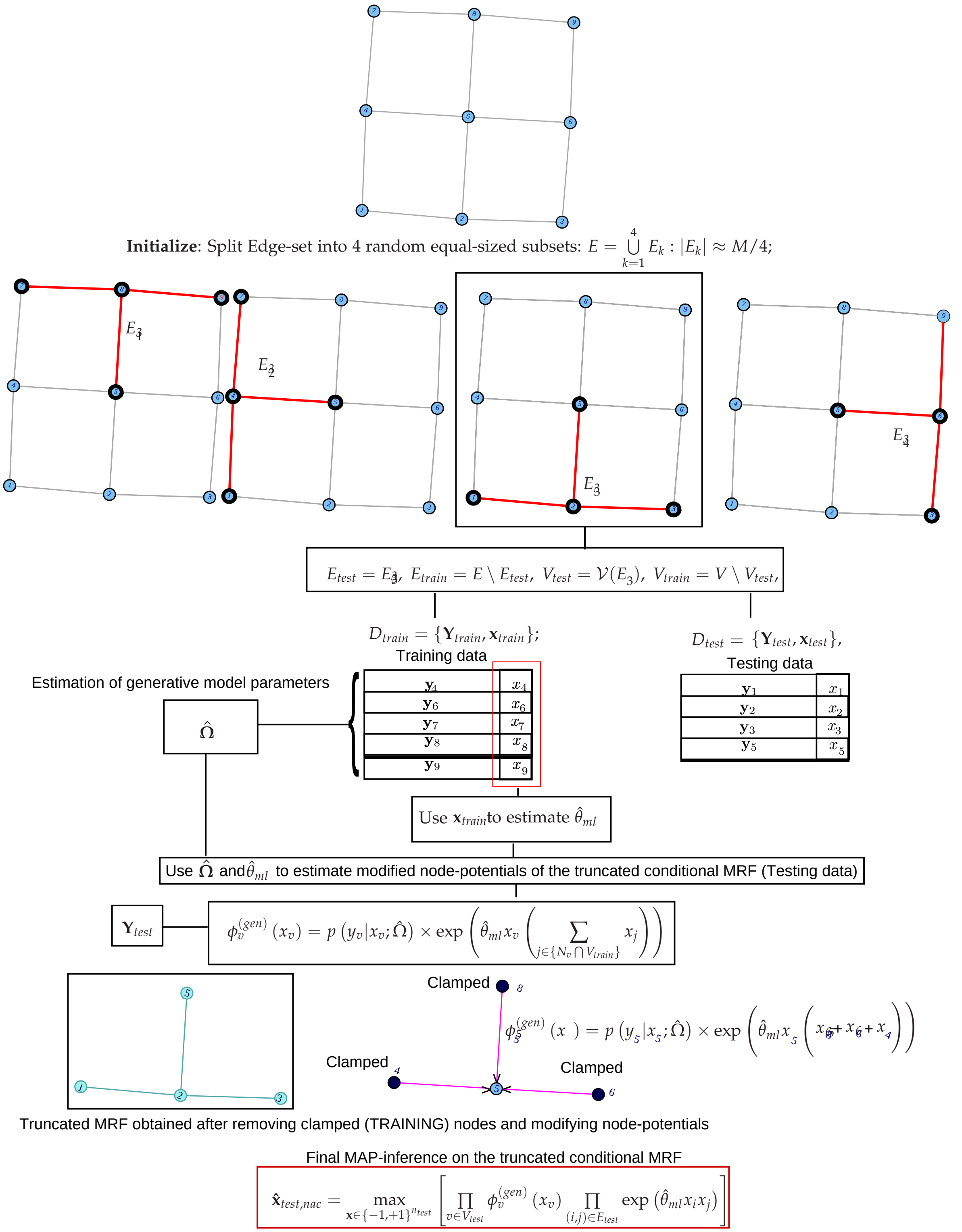

Initialize: Split Edge-set into 4 random equal-sized subsets: $E = \bigcup_{k=1}^{4} E_k : |E_k| \approx M/4$;
E_1
E_2
E_3
E_4
$E_{test} = E_3,\ E_{train} = E \setminus E_{test},\ V_{test} = \mathcal{V}(E_3),\ V_{train} = V \setminus V_{test},$
$D_{train} = \{\mathbf{Y}_{train}, \mathbf{x}_{train}\};$
Training data
$D_{test} = \{\mathbf{Y}_{test}, \mathbf{x}_{test}\},$
Testing data
Estimation of generative model parameters
$\hat{\Omega}$
$\mathbf{y}_4$ x_4
$\mathbf{y}_6$ x_6
$\mathbf{y}_7$ x_7
$\mathbf{y}_8$ x_8
$\mathbf{y}_9$ x_9
$\mathbf{y}_1$ x_1
$\mathbf{y}_2$ x_2
$\mathbf{y}_3$ x_3
$\mathbf{y}_5$ x_5
Use $\mathbf{x}_{train}$to estimate $\hat{\theta}_{ml}$
Use $\hat{\Omega}$ and$\hat{\theta}_{ml}$ to estimate modified node-potentials of the truncated conditional MRF (Testing data)
$\mathbf{Y}_{test}$
$\phi_v^{(gen)}(x_v) = p(y_v|x_v;\hat{\Omega}) \times \exp\left(\hat{\theta}_{ml} x_v \left(\sum_{j \in \{N_v \bigcap V_{train}\}} x_j\right)\right)$
Clamped
Clamped
Clamped
$\phi_5^{(gen)}(x_5) = p(y_5|x_5;\hat{\Omega}) \times \exp\left(\hat{\theta}_{ml} x_5 \left(x_8 + x_6 + x_4\right)\right)$
Truncated MRF obtained after removing clamped (TRAINING) nodes and modifying node-potentials
Final MAP-inference on the truncated conditional MRF
$\hat{\mathbf{x}}_{test,nac} = \max_{\mathbf{x} \in \{-1,+1\}^{n_{test}}} \left[\prod_{v \in V_{test}} \phi_v^{(gen)}(x_v) \prod_{(i,j) \in E_{test}} \exp(\hat{\theta}_{ml} x_i x_j)\right]$

Figure 3.3: Demonstration of the NAC methodology on a 9 node grid graph

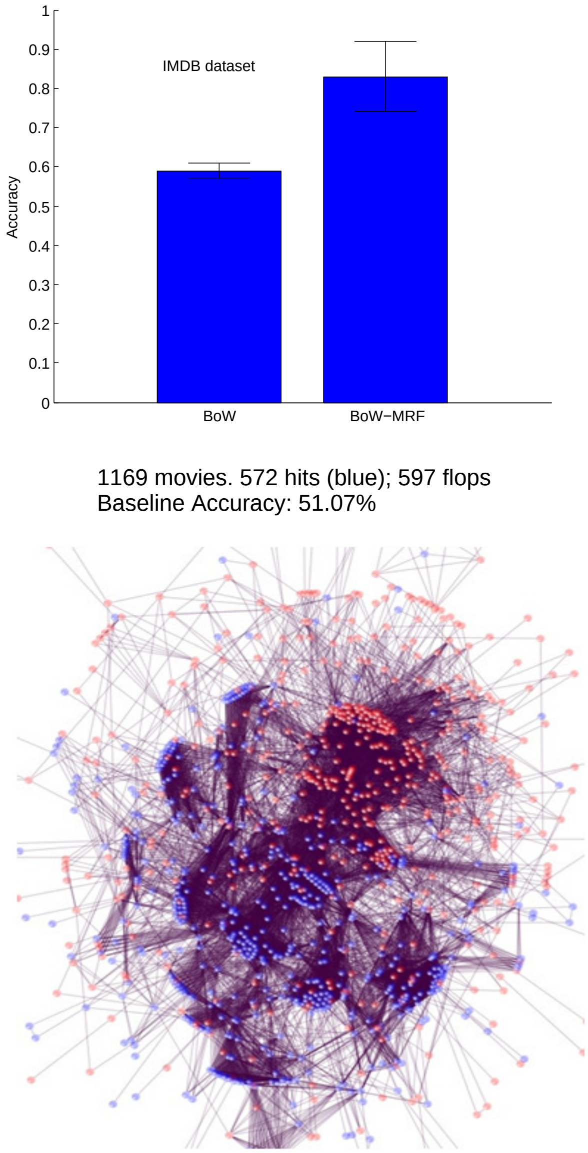
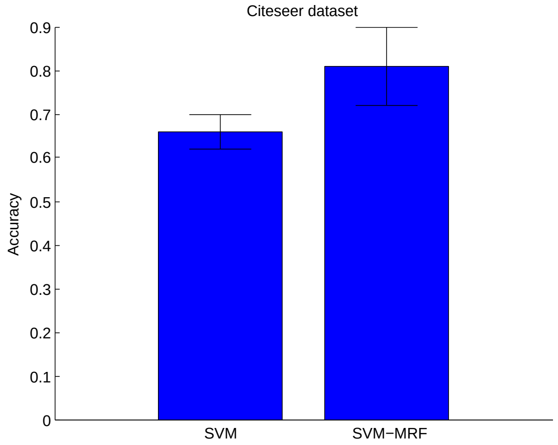


Figure 3.4: Movie returns classification using the IMDB dataset



3312 papers
1435 |1877 split
Baseline Accuracy: 56.67%
Network: 3312 nodes with 4598 edges
Dictionary: 3718 words

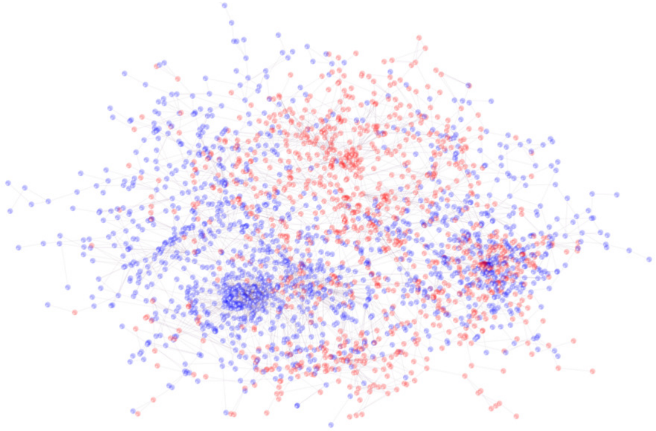


Figure 3.5: Citeseer dataset.

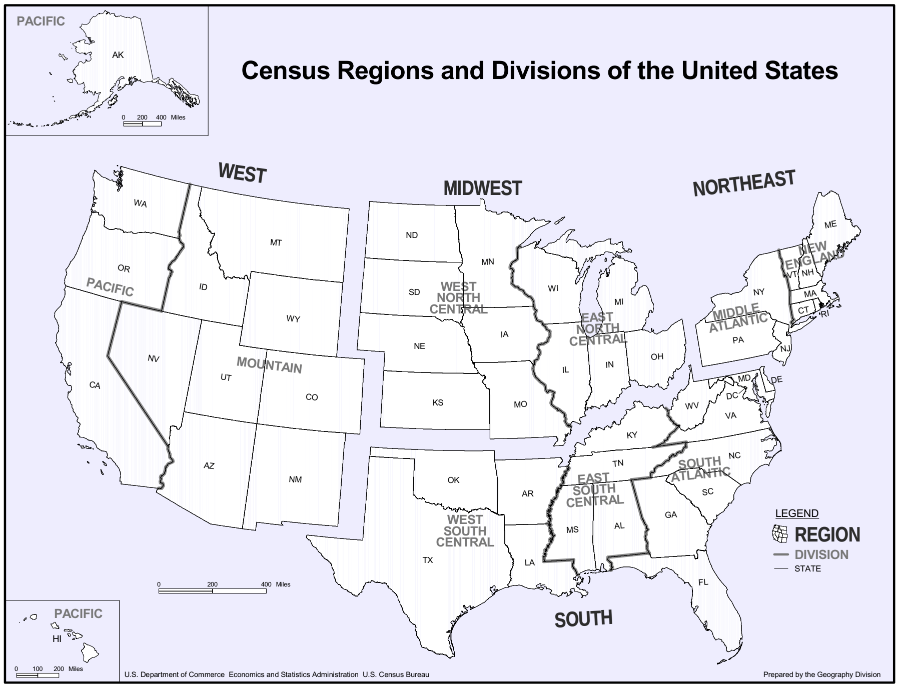


Figure 3.6: The four statistical regions with nine divisions according to the United States Census Bureau.

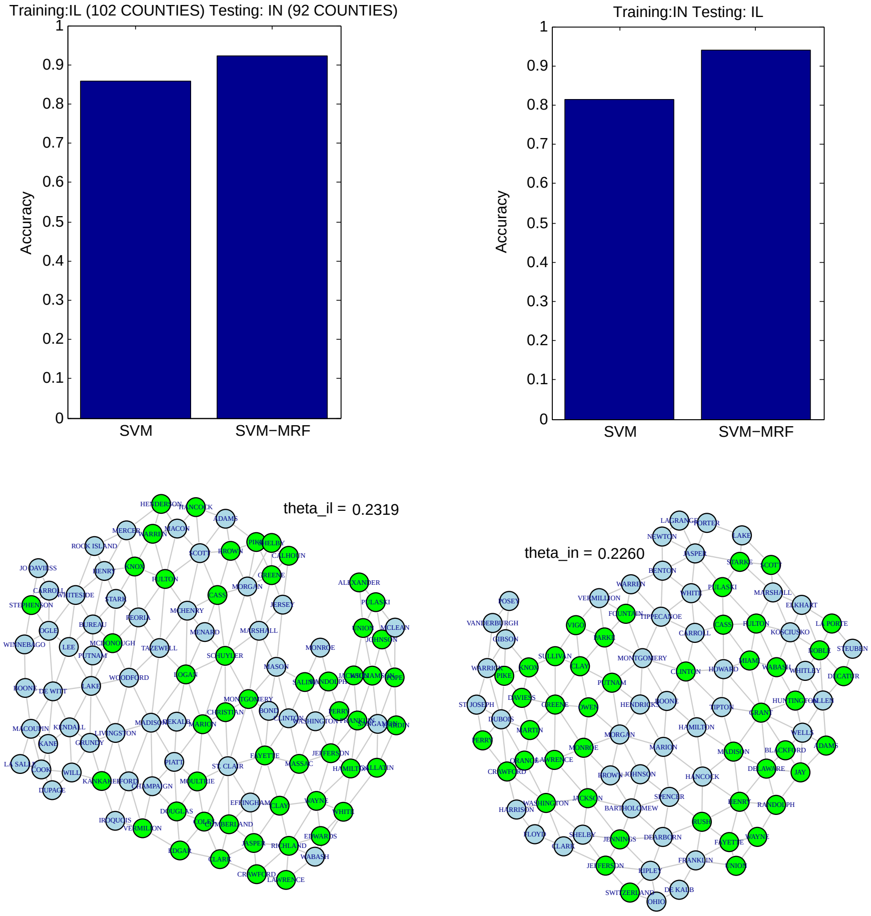


Figure 3.7: Crime level classification in state counties

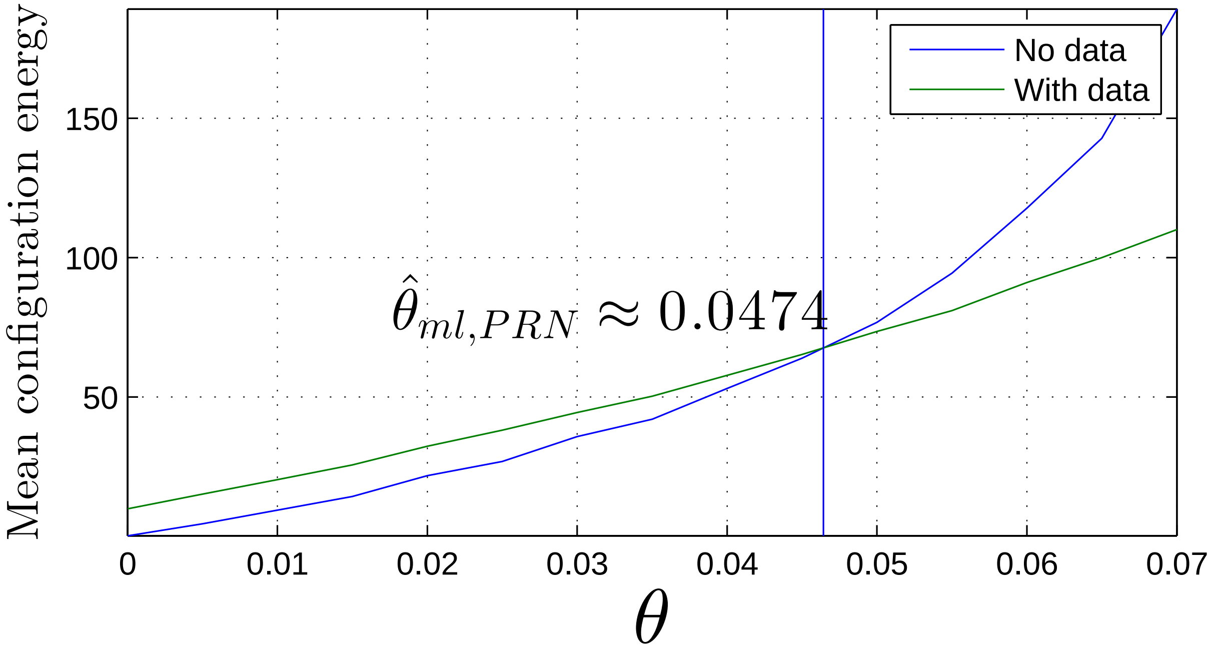


Figure 3.8: ML estimation the inverse temperature: Mean configuration energy with and without data as a function of the inverse temperature chosen

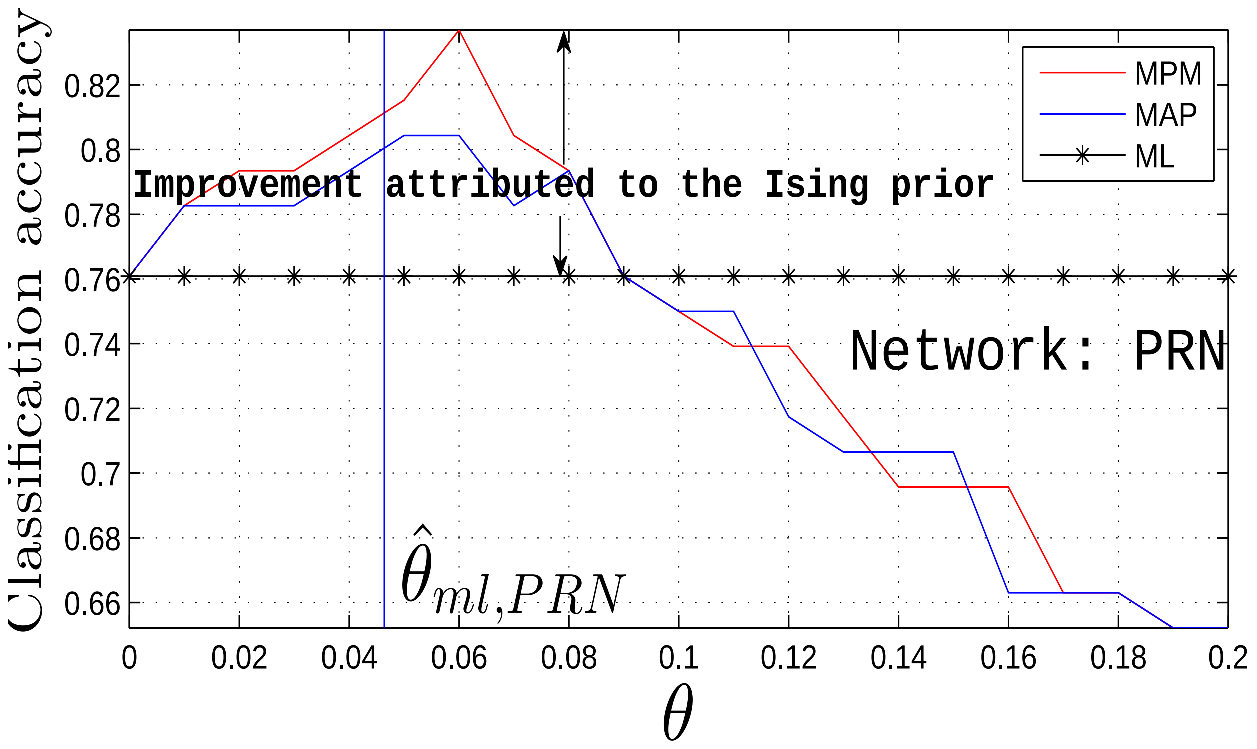


Figure 3.9: Variation of classification accuracy of the MPM and MAP solutions with respect to the inverse temperature chosen.

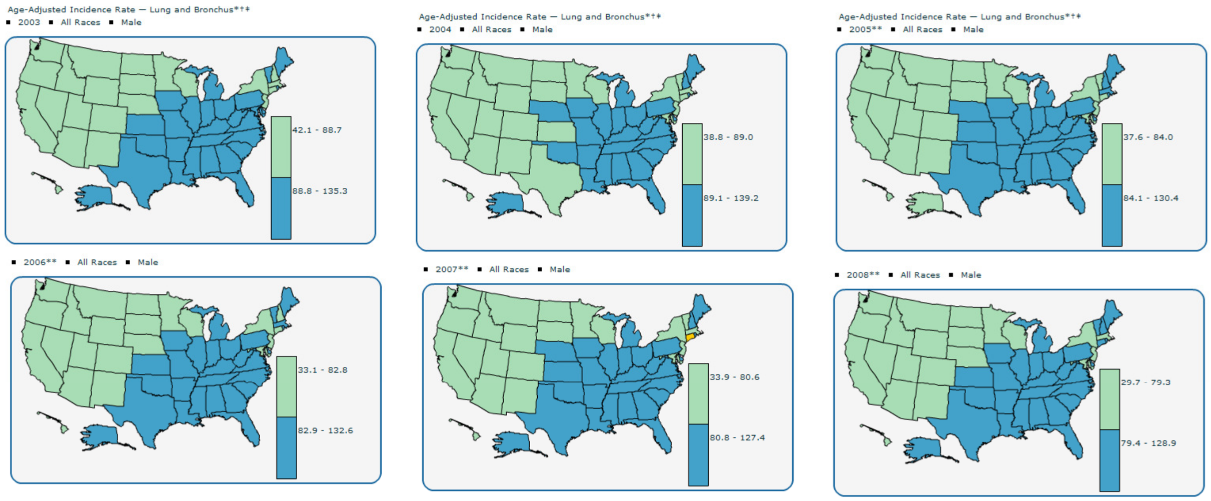

Age-Adjusted Incidence Rate — Lung and Bronchus*†‡
▪ 2003 ▪ All Races ▪ Male
42.1 - 88.7
88.8 - 135.3
Age-Adjusted Incidence Rate — Lung and Bronchus*†‡
▪ 2004 ▪ All Races ▪ Male
38.8 - 89.0
89.1 - 139.2
Age-Adjusted Incidence Rate — Lung and Bronchus*†‡
▪ 2005** ▪ All Races ▪ Male
37.6 - 84.0
84.1 - 130.4
▪ 2006** ▪ All Races ▪ Male
33.1 - 82.8
82.9 - 132.6
▪ 2007** ▪ All Races ▪ Male
33.9 - 80.6
80.8 - 127.4
▪ 2008** ▪ All Races ▪ Male
29.7 - 79.3
79.4 - 128.9

Figure 3.10: MAP showing the age adjusted Lung and Bronchus Cancer incidence rates in states

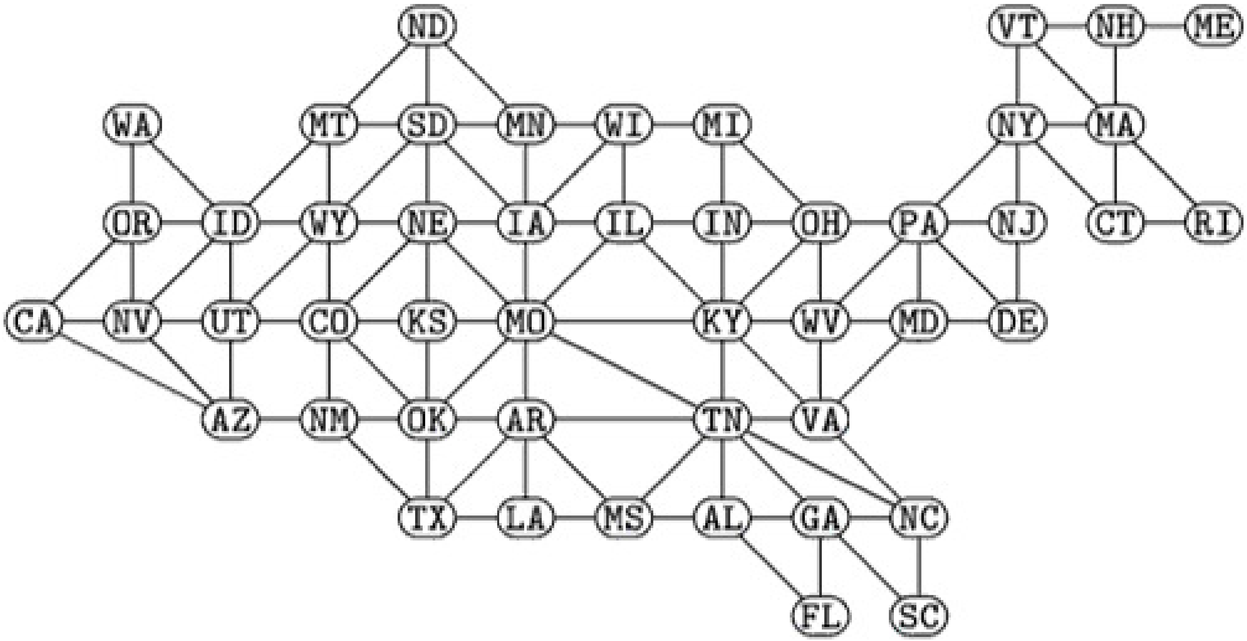
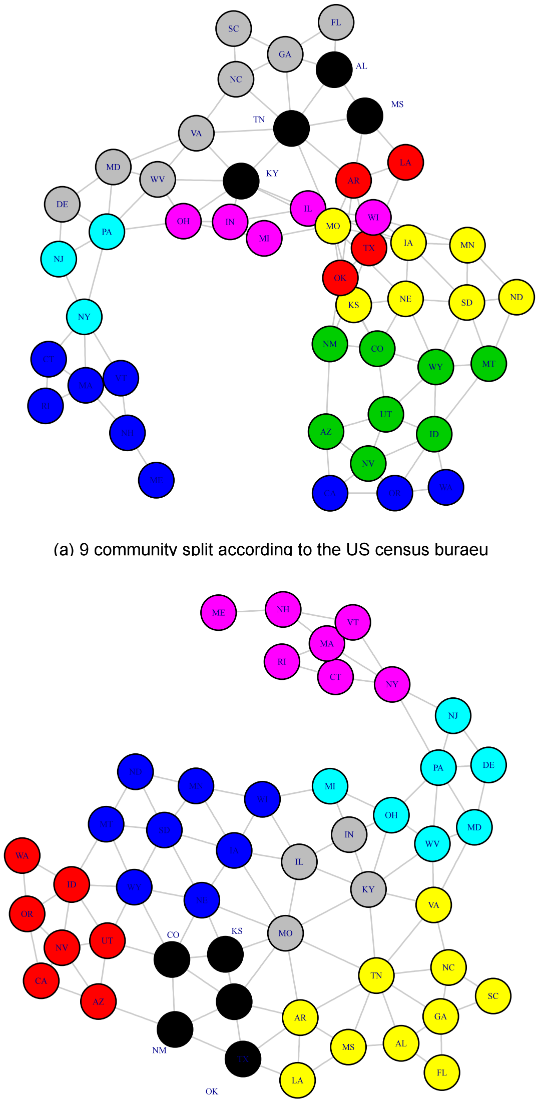


Figure 3.11: The spatial adjacency graph of the 48 contiguous states of USA



(a) 9 community split according to the US census buraeu

(b) 7 community split according to the Newman-Girvan algorithm

Figure 3.12: The community (regions) of the USA according to the US census bureau and the NG community detection algorithm

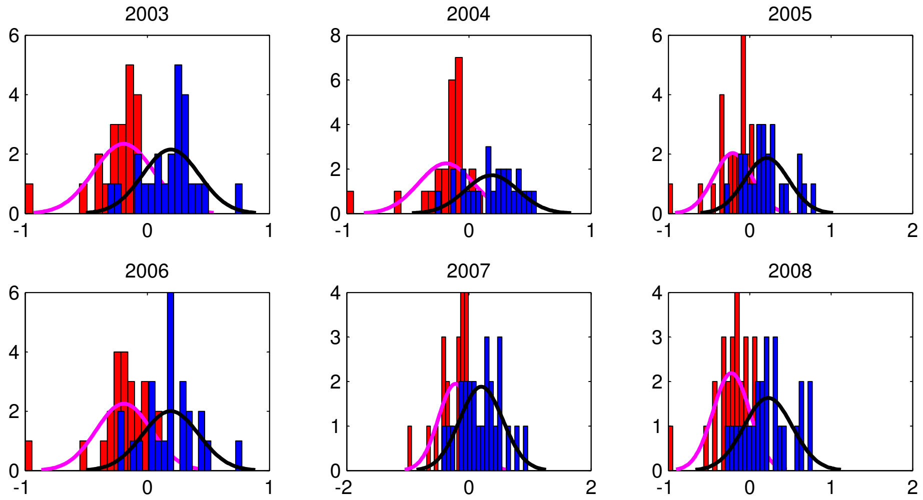


Figure 3.13: Histogram plot of smoking level features across different years

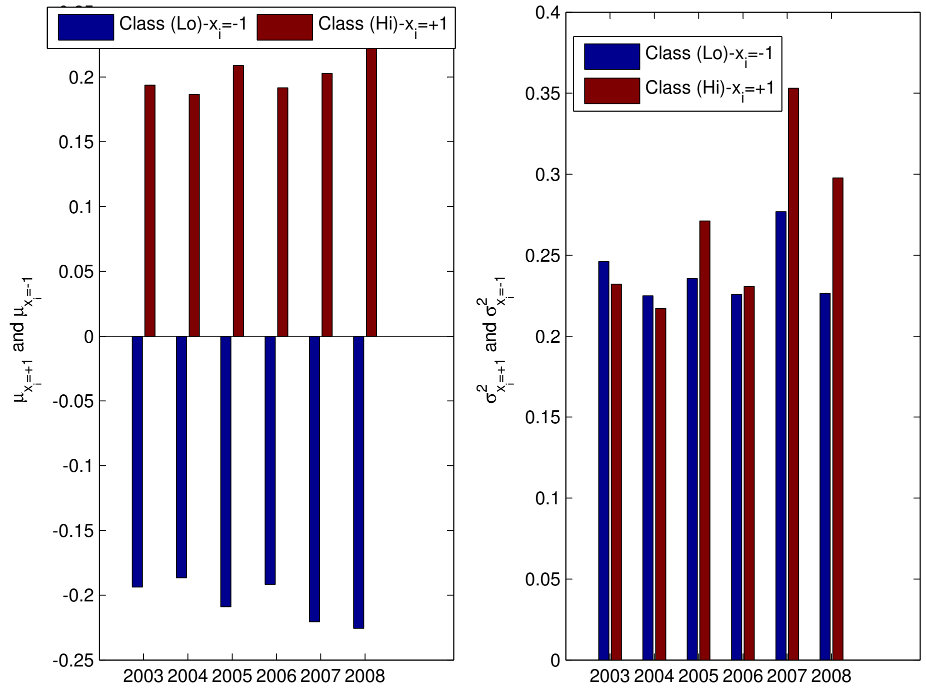


Figure 3.14: Estimated mean (μ_k) and variance (σ_k^2) across different years

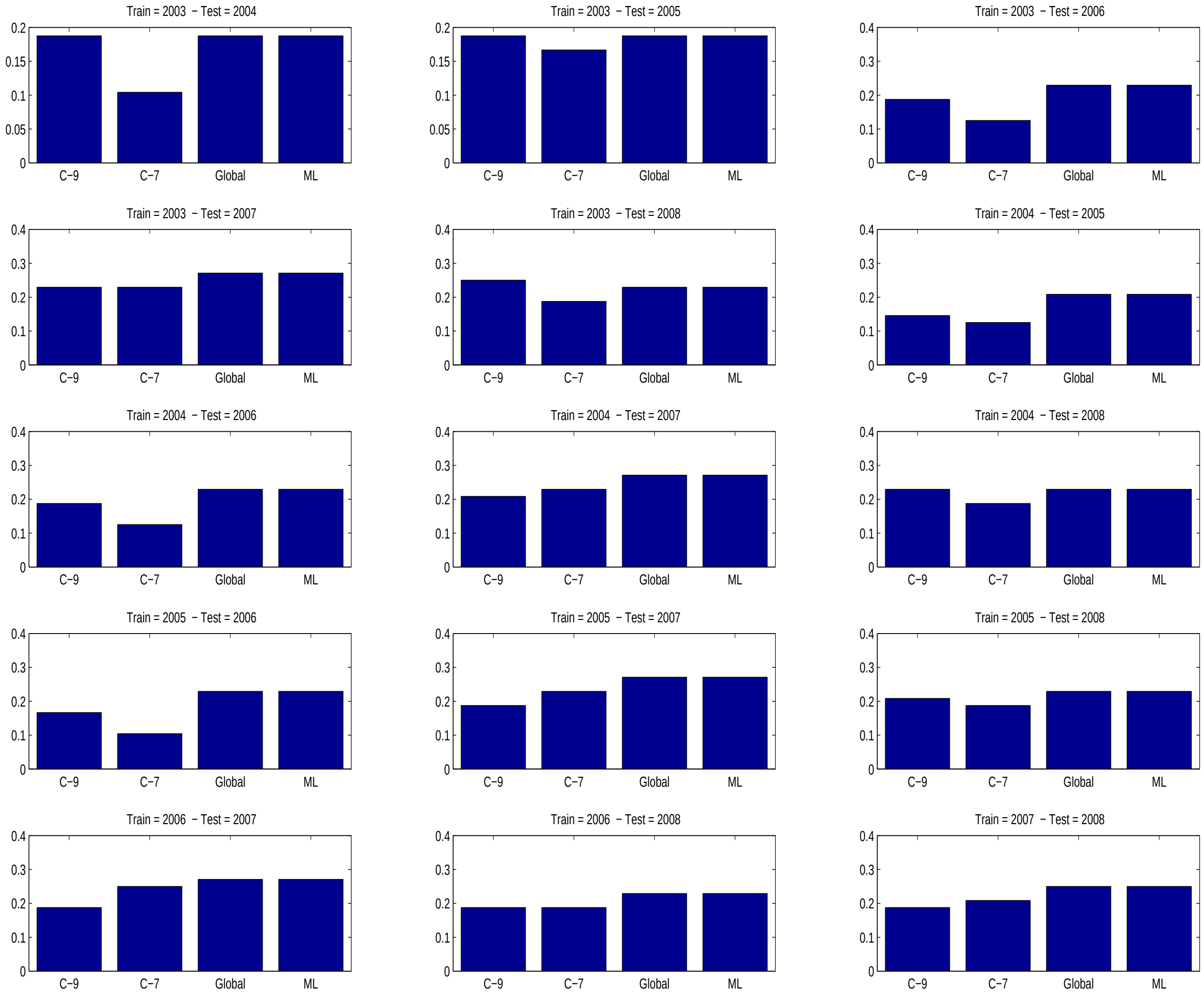


Figure 3.15: Error rate comparisons for across different train-year test-year combinations

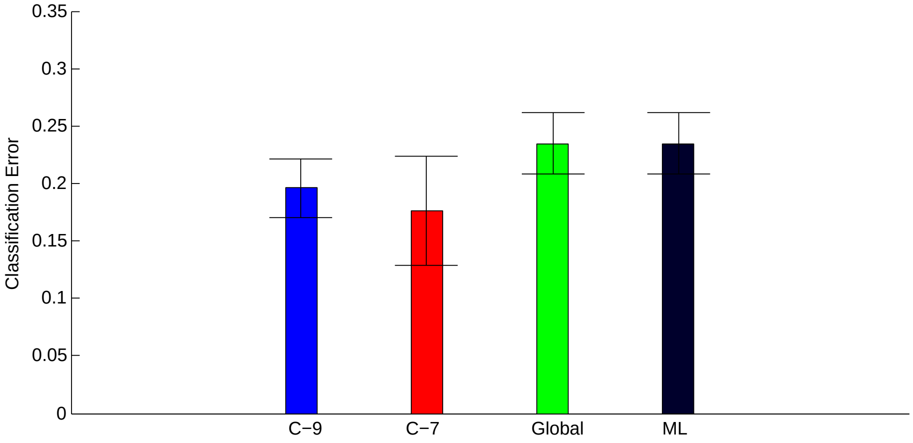


Figure 3.16: Error rate comparisons for the 4 models averaged across the different train-year test-year combinations

Chapter 4

Latent sentiment detection

4.1 Introduction

In Chapter-3, we introduced the Network Aided Classification (NAC) framework where the focus was on solving a ***vector*** classification problem. We showcased several applications of this framework with real world datasets and demonstrated an appreciable improvement in the classification error rate compared to ML based classification. We saw that with the use of the NAC, the classification error rate was of the order of 15% to 20%, which is certainly an improvement over the ML classifiers encountered but still very high compared to the error rates typically encountered in communication theoretic settings where the Bit Error Rates (BER) encountered are of the order of 10^{-k}, with $k \gtrsim 3$.

In this chapter, we now look at binary detection problem where the focus is using the underlying network to detect whether the *global sentiment* prevalent in the social network is either positive or negative instead of vector of individual sentiments. We term this to be Network Aided Detection (NAD). This problem is not just pertinent from a real-world perspective (as we will soon motivate) but also allows for the detection error rates to be low enough to be comparable to those encountered in communication theoretic settings. This in turn facilitates importing certain classical communication-theoretic error probability upper bounding techniques which results in some interesting theoretical insight emerging from the analysis, an aspect that was lacking in Chapter-3.

4.1.1 Sentiment detection in OSNs

As motivated in Chapter-1, Online Social Networks (OSN), such as Twitter[1] have come to heavily influence the way people socially interact.

Recent world events such as the Arab Spring, witnessed cascading democratic revolutions characterized by a strong reliance on online social media such as Twitter and Facebook [107]. Today, there are about 554 million registered active Twitter users with about $135,000$ new Twitter users signing up everyday. Around 58 million tweets are *tweeted* per day and the website attracts over 190 million visitors every month [108]. Such staggering numbers have turned such OSNs into an invaluable data source for organizations and individuals who have a strong social, political or economic interest in maintaining and enhancing their clout and reputation. Therefore, extracting and analyzing the embedded sentiment in the microblogs (or *Tweets*) posted by the tweeters about these organizations or individuals, or specific issues, products and events related to them or their competitors, is of great interest to them. Of particular interest is the *latent sentiment* (as opposed to individual expressed sentiments), which can be either positive or negative with respect to a particular position. We explain this latent sentiment in detail in Section 4.2.

4.1.2 Micro-blog /Tweet level sentiment classification

Strict length restrictions (such as the 140 character-limit per tweet), irregular structure of the microblog content and the usage of sarcasm renders the problem of automatic latent sentiment detection (classifying latent sentiment as positive or negative) from the microblog contents error-prone. As evidenced in literature ([26], [27]), sentiment detection has been approached from an engineering perspective with the main focus being on sentiment detection algorithms, followed by empirical performance comparisons using standard datasets such as Stanford Twitter Sentiment (STS) dataset and the Obama-McCain Debate dataset [109]. Works such as [110] and [25] have focused on harnessing the underlying social network to aid in sentiment analysis. [110] used label propagation to incorporate labels from a maximum entropy classifier trained on noisy labels in combination with the Twitter follower graph. [25] incorporated a semi-supervised frame-work, that used either the follower/followee network or the @-mentions network and applied loopy be-

[1] `https://Twitter.com/`

lief propagation to infer user sentiment labels on unlabeled nodes. In [28], we had considered a similar problem of network aided detection of votes in the senate harnessing the joint press release network. In this chapter, we consider the problem of social-network aided sentiment detection. That is, we use the underlying social network as a graph of sentiment similarity, to improve the performance of latent sentiment detection. In this chapter, we approach such problems from a relatively scientific perspective. That is, we attempt to answer the following important question regarding latent sentiment detection in social networks.

- *How does the social network structure affect the performance of a trivial sentiment detector that is oblivious to the presence of an underlying social network?*

This analysis helps isolate the influence that is introduced by the network prior alone. For this, we use the stylized model of latent sentiment detection, based on the Ising prior proposed in Chapter-1 ((1.8)). We then analyze the performance of the trivial sentiment detector, keeping in mind the underlying social network structure. For this, we are inspired by a communications-oriented viewpoint, where we view the underlying network as providing a *weak channel code*, that transmits one bit of information, which is the latent sentiment. Accordingly, we are able to analyze the performance of the sentiment detector by borrowing tools from information theory. We can then compute and contrast the performance under various stylized social network topologies, thus providing a comprehensive answer to the question posed above. Thus, we show that *communication theorists* can contribute to the growing field of social network analysis. The rest of the chapter is organized as follows. In Section 4.2, we formally describe the latent sentiment detection problem, introduce the formal model and motivate its relevance through real-world scenarios based on Twitter. We also specify the trivial sentiment detector. In Section 4.3, we perform a communications-oriented analysis of the trivial sentiment detector to derive an upper bound on the detection error probability, in terms of an error exponent. We also show how the exponent can be evaluated numerically for various stylized topologies such as the complete network, the star network and the (closed) chain network. In Section 4.4, we present numerical results that show how the error exponent depends on the network topology and other model parameters. We conclude the chapter in Section 4.5.

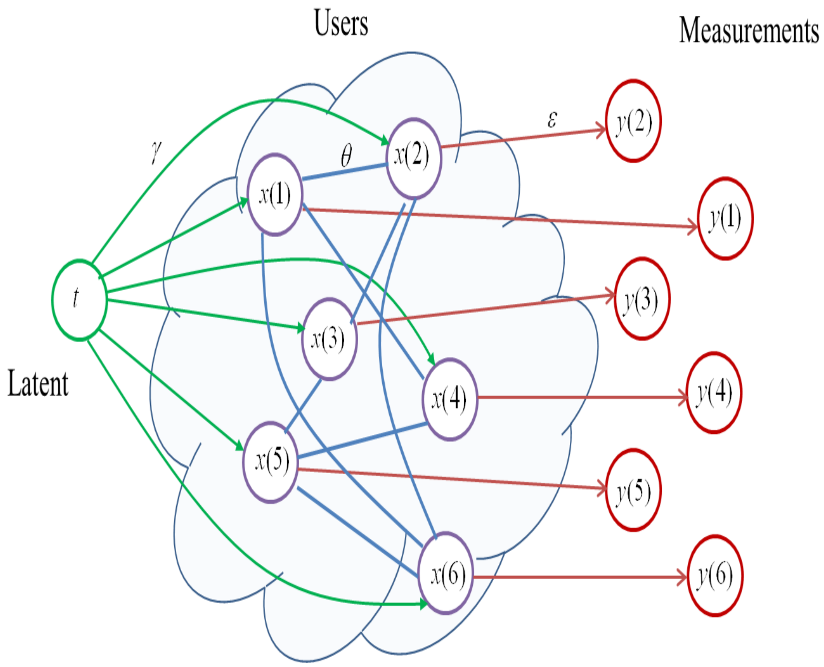


Figure 4.1: Model for latent sentiment detection.

4.2 Latent sentiment detection problem

4.2.1 Model

Here, we present the Ising prior based model first proposed in (1.8) in Chapter-1 (Figure 4.1), in detail. Let $\mathbf{x} \in \{-1,+1\}^n$ be the vector of expressed sentiments of the n members of a social network, with $x_i \in \{-1,+1\}$ being the expressed sentiment of the i^{th} member/node. The social network structure is modeled as an undirected graph $G(V,E)$ characterized by its upper triangular adjacency matrix A. It may be obtained using the follower/followee relationships, or in some cases, using the @-mentions in the tweets [25]. The graph is undirected since we will use it to model correlation, rather than influence flows. The sentiments are assumed to be sampled from an underlying homogeneous MRF [68] with unit edge potential and inverse temperature parameter, θ. In this chapter, we assume $\theta \geq 0$, so that we are restricting ourselves to attractive/ferromagnetic models, which correspond to homophilic networks. In such a ferromagnetic model, the neighboring nodes positively correlate with each other, so that the distribution has more probability in configurations with similar values on the nodes of the graph.

Let $t \in \{-1, +1\}$ indicate the latent sentiment variable which homogeneously influences every node of the network as a local field of strength γt. In the absence of any sentiment bias, we assume t to be equi-probably equal to $+1$ or -1. Thus, the conditional distribution of $\mathbf{x}$ given t, can be written as,

$$p(\mathbf{x}|t) \;=\; \frac{\exp\left\{\theta \mathbf{x}^T A \mathbf{x} + \gamma t \mathbf{e}^T \mathbf{x}\right\}}{\sum\limits_{\mathbf{x}} \exp\left\{\theta \mathbf{x}^T A \mathbf{x} + \gamma t \mathbf{e}^T \mathbf{x}\right\}}. \tag{4.1}$$

Notice that from the communications perspective, $\mathbf{x}$ is a codeword randomly chosen in response to bit t. Let $\mathbf{y}$ be a noisy estimate of $\mathbf{x}$. It may be estimated from the features extracted from the user profiles or could even be the sentiment vector estimated by a given classifier algorithm, such as the ones in [26] and [27]. While the alphabet of each y_i can be arbitrary, in this chapter, for simplicity, we assume that it is binary $\{-1, +1\}$. We model $\mathbf{y}$ to be the output of n-identical and independent Binary Symmetric Channels (BSCs) characterized by the equal cross-over probability p_{bsc}, with the elements of the true sentiment vector $\mathbf{x}$ as the input. Therefore, the conditional distribution, $p(\mathbf{y}|\mathbf{x})$ may be written as,

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{c^n} \exp\left\{\varepsilon \mathbf{y}^T \mathbf{x}\right\}, \tag{4.2}$$

where $\varepsilon = \frac{1}{2} \log\left(\frac{1-p_{bsc}}{p_{bsc}}\right)$ and $c = 2\cosh(\varepsilon)$. The joint distribution of all variables may now be written as,

$$\begin{aligned} p(t, \mathbf{x}, \mathbf{y}) \;&=\; \frac{1}{2\, Z'(t)} \exp\left\{\theta \mathbf{x}^T A \mathbf{x} + \varepsilon \mathbf{y}^T \mathbf{x} + \gamma t \mathbf{e}^T \mathbf{x}\right\}, && (4.3) \\ Z'(t) \;&=\; c^n \sum_{\mathbf{x}} \exp\left\{\theta \mathbf{x}^T A \mathbf{x} + \gamma t \mathbf{e}^T \mathbf{x}\right\}. && (4.4) \end{aligned}$$

4.2.2 Revisiting the `#iloveobamacare` Twitter network

The model (4.3) is applicable to several real-world latent sentiment detection scenarios. We begin by assuming that there exists a latent sentiment ($t \in \{-1, +1\}$), which will cause a certain concrete event in the future. This event may be the passage (or defeat) of a bill in the senate, or an up (or down) movement of the stock market, when $t = +1$ (or -1, respectively). The intention is to predict this real-world event in the present using the expressed sentiments gathered from the *twitterverse* ($\mathbf{y}$ in our model). Thus, it is the same as detecting the value of t (hence the term *latent sentiment detection*).

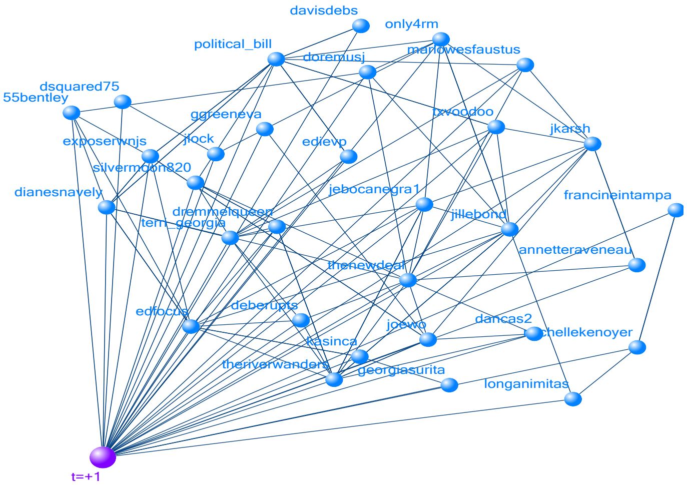


Figure 4.2: The ’supportive’ #iloveobamacare network ($t = +1$) scenario

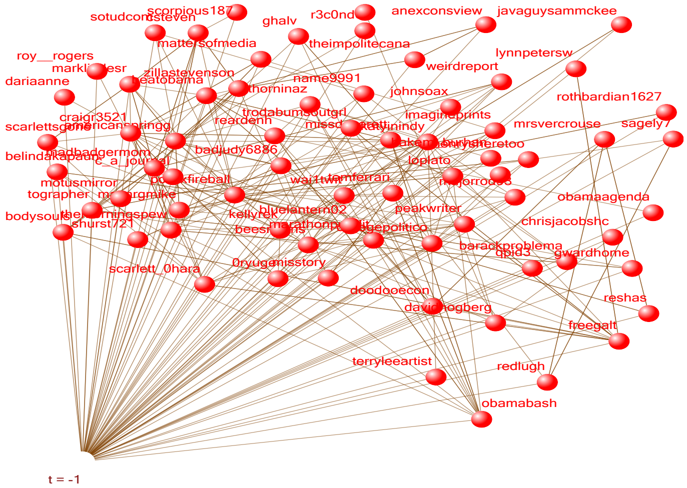


Figure 4.3: The ’anti-supportive’ #iloveobamacare network ($t = -1$) scenario

In Chapter-1 we had used the #iloveobamacare Twitter network (Figure 1.2) to motivate network aided classification. Now, we revisit that example, albeit with the a slightly different approach. As seen in Figure 1.2, the network is composed of two communities of Twitter users who either opposed or supported the hashtag via their tweets. Now, let us consider these communities separately here.

Figure 4.2 represents the community of liberal-minded follower/followee networks of Twitter users, who tweeted in support of the #iloveobamacare hashtag, which was promulgated on Twitter to galvanize more support.

Figure 4.3 represents the follower/followee network of conservative-minded Twitter users who attacked the #`iloveobamacare` hashtag with a series of sharp and sarcastic tweets resulting in what is called **Hashtag-Hijacking** [16].

The national survey conducted by the `Pew Research Center` and `USA TODAY` [38], later confirmed the existence of an underlying sentiment of support ($t = +1$) to the act amongst liberals and opposition ($t = -1$) to the act amongst the conservatives, represented by the pseudo-nodes labeled $t = +1$ and $t = -1$ in Figure 4.2 and Figure 4.3 respectively.

The goal would be for an automatic sentiment detector to predict t for each network. Of course, in this example, one can do this knowing the political stance of the networks, which is side information. However, automatic detection aims to apply a general method based detector on $\mathbf{y}$ without requiring human intervention through specialized side information. As it later revealed in a national survey conducted by the `Pew Research Center` and `USA TODAY` [38], that 75% of Republican party members opposed the PPACA and believed it would negatively affect the country in the coming years, while 63% of Democrats supported it and thought its impact will be positive.

4.2.3 Trivial sentiment detector

The trivial sentiment detector that does not use the knowledge of either the adjacency matrix A or the other system parameters, θ, ε and γ is defined as,

$$\hat{t} = \begin{cases} +1, & \mathbf{e}^T\mathbf{y} \geq \mathbf{0} \\ -1, & \mathbf{e}^T\mathbf{y} < \mathbf{0}, \end{cases} \tag{4.5}$$

where $\mathbf{e}$ is the vector of all ones. As we will see in Section 4.4, the performance of this trivial estimator is still good enough to result in a positive error exponent due to the strong underlying label dependencies.

In the next section, we analyze the performance of this trivial sentiment detector (4.5). We perform a communications-inspired analysis of the probability of error of the detector, using which, we seek to understand the role played by the underlying network topology in the performance of the detector.

4.3 Communications-inspired analysis for the trivial detector

In this section, we perform an analysis of the error probability of the latent sentiment detector (4.5). By the symmetry of the model, the error probability is,

$$\begin{aligned} P_e &= P_{e|t=-1} \\ &= P(\mathbf{e}^T\mathbf{y} \geq 0 | t = -1) \\ &= \sum_{\mathbf{y}} p(\mathbf{y}|t=-1)\, \mathbf{I}[[\mathbf{e}^T\mathbf{y} \geq 0]]. \end{aligned} \tag{4.6}$$

As seen, the exponential sum over $\mathbf{y} \in \{-1, +1\}^n$ makes it infeasible to calculate P_e for large social networks. So, in the next subsection, we present an upper bound for P_e.

4.3.1 P_e upper bound

The main result of this chapter is the following theorem.

Theorem 4.3.1. *For the trivial detector* (4.5)*, an upper bound on the error probability P_e is,*

$$\begin{aligned} P_{e,UB} &= \frac{1}{Z(\theta,\gamma)(\cosh(\varepsilon))^n} \min_b A(b), \quad \textit{where,} \\ A(b) &= \left(\frac{\cosh(2b) + \cosh(2\varepsilon)}{2}\right)^{n/2} Z(\theta, \beta), \\ \beta &= \gamma + \frac{1}{2}\log\left(\frac{\cosh(b-\varepsilon)}{\cosh(b+\varepsilon)}\right), \\ Z(\theta,\beta) &= \sum_{\mathbf{x}} \exp\left\{\theta\mathbf{x}^T A\mathbf{x} - \beta\mathbf{e}^T\mathbf{x}\right\} \quad \textit{and} \\ Z(\theta,\gamma) &= \sum_{\mathbf{x}} \exp\left\{\theta\mathbf{x}^T A\mathbf{x} - \gamma\mathbf{e}^T\mathbf{x}\right\}. \end{aligned} \tag{4.7}$$

Proof: The proof relies on Information-theoretic analysis. (See Appendix A).

4.3.2 Computation of the upper bound

From (4.7), we see that computation of the upper bound requires computing the partition functions related to the underlying MRF. This is an #P-complete problem in general [111]. However, for certain stylized topologies, such as the complete network (Curie-Weiss prior), the star network, the wheel network and the (closed) chain network, we can compute closed form expressions for the partition function exploiting the geometry.

A note on the Curie-Weiss prior

The Curie-Weiss Ising prior with homogeneous edge-potential θ and constant external field h, is defined on a complete graph and is given by,

$$p_{cw}(\mathbf{x}) = \frac{\exp\left\{\frac{\theta}{n}\sum\limits_{1\leqslant i\leqslant j\leqslant n} x_i x_j + h\sum\limits_{1\leqslant i\leqslant n} x_i\right\}}{Z(\theta,h)}.$$

The $1/n$ scaling of the edge-potential in the Hamiltonian is to ensure that the edge-wise contribution, $\mathcal{E}_{edge} = \frac{\theta}{n}\sum\limits_{1\leqslant i\leqslant j\leqslant n} x_i x_j$ is of the order n [112]. While the assumption that the underlying graph is complete, which in physical systems relates to the assumption of an infinite-range interaction, is possibly unphysical, it remains a heavily investigated toy model of choice in statistical physics [112, 113]. The symmetry in the Hamiltonian not only allows one to derive closed form expressions for the partition function and related quantities such as the free entropy density (to be defined in the upcoming section), but also allows one to demonstrate a phase transition in the average magnetization at $\theta_c = 1$ (called the *critical potential*) [113], where the average magnetization is defined as,

$$M(\theta,h) = \lim_{n\to\infty}\left[\sum_{\mathbf{x}} p(\mathbf{x})m(\mathbf{x})\right],$$

where $m(\mathbf{x})$ is termed as the instantaneous magnetization, defined as,

$$m(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

In the forthcoming subsection on error exponent (α) variation with respect to θ, we will show that there is a sharp increase in α when θ is increased beyond the critical edge potential $\theta_c = 1$.

Topology	Partition function $Z(\theta, \gamma)$
Empty (iid)	$(2\cosh(\gamma))^n$
Star	$\exp(\gamma)\,(2\cosh(\theta+\gamma))^{n-1} + \exp(-\gamma)\,(2\cosh(\theta-\gamma))^{n-1}$
Chain	$\lambda_+^n + \lambda_-^n$
Wheel	$\exp(\gamma)\, Z_{chain}^{(n-1)}(\theta, \theta+\gamma) + \exp(-\gamma)\, Z_{chain}^{(n-1)}(\theta, \theta-\gamma)$
Curie-Weiss	$\sum_{m=0}^{n} \left(\begin{array}{c} n \\ m \end{array} \right) {u_c}^{(n-2m)^2-n} {v_c}^{(n-2m)}$

Table 4.1: Table detailing the partition functions for various graph topologies

Now, we present the closed form expressions for the partition functions for these topologies and list them in Table 4.1. The constants used in the table 4.1 are defined as follows.

$$u = \exp(\theta)\,,\; v = \exp(\gamma)$$

$$u_c = \exp\left(\frac{\theta}{2n}\right),\; v_c = \exp(\gamma) \text{ and}$$

$$\lambda_{\pm} = \exp(\theta)\left\{\cosh(\gamma) \pm \sqrt{\sinh^2(\gamma) + \exp(-4\theta)}\right\}. \tag{4.8}$$

Also, $Z_{chain}^{(n-1)}(\theta, \theta \pm \gamma)$ refers to the partition function of the chain graph with $n-1$ nodes, constant edge-potential θ and constant external field being set to $(\theta \pm \gamma)$.

Thus, the error probability upper bound (4.7) can be evaluated for the above network topologies, to provide insight into the impact of social network structure on the sentiment detector performance. Note that, via (4.7), we have reduced the complicated problem of computing an error probability to a problem of calculating an MRF partition function. The partition function calculation is a well researched problem in MRF theory [74], [114] [115], and significant effort has been expended in statistical physics and machine learning to compute it for a variety of graphs. Thus, our theorem facilitates importing ideas from that literature to obtain the error probability bound for a variety of graphs.

4.3.3 Error exponent

For large networks, an error exponent can be defined as,

$$\alpha = \liminf_{n\to\infty} \left\{ \frac{-\log P_e}{n} \right\}. \tag{4.9}$$

Using the bound (4.7), we can show that,

$$\begin{array}{l} \alpha \geq \log(\cosh(\varepsilon)) + C(\theta,\gamma) \\ \quad -\min_{b} \left[\frac{1}{2}\log\left(\frac{\cosh(2b)+\cosh(2\varepsilon)}{2}\right) + C(\theta,\beta)\right], \text{ where} \end{array} . \tag{4.10}$$

$$\begin{aligned}
\beta &= \gamma + \frac{1}{2}\log\left(\frac{\cosh(b-\varepsilon)}{\cosh(b+\varepsilon)}\right) \quad \text{and,} \\
C(\theta,\beta) &= \limsup_{n\to\infty} \frac{1}{n}\log\left(\sum_{\mathbf{x}} \exp\left\{\theta \mathbf{x}^T A \mathbf{x} - \beta \mathbf{e}^T \mathbf{x}\right\}\right), \\
C(\theta,\gamma) &= \liminf_{n\to\infty} \frac{1}{n}\log\left(\sum_{\mathbf{x}} \exp\left\{\theta \mathbf{x}^T A \mathbf{x} - \gamma \mathbf{e}^T \mathbf{x}\right\}\right),
\end{aligned}$$

are limits of the logarithm of partition ('log-partition') functions. This error exponent allows approximate bounding of the error probability at large n by,

$$P_e \gtrsim \exp[\alpha n].$$

4.3.4 Computation of the error exponent

The limit,

$$C(\theta,h) = \lim_{n\to\infty} \left[\frac{\log\left(Z(\theta,h)\right)}{n}\right],$$

used in (4.10) has a special physical significance in statistical physics [113] and is termed as the *Free Entropy density* of the Ising model.

Now, using the closed form expression for the partition functions listed in 4.1, we can compute this *Free entropy density* for these topologies. Table 4.2 enlists the computed free entropy densities. In Table 4.2, we use $\mathcal{H}(p)$ to denote the Shannon-entropy function defined as,

$$\mathcal{H}\left(p\right) = -p\log p - (1-p)\log(1-p).$$

Topology	Free Entropy Density $C(\theta, \gamma)$
Empty (iid)	$\log\left(2\cosh(\gamma)\right)$
Star	$\log\left(2\cosh\left(\theta+\gamma\right)\right)$
Chain	$\theta + \log\left(\cosh\left(\gamma\right) + \sqrt{\sinh^2\left(\gamma\right) + \exp\left(-4\theta\right)}\right)$
Wheel	$\theta + \log\left(\cosh\left(\theta+\gamma\right) + \sqrt{\sinh^2\left(\theta+\gamma\right) + \exp\left(-4\theta\right)}\right)$
Curie-Weiss	$\sup_{m\in\{-1,+1\}}\left[\gamma m + \frac{\theta m^2}{2} + \mathcal{H}\left(\frac{1+m}{2}\right)\right]$

Table 4.2: Table detailing the free entropy density for various graph topologies

Computing α_{iid}

When the network is absent (i.i.d scenario), we can still get a positive error exponent. Using $C_{iid}(\theta, h) = \log(2cosh(h))$ in (4.10), setting the derivative with respect b to 0 and solving for b, we get,

$$\underset{b\geqslant 0}{\operatorname{argmin}}\left[C(\theta,\beta) + \frac{1}{2}\log\left(\cosh(b+\varepsilon)\cosh(b-\varepsilon)\right)\right] = \frac{1}{2}\log\left[\frac{\cosh\left(\varepsilon+\gamma\right)}{\cosh\left(\varepsilon-\gamma\right)}\right] = b_{iid}. \tag{4.11}$$

Now, using (4.11) in (4.10), we get,

$$\alpha_{iid} = \log(\cosh(\varepsilon)) + \log\left(\frac{\cosh\left(\gamma\right)}{\cosh\left(\beta_{iid}\right)}\right) - \left[\frac{1}{2}\log\left(\cosh(b_{iid}+\varepsilon)\cosh(b_{iid}-\varepsilon)\right)\right], \tag{4.12}$$

where $\beta_{iid} = \gamma - \frac{1}{2}\log\left(\frac{\cosh(b_{iid}+\varepsilon)}{\cosh(b_{iid}-\varepsilon)}\right)$. Thus, we can use (4.12) as a benchmark to compare error exponents obtained for the various network topologies via evaluation of (4.10) which is considered in the following subsection.

In the following section, we perform these comparisons by plotting the variation of the error exponent α derived in (4.10) and (4.12) with respect to the model parameters, θ, γ and ε.

4.4 Numerical results: Error exponent of different networks

The aim of this section is to answer the main question raised in the Introduction, that is, demonstrate the effect that the underlying social network structure has on the performance of the trivial sentiment detector which is oblivious of its presence.

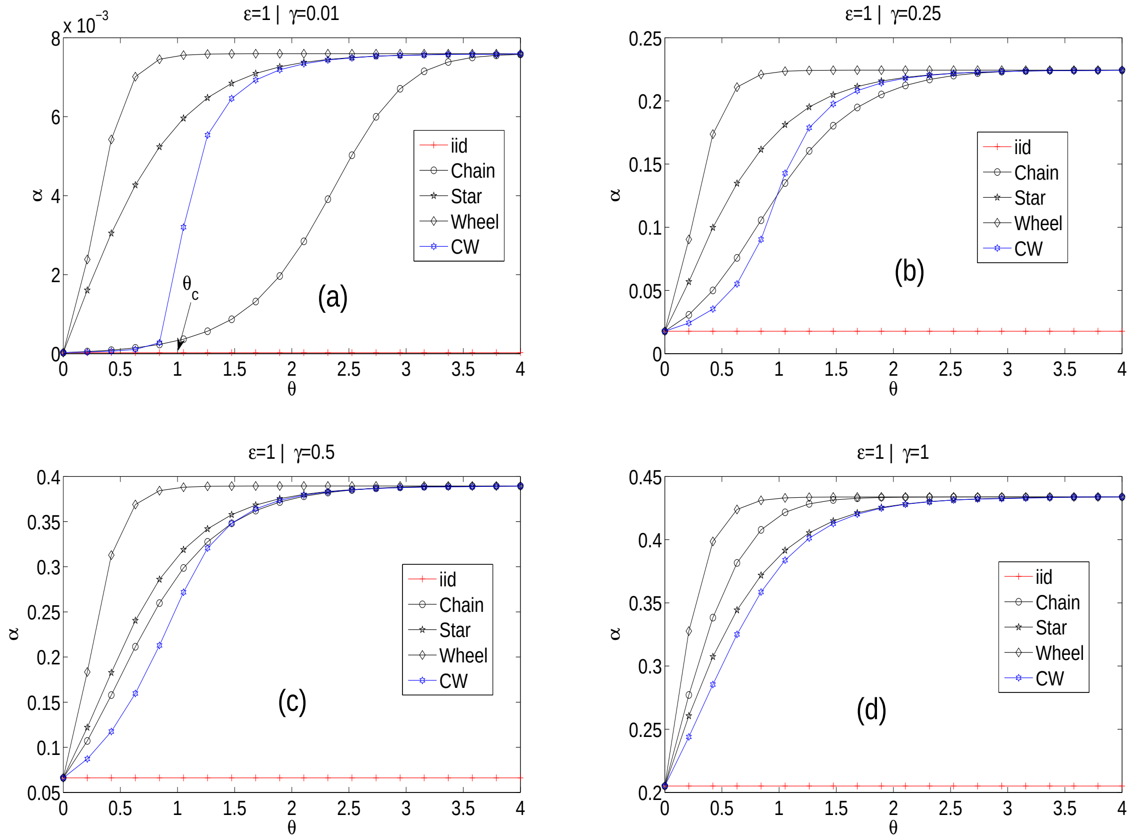


Figure 4.4: Variation of error exponent (α) with θ for different network topologies

4.4.1 Variation of error exponent (α) with edge-potential θ for different network topologies

In Figure 4.4, we set $\epsilon = 1$ (which maps to flipping probability of ~ 0.12 for the BSC channels) and plotted the variation of the of error exponent (α) with respect to the edge-potential θ for different network topologies. Figure 4.4(a) through Figure 4.4(d) vary in terms of the parameter γ being increased from 0.01 (weak latent sentiment) to 1 (strong latent sentiment).

In all the four subplots, we see that the error exponent curve for the empty network case (with legend 'iid'), expectedly remains flat for all values of the edge-potential θ. For all other topologies there is a monotonic rise in α with increasing θ.

In Figure 4.4(a), we see that when γ was set to 0.01 (weak latent sentiment), we can see that sharp rise in the error exponent curve of the Curie-Weiss prior around the *critical* edge-potential $\theta_c = 1$.

It is also rather interesting to note that while the star and the chain both have the same edge density ($n-1$ edges), the chain graph's curve is the one that increases much slower with increase in edge potential. The wheel graph on account of having greater edge-density ($2n-3$ edges) brings in stronger network effect and hence α increases fastest amongst all other topologies considered. It is be noted here that while the Curie-Weiss prior does entail the graph being complete (and hence the most dense topology), the $(1/n)$ scaling in its Hamiltonian ensures that the increase in α remains sluggish when compared to the wheel graph.

Finally, we see that in Figure 4.4(d) when $\gamma = 1$ and $\varepsilon = 1$ (indicating low noise and strong latent sentiment), the error exponent for all topologies increases quickly towards $log(cosh(\varepsilon)) = 0.44$.

4.4.2 Variation of error exponent (α) with latent sentiment strength γ for different network topologies

In Figure 4.5, we again set $\varepsilon = 1$ (which maps to flipping probability of ~ 0.12 for the BSC channels) and plotted the variation of the of error exponent (α) with respect to the strength of the latent sentiment γ, with the parameter θ being increased from 0.01 (weak network effect) to 1.5 (strong network effect) in Figure 4.5(a) through Figure 4.5(d).

In all the four subplots, we see that the error exponent curve for the empty network case (with legend 'iid'), expectedly grows at the same rate with respect to γ irrespective of the edge-potential θ.

We also observe that all topologies exhibit a monotonic rise in α with increasing γ.

In Figure 4.5(a), we see that when θ was set to 0.01 (weak network effect), all the curves are close to that of the *i.i.d curve*. As we increase θ, the curves of α for all topologies are above that of the *i.i.d curve* indicating the ferromagnetic network effect coming from the Ising prior.

It is also rather interesting to note that the curves for star and the chain topologies cross each other at a certain γ. That is, there exists a certain threshold γ above which the chain graph exhibits a higher error exponent compared to the star graph.

Finally, we see that in Figure 4.5(d) when $\theta = 1.5$ and $\varepsilon = 1$ (indicating low noise and strong network effect), the error exponent for all topologies increases quickly towards $log(cosh(\varepsilon)) = 0.44$.

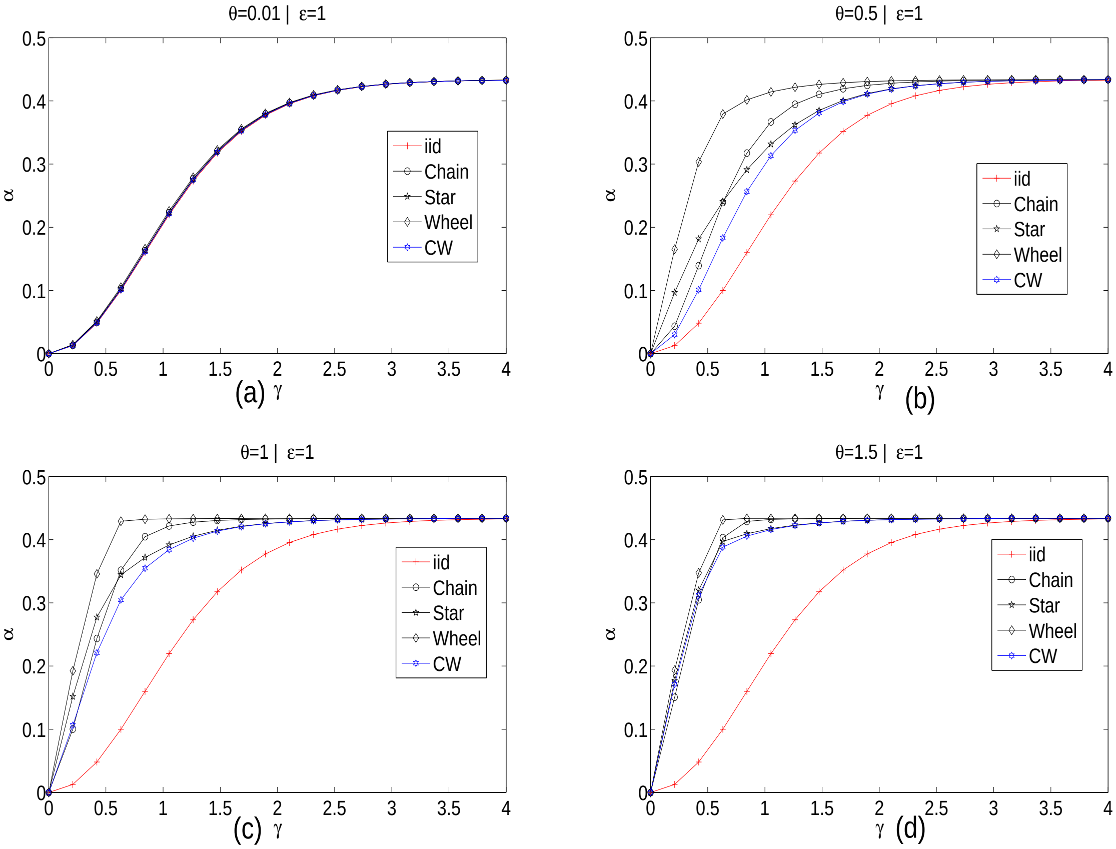


Figure 4.5: Variation of the error exponent (α) with γ for different network topologies

4.4.3 Variation of error exponent (α) with BSC noise level ε for different network topologies

In Figure 4.6, we set $\gamma = 1$ and plotted the variation of the of error exponent (α) with respect to the noise level parameter of the BSC channel(s) ε, with the edge-potential parameter θ being increased from 0.01 (weak network effect) to 1 (strong network effect) in subplots Figure 4.6(a) through Figure 4.6(d). $-\varepsilon$ captures the amount of noise in the BSC in dB (with $\varepsilon \to \infty$ being the zero-noise case). Alternately, ε is the accuracy of the detector used to obtain estimated individual sentiments, $\mathbf{y}$. As expected, increase in ε results in higher α whether the network is present or not.

In Figure 4.6(a), we see that when θ was set to 0.01 (weak network effect), all the curves are close to that of the *i.i.d curve* even at larger ε. As we increase θ, the curves of α for all topologies are

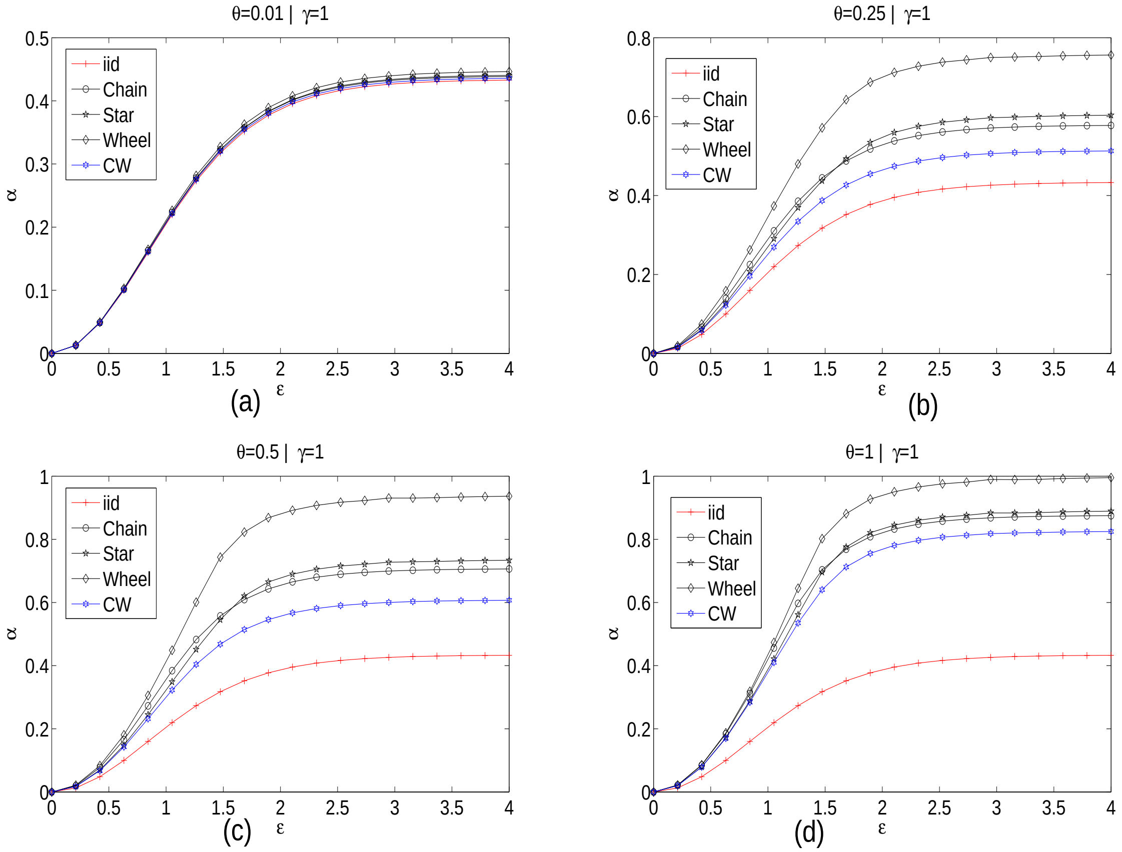


Figure 4.6: Variation of the error exponent (α) with ε for different network topologies

above that of the *i.i.d curve* indicating the harnessing of the network effect coming from the Ising prior.

It is also rather interesting to note that the curves for star and the chain topologies, once again cross each other at a certain ε. This time however, above a certain threshold noise level, ε, above which the star graph exhibits a higher error exponent compared to the chain graph.

Finally, we see that in Figure 4.6(d) when $\theta = 1$ and $\gamma = 1$ (indicating strong latent sentiment and strong network effect), the error exponent for all topologies increases quickly with ε and saturate at different values dependent on the topologies.

4.5 Chapter summary

In this chapter, we have introduced a novel communications-inspired framework for analyzing probability of error of a trivial latent sentiment detector in Online Social Networks. Through this, we have attempted to provide insight into the role played by the network, specifically the topology, in lowering the probability of error of detection, thereby rigorously characterizing the worth of the network as a statistical prior. Firstly, we motivate the practical scenarios where the model is applicable and then provide an analysis of the upper bound on the probability of error, or equivalently, the error exponent for large networks. Finally, we plot the variation of this error exponent with respect to model parameters for the complete network, wheel network, star network and closed chain network topologies, and show the improvement in performance relative to the scenario where the network is absent.

Now, in the upcoming chapter, we move on to the special case of NAD when $\gamma = 0$, that is, there is no latent sentiment but the onus is on detecting the *majority sentiment*.

Chapter 5

Majority sentiment detection

5.1 Introduction

In the previous chapter, we introduced the Network Aided Detection (NAD) framework and analyzed the problem of latent sentiment detection. In this chapter, we consider a special case where $\gamma = 0$ in (1.8), that is, there is no *latent sentiment* but the onus is on detecting the *majority sentiment*. We'd like to emphasize that while this remains a binary detection problem, the technical challenges here are quite different compared to the ones encountered in latent sentiment detection.

With this mind, let us first begin by reviewing the work that has been done under the context of majority sentiment detection in literature.

5.1.1 Majority sentiment detection: Literature review

The study of the majority sentiment prevailing in a network of individuals or the society at large has been an active field of research for sociologists, political scientists, financial analysts and machine learners alike. Research has been carried out in this area under different banners such as public opinion studies [116–119], market sentiment analysis [120], Voting theory [121] and Opinion mining[1] [122].

[1] Keeping in mind the richness of jargon across the fields, the terms: *vote* and *sentiment*, both point towards the same quantum of *opinion* expressed and the three are used interchangeably throughout this chapter. The ambiguity

Estimating the majority *state* also emerges as natural computational problem in Epidemiological studies [123] and in diagnosis of multiprocessor systems [124].

There exists a vast body of literature on Majority sentiment analysis in social networks under the social dynamics/consensus framework [66, 125, 126]. These analyses assume a distributed system of n entities (or nodes) that exchange messages locally according to a pre-devised *consensus protocol* or *population protocol* [127] (synchronously or asynchronously) under the communication constraints imposed by the underlying graph. After every round of communication, the nodes are expected to *update* their opinions based on the message(s) exchanged and the goal here is to understand the nature of convergence of the nodes to a common *majority opinion* and its dependence of the underlying topology of the network.

In the majority dynamics setting [125], after each iteration of the dynamics, each node, $i \in \{V\}$, sets its vote to be the most popular vote (majority vote) among its first order neighbors in the previous iteration.

Similarly, in the classical *DeGroot model* [126], every nodes express its opinion as a real number which is evaluated at each iteration by taking the average of the opinions of its neighbors from the previous iteration. It has been shown in [128] that the nodes will all converge to the same number, which would then serve as a good approximation of the average of the initial opinions when the node-degrees are low.

In sociological research, as succinctly stated in [129], *Majority sentiment is the dark matter of the sociological universe. Because it is amorphous, it is difficult to grasp; yet it exerts a profound influence upon the selection of problems for research, and therefore upon the character of sociological knowledge*.

From the point of view of financial analysis, there exists a vast body of literature dedicated to understanding and guaging the prevalent mood exuded by the financial investors with regard to a particular security or the larger financial market being invested in [120]. The flagship problem here is *market sentiment detection*, which simply put, entails detecting the direction of swing of the crowd psychology existent amongst the investors by sifting through the activity records and price movements of the securities being traded. It targets accurately classifying the market sentiment as being either *bullish* (in the case of upward movement of the prices associated) or *bearish* (in the case of downward movement) and finally, utilizing this information to make informed buying

is clarified from the application and context in which it is used.

and selling decisions [130].

Similarly, judicial scholars have conducted rigorous studies to analyze the relationship between the prevailing majority public sentiment and the judicial selection process [131].

Public policy think tanks such as the Pew Research Center [132], Rand Corporation [133] and the Brookings Institution [134] regularly undertake public opinion polling surveys in order to ascertain the proportion of a population that holds a specific viewpoint which can be crucial in assisting informed inter-cultural dialogue, dismantling of stereotypes and better policy design.

With this background, we now turn our attention to a disruptive new paradigm of social engagement of Online Social Networks (OSNs) that has had profound effects on both the formation as well as detection of the majority sentiment on various social, commercial and political issues.

In the previous chapter, we surveyed the approaches that researchers have undertaken for tweet level sentiment detection in OSNs. With this background in mind, we now focus on the problem set up of majority sentiment detection in the upcoming section.

5.1.2 Problem setup

In many real world scenarios using, for example, Twitter, a *tweet corpus* is used for estimating the global majority sentiment prevalent rather tweet level sentiments. For example, the social media campaign manager of a corporate house who is running a *hashtag* driven product campaign is more worried about the general sentiment surrounding the campaign rather than specific tweet level sentiments of the users of the hashtag. Motivated by this and other similar applications mentioned is section 5.1.1, in this chapter, we focus the majority sentiment detection problem in the network aided detection setting.

For this, we reuse the communication-theoretic approach used for latent sentiment detection, viewing the binary majority sentiment as a bit transmitted via a *weak channel code*, that is the underlying social network. Specifically, this entails modeling the true sentiment vector $\mathbf{x}$ as being a *codeword* sampled from a statistical prior ($p(\mathbf{x})$), resulting in majority sentiment, defined as,

$$m = sign(\mathbf{e}^T\mathbf{x}), \tag{5.1}$$

where $\mathbf{e}$ is the $n \times 1$ vector if 1's.

To this end, harnessing the underlying network as a statistical prior merits using the Maximum A posteriori Probability (MAP) detection framework over the Maximum Likelihood (ML) framework that assumes a uniform prior for $\mathbf{x}$. Simply put, the primary motivation behind using MAP detection is to exploit the underlying *network effect* and lower detection error rates.

In developing this MAP framework, we are influenced by two important observations that inspire us towards focusing on a specific regime of operation where this network aided detection paradigm makes practical sense.

Firstly, there exists a vast body of literature emanating from Social Networks Analysis (SNA) ([135, 136]) that characterize OSNs such as Twitter to be dominated by opportunistic low cost *mildly homophilic weak ties*, where the strength of the tie defined in the *Granovetter sense* [2] [137].

Secondly, as evinced in [109,138], the best of the state of the art machine learning tweet level sentiment classifiers typically provide binary classification error rates of about 25% to 35% which is quite high when benchmarked with the typical eror rates seen in the communications literature.

The combination of the two above stated observations inspired us to fine tune the MAP framework to operate in what we term as the Noisy-Weak network effect (NW) regime. One could also argue that if the tweet level sentiment classifiers were indeed of high-accuracy and if the network effect was really strong, a trivial detector which would just assign the majority sentiment amongst the estimated sentiments, $\mathbf{y}$, would suffice to provide a high accuracy estimate of the true majority sentiment, thus rendering the problem quite trivial to solve.

The rest of the chapter is organized as follows.

In Section 5.2, we introduce the Ising prior based model for majority sentiment detection and describe the trivial,ML and the MAP) detectors.

In Section 5.3, we motivate the relevance of the Positive part Partition function (PPF) computation for MAP detection and derive the High Temperature (HT) expansion form using a novel code puncturing approach.

In Section 5.4 , we derive an approximate MAP detector for the realistic weak network high noise scenario using second order Taylor series approximation.

In Section 5.5 , we present numerical results comparing the probability of error of majority sen-

[2] Granovetter in [137] defines strength of a social tie as a measure that captures *"a combination of the amount of time, the emotional intensity, the intimacy (mutual confiding), and the reciprocal services which characterize the tie"* .

timent detection for the trivial , Maximum Likelihood (ML) and the approximate MAP detector proposed in Section 5.4 under two settings. In the first setting we have a constant noise level across all the nodes, and in the second scenario, the noise levels are chosen based on the node degrees. Via these numerical simulations, we demonstrate the superiority of the approximate MAP detector over the ML and trivial detectors in both the settings.

In Section 5.6, we derive the HT spanning subgraphs expression for a modified Ising model with a strictly edge-wise energy function which is defined on an appended graph with one extra node. This appended graph is constructed by introducing an extra pseudo-node which connects to all the existent n nodes and serves to *absorb* the node-potentials into edge-potentials on the new edges thus formed. Using this framework, we provide a geometric interpretation of the Taylor series approximation of the MAP decision statistic derived in Section 5.4 which provides some interesting intuitive insights.

In Section 5.7 , we cover several interesting applications of the HT expansion framework derived in 5.3 and in 5.6. This includes deriving closed form expressions for the exact probability of error of the trivial detector for various stylized topologies and addressing the problem of generalized majority detection. We also show that the modified HT expression leads to an alternate lattice path sum interpretation of the subgraph weight contributions while also providing for interesting connections between the spanning subgraph weights and Super-Catalan numbers and Krawchuk polynomials. We recap the main contributions and conclude the chapter in Section 5.8 .

5.2 Model for majority vote detection

The model considered in this chapter is as shown in Figure 5.1. As in [139], the social network is modeled as an undirected graph $G(V, E)$ characterized by its symmetric adjacency matrix A. It may be obtained using the follower/followee relationships, or in some cases, using the @-mentions in the tweets [25]. Here, $V = \{1, ..., n\}$ is the vertex set and E is the edge-set with cardinality, $|E| = M$. We assume n is odd to avoid the ambiguous case of equal positive and negative sentiments which might to lead to the majority sentiment variable $m = 0$. Although the social graph in OSNs such as Twitter is directed, we ignore the directionality of the edges given that we will use the underlying social graph to capture statistical correlation, rather than

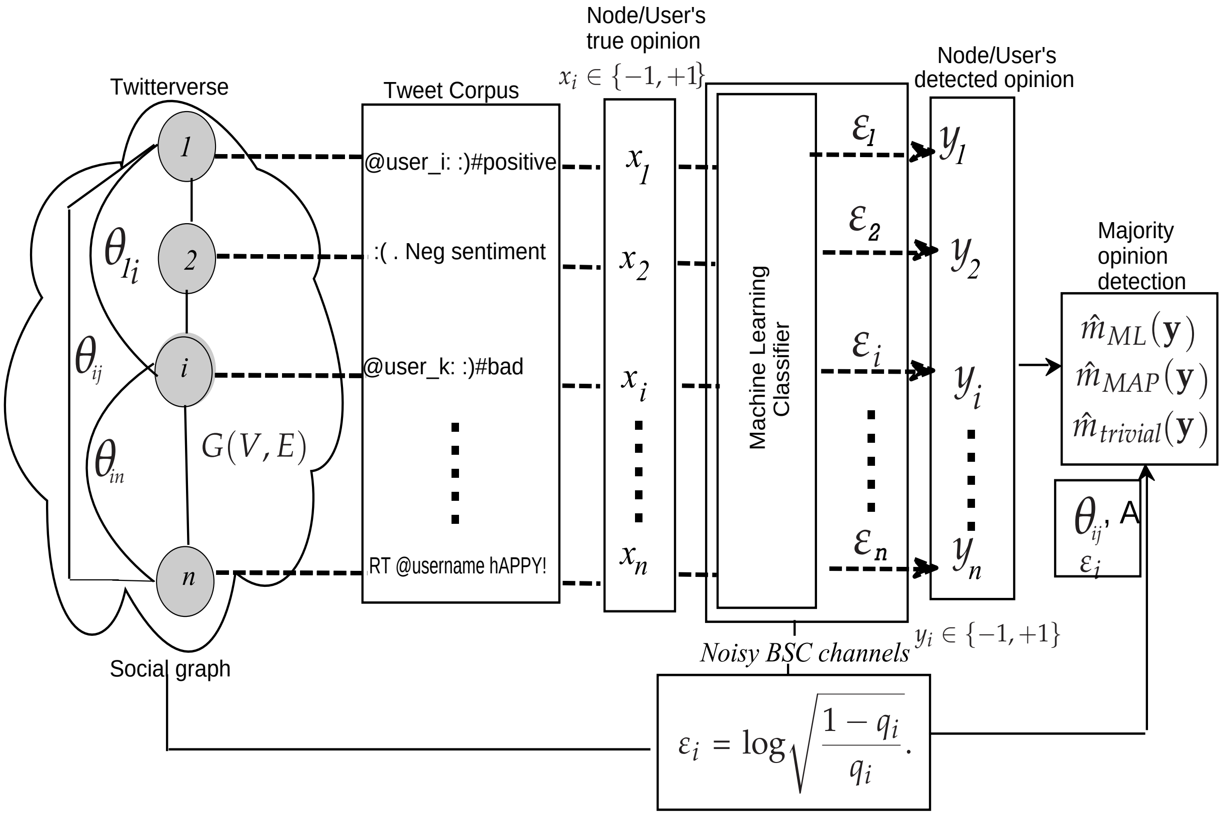


Figure 5.1: The majority vote/opinion model

influence flows.

5.2.1 The non-homogeneous Ising prior

We begin with a more generalized non-homogeneous Ising prior model characterized by the following probability mass function (pmf),

$$p(\mathbf{x}) = \frac{\exp\left\{\sum_{(i,j)\in E} \theta_{ij} x_i x_j\right\}}{Z(\boldsymbol{\theta})}. \tag{5.2}$$

Here, $Z(\boldsymbol{\theta})$ is the so called partition function (PF), which is,

$$Z(\boldsymbol{\theta}) = \sum_{\mathbf{x}\in\{-1,+1\}^n} \exp\left\{\sum_{(i,j)\in E} \theta_{ij} x_i x_j\right\}. \tag{5.3}$$

We still retain the $\theta_{ij} > 0$ ferromagnetic assumption where the neighboring nodes are positively correlated with each other, so that the probability distribution is biased towards configurations with similar values on the nodes of the graph.

The sole reason for allowing this richer specification of the edge-potentials is to facilitate solving the problem of Positive Part Partition function computation (that we will soon encounter) in a more generalized setting which might have important ramifications beyond the specific majority sentiment detection context in which it is being considered.

5.2.2 Revisiting the BSC model for the machine learning classifiers

As in Chapter 4, we model the output of the tweet level machine learning classifier, $\mathbf{y}$, in a classical communication theoretic sense as the output of n-independent Binary Symmetric Channels (BSCs) with true sentiment vector $\mathbf{x}$ as the input, and whose noise level is decided by the classification accuracy of the machine learning classifier. This modeling choice thus provides for a simple way to *plug in* the tweet level machine learning classifier of choice in to the MAP detection framework.

Again, assuming that each of the n-independent Binary Symmetric Channels (BSCs) are characterized by a cross-over (bit-flip) probability q_i, we define ε_i, in terms of q_i as follows,

$$\varepsilon_i = \log\sqrt{\frac{1-q_i}{q_i}}. \tag{5.4}$$

5.2.3 The joint Bayesian modeling of x and y

Using (5.4), we can write the conditional probability $p(y_i|x_i)$ in the exponential form as follows,

$$p(y_i|x_i) = \frac{\exp\{\varepsilon_i y_i x_i\}}{(2\cosh(\varepsilon_i))}. \tag{5.5}$$

Given the conditional independence relationship between the observed vector $\mathbf{y}$ given the input vector $\mathbf{x}$, ($p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} p(y_i|x_i)$), we can write the joint distribution of $\mathbf{x}$ and $\mathbf{y}$ as,

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y}|\mathbf{x}) = \frac{\exp\left\{\sum\limits_{(i,j)\in E} \theta_{ij} x_i x_j\right\}}{Z(\boldsymbol{\theta})} \frac{\exp\left\{\mathbf{h}^T\mathbf{x}\right\}}{\left(2^n \prod\limits_{i=1}^{n} \cosh(\varepsilon_i)\right)}, \tag{5.6}$$

where $h_i = \varepsilon_i y_i, 1 \leq i \leq n$. Note that the main goal behind expressing the BSC conditional probability in the exponential form is to incorporate the observed vector as the external field vector of the Ising model ($\mathbf{h}$). It is also instructive to note that this model in (5.6) is a special case of the general model we introduced in (1.8) with $\gamma = 0$.

For simplicity, as stated earlier, we emphasize that the number of nodes, n, is assumed to be odd so that there are no ties ($\mathbf{e}^T\mathbf{x} = 0$), thereby making the majority sentiment strictly binary.

5.2.4 Majority sentiment detectors

In this subsection, we describe the three majority sentiment detectors used in this chapter.

Trivial detector:

The trivial majority vote detector is one that declares the majority vote to be the sign of the sum of the noisy estimates $\mathbf{y}$. That is,

$$\hat{m}_{trivial}(\mathbf{y}) = \begin{cases} +1 & \text{if } \mathbf{e}^T\mathbf{y} > 0 \\ -1 & \text{otherwise.} \end{cases} \tag{5.7}$$

MAP detector:

The MAP detector is defined as,

$$\hat{m}_{MAP}(\mathbf{y}) = \arg \max_{m\in\{-1,+1\}} \left[p(m|\mathbf{y})\right].$$

It is known ([140]) that the MAP detector maximizes the probability of correct decision for each observation $\mathbf{y}$ compared to any other detector, say ($\mathcal{D}$), and $\forall \mathbf{y} \in \{-1, +1\}^n$. That is,

$$p(\hat{m}_{MAP}(\mathbf{y}) = m|\mathbf{y}) \geqslant p(\hat{m}_{\mathcal{D}}(\mathbf{y}) = m|\mathbf{y}) \;\; \forall(\mathcal{D}, \mathbf{y}).$$

It also maximizes the probability of correct decision averaged over all $\mathbf{y}$ leading to,

$$\sum_{\mathbf{y}} p(y)p(\hat{m}_{MAP}(\mathbf{y}) = m|\mathbf{y}) \geqslant \sum_{\mathbf{y}} p(y)p(\hat{m}_{\mathcal{D}}(\mathbf{y}) = m|\mathbf{y}).$$

Now, let us define the positive part Partition Function (PPF) of an Ising model with edge potential vector, $\boldsymbol{\theta}$, and external field $\mathbf{h}$ as,

$$Z_{+}(\boldsymbol{\theta}, \mathbf{h}) = \sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}>0} \exp\left\{ \sum_{(i,j)\in E} \theta_{ij}x_ix_j + \sum_{(i)\in V} h_ix_i \right\}. \tag{5.8}$$

Now, we see that if $\mathbf{x}$ and $\mathbf{y}$ are jointly distributed according to (5.6), the MAP detector of the majority vote $m = sign(\mathbf{e}^T\mathbf{x})$ given an observation vector $\mathbf{y}$, will take the form,

$$\hat{m}_{MAP}(\mathbf{y}) = \begin{cases} +1 & \text{if } l(\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}) \geq 1 \\ -1 & \text{otherwise,} \end{cases} \tag{5.9}$$

where the MAP decision statistic, $l(\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\varepsilon})$ is the ratio of the *a posteriori* probabilities defined as,

$$l(\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}) = \frac{P\left(\mathbf{e}^T\mathbf{x} > 0|\mathbf{y}\right)}{P\left(\mathbf{e}^T\mathbf{x} < 0|\mathbf{y}\right)} = \frac{P\left(\mathbf{e}^T\mathbf{x} > 0, \mathbf{y}\right)}{P\left(\mathbf{e}^T\mathbf{x} < 0, \mathbf{y}\right)}. \tag{5.10}$$

This can further be simplified as,

$$l(\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}) = \frac{\sum\limits_{\mathbf{x}:\mathbf{e}^T\mathbf{x}>0} \exp\left\{ \sum\limits_{(i,j)\in E} \theta_{ij}x_ix_j + \mathbf{h}^T\mathbf{x} \right\}}{\sum\limits_{\mathbf{x}:\mathbf{e}^T\mathbf{x}<0} \exp\left\{ \sum\limits_{(i,j)\in E} \theta_{ij}x_ix_j + \mathbf{h}^T\mathbf{x} \right\}} = \frac{Z_{+}(\boldsymbol{\theta}, \mathbf{h})}{Z_{+}(\boldsymbol{\theta}, -\mathbf{h})} \tag{5.11}$$

where the external field $\mathbf{h} = \boldsymbol{\varepsilon}.\mathbf{y}$, with $\boldsymbol{\varepsilon}.\mathbf{y}$ being the element-wise Hadamard product of vectors $\boldsymbol{\varepsilon}$ and $\mathbf{y}$.

ML detector:

The ML detector assumes a uniform prior for $\mathbf{x}$ and is a special case of the MAP detector with $\boldsymbol{\theta} = \mathbf{0}$. That is,

$$\hat{m}_{ML}(\mathbf{y}) = \begin{cases} +1 & \text{if } l_{ML}(\mathbf{y}, \boldsymbol{\varepsilon}) \geq 1 \\ -1 & \text{otherwise,} \end{cases} \tag{5.12}$$

where

$$l_{ML}(\mathbf{y},\boldsymbol{\varepsilon}) = l(\mathbf{y},\boldsymbol{\theta} = \mathbf{0},\boldsymbol{\varepsilon}).$$

As seen in (5.11), the MAP detector requires computing the PPF where the external field $\mathbf{h} = \boldsymbol{\varepsilon}.\mathbf{y}$, which turns out to be an extremely challenging computational problem.
In general, for graphs with arbitrary topology, computing the full partition function of an Ising model is #-P complete [111], with recent efforts geared towards finding tight lower and upper bounds for the same [20]. However, there exist no previous efforts in machine learning literature directed towards computing the PPF as defined in (5.8) with external fields $h_i = \varepsilon_i y_i$ having different signs (depending on y_i unless in the trivial case of $\mathbf{y} = \pm\mathbf{e}$). This leads us in to the forthcoming sections where the onus in on tackling the PPF computation problem by firstly coming up with a High Temperature (HT) expansion framework for the PPF and harnessing this to perform Taylor series expansion in the NW regime.

5.3 Positive part Partition function (PPF) and MAP detection

As in seen (5.11), the observations ($\mathbf{y}$) are incorporated as the external field, which implies that computing $l(\mathbf{y},\boldsymbol{\theta},\boldsymbol{\varepsilon})$ entails computing the PPF of the Ising prior, given by,

$$Z_+(\boldsymbol{\theta},\mathbf{h}) = \sum_{\mathbf{x}\in\mathcal{X}_+^{(n)}} \exp\left\{\sum_{(i,j)\in E}\theta_{ij}x_ix_j + \sum_{(i)\in V}h_ix_i\right\}, \tag{5.13}$$

where $\mathcal{X}_+^{(n)} = \left\{\mathbf{x} : \mathbf{x} \in \{-1,+1\}^n, \mathbf{e}^T\mathbf{x} > 0\right\}$ and $h_i = \varepsilon_i y_i$.
In this section, we present the main result of this chapter, which is the high temperature *subgraphs world* representation for the PPF, the motivation for which is as follows .

5.3.1 Motivating the HT framework

In the previous section, we introduced the idea of the natural setting that merits using the MAP detector in the first place, is the Noisy weak network (NW) setting. Here, we begin by showcasing the relevance of the so called High Temperature (HT) expansion framework as the logical starting point for exploring MAP detection in this NW regime.

In statistical physics literature [113], the Ising prior in (5.2) is specified as a special case of the Gibbs-Boltzmann distribution which is,

$$p(\mathbf{x}) = \frac{\exp\left\{-\frac{1}{k_B T}\mathcal{E}(\mathbf{x})\right\}}{Z(T)} \tag{5.14}$$

where T refers to the temperature of the spin system under consideration, $\mathcal{E}(\mathbf{x}) = -\sum_{(i,j)\in E} x_i x_j$ is the energy function of the spins and k_B is the Boltzmann constant. With regards to our model, we see that the relationship between the common edge potential θ and the temperature parameter (T) is simply, $\theta = 1/(k_B T)$. This implies that the high temperature regime of $T >> 1$ relates to the $\theta \approx 0$ (weak network) scenario. Analysis of the partition function in this HT regime usually entails transforming the summation over the 2^n possible spins into a geometrical summation over all possible sub-graphs using the so called *High temperature* or *character expansion* identity, which is, $e^a = cosh(a)(1 + tanh(a))$, and then performing Taylor series expansion followed by ignoring higher order terms, a procedure which is replicated in this chapter.

Having thus motivated the HT expansion framework for the PPF, we now present the main result of this chapter.

5.3.2 Main Result: The HT spanning expansion framework for the PPF

The following theorem provides the main result of the chapter, which is the spanning sub-graphs world representation of the PPF.

Theorem 5.3.1. *The HT expansion expression for the PPF, as defined in* (5.8)*, is given by,*

$$Z_+(\boldsymbol{\theta}, \mathbf{h}) = c(\boldsymbol{\theta}, \mathbf{h})\left[Z_1 + Z_2\right], \tag{5.15}$$

with

$$Z_1 = 2^{n-1} \sum_{S \subseteq E} \left[\prod_{(i,j)\in S} \lambda_{ij} \prod_{i \in odd(V,S)} \mu_i \right], \tag{5.16}$$

and

$$Z_2 = \sum_{S \subseteq E} \sum_{\substack{U \subseteq V \\ |U| \in \left\{\mathbb{Z}_+^{odd}\right\}}} \left(\prod_{(i,j)\in S} \lambda_{ij} \prod_{i\in U} \mu_i \right) w(S, U). \tag{5.17}$$

Here, $\mathbb{Z}_+^{odd} = \{2\mathbb{N}+1\}$ is the set of positive odd integers, $S \subseteq E$ is the edge-set of the spanning sub-graph $G(V,S)$, $\lambda_{ij} = tanh(\theta_{ij})$, $\mu_i = \tanh(h_i)$ and $V_{odd}(S) \subseteq V$ is the subset of odd-degreed nodes in V and $c(\boldsymbol{\theta}, \mathbf{h}) = \left\{ \prod\limits_{(i,j)\in E} \cosh(\theta_{ij}) \right\} \left\{ \prod\limits_{i \in V} \cosh(h_i) \right\}$. The weight $w(S,U)$ is given by,

$$w\left(S,U\right) = \omega\left(n, \left(|odd(U,S)| + |even(V-U,S)|\right)\right), \tag{5.18}$$

where $|odd(U,S)|$ is the number of odd-degreed nodes in the vertex subset, U, with respect to the graph $G(V,S)$, $|even(V-U,S)|$ is the number of even-degreed nodes in the vertex subset, $V-U$, with respect to the graph $G(V,S)$, and the function $\omega(n,p)$ is defined as,

$$\omega\left(n,p\right) = \sum_{i=0}^{\frac{n-1}{2}} (-1)^i \left(\begin{array}{c} n-p \\ i \end{array} \right) \sum_{k=0}^{\left(\frac{n-1}{2}\right)-i} \left(\begin{array}{c} p \\ k \end{array} \right). \tag{5.19}$$

(In (5.19), with regard to the binomial coefficient indexed by two nonnegative integers , a and b, that is, $\binom{a}{b}$, we assume $\binom{a}{b} = 0$ if $b > a$).

Proof. The proof utilizes ideas from coding theory involving calculation of weight distribution of codewords in punctured codebooks and is detailed in Appendix B. We would like to emphasize that the techniques used in this proof are drawn from the classical communication theoretic perspective hitherto unseen in machine learning literature. □

One important insight we gained from this HT expansion framework is that the non-positivity of subgraph weights $w(S,U)$ prohibits the formulation of a probability mass function (pmf) over the subgraphs configuration space. This, in turn, implies that the strategy of sampling spanning subgraphs based configurations instead of spin-configurations leading to a Fully Polynomial Randomized Approximation Scheme (FPRAS) ([141]) for the full ferromagnetic partition function cannot be replicated for the PPF even when the external fields are all of the same sign.

5.4 MAP detection in the Noisy data Weak network (NW) regime with homogeneous edge-potential

Having derived the HT expansion formula for the PPF of a generalized non-homogeneous Ising prior, we now revert back to the homogeneous variant introduced in this thesis in (1.8) in Chapter-

1 with $\theta_{ij} = \theta,\ \forall (i,j) \in E$.

Now using (5.11) and (5.15), we have the MAP decision statistic $l(\theta, \boldsymbol{\varepsilon})$ to be,

$$l\left(\theta, \boldsymbol{\varepsilon}\right) = \frac{Z_1^{'} + Z_2^{'}}{Z_1^{'} - Z_2^{'}}, \tag{5.20}$$

with

$$Z_1^{'} = 2^{n-1} \sum_{S \subseteq E} \left[\lambda^{|S|} \prod_{i \in odd(V,S)} \mu_i \right]; Z_2^{'} = \sum_{S \subseteq E} \sum_{\substack{U \subseteq V \\ |U| \in \left\{ \mathbb{Z}_+^{odd} \right\}}} \lambda^{|S|} \prod_{i \in U} \mu_i w(S, U). \tag{5.21}$$

(Note that we have dropped $\mathbf{y}$ in the argument of the MAP decision statistic $l(\theta, \boldsymbol{\varepsilon})$ for simplicity of notation).

Now, for the NW scenario, we take the second order Taylor series expansion of the MAP decision statistic in the neighborhood of $(\theta, \boldsymbol{\varepsilon}) = (0, \mathbf{0})$. Denoting the partial derivatives by the following notation, $\frac{\partial l(\theta, \boldsymbol{\varepsilon})}{\partial \theta} = l_\theta^1$, $\frac{\partial l(\theta, \boldsymbol{\varepsilon})}{\partial \varepsilon_i} = l_{\varepsilon_i}^1$, $\frac{\partial^2 l(\theta, \boldsymbol{\varepsilon})}{\partial \varepsilon_i^2} = l_{\varepsilon_i^2}^2$, $\frac{\partial^2 l(\theta, \boldsymbol{\varepsilon})}{\partial \theta^2} = l_{\theta^2}^2$ and $\frac{\partial^2 l(\theta \boldsymbol{\varepsilon})}{\partial \varepsilon_i \partial \theta} = l_{\varepsilon_i \theta}^2$, we have,

$$\begin{aligned} \tilde{l}\left(\theta, \boldsymbol{\varepsilon}\right) \approx l\left(0, \mathbf{0}\right) + l_\theta^1\left(0, \mathbf{0}\right) \times \theta + \sum_{i=1}^{n} \left[l_{\varepsilon_i}^1\left(0, \mathbf{0}\right) \times \varepsilon_i \right] \\ + \frac{1}{2} \left[l_{\theta^2}^2\left(0, \mathbf{0}\right) \theta^2 + \sum_{i=1}^{n} \left[l_{\varepsilon_i^2}^2\left(0, \mathbf{0}\right) \times \varepsilon_i^2 \right] \right] \\ + \sum_{i=1}^{n} \sum_{j=1: j \neq i}^{n} l_{\varepsilon_i \varepsilon_j}^2\left(0, \mathbf{0}\right) \varepsilon_i \varepsilon_j + \theta \sum_{i=1}^{n} \left[l_{\varepsilon_i \theta}^2\left(0, \mathbf{0}\right) \times \varepsilon_i \right]. \end{aligned} \tag{5.22}$$

From (5.11), we see that $l\left(0, \mathbf{0}\right) = 1$. Further, performing the required partial differentiations at $(0, \mathbf{0})$, we see that the following partial derivatives are, in fact 0 at $(\theta, \boldsymbol{\varepsilon}) = (0, \mathbf{0})$:

$$l_\theta^1\left(0, \mathbf{0}\right) = 0, l_{\theta^2}^2\left(0, \mathbf{0}\right) = 0, l_{\varepsilon_i^2}^2\left(0, \mathbf{0}\right) = 0, l_{\varepsilon_i \varepsilon_j}^2\left(0, \mathbf{0}\right) = 0. \tag{5.23}$$

The only partial derivatives that are non-zero turn out to be,

$$\begin{aligned} l_{\varepsilon_i}^1\left(0, \mathbf{0}\right) &= \frac{2}{2^{n-1}} y_i \omega\left(n, n-1\right), \\ l_{\varepsilon_i \theta}^2\left(0, \mathbf{0}\right) &= \frac{2}{2^{n-1}} y_i \left\{ \Delta_i \omega\left(n, n-1\right) + (M - \Delta_i) \omega\left(n, n-3\right) \right\}, \end{aligned} \tag{5.24}$$

with Δ_i being the degree of node i and M being the number of edges while $\omega(n, p)$ is as defined in (5.19).

Now, substituting (5.23) and (5.24) in (5.22) and simplifying, we have,

$$\tilde{l}\left(\theta = 0, \boldsymbol{\varepsilon} = \mathbf{0}\right) \approx 1 + \varpi \left[\sum_{i=1}^{n} \left[y_i \varepsilon_i \overbrace{\left(1 + \theta \frac{\Delta_i\left(n-1\right) - M}{n-2} \right)}^{v_i} \right] \right], \tag{5.25}$$

where,

$$\varpi = \frac{2\omega\left(n, n-1\right)}{2^{n-1}} = \frac{(n-1)!}{2^{n-2}\left(\left(\frac{n-1}{2}\right)!\right)^2}. \tag{5.26}$$

Given that ϖ in (5.26) above is positive for $n \geq 1$, we see that the Taylor approximation in (5.25), $\tilde{l}\left(\theta, \boldsymbol{\varepsilon}\right) > 1$ when

$$\left[\sum_{i=1}^{n} y_i v_i\right] > 0. \tag{5.27}$$

That is, the approximate MAP estimate is linear in $\mathbf{y}$ where the node-wise weights v_i can be written as the sum of ε_i and an additive term that captures the contribution coming from the network-effect. That is,

$$v_i = \overbrace{\varepsilon_i}^{Node-effect} + \overbrace{\varepsilon_i \theta \left\{\frac{\Delta_i\left(n-1\right) - M}{n-2}\right\}}^{Network-effect}. \tag{5.28}$$

It is also clear from (5.28) that, besides M, the number of edges in the network (global graph statistic), the approximate MAP detector takes in to consideration only the first order network information (node-degree (Δ_i)). This renders the approximate MAP detector amenable for distributed implementation too.

5.5 Probability of error of majority sentiment detection in the NW regime: Numerical results

In the section, we present simulation results which demonstrate the utility of the approximate MAP detector, which is defined as,

$$\hat{m}_{MAP-TAYLOR}(\mathbf{y}) = \begin{cases} +1 & \text{if } \tilde{l}(\theta, \boldsymbol{\varepsilon}) > 1 \\ -1 & \text{otherwise,} \end{cases} \tag{5.29}$$

in comparison with the Maximum Likelihood (ML) detector which assumes a uniform prior instead of the Ising model, or in other other words, the MAP detector with $\theta = 0$. We specifically focus on the realistic NW regime which was motivated in the previous sections of this chapter.From (5.28), we see that this ML detector takes the form,

$$\hat{m}_{ML}(\mathbf{y}) = \begin{cases} +1 & \text{if } \left[\sum\limits_{i=1}^{n} y_i \varepsilon_i\right] > 0 \\ -1 & \text{otherwise,} \end{cases} \tag{5.30}$$

We consider two scenarios. The first scenario is where we set $\varepsilon_i = \varepsilon$ across all nodes. In this case, the ML detector is the trivial detector in (5.7). In the second scenario, we set ε_i for each node according to its degree (a measure of importance in the network) by,

$$\varepsilon_i = \varepsilon_{avg} \times log(\Delta_i + 1), \tag{5.31}$$

where ε_{avg} is akin to the *average SNR* of the network.

For simulations, we consider the (n_1, n_2)-Lollipop graph (Figure 5.2), which is obtained by joining a complete graph $G_{n_1}^{(complete)}$ with a chain(or path) graph $G_{n_2}^{(chain)}$ via a bridge edge.

These lollipop graphs have been widely studied in areas such as majority consensus analysis [127], random walk convergence time analysis [142, 143] and convergence time for quantized consensus. It has been shown in [144] that lollipop graphs maximize the mean consensus time under the so called *link dynamics* update rules and the authors in [143] show that Lollipop graphs are indeed extremal for commute times. The significance of this topology with regard to complex networks analysis and social networks analysis is that it captures the extremal case of two highly topologically imbalanced communities with one being densely connected and the other being sparse. It has been shown in [145,146], that the street layouts typical of the suburban early sixties in United States were quintessentially *lollipop layouts* and this legacy has had deep ramifications on issues such as safety, transport efficiency and general livability.

With this motivation in mind, we detail the Monte Carlo simulation procedure as follows.

Firstly, we fixed the lollipop graph parameters, $n_1 = 50$ and $n_2 = 951$, (n=1001) and set the common Ising edge-potential parameter to be $\theta = 0.1$. For each choice of $\boldsymbol{\varepsilon}$, $N_s = 10^6$ samples were sampled from,

$$p(\mathbf{x}) = \frac{\exp\left\{0.1 \sum_{(i,j)\in E_{lollipop}} x_i x_j\right\}}{Z(\theta = 0.1)}, \tag{5.32}$$

using the Gibbs sampler in [147] (with Burn-In set at 10^6), and the obtained samples $\{\mathbf{x}^{(s)};\ s = 1, ..., N_s\}$ were then flipped according to (5.4) to obtain the noisy observations $\{\mathbf{y}^{(s)};\ s = 1, ..., N_s\}$. Then, the detectors as defined in (5.7),(5.29) and (5.30), were used to obtain the majority vote estimates $\{\hat{m}^{(s)};\ s = 1, ..., N_s\}$, and finally the mean probability of error ($\bar{P}_{err}$), was evaluated according to,

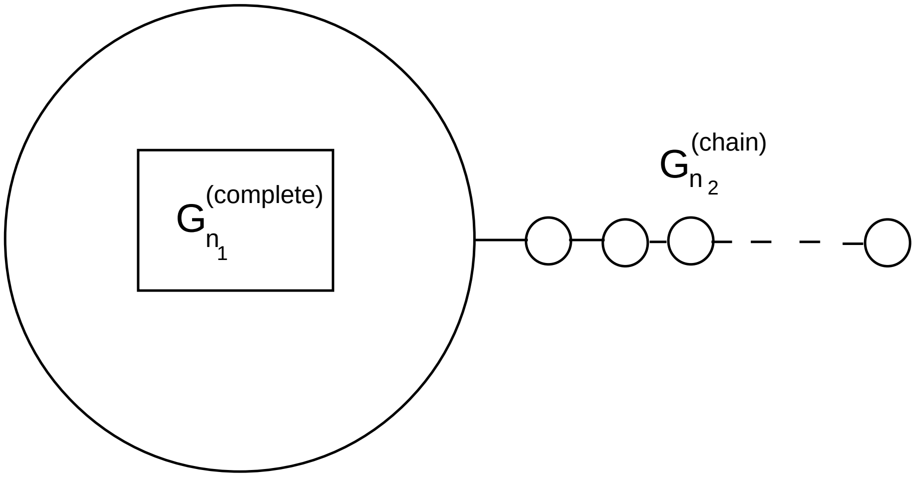


Figure 5.2: The Lollipop graph

$$\bar{P}_{err} = \frac{\sum_{s=1}^{N_s} \mathbf{I}\left[\left[sign\left(\mathbf{e}^T\mathbf{x}^{(s)}\right) \neq \hat{m}^{(s)}\right]\right]}{N_s}, \tag{5.33}$$

where $\mathbf{I}[\![.]\!]$ is the indicator function. The results are as shown in Figure 5.3 and Figure 5.4, capturing the homogeneous $\boldsymbol{\varepsilon}$ and varying $\boldsymbol{\varepsilon}$ cases respectively.

5.5.1 Network prior with constant ε

As seen in Figure 5.3, the constant ε across the nodes was varied from 0 to 0.09, which translated into the bit flip probability of the BSC channels being varied in the high noise regime from 0.5 to 0.46. The trivial detector's $\bar{P}_{err}$ curve (denoted by 'TRIVIAL' in the legend') is close to the flip probability curve with $\bar{P}_{err} = 0.44$ at $\varepsilon = 0.09$.

Now, even in this hostile regime with weak network effect ($\theta = 0.1$), we see that the approximate MAP detector of (5.29) (denoted by 'MAP-Taylor' in the legend') provides a near 13% improvement over the trivial detector and achieves an error rate of 0.32 which is admirable bearing in mind that we fed in the observations emanating from BSCs with bit flip probability of nearly 0.5. This in turn motivates the claim that the underlying network, even when homogeneously parameterized with a weak global $\theta = 0.1$, still propels network aided detection leading

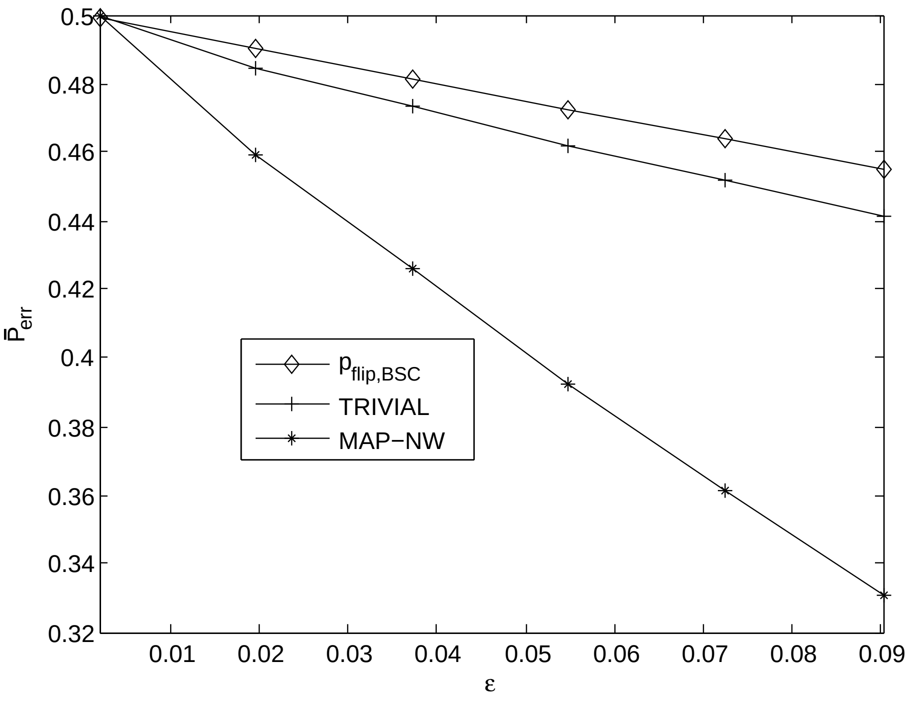


Figure 5.3: Probability of error comparison between the ML (trivial) and the approximate MAP detectors for the 1001 node lollipop graph with $\theta = 0.1$ and fixed ε across the nodes.

to some reasonably useful inference in scenarios where the **GIGO** (Garbage In, Garbage Out) paradigm [148] is expected to hold sway.

5.5.2 Network prior with varying ε_i

For the second scenario where ε_i varies across the nodes (according to (5.31)), and the ML detector has exact knowledge of these ε_i and can weigh the observations according to (5.30), the question remains if the *network-effect* term added in (5.28) makes any difference to the error rate obtained. With respect to Figure 5.4, we see that the *network-effect* term does indeed makes a difference with the approximate MAP detector (which is also linear in **y** like the ML) comfortably outperforming the ML detector with a improvement of nearly 7.5% when $\varepsilon_{avg} = 0.03$.

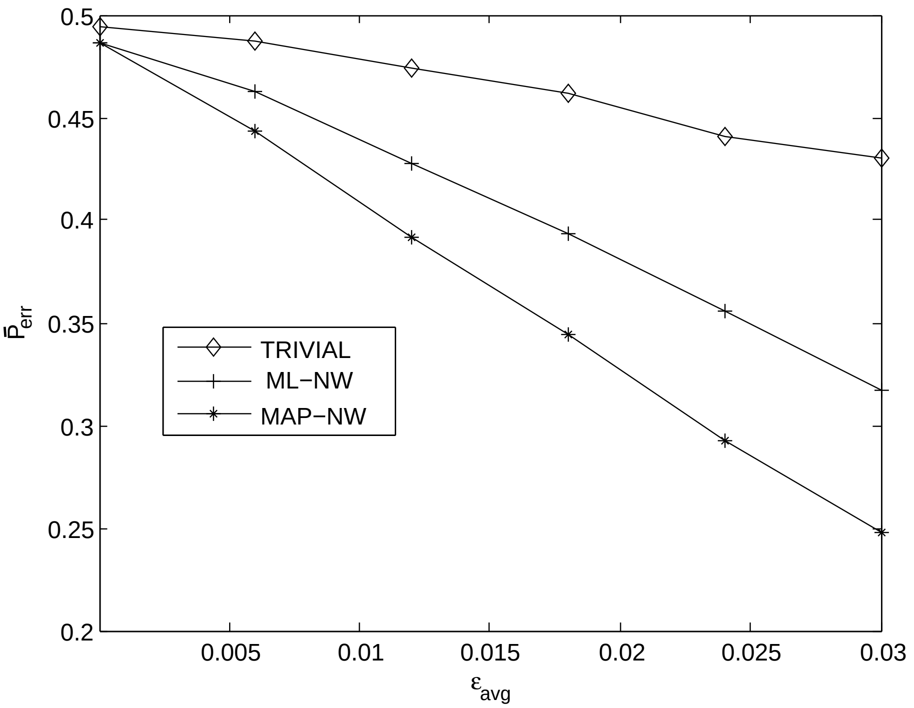


Figure 5.4: Probability of error comparison between the Trivial, ML and the approximate MAP detectors for the 1001 node lollipop graph with $\theta = 0.1$ and $\varepsilon_i = \varepsilon_{avg} \times log(\Delta_i + 1)$.

5.6 Alternate HT expansion expression with appended pseudo-node model

As evinced in related machine literature ([149], [114]), it is often beneficial to rewrite the Ising energy function $\left\{\sum\limits_{(i,j)\in E} \theta_{ij}x_ix_j + \sum\limits_{(i)\in V} \theta_i x_i\right\}$ over n binary variables into its equivalent strictly edge-wise Ising energy function of the form $\left\{\sum\limits_{(i,j)\in E'} \theta'_{ij}x_ix_j\right\}$ by introducing an extra clamped pseudo-node which connects to all the existent n nodes and serves to convert the node-potentials into edge-potentials on the new edges thus formed.

In this section, we re-derive the HT spanning subgraphs expression for this modified Ising model, which in turn, helps in the following four ways. Firstly, it helps convert the double summation over S and U in (5.15) into a single summation over the modified spanning subgraphs, which leads to a simpler single summation form of the HT expansion. Secondly, it helps provide an elegant geometric interpretation of the Taylor series approximation of the MAP decision statistic derived. Thirdly, the simplified expressions make it easier to apply the HT framework to derive closed form expressions for the exact probability of error of the trivial detector for various topologies. And finally, it helps provide an alternate lattice path sum interpretation of the subgraph weight contributions in addition to the Super-Catalan number/Krawchuk polynomial interpretations.

5.6.1 Virtual node appending to absorb the node potentials

To begin with, we *eliminate* the node-potentials in (5.2) by introducing a virtual-node (labeled 0) which is clamped at $+1$ and treating the node potentials (h_i) as edge potentials on the virtual edges connecting this virtual node ($'0'$) to all the existent nodes.
Let $E_{ext} = \{(0,i);\ i \in V\}$, be the set of *external* (or *virtual*) edges connecting the nodes in V with the pseudo-node 0 and the extended edge-set of the pseudo-node appended model be $E' = E \cup E_{ext}$. Similarly, let us define the extended vertex-set to be $V' = V \cup \{0\}$.
The edge-potentials of the appended model would be,

$$\theta^{'}_{ij} = \begin{cases} h_i & \text{if } j = 0 \\ h_j & \text{if } i = 0 \\ \theta_{ij} & \text{otherwise.} \end{cases} \tag{5.34}$$

For the appended model, we have the PPF to be,

$$Z_{+}(\boldsymbol{\theta}, \mathbf{h}) = \sum_{\mathbf{x} \in \mathcal{X}^{(n+1)}_{0,+}} \exp\left\{ \sum_{(i,j) \in E'} \theta^{'}_{ij} x_i x_j \right\}, \tag{5.35}$$

where $\mathcal{X}^{(n+1)}_{0,+} = \left\{ \mathbf{x} : \mathbf{x} \in \{-1,+1\}^{n+1},\ x_0 = 1, \sum\limits_{i=1}^{n} x_i > 0 \right\}$.
This appending is as shown in Figure 5.5.

The following proposition presents the HT spanning sub-graphs world representation of the PPF with the appended pseudo-node .

Proposition 5.6.1. *For the Ising model defined in* (5.2)*, the HT expansion expression for the PPF, as*

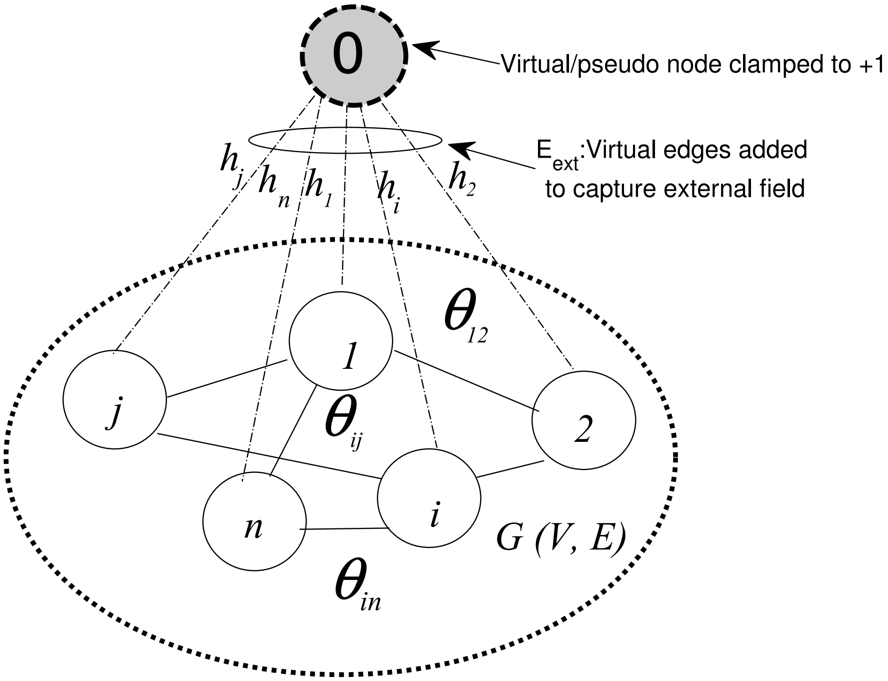

0
Virtual/pseudo node clamped to +1
E_ext:Virtual edges added
to capture external field
h_j
h_n
h_1
h_i
h_2
θ_12
1
j
θ_ij
2
n
i
G (V, E)
θ_in

Figure 5.5: Handling the external fields by using a clamped pseudo node

defined in (5.8)*, is given by,*

$$Z_{+}(\boldsymbol{\theta},\mathbf{h}) = c(\boldsymbol{\theta},\mathbf{h}) \overbrace{\sum_{S'\subseteq E'} \left[\left\{ \prod_{(i,j)\in S'} \lambda_{ij} \right\} \omega(n, n-|V_{odd}(S')|) \right]}^{Z'} . \tag{5.36}$$

Here, $S' \subseteq E'$ is the edge-set of the spanning sub-graph $G(V',E')$, $\lambda_{ij} = tanh(\theta_{ij})$, $V_{odd}(S') \subseteq V$ is the subset of odd-degreed nodes in V (not counting the pseudo-node) in the spanning sub-graph $G(V',S')$ and $\omega(.,.)$ is as defined in (5.19)*.*

Proof. The proof for this proposition entails using the same *code-puncturing* based procedure with some simple algebraic manipulations as for the proof of Theorem-1 and requires no new theoretical techniques or ideas. □

Now, combining (5.19) with the high-temperature spanning subgraphs expansion based expression for the PPF in (5.36), we have,

$$l(\mathbf{y},\boldsymbol{\theta},\boldsymbol{\varepsilon}) = \frac{Z_{+}(\theta,\boldsymbol{\varepsilon}\circ\mathbf{y})}{Z_{+}(\theta,-\boldsymbol{\varepsilon}\circ\mathbf{y})} = \frac{\sum\limits_{S'\subseteq E'}\left[\left\{\prod\limits_{(i,j)\in S'}\lambda^{+}_{ij}\right\}\omega\left(n,n-\left|V_{odd}\left(S'\right)\right|\right)\right]}{\sum\limits_{S'\subseteq E'}\left[\left\{\prod\limits_{(i,j)\in S'}\lambda^{-}_{ij}\right\}\omega\left(n,n-\left|V_{odd}\left(S'\right)\right|\right)\right]}. \tag{5.37}$$

Here,

$$\lambda^{+}_{ij} = \begin{cases} tanh(\varepsilon_i y_i) & \text{if } j = 0 \\ tanh(\varepsilon_j y_j) & \text{if } i = 0 \\ tanh(\theta) & \text{otherwise} \end{cases} \tag{5.38}$$

and

$$\lambda^{-}_{ij} = \begin{cases} -tanh(\varepsilon_i y_i) & \text{if } j = 0 \\ -tanh(\varepsilon_j y_j) & \text{if } i = 0 \\ tanh(\theta) & \text{otherwise} \end{cases}. \tag{5.39}$$

5.7 Applications of the HT expansion of the PPF

5.7.1 Application 1: Geometric interpretation of the approximate MAP detection

From (5.37), we gather that computing $l(\mathbf{y},\boldsymbol{\theta},\boldsymbol{\varepsilon})$ requires computing Z', which is the hard part. So, in this section we approximate Z' in two stages. In the first stage, we split Z' into two summations, termed *internal* and *external* and then look at ways to approximate each of the two.

Splitting Z' into internal and external summations

To begin with, let us write down the expression for Z',

$$Z' = \sum_{S'\subseteq E'}\left[\left\{\prod_{(i,j)\in S'}\lambda_{ij}\right\}\omega\left(n,n-\left|V_{odd}\left(S'\right)\right|\right)\right]. \tag{5.40}$$

Let $S_{ext} \subseteq E_{ext}$ denote an edge sub-set of the set of *external* edges, E_{ext}. The summation in (5.40) over 2^{M+n} spanning subsets of the appended graph, $G(V\cup\{0\}, E\cup E_{ext})$, can be split into two summations as follows.

$$Z' = \overbrace{\sum_{S\subseteq E}\left[\lambda^{|S|}\omega\left(n, n-\left|V_{odd}\left(S\right)\right|\right)\right]}^{Z_{int}} + \overbrace{\sum_{S_{ext}\subseteq E_{ext}}\left[\lambda^{|S_{int}|}\left\{\prod_{(0,i)\in S_{ext}}\mu_i\right\}\omega\left(n, n-\left|V_{odd}\left(S_{ext}\right)\right|\right)\right]}^{Z_{ext}}, \quad (5.41)$$

where $\lambda = tanh(\theta)$ and $\mu_i = tanh(h_i)$.

Thus, from (5.41), it is straightforward to see that,

$$l\left(\mathbf{y}, \theta, \boldsymbol{\varepsilon}\right) = \frac{(Z_{int} + Z_{ext})}{(Z_{int} - Z_{ext})}, . \quad (5.42)$$

The first term in (5.41), $Z_{int} = \sum_{S\subseteq E}\left[\lambda^{|S|}\omega\left(n, n-\left|V_{odd}\left(S\right)\right|\right)\right]$ is the summation over the 2^M *internal* spanning edge subsets of $G(V, E)$, and the second term,

$$Z_{ext} = \sum_{S_{ext}\subseteq E_{ext}}\left[\lambda^{|S_{int}|}\left\{\prod_{(0,i)\in S_{ext}}\mu_i\right\}\omega\left(n, n-\left|V_{odd}\left(S_{ext}\right)\right|\right)\right], \quad (5.43)$$

is a summation defined over the remaining $2^M(2^n - 1)$ spanning edge-subsets including edges from the external edge subset E_{ext}. This is explained in Figure 5.6 where the native graph $G(V, E)$ is a 3-node chain graph with $V = \{1, 2, 3\}$ and $E = \{(1, 2), (2, 3)\}$ and the *appended* graph being $G(V \cup \{0\}, E \cup E_{ext})$ with $E_{ext} = \{(0, 1), (0, 2), (0, 3)\}$. This appended graph has $2^5 = 32$ spanning subgraphs split into 3 different boxes as shown. The first box (in the darkest shade) pertains to the subgraphs consisting strictly of internal edges which maps to Z_{int}. The second and the third boxes have at-least one external edge included and map to Z_{ext}.

Computing Z_{int}

Given that the number of odd-degree nodes in a simple graph is always even [150],$|V_{odd}\left(S\right)| \in \{0, 2, 4, ..., n-1\}$. Let us define $\mathcal{N}(e, o)$ to be the number of spanning sub-graphs of the graph G with e edges and o nodes with odd-degree. Thus, we can write,

$$\begin{aligned} Z_{int} &= \sum_{S\subseteq E}\left[\lambda^{|S|}\omega\left(n, n-\left|V_{odd}\left(S\right)\right|\right)\right] \\ &= \sum_{e=0}^{M}\lambda^e \sum_{o\in\{0,2,..,n-1\}}\mathcal{N}(e, o)\omega(n, n-o).; \end{aligned} \quad (5.44)$$

While $\mathcal{N}(e,o)$ can be computed for certain stylized topologies such as chain and the cycle, it is quite hard to estimate in the general case. In statistical physics, the following formula has been used to approximate $\mathcal{N}(e,o)$ and is known as the binomial approximation formula [151],

$$\mathcal{N}^{(approx,binom)}(e,o) = \binom{M}{e}\binom{n}{o}2^{-n+1}. \tag{5.45}$$

Hence, the binomial approximation formula for Z_{int} would be,

$$Z_{int,binom} = \sum_{e=0}^{M} \lambda^e \sum_{o\in\{0,2,..,n-1\}} \binom{M}{e}\binom{n}{o}2^{-n+1}\omega(n,n-o). \tag{5.46}$$

The high noise weak network effect regime

In the NW regime, we can ignore terms involving higher powers (≥ 2) of λ and μ_i in the PPF calculations. This renders the approximation of Z_{int} in the NW regime to be,

$$Z_{int,nw} = \sum_{e=0}^{1} \lambda^e \sum_{o\in\{0,2,..,n-1\}} \mathcal{N}(e,o)\omega(n,n-o). \tag{5.47}$$

Now, it is straight forward to see that for a simple graph[3],

$$\mathcal{N}(0,o) = \begin{cases} 1 & \text{if } o = 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.48}$$

and

$$\mathcal{N}(1,o) = \begin{cases} M & \text{if } o = 2 \\ 0 & \text{otherwise.} \end{cases} \tag{5.49}$$

Using substituting (5.48) and (5.49) in (5.47), we have,

$$Z_{int,nw} = 2^{n-1} + M\lambda\omega(n,n-2). \tag{5.50}$$

Now, using (5.76a), we see that $\omega(n,n-2) = 0$, which renders,

$$Z_{int,nw} = 2^{n-1}. \tag{5.51}$$

[3] A simple graph, also called a strict graph is an unweighted, undirected graph with no graph loops or multiple edges.

Approximating Z_{ext}

Now, let us turn our attention towards Z_{ext}, which is,

$$Z_{ext} = \sum_{S_{ext} \subseteq E_{ext}} \left[\lambda^{|S_{int}|} \left\{ \prod_{(0,i) \in S_{ext}} \mu_i \right\} \omega \left(n, n - |V_{odd} \left(S_{ext} \right)| \right) \right]. \tag{5.52}$$

With regard to Figure 5.6, we see that including only those spanning subsets with just one *external* edge in the summation, we have,

$$Z_{ext,1} = \sum_{i=1}^{n} \mu_i \left\{ \sum_{e=0}^{m} \lambda^e \sum_{o \in \{0,2,..,n-1\}} \left\{ \begin{array}{c} \mathscr{N}_{odd}(e,o,i)\omega(n, n-(o-1)) \\ + \\ \mathscr{N}_{even}(e,o,i)\omega(n, n-(o+1)) \end{array} \right\} \right\}. \tag{5.53}$$

Here, $\mathscr{N}_{odd}(e,o,i)$ computes the number of spanning sub-graphs with e edges and o odd-degreed nodes where the degree of node i is odd and $\mathscr{N}_{even}(e,o,i)$ computes the number of spanning sub-graphs with e edges and o odd-degreed nodes where the degree of node i is even.

As seen in the case of Z_{int}, in the NW regime, we ignore terms involving higher powers (≥ 2) of λ and μ_i in the PPF calculations, resulting in,

$$Z_{ext,nw} = \sum_{i=1}^{n} \mu_i \left\{ \sum_{e=0}^{1} \lambda^e \sum_{o \in \{0,2,..,n-1\}} \left\{ \begin{array}{c} \mathscr{N}_{odd}(e,o,i)\omega(n, n-(o-1)) \\ + \\ \mathscr{N}_{even}(e,o,i)\omega(n, n-(o+1)) \end{array} \right\} \right\}. \tag{5.54}$$

Computing $\mathscr{N}_{odd}(e,o,i)$ and $\mathscr{N}_{even}(e,o,i)$ for $0 \leq e \leq 1$

Firstly, if $e = 0$, the only sub-graph possible is the empty graph where all the nodes have $\Delta_i = 0$ (even). This implies,

$$\mathscr{N}_{even}(0,o,i) = \begin{cases} 1 & \text{if } o = 0, 1 \leq i \leq n \\ 0 & \text{otherwise} \end{cases} \tag{5.55}$$

and

$$\mathscr{N}_{odd}(0,o,i) = 0, \forall o \in \{0,2,..,n-1\}, 1 \leq i \leq n. \tag{5.56}$$

Also, for $e = 1$, we have M spanning sub-graphs with 1 edge, each containing 2 nodes that are connected by that sole edge having degree 1 and the rest of the unconnected singleton nodes having degree 0. So, a node-i with degree Δ_i will appear in Δ_i of these spanning-subgraphs being connected to one of its Δ_i neighbors and will appear in the rest of the $M - \Delta_i$ spanning sub-graphs unconnected (even degreed). That is,

$$\mathcal{N}_{odd}(1, o, i) = \begin{cases} \Delta_i & \text{if } o = 2, 1 \leq i \leq n \\ 0 & \text{otherwise} \end{cases} \tag{5.57}$$

and

$$\mathcal{N}_{even}(1, o, i) = \begin{cases} (M - \Delta_i) & \text{if } o = 2, 1 \leq i \leq n \\ 0 & \text{otherwise} \end{cases} \tag{5.58}$$

Now substituting (5.56),(5.55),(5.57) and (5.58) in (5.54), we have,

$$Z_{ext,nw} = \sum_{i=1}^{n} \mu_i w_i, \tag{5.59}$$

where the node weights $w_i, 1 \leq i \leq n$, is given by,

$$\begin{aligned} w_i &= \omega\left(n, n-1\right) + \lambda \left\{\Delta_i \omega\left(n, n-1\right) + (M - \Delta_i)\omega\left(n, n-3\right)\right\}; \\ &= 1 + \lambda \left\{\frac{\Delta_i\left(n-1\right) - M}{n-2}\right\}. \end{aligned} \tag{5.60}$$

Finally combining (5.51) and (5.59), we have the following expression for the decision statistic $l(\mathbf{y}, \theta, \boldsymbol{\varepsilon})$ in the NW regime,

$$l\left(\mathbf{y}, \theta, \boldsymbol{\varepsilon}\right) = \frac{\left(2^{n-1} + \sum\limits_{i=1}^{n} \mu_i w_i\right)}{\left(2^{n-1} - \sum\limits_{i=1}^{n} \mu_i w_i\right)}. \tag{5.61}$$

Now, using (5.61), the approximate MAP detector in the NW regime would be

$$\hat{m}_{MAP,nw}(\mathbf{y}) = \begin{cases} +1 & \text{if } \sum\limits_{i=1}^{n} \mu_i w_i > 0 \\ -1 & \text{otherwise} \end{cases} \tag{5.62}$$

Now, let us look at the decision statistics in (5.62). Using the expression for w_i derived in (5.60), we have,

$$\sum_{i=1}^{n} \mu_i w_i = \sum_{i=1}^{n} y_i \delta_i, \tag{5.63}$$

where

$$\nu_i^{'} = \overbrace{\tanh(\varepsilon_i)}^{\text{Node - effect}} + \overbrace{\tanh(\varepsilon_i)\lambda\left\{\frac{\Delta_i\,(n-1)-M}{n-2}\right\}}^{\text{Network - effect}}. \tag{5.64}$$

Now, using the approximation $\tanh(a) \approx a$ for $\tanh(\varepsilon_i)$ and $\lambda = tanh(\theta)$ in (5.64), we have,

$$\nu_i^{'} = \overbrace{(\varepsilon_i)}^{\text{Node - effect}} + \overbrace{(\varepsilon_i)\theta\left\{\frac{\Delta_i\,(n-1)-M}{n-2}\right\}}^{\text{Network - effect}}. \tag{5.65}$$

Finally comparing (5.28) and (5.65), we see that the two weights are the same.

5.7.2 Application 2: Extension of the HT framework for the generalized majority detection problem

In many real world scenarios, such as the cloture rule case of the United States Senate, the vote/opinion threshold to be achieved for the declaration of majority status is not strictly half, but some value, σ_{th}. In the specific case of the cloture rule XXII, the Senate may limit consideration of a matter that is currently pending to 30 additional hours if and only if the 60 vote (three-fifths) majority is in favor [100]. In such scenarios, we have to estimate the probability of *super-majority*, $\Pr(\mathbf{e}^T\mathbf{x} > \sigma_{th})$, which under the current set-up is,

$$\begin{aligned}\Pr(\mathbf{e}^T\mathbf{x} > \sigma_{th}) &= \frac{\sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}>\sigma_{th}} \exp\left\{\sum_{(i,j)\in E}\theta_{ij}x_ix_j + \sum_{(i)\in V}\theta_i x_i\right\}}{Z\,(\theta,\mathbf{h})} \\ &= \frac{Z_{\sigma_{th}}\,(\theta,\mathbf{h})}{Z\,(\theta,\mathbf{h})},\end{aligned} \tag{5.66}$$

where the numerator is the Generalized Partial Partition Function (GPPF),

$$Z_{\sigma_{th}}\,(\boldsymbol{\theta},\mathbf{h}) = \sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}>\sigma_{th}} \exp\left\{\sum_{(i,j)\in E}\theta_{ij}x_ix_j + \sum_{(i)\in V}\theta_i x_i\right\}. \tag{5.67}$$

Now, we generalize the result for the PPF in 5.6.1 via the following lemma.

Lemma 5.7.1. *For the Ising model defined in* (5.2)*, the HT expansion expression for the GPPF, as defined in* (5.67)*, is given by,*

$$Z_{\sigma_{th}}\,(\boldsymbol{\theta},\mathbf{h}) = c(\boldsymbol{\theta},\mathbf{h}) \sum_{S'\subseteq E'} \left[\left[\prod_{(i,j)\in S}\lambda_{ij}\right]\omega_{\sigma_{th}}\left(n, n-|V_{odd}(S')|\right)\right]. \tag{5.68}$$

Here, $S' \subseteq E'$ is the edge-set of the spanning sub-graph $G(V', E')$, $\lambda_{ij} = tanh(\theta_{ij})$, $V_{odd}(S') \subseteq V$ is the subset of odd-degreed nodes in V (not counting the pseudo-node) in the spanning sub-graph $G(V', S')$ and

$$\omega_{\sigma_{th}}(n, p) = \sum_{i=0}^{\left\lfloor \frac{n-\sigma_{th}}{2} \right\rfloor} \left\{ (-1)^i \mathfrak{a}_{\sigma_{th}}(n, p, i) \right\}, \tag{5.69}$$

where,

$$\mathfrak{a}_{\sigma_{th}}(n, p, i) = \begin{pmatrix} n-p \\ i \end{pmatrix} \sum_{k=0}^{\left\lfloor \frac{n-\sigma_{th}}{2} \right\rfloor - i} \begin{pmatrix} p \\ k \end{pmatrix}. \tag{5.70}$$

Proof. The proof is presented in Appendix-B. It is again based on the code puncturing idea developed in the proof for Theorem 5.3.1 in the appendix-A with a small tweak. □

Writing $Z_+(\boldsymbol{\theta}, \mathbf{h})$ in terms of $Z(\boldsymbol{\theta}, \mathbf{h})$

It is noteworthy to observe that the HT expansion for the complete partition function [141] is given by,

$$Z(\boldsymbol{\theta}, \mathbf{h}) = 2^n c(\boldsymbol{\theta}, \mathbf{h}) \sum_{S \subseteq E} \left[\prod_{(i,j) \in S} \lambda_{ij} \prod_{i \in V_{odd}(S)} \mu_i \right]. \tag{5.71}$$

Now, combining (5.71) and (5.15), we have,

$$Z_+(\boldsymbol{\theta}, \mathbf{h}) = \frac{1}{2} Z(\boldsymbol{\theta}, \mathbf{h}) + Z_\delta(\boldsymbol{\theta}, \mathbf{h}), \tag{5.72}$$

where $Z_\delta(\boldsymbol{\theta}, \mathbf{h}) = c(\boldsymbol{\theta}, \mathbf{h}) \sum\limits_{S \subseteq E} \sum\limits_{\substack{U \subseteq V \\ |U| \in \left\{ \mathbb{Z}^+_{odd} \right\}}} \left(\prod\limits_{(i,j) \in S} \lambda_{ij} \prod\limits_{i \in U} \mu_i \right) w(S, U)$.

We make two important observations with regard to (5.72). Firstly, in the absence of externals field, we have, ($\mu_i = 0; i \in V$), which renders $Z_\delta(\boldsymbol{\theta}, \mathbf{h}) = 0$ and $Z_+(\boldsymbol{\theta}, \mathbf{h}) = \frac{1}{2} Z(\boldsymbol{\theta}, \mathbf{h})$. Secondly, from eq. (5.76a), we see that $Z_\delta(\boldsymbol{\theta}, \mathbf{h})$ is not necessarily positive.

5.7.3 Application 3: Relationship between the subgraph weights $w(S,U)$ and Super-Catalan numbers and Krawchuk polynomials

To begin with, for $N \in \mathbb{Z}_{+}, 0 \leqslant k \leqslant N$, the Krawchuk polynomial [152] in variable ξ is defined by the following equivalent summations,

$$\begin{aligned}\mathcal{K}\left(N,k,\xi\right) &= \sum_{j=0}^{k}(-1)^{j}\begin{pmatrix}\xi\\ j\end{pmatrix}\begin{pmatrix}N-\xi\\ k-j\end{pmatrix}\\ &= \sum_{j=0}^{k}(-2)^{j}\begin{pmatrix}N-j\\ k-j\end{pmatrix}\begin{pmatrix}\xi\\ j\end{pmatrix}\\ &= \sum_{j=0}^{k}(-1)^{j}2^{k-j}\begin{pmatrix}N-k+j\\ j\end{pmatrix}\begin{pmatrix}N-\xi\\ k-j\end{pmatrix}.\end{aligned} \tag{5.73}$$

Also, given $a, b \in \{0\} \cup \mathbb{Z}^{+}$, we define Super-Catalan numbers [153], $\mathcal{S}\left(a,b\right)$ as,

$$\mathcal{S}\left(a,b\right) = \frac{(2a)!\,(2b)!}{a!\,(a+b)!b!}. \tag{5.74}$$

The Super-catalan numbers and Krawchuk polynomials are related to each other by the following relationship [154]:

$$\mathcal{K}\left(2(a+b), a+b, 2a\right) = (-1)^{a}\mathcal{S}\left(a,b\right). \tag{5.75}$$

We have shown that (refer to the appendix),

$$\omega(n,p) = \begin{cases}(-1)^{\left(\frac{n-p-1}{2}\right)}\dfrac{(n-p-1)!\,(p)!}{\left(\frac{p}{2}\right)!\left(\frac{n-p-1}{2}\right)!\left(\frac{n-1}{2}\right)!} & \text{if } p \text{ is even} \quad (5.76a)\\ 2^{n-1} & \text{if } p=n \quad (5.76b)\\ 0 & \text{otherwise} \quad (5.76c)\end{cases}$$

We observe that as long as p is even, $\varsigma = \frac{n-p-1}{2} \in \mathbb{Z}^{+}$. Now, using the relationship detailed in (5.75) and (5.76a), we have,

$$\omega\left(n,p\right) = \mathcal{K}\left(n-1, \left(\frac{n-1}{2}\right), n-p-1\right) = (-1)^{\varsigma}\mathcal{S}\left(\varsigma, \left(\frac{n-1}{2}\right) - \varsigma\right), \tag{5.77}$$

where $\varsigma = \left(\frac{n-p-1}{2}\right)$.

This result, we believe is, an interesting contribution to the body of mathematical literature surrounding the Krawchuk polynomials and Super-Catalan numbers in the following ways. Firstly, the relationship detailed in (5.77) would yield one more computational formula hitherto unseen for computing Krawchuk polynomials involving partial row sums of the pascal's triangle. For $a, b \in \mathbb{Z}^+ : a \geq b$, let us define, the pascal's triangle row sum function, $\tau(a, b)$ as,

$$\tau\left(a, b\right) = \sum_{k=0}^{b} \left(\begin{array}{c} a \\ k \end{array}\right). \tag{5.78}$$

As seen, it is simply the sum of elements of the a^{th} row of the pascal's triangle. It is a well known result that $\tau(a, a) = 2^a$. However, for $b < a$, we are left with the partial row sum and the sequence of such partial row sums are known to have interesting properties as detailed in [155].
From (5.19) and (5.77), we have,

$$\mathcal{K}\left(n-1, \left(\frac{n-1}{2}\right), n-p-1\right) = \sum_{i=0}^{\frac{n-1}{2}} \left\{ (-1)^i \left(\begin{array}{c} n-p \\ i \end{array}\right) \tau\left(p, \left(\frac{n-1}{2}\right) - i\right) \right\}. \tag{5.79}$$

Secondly, (5.77) would also yield an interesting physical interpretation for Super-Catalan numbers. There is active research going on regarding physical or geometric interpretation for these numbers and success has only been achieved thus far for specific values [156, 157]. While there exists a physical (geometric) interpretation of the related *Catalan numbers* [156] in terms of the number of *good paths* from points (n, n) to $(0, 0)$ on a grid which do not cross the diagonal line, there is active research underway trying to unearth a similar physical interpretation of the Super-Catalan numbers and success has only been achieved for certain special cases as detailed in [156, 157]. We believe that this result in (5.77) contributes an interesting interpretation for Super-Catalan numbers hitherto unseen in mathematics literature.

5.7.4 Application 4: Error probability of the trivial detector

The total probability theorem renders the probability of error of majority vote detection to be,

$$p_{err} = p_{err|m=-1} p\left(m = -1\right) + p_{err|m=+1} p\left(m = +1\right), \tag{5.80}$$

where the conditional probabilities of error, $p_{err|m=\pm 1}$, are defined as,

$$p_{err|m=\pm 1} = \sum_{\mathbf{y}} p\left(\mathbf{y}|m = \pm 1\right) \mathbf{I}\left[\!\left[dec(\mathbf{y}) \lesseqgtr 1\right]\!\right], \tag{5.81}$$

with $dec(\mathbf{y})$ being the decision statistic used by the detector under analysis. Given the symmetry in the Ising prior without external fields (in (5.2), we have, $p(\mathbf{e}^T\mathbf{x} > 0) = p(\mathbf{e}^T\mathbf{x} < 0) = 0.5$, which implies that $p(m = -1) = p(m = +1) = 0.5$, which in turn renders the probability of error in (5.80) to be simply,

$$p_{err} = \sum_{\mathbf{y}} p\left(\mathbf{y}|m = -1\right) \mathbf{I}\left[\!\left[dec(\mathbf{y}) > 1\right]\!\right]. \tag{5.82}$$

In (5.82) above, the conditional probabilities, $p(\mathbf{y}|m = \pm 1) = p(\mathbf{y}|\mathbf{e}^T\mathbf{x} \gtrless 0)$, are evaluated by,

$$\begin{aligned} p\left(\mathbf{y}|\mathbf{e}^T\mathbf{x} \gtrless 0\right) &= 2p\left(\mathbf{y}, \mathbf{e}^T\mathbf{x} \gtrless 0\right) = 2\sum_{\mathbf{e}^T\mathbf{x}\gtrless 0} p\left(\mathbf{x}, \mathbf{y}\right) \\ &= \frac{2\sum\limits_{\mathbf{e}^T\mathbf{x}\gtrless 0} \exp\left\{\theta \sum\limits_{(i,j)\in E} x_i x_j + \sum\limits_{i=1}^{n} \varepsilon_i y_i x_i\right\}}{Z(\theta)\left(2^n \prod\limits_{i=1}^{n} \cosh(\varepsilon_i)\right)} = \frac{2Z_{\pm}(\theta, \boldsymbol{\varepsilon} \circ \mathbf{y})}{Z(\theta)\left(2^n \prod\limits_{i=1}^{n} \cosh(\varepsilon_i)\right)}. \end{aligned} \tag{5.83}$$

Now, substituting (5.83) in (5.82), we have,

$$\begin{aligned} p_{err} &= \sum_{\mathbf{y}} p\left(\mathbf{y}|m = -1\right) \mathbf{I}\left[\!\left[dec(\mathbf{y}) > 1\right]\!\right] \\ &= \frac{2}{Z(\theta)\left(2^n \prod\limits_{i=1}^{n} \cosh(\varepsilon_i)\right)} \sum_{\mathbf{y}} Z_{-}(\theta, \boldsymbol{\varepsilon}.\mathbf{y}) \mathbf{I}\left[\!\left[dec(\mathbf{y}) \lesseqgtr 1\right]\!\right]. \end{aligned} \tag{5.84}$$

Probability of error of the trivial detector with identical ε_i across the BSCs

To begin with, we set the bit-flip probability (q_i) of all the BSCs to be identical which implies,$\varepsilon_i = \varepsilon;\ 1 \le i \le n$.

With uniform edge potential θ for the Ising prior and uniform ε for all the BSCs, the joint distribution of $\mathbf{x}$ and $\mathbf{y}$ in (5.6) becomes,

$$p\left(\mathbf{y}, \mathbf{x}\right) = p\left(\mathbf{x}\right) p\left(\mathbf{y}|\mathbf{x}\right) = \frac{\exp\left(\theta \mathbf{x}^T A \mathbf{x} + \varepsilon \mathbf{y}^T \mathbf{x}\right)}{Z(\theta)(2\cosh(\ \varepsilon))^n}. \tag{5.85}$$

For the trivial detector, we see that $p_{err,trivial}$ can be evaluated as,

$$\begin{aligned}
p_{err} &= \sum_{\mathbf{y}} p\left(\mathbf{y}|\mathbf{e}^T\mathbf{x} < 0\right) \mathbf{I}\left[\left[\left(\mathbf{e}^T\mathbf{y} > 0\right)\right]\right] \\
&= 2\sum_{\mathbf{y}} p\left(\mathbf{y}, \mathbf{e}^T\mathbf{x} < 0\right) \mathbf{I}\left[\left[\left(\mathbf{e}^T\mathbf{y} > 0\right)\right]\right] \\
&= 2 \sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}<0} \sum_{\mathbf{y}:\mathbf{e}^T\mathbf{y}>0} p\left(\mathbf{y}, \mathbf{x}\right) \\
&= \frac{2}{Z(\theta)(2\cosh(\varepsilon))^n} \sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}<0} \exp\left(\theta\mathbf{x}^T A\mathbf{x}\right) \left\{ \sum_{\mathbf{y}:\mathbf{e}^T\mathbf{y}>0} \exp\left(\varepsilon\mathbf{y}^T\mathbf{x}\right) \right\}.
\end{aligned} \tag{5.86}$$

Now, we show that the use of the HT expansion formulation of the PPF allows us to compute the exact closed form expression of the probability of error for certain topologies such as complete network, star network as well as the case when the network is absent. This will in fact, help us showcase that the problem of majority vote detection inherently results in a higher probability of error compared the classical single channel bit detection problem for the no-network scenario.

Closed form expressions for p_{err}

We derive the closed form expressions in 2 stages. In the first stage, we derive closed-form expression for the inner summation, $\sum_{\mathbf{y}:\mathbf{e}^T\mathbf{y}>0} \exp\left(\varepsilon\mathbf{y}^T\mathbf{x}\right)$ and then in the second stage, we focus on the outer summation.

Lemma 5.7.2. *The summation* $\sum_{\mathbf{y}:\mathbf{e}^T\mathbf{y}>0} \exp\left(\varepsilon\mathbf{y}^T\mathbf{x}\right)$ *can be computed by the following closed form expression,*

$$\sum_{\mathbf{y}:\mathbf{e}^T\mathbf{y}>0} \exp\left(\varepsilon\mathbf{y}^T\mathbf{x}\right) = (\cosh(\varepsilon))^n \sum_{e=0}^{n} \lambda^e \omega\left(n, n-e\right) \sum_{k=0}^{e} (-1)^k \binom{n_{neg}(\mathbf{x})}{k} \binom{n - n_{neg}(\mathbf{x})}{e-k}, \tag{5.87}$$

where $\omega(n, p)$ *is as defined in* (5.19) *and* $n_{neg}(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{I}\left[\!\left[x_i = -1\right]\!\right]$, *which basically counts the number of* -1*s in the vector* $\mathbf{x}$.

Proof. To begin with, we see that the inner summation, $\left\{ \sum_{\mathbf{y}:\mathbf{e}^T\mathbf{y}>0} \exp\left(\varepsilon\mathbf{y}^T\mathbf{x}\right) \right\}$ is in fact the positive part partition function of an empty network with field $\{\varepsilon\mathbf{x}\}$ which can be re-written as the positive part partition function of a star network where the central node is the pseudo-node (clamped to $+1$) and the edge-potentials being defined as shown,

$$\begin{aligned}\sum_{\mathbf{y}:\mathbf{e}^T\mathbf{y}>0} \exp\left(\varepsilon \mathbf{y}^T \mathbf{x}\right) &= Z_{+}^{(star)}(\boldsymbol{\theta}); \\ \boldsymbol{\theta} &= \left\{\theta_{i0}; i = 1, ..., n : \theta_{i0} = \varepsilon x_i\right\}.\end{aligned} \tag{5.88}$$

Using the high-temperature expansion formula we have developed in (5.15), we see that,

$$Z_{+}^{(star)}(\boldsymbol{\theta}) = \prod_{i=1}^{n} \cosh(\varepsilon x_i) \sum_{S' \subseteq E'} \left[W(S')\right] = (\cosh(\,\varepsilon))^n \sum_{S' \subseteq E'} \left[W(S')\right], \tag{5.89}$$

where,

$$W(S') = \left\{\prod_{(i,0)\in S'} \lambda_{i0}\right\} \omega\left(n, n - |V_{odd}\left(\,S'\right)|\right). \tag{5.90}$$

Here, $\lambda_{i0} = \tanh\left(\theta_{i0}\right)$, $V_{odd}(S') \subseteq V$ is the subset of odd-degreed nodes in V (not-counting the pseudo-node) in the spanning sub-graph $G(V \cup \{0\}, S')$ and the function $\omega(n, p)$ is as defined in (5.19). Now, we make the following observations.

1. **Observation 1**: *The number of odd-degreed nodes (not-counting the central pseudo-node) in the spanning sub-graph S' of a star-network is the number of edges in the spanning-subgraph itself($|S'|$).*

   That is,

$$|V_{odd}\left(\,S'\right)| = |S'|. \tag{5.91}$$

2. **Observation 2**:*Given that the edge-potentials of the star-network under consideration are all of the same magnitude varying only in signs, for each spanning sub-graph S', we can write the product $\left\{\prod_{(i,0)\in S'} \lambda_{i0}\right\}$ to be,*

$$\left\{\prod_{(i,0)\in S'} \lambda_{i0}\right\} = \lambda^{|S'|}(-1)^{n_{-}(S')}, \tag{5.92}$$

   where $n_{-}\left(S'\right) = \sum_{(i,0)\in S'} \mathbf{I}\left[\!\left[sign(\lambda_{i0}) = -1\right]\!\right]$.

Combining the above observations, we have,

$$W(S') = \lambda^{|S'|}(-1)^{n_{-}(S')}\omega\left(n, n - |S'|\right). \tag{5.93}$$

We see that the contribution by a spanning sub-graph S' depends on 2 things. The number of edges in it (which also is the number of odd-degreed non-central nodes in it) and the number of edges with negative λ as the edge-weight.

Now, let us define,

$$n_{neg}(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{I} \left[\!\left[x_i = -1 \right]\!\right], \tag{5.94}$$

and $n_{pos}(\mathbf{x}) = n - n_{neg}(\mathbf{x})$.
Now, consider all spanning sub-graphs with $|S^{'}|$ edges. There are $\binom{n}{|S^{'}|} = \binom{n_{pos}(\mathbf{x})+n_{neg}(\mathbf{x})}{|S'|}$ of them. Now, let us assume that are k negative-edges in a given sub-graph with $|S'|$ edges. Using the Chu-Vandermonde identity, we can write this as a summation over varying number of negative edges indexed by k as,

$$\binom{n_{neg}(\mathbf{x}) + n_{pos}(\mathbf{x})}{|S'|} = \sum_{k=0}^{|S'|} \binom{n_{neg}(\mathbf{x})}{k} \binom{n_{pos}(\mathbf{x})}{|S'| - k}. \tag{5.95}$$

The contributions coming from all spanning sub-graphs can be written as a summation over sub-graphs with $|S|^{'}$ edges as,

$$\sum_{S' \subseteq E'} \left[W(S') \right] = \sum_{|S'|=0}^{n} \sum_{k=0}^{|S'|} \binom{n_{neg}(\mathbf{x})}{k} \binom{n_{pos}(\mathbf{x})}{|S'| - k} \lambda^{|S'|} (-1)^k \omega \left(n, n - |S'| \right). \tag{5.96}$$

Therefore, the positive part partial partition function of the star-network under consideration would be,

$$Z_{+}^{(star)}(\varepsilon \mathbf{x}) = (\cosh(\,\varepsilon))^n \sum_{|S'|=0}^{n} \sum_{k=0}^{|S'|} \binom{n_{neg}(\mathbf{x})}{k} \binom{n - n_{neg}(\mathbf{x})}{|S'| - k} \lambda^{|S'|} (-1)^k \omega \left(n, n - |S'| \right). \tag{5.97}$$

As seen, it only depends on ε and number of -1s in $\mathbf{x}$, that is $n_{neg}(\mathbf{x})$. □

Now using (5.87) in (5.86), we have the following expression for $p_{err,trivial}$ which would serve as the starting point to derive exact expressions for the various topologies to follow:

$$p_{err,trivial} = \frac{1}{2^{n-1} Z(\theta)} \sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}<0} \left[\begin{array}{l} \exp\left(\theta \mathbf{x}^T A \mathbf{x} \right) \times \\ \left\{ \sum_{e=0}^{n} \lambda^e \omega \left(n, n-e \right) \sum_{k=0}^{e} (-1)^k \binom{n_{neg}(\mathbf{x})}{k} \binom{n - n_{neg}(\mathbf{x})}{e - k} \right\} \end{array} \right]. \tag{5.98}$$

Now, let us focus on the outer-summation (defined over $\mathbf{x}$).

From (5.87), we see that the inner summation (defined over $\mathbf{y}$) depends only on $n_{neg}(\mathbf{x})$. Further, the constraint $\mathbf{e}^T\mathbf{x} < 0$ implies that $n_{neg}(\mathbf{x}) \in \{\frac{n+1}{2}, ..., n\}$. and there are $\binom{n}{n_-}$ vectors in the set $\{\mathbf{x} : \mathbf{e}^T\mathbf{x} < 0\}$ with $n_{neg}(\mathbf{x}) = n_-$. If, we can write the quadratic $\mathbf{x}^T A\mathbf{x}$ as a function of $n_{neg}(\mathbf{x})$, i.e, $\mathbf{x}^T A\mathbf{x} = g\left(n_{neg}(\mathbf{x})\right)$, we can re-write (5.98) as,

$$p_{err,trivial} = \frac{1}{2^{n-1}Z(\theta)} \sum_{n_-=\lceil \frac{n}{2} \rceil}^{n} \left[\begin{array}{l} \binom{n}{n_-} \exp\left(\theta g\left(n_-\right)\right) \times \\ \left\{ \sum_{e=0}^{n} \lambda^e \omega\left(n, n-e\right) \sum_{k=0}^{e} (-1)^k \binom{n_-}{k} \binom{n-n_-}{e-k} \right\} \end{array} \right]. \tag{5.99}$$

Eq.(5.99) basically showcases a framework which allows calculation of the closed-form expression of p_{err} for all network topologies where the quadratic $\mathbf{x}^T A\mathbf{x}$ can be written as a function of $n_{neg}(\mathbf{x})$, i.e, $\mathbf{x}^T A\mathbf{x} = g\left(n_{neg}(\mathbf{x})\right)$. Now, we will utilize (5.99) to derive closed-form expressions of p_{err} for various topologies such as the Star and complete network as well as for the empty network case.

No-network ($\theta = 0$) In the no-network scenario, $Z(\theta) = 2^n$ and $\mathbf{x}^T A\mathbf{x} = 0$. Using these results in (5.99), we have,

$$p^{(no-net)}_{err,trivial} = \frac{1}{2^{2n-1}} \sum_{n_-=\lceil \frac{n}{2} \rceil}^{n} \left\{ \binom{n}{n_-} \left\{ \sum_{e=0}^{n} \lambda^e \omega\left(n, n-e\right) \sum_{k=0}^{e} (-1)^k \binom{n_-}{k} \binom{n-n_-}{e-k} \right\} \right\}. \tag{5.100}$$

Complete network In the case of the complete network, the adjacency matrix is $A^{(complete)} = \mathbf{e}_n\mathbf{e}_n^T - I_n$, which implies that,

$$\theta\mathbf{x}^T A^{(complete)}\mathbf{x} = \frac{\theta}{2}\left(\left(n - 2n_{neg}(\mathbf{x})\right)^2 - n\right). \tag{5.101}$$

Now, using (5.101) in (5.99), we have,

$$p^{(complete)}_{err} = \frac{1}{2^{n-1}Z^{(complete)}(\theta)} \sum_{n_-=\lceil \frac{n}{2} \rceil}^{n} \left\{ \begin{array}{l} \binom{n}{n_-} \exp\left(\frac{\theta}{2}\left(\left(n - 2n_-\right)\right)^2 - n\right)\right) \times \\ \left\{ \sum_{e=0}^{n} \lambda^e \omega\left(n, n-e\right) \sum_{k=0}^{e} (-1)^k \binom{n_-}{k} \binom{n-n_-}{e-k} \right\} \end{array} \right\}, \tag{5.102}$$

where the (full) partition function, $Z^{(complete)}(\theta)$ is given by [139],

$$Z^{(complete)}(\theta) = \sum_{m=0}^{n} \left\{ \binom{n}{m} \exp\left(\frac{\theta}{2} \left((n-2m))^2 - n \right) \right) \right\}. \tag{5.103}$$

Star network Without loss of generality, Let us fix 1 to be the central *hub node* and nodes 2 through n be the *spoke-nodes*. Now, we define $\nu_{neg}(\mathbf{x})$ as,

$$\nu_{neg}(\mathbf{x}) = \sum_{i=2}^{n} \mathbf{I} [\![x_i = -1]\!]. \tag{5.104}$$

It is straight-forward to see that $n_{neg}(\mathbf{x})$ and $\nu_{neg}(\mathbf{x})$ are related by,

$$n_{neg}(\mathbf{x}) = \begin{cases} \nu_{neg}(\mathbf{x}) & \text{if } x_1 = +1 \\ \nu_{neg}(\mathbf{x}) + 1 & \text{if } x_1 = -1. \end{cases} \tag{5.105}$$

For the star network, the quadratic $\mathbf{x}^T A^{(star)} \mathbf{x}$ can be expressed as a function of $\nu_{neg}(\mathbf{x})$ dependent on whether the central hub node is clamped at $x_1 = +1$ or $x_1 = -1$ as,

$$\mathbf{x}^T A^{(star)} \mathbf{x} = \begin{cases} n - 1 - 2\nu_{neg}(\mathbf{x}) & \text{if } x_1 = +1 \\ 2\nu_{neg}(\mathbf{x}) - (n-1) & \text{if } x_1 = -1. \end{cases} \tag{5.106}$$

Now, let us define the function $\mathfrak{f}(n, \lambda, n_-)$ (related to the inner summation in (5.86)) to be,

$$\mathfrak{f}(n, \lambda, n_-) = \left\{ \sum_{e=0}^{n} \lambda^e \omega(n, n-e) \sum_{k=0}^{e} (-1)^k \binom{n_-}{k} \binom{n - n_-}{e-k} \right\}, \tag{5.107}$$

and the terms T_+ and T_- to be,

$$\begin{aligned} T_+ &= \sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}<0; x_1=+1} \left[\exp\left(\theta \mathbf{x}^T A \mathbf{x} \right) \left\{ \sum_{e=0}^{n} \lambda^e \omega(n, n-e) \sum_{k=0}^{e} (-1)^k \binom{n_{neg}(\mathbf{x})}{k} \binom{n - n_{neg}(\mathbf{x})}{e-k} \right\} \right], \\ T_- &= \sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}<0; x_1=-1} \left[\exp\left(\theta \mathbf{x}^T A \mathbf{x} \right) \left\{ \sum_{e=0}^{n} \lambda^e \omega(n, n-e) \sum_{k=0}^{e} (-1)^k \binom{n_{neg}(\mathbf{x})}{k} \binom{n - n_{neg}(\mathbf{x})}{e-k} \right\} \right]. \end{aligned} \tag{5.108}$$

Observe that, with regard to the summation in (5.98),

$$T_+ + T_- = \sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}<0} \exp\left(\theta \mathbf{x}^T A \mathbf{x} \right) \left\{ \sum_{e=0}^{n} \lambda^e \omega(n, n-e) \sum_{k=0}^{e} (-1)^k \binom{n_{neg}(\mathbf{x})}{k} \binom{n - n_{neg}(\mathbf{x})}{e-k} \right\}. \tag{5.109}$$

Now, rewriting the terms T_+ and T_- in (5.108) in terms of $\mathfrak{f}(n, \lambda, n_-)$ from (5.107), and using (5.106) and (5.105), we have,

$$\begin{aligned} T_+ &= \sum_{n_-=\frac{n+1}{2}}^{n-1} \left\{ \binom{n-1}{n_-} \exp\left(\theta(n-1-2n_-)\right) \mathfrak{f}\left(n, \lambda, n_-\right) \right\}, \\ T_- &= \sum_{\nu_-=\frac{n-1}{2}}^{n-1} \left\{ \binom{n-1}{\nu_-} \exp\left(-\theta(n-1-2\nu_-)\right) \mathfrak{f}\left(n, \lambda, \nu_-+1\right) \right\}. \end{aligned} \tag{5.110}$$

Finally, using (5.109) and substituting (5.110) in (5.98), we have the following expression for p_{err} for star topology,

$$p_{err,trivial}^{(star)} = \frac{1}{2^{n-1} Z^{(star)}(\theta)} \left(T_+ + T_-\right). \tag{5.111}$$

Curie-Weiss Prior The Curie-Weiss Ising model [112] is basically the Ising model defined on the complete graph with the difference being that the edge-potential scaled by $1/n$. That is, the prior $p(\mathbf{x})$ becomes,

$$p\left(\mathbf{x}\right) = \frac{\exp\left(\frac{\theta}{n} \mathbf{x}^T A^{(complete)} \mathbf{x}\right)}{\sum\limits_{\mathbf{x}} \exp\left(\frac{\theta}{n} \mathbf{x}^T A^{(complete)} \mathbf{x}\right)}. \tag{5.112}$$

Using (5.102), we see that,

$$p_{err}^{(curie-weiss)} = \frac{1}{2^{n-1} Z^{(complete)}(\theta)} \sum_{n_-=\lceil \frac{n}{2} \rceil}^{n} \left\{ \begin{array}{l} \binom{n}{n_-} \exp\bigg(\frac{\theta}{2n}\Big((n-2n_-))^2 - n\Big)\bigg) \times \\ \left\{ \sum\limits_{e=0}^{n} \lambda^e \omega\left(n, n-e\right) \sum\limits_{k=0}^{e} (-1)^k \binom{n_-}{k} \binom{n-n_-}{e-k} \right\} \end{array} \right\}, \tag{5.113}$$

where,

$$Z^{(curie-weiss)}(\theta) = \sum_{m=0}^{n} \left\{ \binom{n}{m} \exp\bigg(\frac{\theta}{2n}\Big((n-2m))^2 - n\Big)\bigg) \right\}. \tag{5.114}$$

Network prior with common ε across the BSCs

Variation of p_{err} with change in network topology In order to understand how p_{err} for the trivial detector varies with the change in network topology, we plotted the $p_{err,trivial}$ obtained by

computing the closed-form expressions (5.100), (5.102), (5.111) and (5.113).
Figure 5.7 presents the plot of p_{err} vs q_{flip}, which is related to ε by, $\varepsilon = \log\sqrt{\frac{1-q_{flip}}{q_{flip}}}$, with $n = 25$ and θ being set to 0.1, 0.75, 1 and 2 in the 4 sub-plots respectively. The important observations are:

1. The curve for the probability of error of majority vote detection for the no-network case (indicated by 'No net' in the legend(s)) lies above the q_{flip} line for $0 < q_{flip} < 0.5$ which implies that the problem of majority bit detection inherently results in a higher probability of error when compared to the classical single bit detection problem.

2. The presence of a network prior results in a lower probability of error compared to the no-network prior scenario. Also, expectedly, the densest network, which is the complete network, yields the lowest p_{err}.

3. For the star-topology, if the network effect is weak ($\theta = 0.1$ case), the $p_{err,trivial}$ curve lies above the q_{flip} line. However, as we increase θ (thereby increasing the network effect), the $p_{err,trivial}$ curve falls above the q_{flip} line.

4. The $p_{err,trivial}$ curves pertaining to the Curie-Weiss prior are the most interesting. As seen, when $\theta = 0.1$, the $p_{err,trivial}^{(curie-weiss)}$ curve is close to the no-network curve lying above the $p_{err,trivial}^{star}$ curve. But when θ is increased to 0.5, we see that there exists a threshold q_{flip} above which the $p_{err,trivial}^{(curie-weiss)}$ curve falls below the q_{flip} line.

5. When θ is large (=2.0), the curves for the Curie-Weiss prior, the complete network prior and the star prior all merge.

5.7.5 Application 5: Lattice path sum interpretation of $w(S')$

In this section, we will explore the Lattice path sum interpretation of $w(S')$ which would be complementary to the code-puncturing approach.

Spin-world viewpoint

In the *spin-world* representation, $w(S')$ is defined as,

$$w(S') = \sum_{\mathbf{x}\in\mathcal{X}_{+}^{(n)}} \prod_{i\in V_{odd}(S')} x_i, \tag{5.115}$$

where $\mathcal{X}_{+}^{(n)} = \left\{\mathbf{x} : \mathbf{x} \in \{-1,+1\}^n, \mathbf{e}^T\mathbf{x} > 0\right\}$.

Lattice path sum viewpoint

[154] establishes the following expression for super-Catalan numbers,

$$\mathcal{S}\left(a,b\right) = (-1)^a \sum_{P\in\Pi_{a+b}} (-1)^{H(2a,P)}. \tag{5.116}$$

Here the summation is defined over the set Π_{a+b} of all lattice paths from $(0,0)$ to $(a+b, a+b)$ consisting of unit steps taken to the right and up the grid, and $H(2a,P)$ denotes the height of the path $P = (P_0; P_1; ...; P_{2(a+b)}) \in \Pi_{a+b}$ after the $2a^{th}$ step, i.e., the y-coordinate of P_{2a}.

Now, combining (5.116), (5.15) and (5.77), we have for $|V_{odd}(S')| > 0$,

$$w(S') = \omega\left(n, n - |V_{odd}(S')|\right) = \sum_{P\in\Pi_{\left(\frac{n-1}{2}\right)}} (-1)^{H(|V_{odd}(S')|-1,P)}. \tag{5.117}$$

The above equation captures the lattice path sum interpretation of $w(S')$. As seen above, the summation is defined over the set $\Pi_{(n-1)/2}$ of all lattice paths from $(0,0)$ to $((n-1)/2, (n-1)/2)$ consisting of unit steps taken to the right and up the grid, and $H(|V_{odd}(S')|-1, P)$ denotes the height of the path $P = (P_0; P_1; ...; P_{n-1}) \in \Pi_{(n-1)/2}$ after the $(|V_{odd}(S')|-1)^{th}$ step.

5.8 Chapter Summary

In this chapter, we have tackled the problem of majority sentiment detection in social networks. We began by motivating the use of the ferromagnetic Ising prior for modeling the expressed

sentiments in a homophilic social network and also proposed using the classical communication theoretic BSC model to model the noisy estimated sentiments. We then demonstrate the need for computing the Ising PPF for MAP detection and derived the High Temperature (HT) expansion formula using a novel code-puncturing based approach. We then derived an approximate MAP detector for the weak network high noise scenario based on second order Taylor series expansion and showed the improvement obtained in terms of the probability of error by this approximate MAP detector via numerical results.
The main contributions of this chapter can be summarized follows.

1. Deriving the HT expansion formula for the PPF using ideas from coding theory, which in itself is an interesting contribution towards the statistical physics and Machine Learning literature.

2. Extending the HT expansion formula for computing the GPPF.

3. Exposing an interesting link between the weights contributed by each subgraph configuration in the PPF computation and Super-Catalan numbers/Krawchuk polynomials.

4. Deriving an approximate linear MAP detector that requires just the first order network information (degree).

5. Providing numerical results that demonstrate the probability of error improvement obtained by harnessing the network prior even in the Weak network Highly noisy labels regime.

6. Using the HT expansion formula to derive the closed form expression for the error probability of the trivial detector for certain stylized graph topologies.

This concludes the NAD part of the this thesis.
Now, we move on to the concluding chapter of this thesis where we recap the important contributions and also list out certain interesting avenues for future research in the topics dealt with.

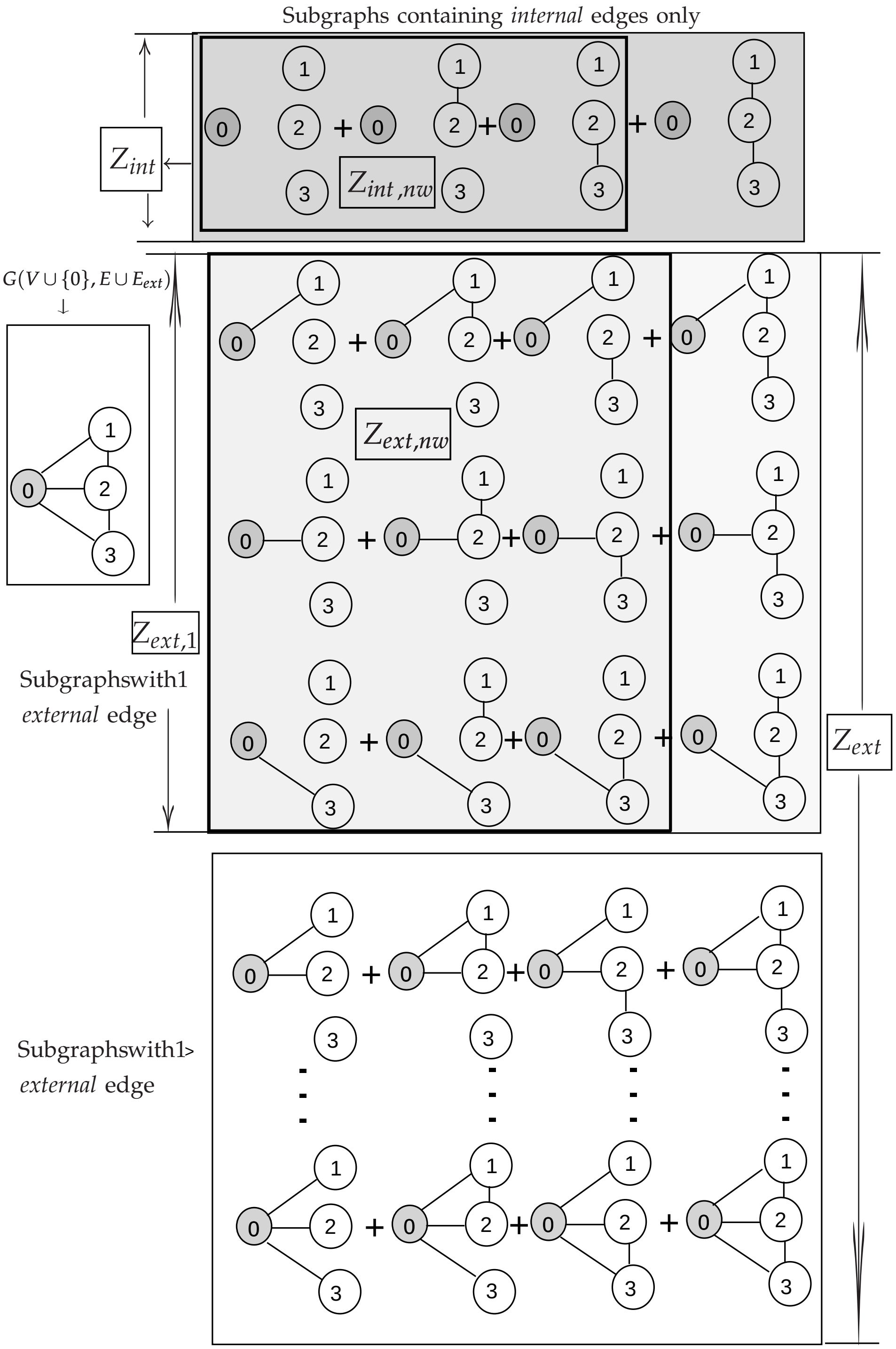


Figure 5.6: The spanning subsets of an appended 3-node chain graph showcasing the splitting of Z' into internal and external summations.

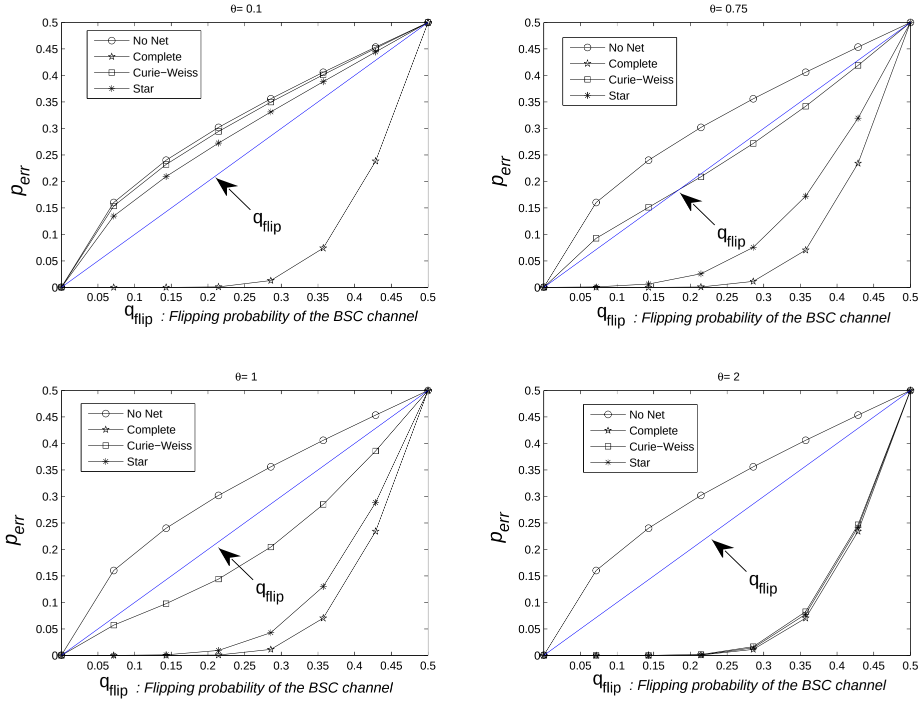


Figure 5.7: $P_{err|\mathbf{e}^T\mathbf{x}<0}$ vs q_{flip} for different topologies

Chapter 6

Thesis summary and Future Work

6.1 Main Contributions

This thesis makes a case for Network Aided Classification and Detection of data. The results presented are both experimental as well as theoretical. We were able to experimentally showcase a series of examples with real world data where Network Aided Classification resulted in substantial improvements in classification accuracy. In the context of Network Aided Detection, we were able to provide some theoretical insights into the improvement brought in error rates by the network priors by means of communication-theory inspired error probability and error exponent analysis. In doing so, we introduced several classical communication theoretic ideas and methodologies that pave the way for deeper importing of ideas and proof methodologies between communications theory and social networks analysis in general.
The main contributions of this thesis can be summarized thus.

1. Introduced a p-value based hypothesis testing framework to validate the homogeneous ferromagnetic Ising prior for the NAC and NAD models. This also involved showcasing real world data where the null hypothesis of the data emanating from the i.i.d flat prior distribution is rejected with high statistical significance in favor of the Ising prior.

2. Showcased a series of examples with real world data where Network Aided Classification resulted in substantial improvements in classification accuracy in disparate areas such as

political science (vote prediction), health policy (lung and cancer level detection) and crime level detection.

3. Introduced the idea of the social network as a *weak channel code* and used a novel communications-inspired framework for analyzing probability of error of a trivial latent sentiment detector in Online Social Networks.

4. Performed error exponent analysis for the trivial latent sentiment detector for standard graph topologies such as star, chain, wheel and complete graphs.

5. Derived the HT expansion formula for the PPF using ideas from coding theory, which in itself is an interesting contribution towards the statistical physics and machine learning literature.

6. Derived an approximate linear MAP detector in the highly pertinent Weak network Highly noisy labels regime and providing numerical results that demonstrated the probability of error improvement obtained by using this detector.

7. Used the HT expansion formula to derive the closed form expression for the error probability of the trivial detector for certain stylized graph topologies.

Now, we conclude this thesis by providing some interesting observation and empirical comparisons which we could not translate in to rigorous theoretic results, but ones which we feel might serve to be a good starting point for further investigation.

6.2 Future work: OSN topology versus inference algorithm

Through the course of the research undertaken, we observed that the classification accuracy obtained on using an underlying graph in the NAC and NAD frameworks described in the thesis, did depend on the approximate inference algorithm being used. For example, in some cases when the graph had high *local transitivity*, it was advisable to use the Mean Field (MF) inference algorithm over the Loopy Belief Propagation (LBP) algorithm. Also, the execution time for each of these algorithms varied immensely as shown in the upcoming section.

In this section, we present empirical results relating the classification error rate (within the NAC framework) obtained by a certain inference algorithm with the variation in parameters that control the underlying topology of the graph constructed using two standard synthetic graph models used heavily in Network Science and SNA literature [14,15].

In doing so, we hope that these empirical results might serve to be a good starting point for further research in this area.

We also hope that this section can serve as a repository of empirical guidance which will help a machine learning practitioner to cherry pick the right inference algorithm based on the topology of the graph that he is planning to use in the NAC and the NAD frameworks described in the thesis.

We now begin by briefly describing the two network generation models used in Network Sciences, that is the Barabasi-Albert (BA) model used to generate networks with scale-free degree distribution and the Watts-Strogatz (WS) model used to generate networks with *small-world* characteristics.

Before we describe these models, we define a few network-theoretic terms we will be using to describe our results.

6.2.1 Definitions

Let $G(V, E)$ be an undirected graph. The neighborhood N_i for a node i is defined as the subset of vertices that it connected to. That is, $N_i = \{j \in V : (i, j) \in E\}$. The degree of the node i is denoted by $\Delta_i = |N_i|$.

Local clustering coefficient of a node i (C_i)

If a vertex i has Δ_i neighbors, a maximum of $\frac{\Delta_i(\Delta_i-1)}{2}$ edges could exist among the nodes within the neighborhood of this node i. The local clustering coefficient for undirected graphs is be defined as

$$c_i = \frac{2|\{(j,k) : j,k \in N_i, (j,k) \in E\}|}{\Delta_i(\Delta_i - 1)}.$$

As seen, it is a measure of the extent to which the nodes of the graph tend to cluster together and is also termed *local transitivity*. Having defined the local clustering coefficient for a node i, we can define the global clustering coefficient in a network (Watts and Strogatz [41]) as the average of the local clustering coefficients of all the n vertices,

$$C = \frac{1}{n} \sum_{i=1}^{n} c_i. \tag{6.1}$$

This metric has deep significance in SNA as social graphs are expected to have a higher level of clustering compared to random graphs or even technological graphs based on the principle of homophily [18].

Average path length

Average path length is defined as the average number of *hops* (or *steps*) along the shortest paths for all possible pairs of graph nodes and is used as a measure of the efficiency of information or mass *flow* on a network.
For any given $i, j \in V$, let us denote $d(i, j)$ to be the shortest distance between the nodes i and j (With the convention that $d(i, j) = 0$ if the node j cannot be reached from i). Now, the average path length L is defined as,

$$L = \frac{1}{n \cdot (n-1)} \cdot \sum_{i \neq j} d(i, j). \tag{6.2}$$

Generalized loop of a graph

A generalized loop in a graph $G(V, E)$ is any subgraph $C = (V', E')$, $V' \subseteq V, E' \subseteq (V' \times V')) \cap E$ such that each node in V' has degree two or larger.

Having defined the required terms, we now briefly describe the BA and the WS graph models.

6.2.2 Barabasi-Albert (BA) model

One of the earlier discoveries in the field of network sciences was the wide spread existence of networks whose degree distributions followed a power-law degree distribution (or *scale-free distributed*) in both natural and human-made systems. This included the Internet, the world wide web (WWW), academic citation networks and social networks [14, 15].
The Barabasi-Albert (BA) model [42] was a simple stochastic model proposed to generate such a scale-free graph based on a very simple and intuitive *preferential attachment* concept.

It entails a discrete time step model and in each time step a single node is introduced to the graph which makes preferential connections with the already existent nodes of the graph by a mechanism termed as *preferential attachment*.

We begin with one node and 0 edges in the first time step. Then one node is appended in each time step and this newly added node adds some connections (edges) to the pre-existent nodes in the graph. The number of edges added in each time step is a parameter of the model, denoted by m_{ba}. Now, the *preferential attachment* mechanism is incorporated via the probability with which a given pre-existent node, say i, is chosen by the new *incoming* node and is given,

$$p_{chosen}(i) \sim \Delta_i^{pow} + p_0, \tag{6.3}$$

where Δ_i is the degree of node i in the current time step, pow is the power parameter of the BA model and p_0 is termed as the 'zero appeal' argument usually set to 1. If $pow = 1$, it is said to be a linear-preferential BA model and if $pow = 2$, we say that the preferential-attachment was quadratic.

Defining the degree distribution $P(\Delta)$ of a network to be the fraction of nodes in the network with degree Δ, it has been shown in [14,15,42], that a graph generated by this linear-preferential BA model has a degree distribution that is *scale free* and the degree distribution can be written as a power law distribution of the form, $P(\Delta) \sim \Delta^{-3}$.

Now, as seen in Figure 6.1,(with $n = 100$ and $m_{ba} = 2$), when the power parameter pow is increased from 1 to 2, that is we switch from linear preferential attachment to quadratic preferential attachment, we see two things. Firstly, the topology becomes more *hub-centric* with a few important *hub nodes* connecting to most on account of the fact that these were the initial nodes that enjoyed the quadratic preferential attachment in their favor as the model evolved through the discrete time steps. Secondly, we also see that there is a dramatic increase in the mean local transitivity as defined in (6.1).

6.2.3 Watts-Strogatz (WS) model

The WattsStrogatz (WS) model [41] is another popular stochastic graph generation model that produces graphs with *small-world* properties. That is, the graphs produced have small average path lengths while maintaining high level of clustering or transitivity.

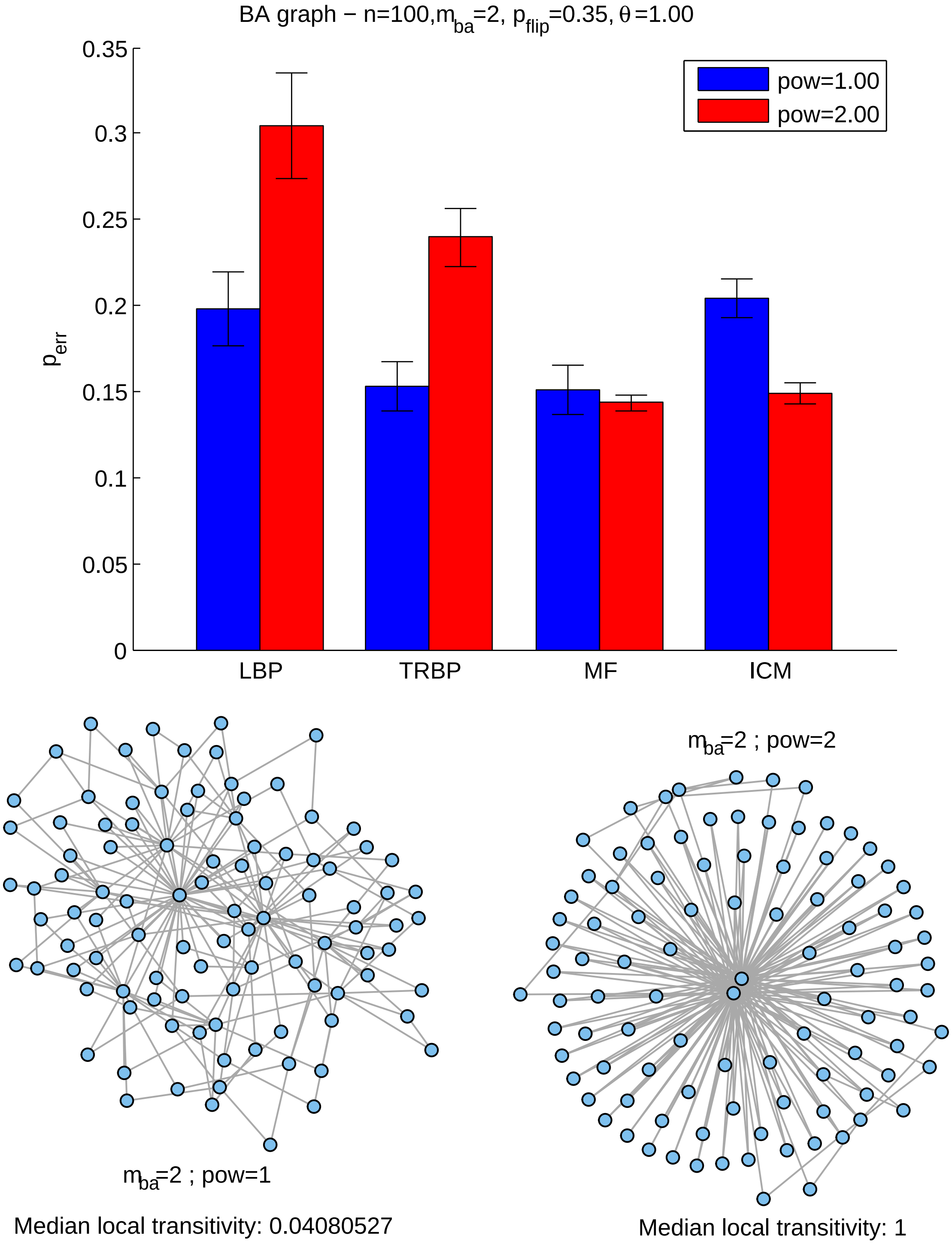
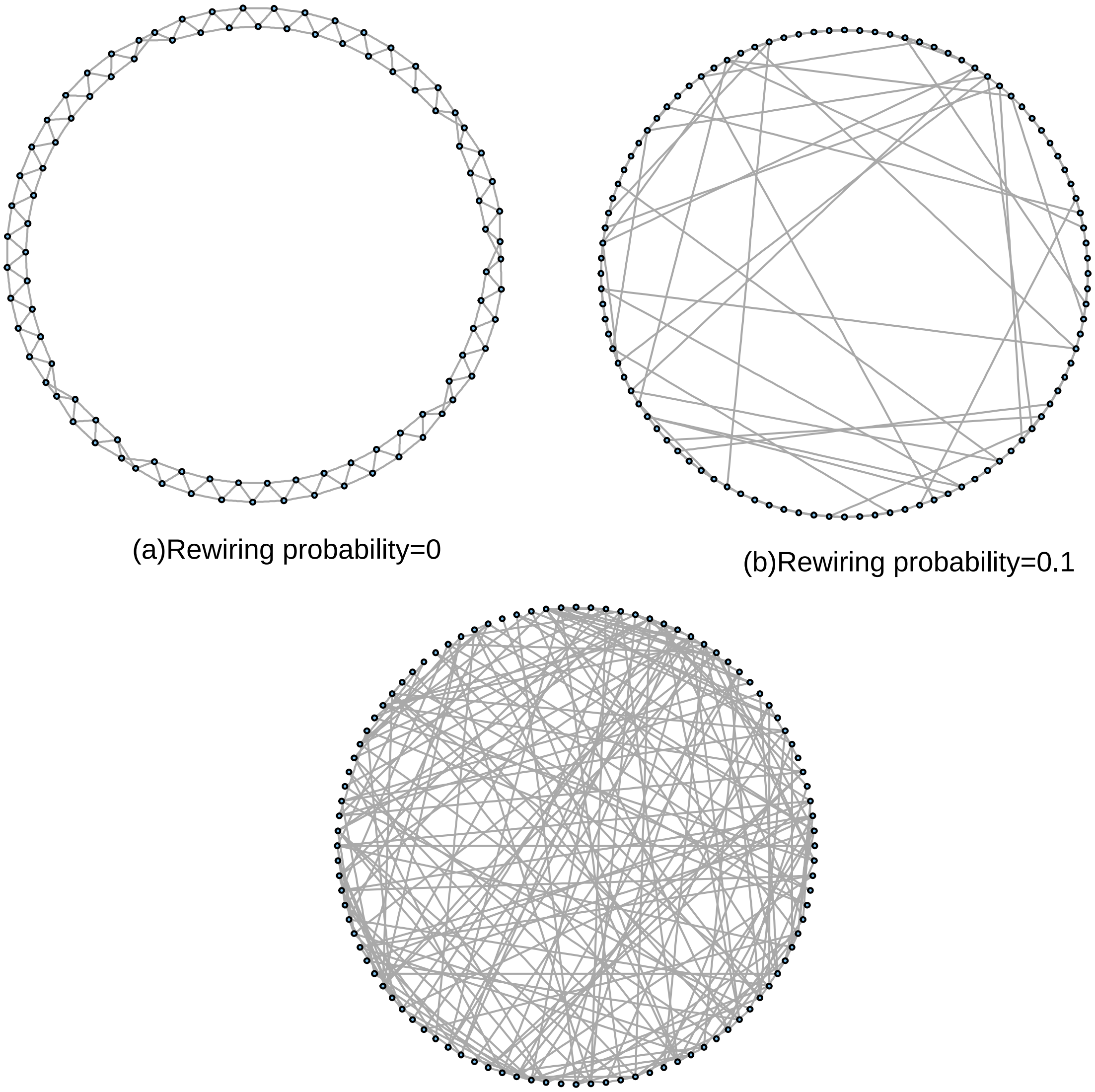


Figure 6.1: Variation of classification error rate with varying power parameter of the BA model (with $n = 100$ and $m_{ba} = 2$) for the 4 inference algorithms considered.



(a)Rewiring probability=0

(b)Rewiring probability=0.1

(c)Rewiring probability=1

Figure 6.2: Figure depicting graphs constructed by using the WS model with differing rewiring probabilities.

The input parameters in to the model are n (the number of nodes), Δ_{mean}, the mean degree (which is assumed to be an even integer), and the re-wiring probability parameter p_{WS}, satisfying $0 \leq p_{WS} \leq 1$ and $n \gg \Delta_{mean} \gg \ln(n) \gg 1$. The model outputs an undirected graph with n nodes and $\frac{n\Delta_{mean}}{2}$ edges using the following two step procedure.

Step 1: Construction of the Regular Lattice

A regular *ring lattice graph* is constructed with n nodes, with each node connected to Δ_{mean} neighbors, with $\Delta_{mean}/2$ on each side. This is as shown in Figure 6.2(a) with $n = 100$ and $\Delta_{mean} = 4$.

Step 2: Rewiring of the regular lattice

For every node $i \in V$, we take every edge (i, j) with $i < j$, and *rewire* it with probability p_{WS}. This *rewiring* is done by replacing (i, j) with (i, k) where the new *destination node k* is chosen with uniform probability from all possible nodes that avoid self-loops ($k \neq i$) and *link duplication*.(That is, there should exists no edge (i, k') with $k' = k$).
Figure 6.2(b) and (c) showcase graphs with the rewiring probabilities being set at $p_{ws} = 0.1$ and $p_{WS} = 1$ respectively. It is to be noted that p_{WS} acts as a control parameter that tunes the randomness of the graph topology from zero-randomness (regular lattice) to completely randomness (in the Erdos-Renyi sense [14]).
The interesting discovery made in [41] is that when the rewiring probability p_{WS} is incrementally increased in a certain small range ($\sim$ 0.01 to 0.1), we see the average path-length (as defined in (6.2)) decrease dramatically on account of these long-range shortcut connections emerging because of the random rewirings, but with the transitivity (defined according to (6.1)) still remaining relatively unchanged. In this range of p_{WS}, the graph is said to be a small-world graph combining the high transitivity characteristic of the lattice graphs with the relatively small average path length of the ER random graphs. This is as shown in Figure 6.3 where $L(p_{WS})$ denotes the average path-length according to (6.2) of the graphs obtained with the rewiring probability fixed at p_{WS} and $C(p_{WS})$ denotes the mean global clustering coefficient according to (6.1) of the graphs obtained with the rewiring probability fixed at p_{WS}.

6.2.4 Empirical Results

The experimental setup used was as follows. To begin with, the stochastic model was chosen (BA or WS) and the parameters of the model were initialized to the appropriate values and n_g graph instances were derived. Now, each of these n_g graph instances were used to specify the Ising prior

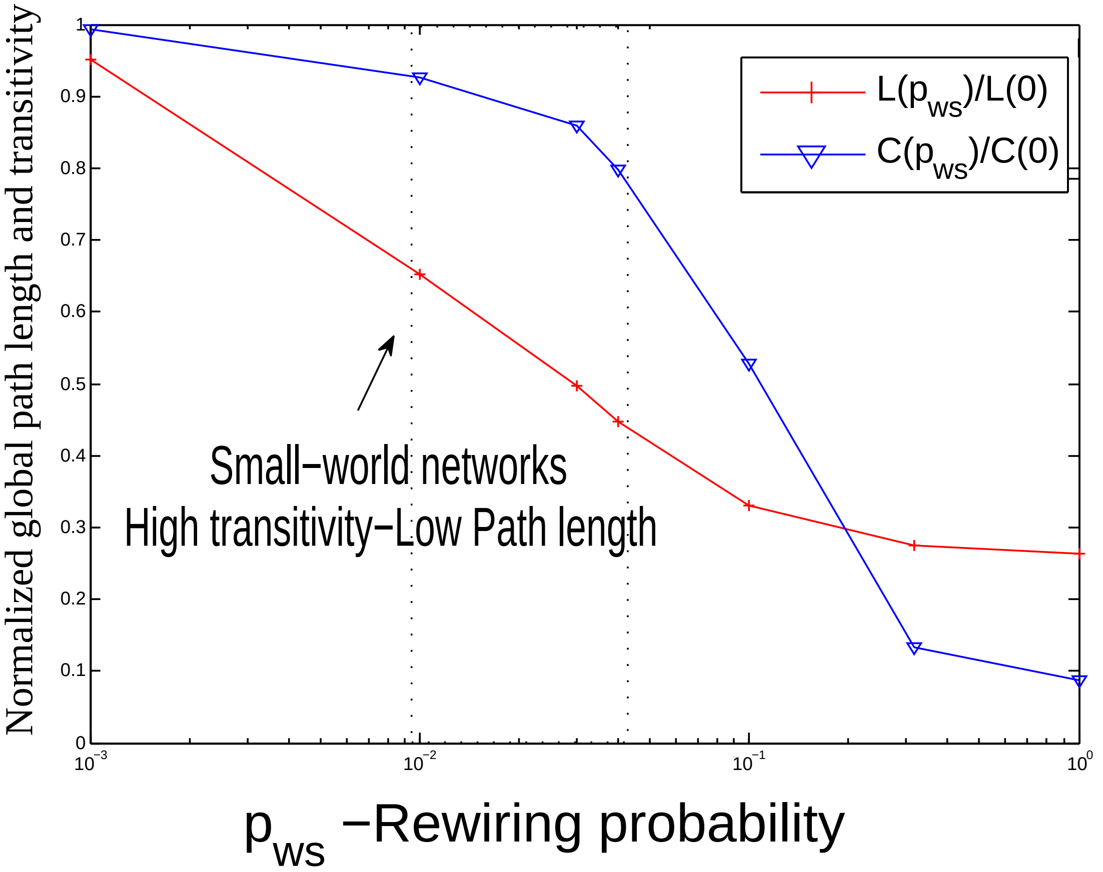


Figure 6.3: Variation of Clustering coefficient and average path length with varying rewiring probability of the WS model (with $n = 100$)

in (1.5) along with the chosen θ and n_s samples were sampled (via Gibb's sampling [147]).Then, these samples were flipped at a rate specified by p_{flip} (or its equivalent ε) in order to simulate the noisy labels derived out of a discriminative classifier with symmetric misclassification rate of p_{flip}. The noisy labels thus derived were used to specify the node potentials of an RFIM (See Chapter-3) and MAP-inference was performed using the following 4 approximate inference algorithms: Loopy Belief propagation (LBP) [158], Tree-Reweighted Belief Propagation (TRBP) [75], Mean Field (MF) [79] and Iterated Conditional Modes (ICM) [39].

Let $\mathbf{x}^{(g,s)}$ denoted the s^{th} Gibbs sample sampled from the Ising prior defined by the g^{th} instance graph derived from the BA/WS model being used. The error rate (p_{err}) was evaluated according to,

$$p_{err} = \frac{\sum\limits_{g=1}^{n_g} \sum\limits_{s=1}^{n_s} \bar{d}_H(\mathbf{x}^{(g,s)}, \hat{\mathbf{x}}^{(g,s)})}{n_g \times n_s}, \tag{6.4}$$

where $\bar{d}_H(\mathbf{x}, \hat{\mathbf{x}})$ is the normalized hamming-distance between the true label $\mathbf{x}$ and its estimate, $\hat{\mathbf{x}}$, defined as,

$$\bar{d}_H(\mathbf{x}, \hat{\mathbf{x}}) = \frac{\sum\limits_{i=1}^{n} \mathbf{I}\,[[x_i \neq \hat{x}_i]]}{n}.$$

Having described the experimental setup, let us now focus on the results derived.

In Figure 6.1, we plot the variation of p_{err} with respect to the preferential attachment power parameter (*pow*) of the BA model. The system parameters were intialized as follows. The number of nodes, $n = 100$, the common edge-potential (θ) was set at $\theta = 1$ and the BSC flipping probability was set at $p_{flip} = 0.35$.

We see that when *pow* is increased from 1 to 2 (linear preferential attachment to quadratic preferential attachment), we see that the mean local clustering increases sharply that results in p_{err} increasing sharply for the Belief Propagation based algorithms (LBP as well as TRBP) while the MF and the ICM algorithms' performance is more robust.

One justification for this degrading of the performance of LBP and TRBP algorithms can be traced back to the introduction of more *loops* (closed circuits) or *generalized loops* in to the underlying graph structured when *pow* is increased, which in SNA terms translates as increase in the local transitivity or clustering levels in the graph. Results in machine learning literature show that BP based inference is exact in *loop free* graphs (trees) [20] while the introduction of loops in to the structure of the underlying graph [159, 160] results in inaccuracies. However, these results are focused on capturing the deviation of the partition function and not on the resultant hamming distance between the true MAP configuration and the approximate MAP configuration predicted by the BP based algorithm. Also, the deviation is captured by a multiplicative factor which is either in the form of *Chertkov and Chernyak loop series* expansion [160] or the bi-variate Watanabe polynomial [159], neither of which have a straightforward interpretation in terms of the global topological measures such as transitivity and path length used by the SNA community. This provides for an exciting avenue of research which would target translating some of the machine learning results (such as those in [159, 160]) in SNA terms.

These results also showcase the fact that it might be better to use simple approximate inference algorithms such as MF and ICM instead of the BP variants when it is known that the underlying graph has BA type characteristics (scale-free degree distribution) along with high local transitivity. This is important as the execution time required by the BP based algorithms can be substantially higher compared to the MF and ICM .In Figure 6.4 we show the variation of decoding time with varying power parameter of the BA model (with $n = 100$ and $m_{ba} = 2$) for the 4 inference algorithms considered. As seen, TRBP requires the largest execution time followed by LBP and then MF and ICM.

Now, we turn our attention to empirical results involving the WS graphs. As in the case of BA graphs, the system parameters were chosen as, $n = 100$ and $p_{flip} = 0.35$. In Figure 6.5, we show the variation of classification error rate with varying rewiring probability of the WS model for the 4 inference algorithms considered with Figure 6.5(a) containing results with $\theta = 0.5$ and Figure 6.5(b) covering the $\theta = 1$ scenario. In each of the two subplots, the rewiring probability p_{WS} was increased from 0.01 to 1, with $p_{WS} \in \{0.01, 0.05, 0.1\}$ covering the phase where the graph exhibits small-world characteristics.

As expected, we see a marked improvement in the performance of the BP based algorithms when p_{WS} was increased which results in the graph becoming more random with lowered local (and global) transitivity. In the strong network effect regime with $\theta = 1$ (Figure 6.5(b)), we see that the TRBP gives the best performance amongst all the inference algorithms thus implying that it might be the inference algorithm of choice when the network effect is strong (large θ) and clustering is low. The reasoning for this can be traced to the work in [20], where the authors show that for this case of pure attractive couplings and a 2-D grid topology, the TRBP bound becomes tight as θ tends to infinity. Also, as seen in the simulation results in [75], TRBP does outperform LBP and MF (albeit in terms of accuracy of the estimated partition function) when the edge strength is above a certain threshold value.

It is also interesting to note that the change in performance for both the BP based variants as well as MF and ICM algorithms was not monotonic with the increase in p_{WS}. This indicates the presence of an underlying topological characteristic beyond transitivity or path-length that results in the performance of these algorithms improving when p_{WS} is increased till a certain p_{WS} and the gradually worsening with increasing p_{WS}. This interesting behavior might well be an avenue of further investigation.

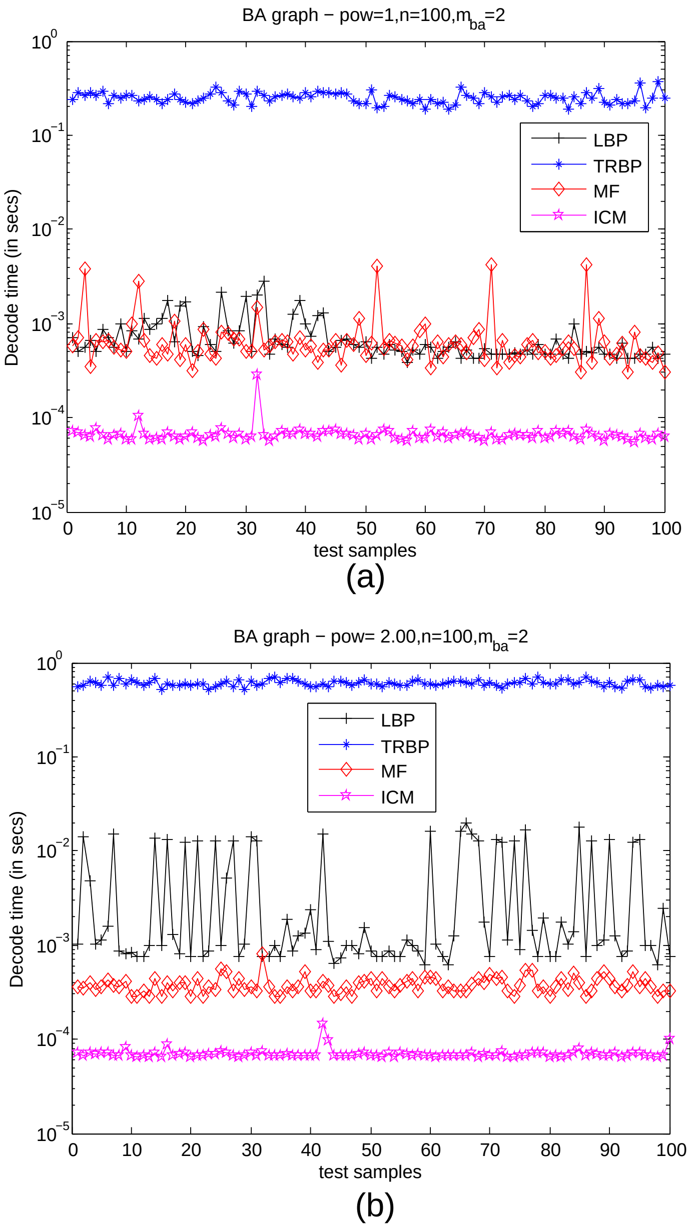


Figure 6.4: Variation of decoding time with varying power parameter of the BA model (with $n = 100$ and $m_{ba} = 2$) for the 4 inference algorithms considered.

With these results, we conclude the thesis.

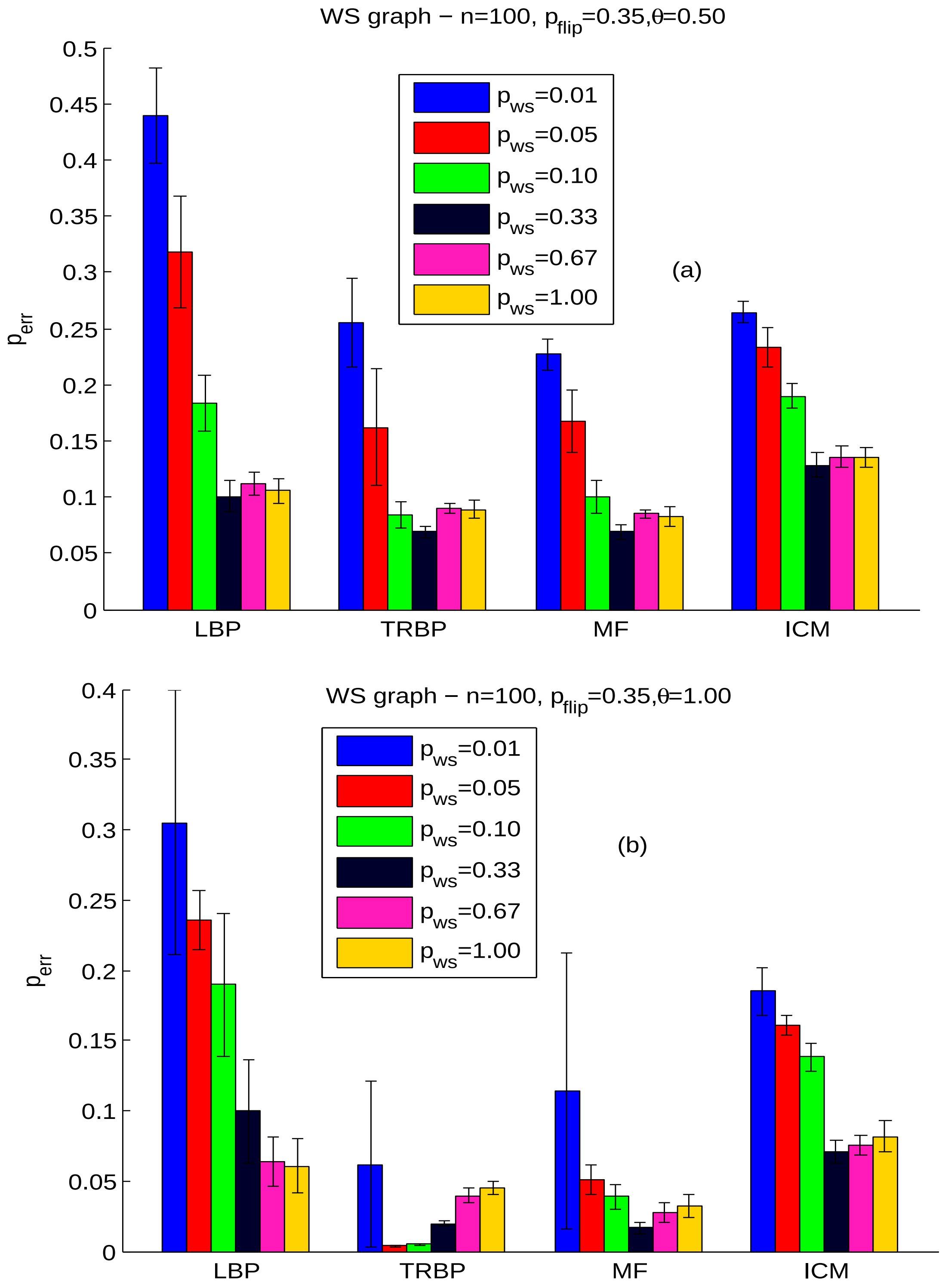


Figure 6.5: Variation of classification error rate with varying rewiring probability of the WS model (with $n = 100$) for the 4 inference algorithms considered.

Appendices

A Proof of Theorem 4.3.1

Following [161], we begin by upper bounding the probability of error in (4.6) by replacing the indicator function with an exponential. That is, using the result

$$\mathbf{I}[[\mathbf{e}^T\mathbf{y} > 0]] \leqslant \exp\left\{b\mathbf{e}^T\mathbf{y}\right\}$$

for $b > 0$ in (4.6) leads to,

$$\begin{aligned} P_e &\leqslant \sum_{\mathbf{y}} p(\mathbf{y}|t=-1)\exp\left\{b\mathbf{e}^T\mathbf{y}\right\} \\ &\leqslant 2\sum_{\mathbf{y}} p(\mathbf{y},t=-1)\exp\left\{b\mathbf{e}^T\mathbf{y}\right\}. \end{aligned} \tag{5}$$

Now, using $p(\mathbf{y},t=-1) = \sum_{\mathbf{x}} p(t=-1,\mathbf{x},\mathbf{y})$ and substituting the joint pdf for $p\left(t,\mathbf{x},\mathbf{y}\right)$ from (1.8) in (5), we have,

$$\begin{aligned} P_e &\leqslant \frac{1}{c^n Z\left(\theta,\gamma\right)}\sum_{\mathbf{y}}\sum_{\mathbf{x}}\exp\left\{\begin{matrix}\theta\mathbf{x}^T A\mathbf{x} - \gamma\mathbf{e}^T\mathbf{x} \\ +\mathbf{y}^T(b\mathbf{e}+\varepsilon\mathbf{x})\end{matrix}\right\} \\ &\leqslant \frac{1}{c^n Z\left(\theta,\gamma\right)}\sum_{\mathbf{x}}\left\{\begin{matrix}\exp\left\{\theta\mathbf{x}^T A\mathbf{x} - \gamma\mathbf{e}^T\mathbf{x}\right\} \\ \sum_{\mathbf{y}}\exp\left\{\mathbf{y}^T(b\mathbf{e}+\varepsilon\mathbf{x})\right\}\end{matrix}\right\}. \end{aligned} \tag{6}$$

where $Z\left(\theta,\gamma\right) = \sum_{\mathbf{x}}\exp\left\{\theta\mathbf{x}^T A\mathbf{x} + \gamma\mathbf{e}^T\mathbf{x}\right\}$.

Now, we notice that the summation $\sum_{\mathbf{y}}\exp\left\{\mathbf{y}^T(b\mathbf{e}+\varepsilon\mathbf{x})\right\}$ is the partition function of an Ising model with an empty network and can be evaluated by the following closed form expression,

$$\sum_{\mathbf{y}}\exp\left\{\mathbf{y}^T(b\mathbf{e}+\varepsilon\mathbf{x})\right\} = 2^n\prod_{i=1}^{n}\cosh(b+\varepsilon x_i).$$

Further rewriting the cosh product term as, $\prod_{i=1}^{n} \cosh(b+\varepsilon x_i) = \left\{ \begin{array}{l} \left(\frac{\cosh(2b)+\cosh(2\varepsilon)}{2}\right)^{n/2} \\ \times \left(\sqrt{\frac{\cosh(b+\varepsilon)}{\cosh(b-\varepsilon)}}\right)^{\mathbf{e}^T\mathbf{x}} \end{array} \right\}$,

we have,

$$\sum_{\mathbf{y}} \exp\left\{\mathbf{y}^T(b\mathbf{e}+\varepsilon\mathbf{x})\right\} = 2^n \times \left\{ \begin{array}{l} \left(\frac{\cosh(2b)+\cosh(2\varepsilon)}{2}\right)^{n/2} \\ \times \left(\sqrt{\frac{\cosh(b+\varepsilon)}{\cosh(b-\varepsilon)}}\right)^{\mathbf{e}^T\mathbf{x}} \end{array} \right\}. \tag{7}$$

Finally, substituting (7) in (6) and simplifying yields,

$$\begin{aligned} P_e &\leq \frac{1}{Z(\theta,\gamma)(\cosh(\varepsilon))^n} \min_b A(b), \quad \text{where,} \\ A(b) &= \left(\frac{\cosh(2b)+\cosh(2\varepsilon)}{2}\right)^{n/2} Z(\theta,\beta), \\ \beta &= \gamma + \frac{1}{2}\log\left(\frac{\cosh(b-\varepsilon)}{\cosh(b+\varepsilon)}\right), \\ Z(\theta,\beta) &= \sum_{\mathbf{x}} \exp\left\{\theta\mathbf{x}^T A\mathbf{x} - \beta\mathbf{e}^T\mathbf{x}\right\}, \end{aligned} \tag{8}$$

which is the RHS of Theorem 4.3.1.

B Proof of Theorem 5.3.1

In this appendix, we prove the HT expansion Theorem 5.3.1. The proof is developed in four stages.

B.1 Stage 1: Using the high temperature character expansion identity

To begin with, let us define $\lambda_{ij} = \tanh(\theta_{ij})$, $\mu_i = \tanh(h_i)$ and

$$c(\boldsymbol{\theta},\mathbf{h}) = \left\{ \prod_{(i,j)\in E} \cosh(\theta_{ij}) \prod_{i\in V} \cosh(h_i) \right\}. \tag{9}$$

Now, using the HT (character) expansion identity of $e^a = cosh(a)(1 + tanh(a))$ in (5.8) and interchanging the order of order of summation, we have,

$$\begin{aligned}
Z_+(\boldsymbol{\theta}, \mathbf{h}) &= \sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}>0} \left\{ \prod_{(i,j)\in E} \mathrm{e}^{\theta_{ij}x_i x_j} \prod_{i\in V} \mathrm{e}^{h_i x_i} \right\} \\
&= \overbrace{\left\{ \prod_{(i,j)\in E} \cosh(\theta_{ij}) \prod_{i\in V} \cosh(h_i) \right\}}^{c(\boldsymbol{\theta},\mathbf{h})} \sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}>0} \left\{ \prod_{(i,j)\in E} \left(1 + \lambda_{ij} x_i x_j\right) \prod_{i\in V} (1 + \mu_i x_i) \right\} \qquad (10) \\
&= c(\boldsymbol{\theta}, \mathbf{h}) \sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}>0} \left\{ \sum_{S\subseteq E} \prod_{(i,j)\in S} \lambda_{ij} x_i x_j \sum_{U\subseteq V} \prod_{i\in U} (\mu_i x_i) \right\}.
\end{aligned}$$

Now, defining $w(S, U)$ as,

$$w\left(S, U\right) = \left[\sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}>0} \left[\prod_{(i,j)\in S} x_i x_j \prod_{v\in U} x_v \right] \right], \qquad (11)$$

we have,

$$Z_+(\boldsymbol{\theta}, \mathbf{h}) = c(\boldsymbol{\theta}, \mathbf{h}) \sum_{S\subseteq E} \sum_{U\subseteq V} \left\{ \prod_{(i,j)\in S} \lambda_{ij} \prod_{i\in U} \mu_i w\left(S, U\right) \right\}. \qquad (12)$$

Now, looking at (12) above, we see that every edge subset-vertex subset combination, (S, U) contributes a certain weight to the overall partial partition function computation. This weight is the product of edge-weight contributions and node-weight contributions captured by ($\prod_{(i,j)\in S} \lambda_{ij} \prod_{i\in U} \mu_i$), multiplied by another weight $w(S, U)$ dependent only on the topological artifacts in the (S, U) combination.

The second stage focuses on evaluating these weights,

$$w\left(S, U\right) = \left[\sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}>0} \left[\prod_{(i,j)\in S} x_i x_j \prod_{v\in U} x_v \right] \right]. \qquad (13)$$

B.2 Stage 2: Computing the weights - $w(U, S)$: The binary code puncturing approach

The first step in evaluating $w(S, U)$ is writing edge-wise product $\prod_{(i,j)\in S} \left\{x_i x_j\right\}$ in terms of a node-wise product. Denoting Δ_i as the degree of node i and $V_{odd}(S) \subseteq V$ to be the subset of odd-degreed nodes in the graph $G(V, S)$, we have,

$$\prod_{(i,j)\in S} x_i x_j = \prod_{i\in V} {x_i}^{\Delta_i} = \prod_{i\in V_{odd}(S)} x_i. \tag{14}$$

We see that the indices pertaining to the even-degree nodes are duly ignored in the product given that $x_i \in \{-1, +1\}$ and $(x_i)^{\Delta_i} = 1$ if Δ_i is even.

Thus, we have,

$$w\left(S, U\right) = \left[\sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}>0} \left[\prod_{i\in V_{odd}(S)} x_i \prod_{v\in U} x_v\right]\right] \tag{15}$$

Now, we see that the product $\left[\prod_{i\in V_{odd}(S)} x_i \prod_{v\in U} x_v\right]$ can be written as,

$$\left[\prod_{i\in V_{odd}(S)} x_i \prod_{v\in U} x_v\right] = \left[\prod_{i=1}^{n-p} x_i\right], \tag{16}$$

where p denotes the number of nodes that have been *punctured out* of the product on account of the fact that there were raised to an even power. Also, we note that the indices of the punctures vertices ($v \in V$) that were punctured out does not matter and only the number of punctures (p) is taken into account.

We see that nodes that will be *punctured out* are those that were either odd-degreed and included in U (denoted as $U_{odd}(S)$) or were even degreed in $V - U$ (denoted as $(V - U)_{even}(S)$). Thus, we have,

$$w\left(S, U\right) = \left[\sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}>0} \left[\prod_{i=1}^{n-p} x_i\right]\right], \tag{17}$$

with $p = (|U_{odd}(S)| + |(V - U)_{even}(S)|)$. Also, denoting $\mathbf{M}_+^{(n)}$ to be the $n \times 2^{n-1}$ matrix, whose columns are the 2^{n-1} vectors from $\{\mathbf{x} \in \{-1, +1\}^n : \mathbf{e}^T\mathbf{x} > 0\}$, we can extract $\mathbf{M}_+^{(n,p)}$ to be the $(n - p) \times 2^{n-1}$ sub-matrix of $\mathbf{M}_+^{(n)}$ obtained by deleting the p rows pertaining to the $p = (|U_{odd}(S)| + |(V - U)_{even}(S)|)$ node that were punctured out.

Now, we have,

$$\sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}>0} \prod_{i\in V_{odd}(S')} x_i = \sum_{\mathbf{x}:iscol\left(\mathbf{x},\mathbf{M}_+^{(n,p)}\right)=1} \prod_{i=1}^{n-p} x_i, \tag{18}$$

where $iscol\left(\mathbf{x}, \mathbf{A}\right)$ is an indicator function that returns 1 if $\mathbf{x}$ is a column of the matrix $\mathbf{A}$. In coding theoretic terms, this matrix represents a *punctured codebook* with p-punctures.

Given that $x_i \in \{-1, +1\}$, the product $\prod_{i=1}^{n-p} x_i$ can taken values only in $\{-1, +1\}$. Simply put, the product $\prod_{i=1}^{n-p} x_i$ just involves checking if the number of of -1s in a given $\mathbf{x}$ is even or odd. That is,

$$\prod_{i=1}^{n-p} x_i = \begin{cases} +1 & \text{if } \sum_{i=1}^{n-p} \mathbf{I} [\![x_i = -1]\!] \text{ is even} \\ -1 & \text{otherwise} \end{cases} \tag{19}$$

Also, the summation $\sum_{\mathbf{x}:iscol\left(\mathbf{x},\mathbf{M}_{+}^{(n,p)}\right)=1} \prod_{i=1}^{n-p} x_i$ involves counting how many of the $\mathbf{x}$'s had even number of -1s in them. Therefore,

$$w(S,U) = \sum_{\mathbf{x}:iscol\left(\mathbf{x},\mathbf{M}_{+}^{(n,p)}\right)=1} \prod_{i=1}^{n-p} x_i = N_{+} - N_{-}, \tag{20}$$

where, $N_{+} = \sum_{\mathbf{x}:iscol\left(\mathbf{x},\mathbf{M}_{+}^{(n,p)}\right)=1} \mathbf{I} \left[\!\left[\prod_{i=1}^{n-p} x_i = +1 \right]\!\right]; N_{-} = 2^{(n-1)} - N_{+}$.

To make things clearer, we switch from the spins notation to the classical binary notation by defining

$$\mathbf{C}_{+}^{(n,p)} = \frac{\mathbf{1} - \mathbf{M}_{+}^{(n,p)}}{2} \tag{21}$$

as the *binarized* punctured codebook matrix. Here, $\mathbf{1}$ is the $(n-p) \times 2^{n-1}$ matrix of all 1's. Thus, in the *binary world*, we can write,

$$w(S,U) = \mathcal{N}_{even} - \mathcal{N}_{odd}, \tag{22}$$

where, $\mathcal{N}_{even} = \sum_{\mathbf{b}:iscol\left(\mathbf{b},\mathbf{C}_{+}^{(n,p)}\right)=1} \mathbf{I} [\![d_H(\mathbf{b}) \in \{2\mathbb{N}\}]\!]; \mathcal{N}_{odd} = 2^{(n-1)} - \mathcal{N}_{even}$ and $d_H(\mathbf{b})$ denotes the hamming weight of the binary vector $\mathbf{b}$.

Now, the problem of calculating $w(S,U)$ can be stated simply as the number of binary vectors (*codewords*) in the p-punctured codebook $\mathbf{C}_{+}^{(n,p)}$ of even weight minus the number of binary vectors of odd weight. Let $\mathfrak{a}_{+}(n,p,i)$ be a function that evaluates the number of codewords of hamming weight i in the codebook $\mathbf{C}_{+}^{(n,p)}$ obtained after p punctures of $\mathbf{C}_{+}^{(n,0)}$ of length n. Then, we have,

$$w(S,U) = \mathcal{N}_{even} - \mathcal{N}_{odd} = \sum_{i=0}^{\left(\frac{n-1}{2}\right)} (-1)^i \mathfrak{a}_{+}(n, |U_{odd}(S)| + |(V-U)_{even}(S)|, i). \tag{23}$$

Evaluating $\mathfrak{a}_+(n, p, i)$

To begin with, let us consider the un-punctured code book $\mathbf{C}_+^{(n,0)} \in \{0,1\}^{n \times 2^{n-1}}$. It is obtained by *expurgating* all codewords of hamming weight greater than $\frac{n-1}{2}$ from the universal codebook $\{0,1\}^n$ and retains the homogeneity property of the universal codebook. Let $\mathbf{C}_+^{(n,p)}$ be the codebook obtained after p punctures of $\mathbf{C}_+^{(n,0)}$. $\forall 0 \leq j \leq 2^{n-1}$, let $\tilde{\mathbf{b}}^{(j)}$ be the j^{th} code-word in $\mathbf{C}_+^{(n,p)}$ obtained by the puncturing of $\mathbf{b}^{(j)}$ from $\mathbf{C}_+^{(n,0)}$.

The following theorem provides for a closed form formula to evaluate $\mathfrak{a}_+(n, p, i)$:

Theorem B.1. *Let $\mathfrak{a}_+(n, p, i)$ be a function that evaluates the number of codewords of hamming weight i in the codebook $\mathbf{C}_+^{(n,p)}$ obtained after p punctures of $\mathbf{C}_+^{(n,0)}$ of length n. Then,*

$$\mathfrak{a}_+(n, p, i) = \binom{n-p}{i} \sum_{k=0}^{\left(\frac{n-1}{2}\right)-i} \binom{p}{k}; i = 0, ..., \frac{n-1}{2}. \tag{24}$$

Proof. To begin with, we see that,

$$\mathfrak{a}_+(n, p, i) = E_\pi \left(\sum_{j=0}^{2^{n-1}} \mathbf{I} \left[\left[d_H(\tilde{\mathbf{b}}^{(j)}) = i \right]\right] \right). \tag{25}$$

Here, the expectation is taken over all possible permutations (π) of the vertices of the graph (or rows of $\mathbf{C}_+^{(n,0)}$) chosen to be punctured. Note that on account of the homogeneity property of the codebook $\mathbf{C}_+^{(n,0)}$, the weight distribution of the punctured code $\mathfrak{a}_+(n, p, i)$ is permutation invariant.

Now, re-writing (25) as a conditional expectation, we have,

$$\mathfrak{a}_+(n, p, i) = \sum_{j=0}^{2^{n-1}} \sum_{k=i}^{\min\left(p+i, \frac{n-1}{2}\right)} \underbrace{E_\pi \left[\mathbf{I} \left[\left[d_H(\tilde{\mathbf{b}}^{(j)}) = i \right]\right] | d_H(\mathbf{b}^{(j)}) = k \right]}_{P\left(d_H(\tilde{\mathbf{b}}^{(j)}) = i | d_H(\mathbf{b}^{(j)}) = k\right)} \mathbf{I} \left[\left[d_H(\mathbf{b}^{(j)}) = k \right]\right], \tag{26}$$

where is the conditional expectation $E_\pi \left[\mathbf{I} \left[\left[d_H(\tilde{\mathbf{b}}^{(j)}) = i \right]\right] | d_H(\mathbf{b}^{(j)}) = k \right]$ is the conditional probability,

$$E_\pi \left[\mathbf{I} \left[\left[d_H(\tilde{\mathbf{b}}^{(j)}) = i \right]\right] | d_H(\mathbf{b}^{(j)}) = k \right] = P\left(d_H(\tilde{\mathbf{b}}^{(j)}) = i | d_H(\mathbf{b}^{(j)}) = k \right). \tag{27}$$

Figure 6 helps explain how the limits of the inner summation in (26) are obtained. Given that puncturing only reduces the hamming weight, it is clear that the lower limit should be i. In order to obtain the upper limit, we look at the *worst-case* scenario where we have for some j, a codeword, $\mathbf{b}^{(j)}$, where all of the p-punctures resulted in removal of 1s thus reducing its hamming weight from $p+i$ to p. Given that the maximum hamming weight in $\mathbf{C}_{+}^{(n,0)}$ is $\frac{n-1}{2}$, we have the upper-limit of the summation to be $min(p+i, \frac{n-1}{2})$. Now, using combinatorics, we see that the

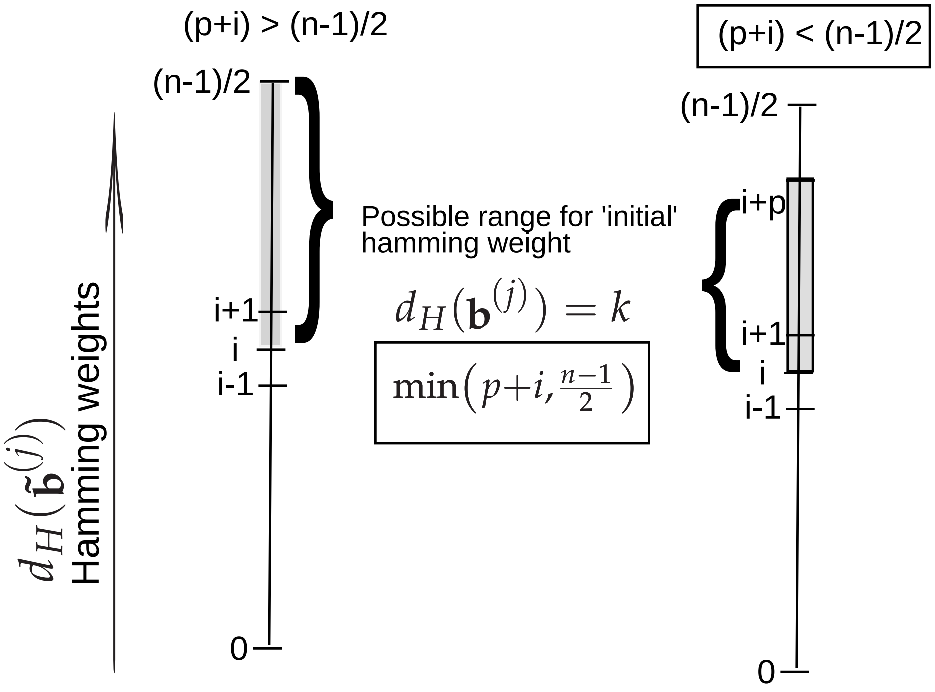

(p+i) > (n-1)/2
(p+i) < (n-1)/2
(n-1)/2
Possible range for 'initial' hamming weight
$d_H(\mathbf{b}^{(j)}) = k$
$\min\left(p+i, \frac{n-1}{2}\right)$
i+1
i
i-1
0
(n-1)/2
i+p
i+1
i
i-1
0
$d_H(\tilde{\mathbf{b}}^{(j)})$
Hamming weights

Figure 6: Possible range of pre-puncturing hamming weights possible for arriving at a post p-punctured codeword, $\tilde{\mathbf{b}}$, with hamming weight $d_H(\tilde{\mathbf{b}}) = i$.

conditional probability $P\left(d_H(\tilde{\mathbf{b}}^{(j)}) = i | d_H(\mathbf{b}^{(j)}) = k\right)$ can be evaluated as,

$$P\left(d_H(\tilde{\mathbf{b}}^{(j)}) = i | d_H(\mathbf{b}^{(j)}) = k\right) = \frac{\binom{k}{k-i}\binom{n-k}{p-(m-i)}}{\binom{n}{p}} \tag{28}$$

Now, substituting (28) in (26), we have,

$$\begin{aligned}\mathfrak{a}_{+}(n,p,i) &= \sum_{j=0}^{2^{n-1}} \sum_{k=i}^{\min\left(p+i,\frac{n-1}{2}\right)} \frac{\binom{k}{k-i}\binom{n-k}{p-(m-i)}}{\binom{n}{p}} \mathbf{I}\left[\left[d_H(\mathbf{b}^{(j)}) = k\right]\right] \\ &= \sum_{k=i}^{\min\left(p+i,\frac{n-1}{2}\right)} \left\{\frac{\binom{k}{k-i}\binom{n-k}{p-(m-i)}}{\binom{n}{p}}\right\} \sum_{j=0}^{2^{n-1}} \mathbf{I}\left[\left[d_H(\mathbf{b}^{(j)}) = k\right]\right] .;\end{aligned} \tag{29}$$

Now, with regard to the un-punctured codebook, $\mathbf{C}_{+}^{(n,0)}$, it is straightforward to see that it has $\binom{n}{k}$ codewords of hamming weight k, $\forall k \in \{0, (n-1)/2\}$. That is,

$$\sum_{j=0}^{2^{n-1}} \mathbf{I}\left[\left[d_H(\mathbf{b}^{(j)}) = k\right]\right] = \binom{n}{k} \tag{30}$$

Now substituting (30) in (29) and simplifying, we have,

$$\begin{aligned}
\mathfrak{a}_{+}(n,p,i) &= \sum_{k=i}^{\min\left(p+i,\frac{n-1}{2}\right)} \left\{ \frac{\binom{k}{k-i}\binom{n-k}{p-(m-i)}}{\binom{n}{p}} \right\} \binom{n}{k} \\
&= \binom{n-p}{i} \sum_{k=i}^{\min\left(p+i,\frac{n-1}{2}\right)} \binom{p}{k-i} \\
&= \binom{n-p}{i} \sum_{k=0}^{\min\left(p,\frac{n-1}{2}-i\right)} \binom{p}{k}.
\end{aligned} \tag{31}$$

In (31), for the binomial coefficient indexed by two nonnegative integers , a and b, that is, $\binom{a}{b}$, assuming that $\binom{a}{b} = 0$ if $b > a$, we have the simplification of the upper limit of the summation as,

$$\mathfrak{a}_{+}(n,p,i) = \binom{n-p}{i} \sum_{k=0}^{\frac{n-1}{2}-i} \binom{p}{k} \tag{32}$$

□

Now, defining the function $\omega(n,p)$ as,

$$\omega\left(n,p\right) = \sum_{i=0}^{\frac{n-1}{2}} (-1)^{i} \binom{n-p}{i} \sum_{k=0}^{\left(\frac{n-1}{2}\right)-i} \binom{p}{k}, \tag{33}$$

and using (32) in (23), we have the following expression for $w(S,U)$,

$$w\left(S,U\right) = \omega\left(n, \left(|U_{odd}(S)| + |(V-U)_{even}(S)|\right)\right). \tag{34}$$

B.3 Stage 3: Establishing the Krawchuk Polynomials/Super-catalan number connection.

For $N \in \mathbb{Z}_{+}, 0 \leqslant k \leqslant N$, the Krawchuk polynomial [152] in a variable ξ is defined by the following equivalent summations,

$$\begin{aligned}\mathcal{K}\left(N,k,\xi\right) &= \sum_{j=0}^{k}(-1)^{j}\left(\begin{array}{c}\xi\\ j\end{array}\right)\left(\begin{array}{c}N-\xi\\ k-j\end{array}\right)\\ &= \sum_{j=0}^{k}(-2)^{j}\left(\begin{array}{c}N-j\\ k-j\end{array}\right)\left(\begin{array}{c}\xi\\ j\end{array}\right)\\ &= \sum_{j=0}^{k}(-1)^{j}2^{k-j}\left(\begin{array}{c}N-k+j\\ j\end{array}\right)\left(\begin{array}{c}N-\xi\\ k-j\end{array}\right).\end{aligned} \tag{35}$$

Given $a, b \in \mathbb{Z}^{*} = \{0\} \cup \mathbb{Z}^{+}$, we define Super-Catalan numbers [153], $\mathcal{S}\left(a,b\right)$ as,

$$\mathcal{S}\left(a,b\right) = \frac{(2a)!\,(2b)!}{a!\,(a+b)!b!}. \tag{36}$$

The authors in [154] derived an interesting relationship between Super-catalan numbers and Krawchuk polynomials captured by,

$$\mathcal{K}\left(2(a+b), a+b, 2a\right) = (-1)^{a}\mathcal{S}\left(a,b\right). \tag{37}$$

Proposition B.2. *The function* $\omega(n,p)$ *as defined in* (5.19) *can be evaluated as,*

$$\omega(n,p) = \begin{cases} (-1)^{\left(\frac{n-p-1}{2}\right)}\dfrac{(n-p-1)!\,(p)!}{\left(\frac{p}{2}\right)!\left(\frac{n-p-1}{2}\right)!\left(\frac{n-1}{2}\right)!} & \textit{if } p \textit{ is even} \quad \text{(38a)}\\ 2^{n-1} & \textit{if } p = n \quad \text{(38b)}\\ 0 & \textit{otherwise} \quad \text{(38c)}\end{cases}$$

Proof. **Proof for eq. (38a)**

To begin with, let us look at (5.19). Using pascal's rule,

$$\left(\begin{array}{c}n-p\\ i\end{array}\right) = \left(\begin{array}{c}n-p-1\\ i\end{array}\right) + \left(\begin{array}{c}n-p-1\\ i-1\end{array}\right), \tag{39}$$

in (5.19) results in,

$$\omega\left(n,p\right)=\sum_{i=0}^{\frac{n-1}{2}}\sum_{k=0}^{\left(\frac{n-1}{2}\right)-i}\left(-1\right)^{i}\left(\begin{array}{c}p\\k\end{array}\right)\left(\left(\begin{array}{c}n-p-1\\i\end{array}\right)+\left(\begin{array}{c}n-p-1\\i-1\end{array}\right)\right). \tag{40}$$

Now, splitting the RHS of (40) into 2 summations, we have,

$$\begin{aligned}\omega\left(n,p\right)&=\sum_{i=0}^{\frac{n-1}{2}}\sum_{k=0}^{\left(\frac{n-1}{2}\right)-i}\left(-1\right)^{i}\left(\begin{array}{c}n-p-1\\i\end{array}\right)\left(\begin{array}{c}p\\k\end{array}\right)+\sum_{i=0}^{\frac{n-1}{2}}\sum_{k=0}^{\left(\frac{n-1}{2}\right)-i}\left(-1\right)^{i}\left(\begin{array}{c}n-p-1\\i-1\end{array}\right)\left(\begin{array}{c}p\\k\end{array}\right)\\&=\sum_{i=0}^{\frac{n-1}{2}}\sum_{k=0}^{\left(\frac{n-1}{2}\right)-i}\left(-1\right)^{i}\left(\begin{array}{c}n-p-1\\i\end{array}\right)\left(\begin{array}{c}p\\k\end{array}\right)+\sum_{i=1}^{\frac{n-1}{2}}\sum_{k=0}^{\left(\frac{n-1}{2}\right)-i}\left(-1\right)^{i}\left(\begin{array}{c}n-p-1\\i-1\end{array}\right)\left(\begin{array}{c}p\\k\end{array}\right).\end{aligned} \tag{41}$$

Now, consider the first summand, t_1,

$$t_1=\sum_{i=0}^{\frac{n-1}{2}}\left\{\sum_{k=0}^{\left(\frac{n-1}{2}\right)-i}\left(-1\right)^{i}\left(\begin{array}{c}n-p-1\\i\end{array}\right)\left(\begin{array}{c}p\\k\end{array}\right)\right\}. \tag{42}$$

Now, splitting the inner summand of t_1 into two parts, the first term for $k=(n-1)/2-i$ and the second for $k \in \{0, 1, ..., (n-1)/2 - i - 1\}$, we have,

$$
\begin{aligned}
t_1 &= \sum_{i=0}^{\frac{n-1}{2}} \left\{ \sum_{k=0}^{\left(\frac{n-1}{2}\right)-i} (-1)^i \binom{n-p-1}{i} \binom{p}{k} \right\} \\
&= \sum_{i=0}^{\frac{n-1}{2}} \left\{ (-1)^i \binom{n-p-1}{i} \binom{p}{\left(\frac{n-1}{2}\right)-i} + \sum_{k=0}^{\left(\frac{n-1}{2}\right)-i-1} (-1)^i \binom{n-p-1}{i} \binom{p}{k} \right\} \\
&= \sum_{i=0}^{\frac{n-1}{2}} (-1)^i \binom{n-p-1}{i} \binom{p}{\left(\frac{n-1}{2}\right)-i} + \sum_{i=0}^{\left(\frac{n-1}{2}\right)} \sum_{k=0}^{\left(\frac{n-1}{2}\right)-(i+1)} (-1)^i \binom{n-p-1}{i} \binom{p}{k}.
\end{aligned}
\tag{43}
$$

Now, with regard to the second term on the RHS of (43), we see that for $i = \frac{n-1}{2}$, the upper limit of the inner summation becomes -1. Hence, we can write,

$$
\sum_{i=0}^{\frac{n-1}{2}} \sum_{k=0}^{\left(\frac{n-1}{2}\right)-(i+1)} (-1)^i \binom{n-p-1}{i} \binom{p}{k} = \sum_{i=0}^{\left(\frac{n-1}{2}\right)-1} \sum_{k=0}^{\left(\frac{n-1}{2}\right)-(i+1)} (-1)^i \binom{n-p-1}{i} \binom{p}{k}.
\tag{44}
$$

Now, defining a new index, $j = i - 1$, we have,

$$
\sum_{i=0}^{\frac{n-1}{2}} \sum_{k=0}^{\left(\frac{n-1}{2}\right)-(i+1)} (-1)^i \binom{n-p-1}{i} \binom{p}{k} = -\sum_{j=1}^{\left(\frac{n-1}{2}\right)} \sum_{k=0}^{\left(\frac{n-1}{2}\right)-j} (-1)^j \binom{n-p-1}{j-1} \binom{p}{k}.
\tag{45}
$$

Finally, substituting (43) and (45) into (41), we establish the following result,

$$
\sum_{i=0}^{\frac{n-1}{2}} (-1)^i \binom{n-p}{i} \sum_{k=0}^{\left(\frac{n-1}{2}\right)-i} \binom{p}{k} = \sum_{j=0}^{\frac{n-1}{2}} (-1)^j \binom{n-p-1}{j} \binom{p}{\left(\frac{n-1}{2}\right)-j}.
\tag{46}
$$

Now, we see that the RHS of (46) can indeed be packaged in the first alternative summation form of the Krawchuk polynomial in (35). Therefore,

$$\omega\left(n,p\right)=\mathcal{K}\left(n-1,\left(\frac{n-1}{2}\right),n-p-1\right). \tag{47}$$

Now notice that as long as p is even, $\varsigma=\frac{n-p-1}{2}\in\mathbb{Z}^{+}$ and we can use the relationship detailed in (37) which connects Krawchuk polynomials to Super-Catalan numbers, and thus we have,

$$\omega\left(n,p\right)=\mathcal{K}\left(n-1,\left(\frac{n-1}{2}\right),n-p-1\right)=(-1)^{\varsigma}\mathcal{S}\left(\varsigma,\left(\frac{n-1}{2}\right)-\varsigma\right), \tag{48}$$

where $\varsigma=\left(\frac{n-p-1}{2}\right)$.
Finally, by using the definition of the Super-Catalan numbers in (36), eq. (38a) follows.

Proof for eq. (38b)

By substituting $p=n$ in the third alternative summation form of the Krawchuk polynomial eq. (38b) follows.

Proof for eq. (38c)

One interesting property of Krawtchouk polynomials is that they satisfy a linear recurrence relation with linear coefficients in every variable [162]. Specifically, we have,

$$(N-x)\mathcal{K}(N,k,x+1)=(N-2k)\mathcal{K}(N,k,x)-x\mathcal{K}(N,k,x-1). \tag{49}$$

Now, substituting $N=n-1$, $k=\frac{n-1}{2}$ and $x=n-p-2$ in (49) above, we have,

$$p\mathcal{K}(n-1,\frac{n-1}{2},n-p-1)=-(n-p-1)\mathcal{K}(n-1,\frac{n-1}{2},n-(p+2)-1). \tag{50}$$

From (50), we gather that in order to $\omega(n,p)=0$ for $p\in\{1,3,5,...,n-2\}$, it is sufficient to prove that $\omega(n,1)=0$ and use the recurrence relationship to prove eq. (5.76c). Now, we proceed to prove that $\omega(n,1)=0$.
From (47), we see that,

$$\omega\left(n,p\right)=\sum_{j=0}^{\frac{n-1}{2}}(-1)^{j}\left(\begin{array}{c}n-p-1\\ j\end{array}\right)\left(\begin{array}{c}p\\ (\frac{n-1}{2})-j\end{array}\right). \tag{51}$$

Now, substituting $p = 1$ in (51) above, we have,

$$\begin{aligned}\omega(n,1) &= \sum_{j=0}^{\frac{n-1}{2}} (-1)^j \binom{n-2}{j} \binom{1}{\left(\frac{n-1}{2}\right) - j} \\ &= (-1)^{\frac{n-3}{2}} \binom{n-2}{\frac{n-3}{2}} + (-1)^{\frac{n-1}{2}} \binom{n-2}{\frac{n-1}{2}} \\ &= 0.\end{aligned} \tag{52}$$

Now, using (52) in (50), eq. (38c) follows. □

B.4 Stage 4: Simplifications based on *oddity* of the cardinality of the vertex subset U

1. **Scenario 1: $|U|$ is even and $U = V_{odd}(S)$**

   If $U = V_{odd}(S)$, we see that $\left[\prod_{i \in V_{odd}(S)} x_i \prod_{v \in U} x_v\right] = 1$, which renders,

   $$w(S,U) = \left[\sum_{\mathbf{x}:\mathbf{e}^T\mathbf{x}>0} \left[\prod_{i \in V_{odd}(S)} x_i \prod_{v \in U} x_v\right]\right] = 2^{n-1} \tag{53}$$

2. **Scenario 2: $|U|$ is even and $U \neq V_{odd}(S)$**

   We show that in this scenario $p = (|U_{odd}(S)| + |(V-U)_{even}(S)|)$ is always odd irrespective of whether $|U_{odd}(S)|$ is odd or even. If $|U_{odd}(S)|$ odd, given that $|V_{odd}(S)|$ is always even, $|(V-U)_{odd}(S)|$ will be odd. This, in turn, renders $|(V-U)_{even}(S)|$ to be even (given that n is assumed to be odd). Thus, $p = (|U_{odd}(S)| + |(V-U)_{even}(S)|)$ is odd.

   Similarly, we can argue that if $|U_{odd}(S)|$ even, given that $|V_{odd}(S)|$ is always even, $|(V-U)_{odd}(S)|$ will also be even. This, in turn renders $|(V-U)_{even}(S)|$ to be odd (given that n is assumed to be odd). Thus,

   $$p = (|U_{odd}(S)| + |(V-U)_{even}(S)|), \tag{54}$$

   is odd is this case too. Now, exploiting eq. (38c), we see that, $w(S,U) = 0$

3. **Scenario 3: $|U|$ is odd**

   Making similar arguments as above, we see that $p = (|U_{odd}(S)| + |(V-U)_{even}(S)|)$ is *even* when $|U|$ is *odd*.

The above arguments imply that the summation over all (S, U) combinations can be split into two summations, one adhering to the case that $U = V_{odd}(S)$ and the other adhering to the case that $|U|$ is odd. That is,

$$\sum_{S\subseteq E}\sum_{U\subseteq V}\left[\prod_{(i,j)\in S}\lambda_{ij}\prod_{i\in U}\mu_i w\left(S,U\right)\right] = Z_1 + Z_2, \tag{55}$$

where,

$$Z_1 = 2^{n-1}\sum_{S\subseteq E}\left[\prod_{(i,j)\in S}\lambda_{ij}\prod_{i\in V_{odd}(S)}\mu_i\right], \tag{56}$$

and

$$Z_2 = \sum_{S\subseteq E}\sum_{\substack{U\subseteq V\\ |U|\in\left\{\mathbb{Z}^+_{odd}\right\}}}\left(\prod_{(i,j)\in S}\lambda_{ij}\prod_{i\in U}\mu_i\right)w(S,U). \tag{57}$$

Finally, combining (56) and (57), we have,

$$Z_+(\boldsymbol{\theta}, \mathbf{h}) = c(\boldsymbol{\theta}, \mathbf{h})\left[Z_1 + Z_2\right] \tag{58}$$

Thus the theorem for the PPF stated in (5.15) stands proved.

C Proof of Lemma 5.7.1

The proof for Lemma 5.7.1 entails following the same steps as in the proof for Theorem 5.3.1, with a minor tweak in the *Stage-2* of the proof, which we explain here.

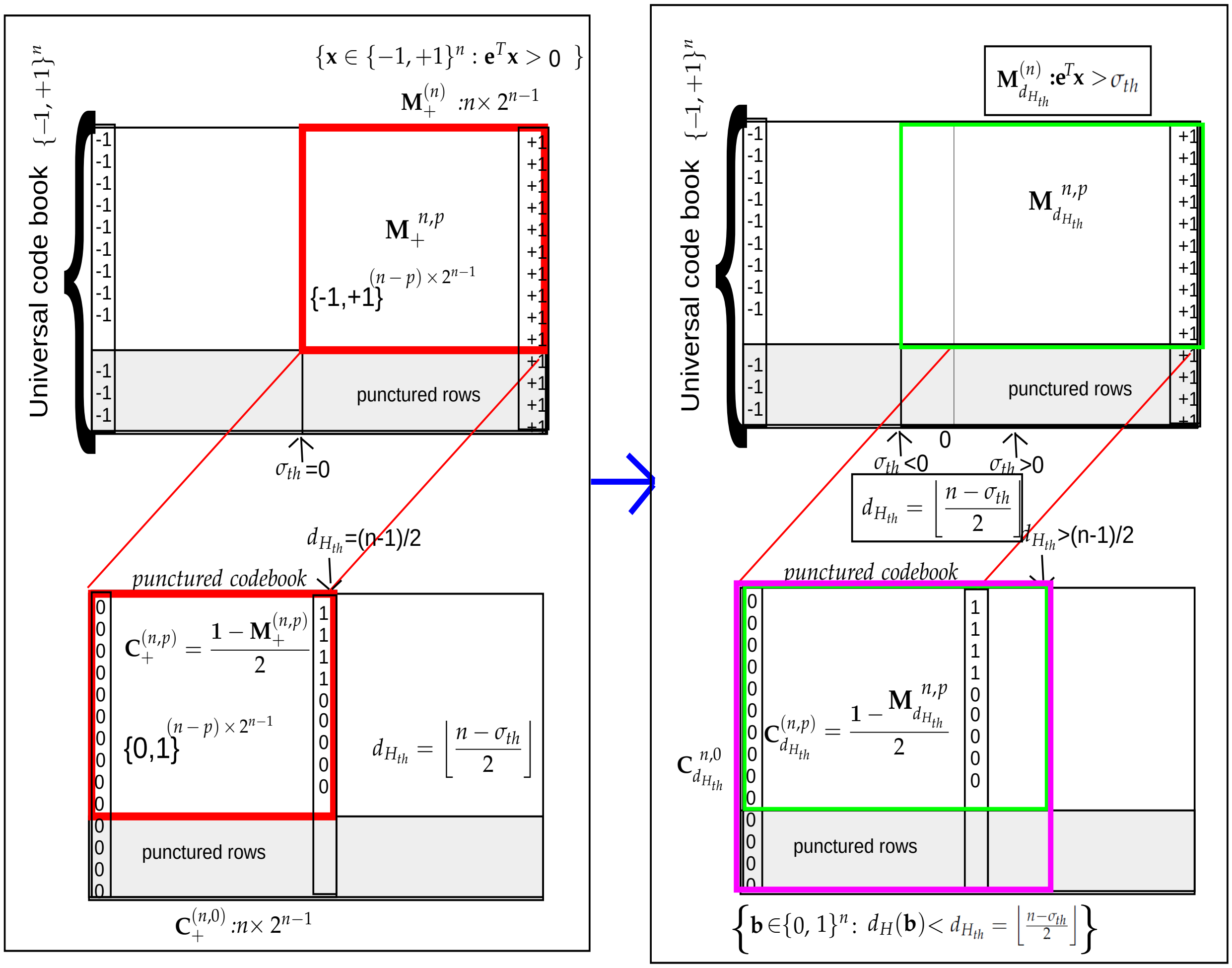
Akin to the proof of Theorem 5.3.1, we begin by considering an un-punctured code book $\mathbf{C}^{(n,0)}_{d_{H_{th}}}$ which is obtained by expurgating all codewords of hamming weight greater than $d_{H_{th}}$ from the universal codebook $\{0,1\}^n$ and which retains the homogeneity property of the universal codebook. Now, let $\mathbf{C}^{(n,p)}_{d_{H_{th}}}$ be the codebook obtained after p punctures of this $\mathbf{C}^{(n,0)}_{d_{H_{th}}}$.

Figure 7 provides a visualization of these codebook matrices. On the left half of the picture, we see the codebooks (highlighted in red) for the case studied earlier in the proof for 5.3.1 (covering the case $\sigma_{th} = 0$). It is easy to see that the threshold sum in the spins representation, σ_{th}, translates as the threshold hamming weight,

$$d_{H_{th}} = \left\lfloor \frac{n - \sigma_{th}}{2} \right\rfloor,$$

in the binary domain.

We note that the parameter $d_{H_{th}}$ (or σ_{th}) acts as a slider variable controlling only the width (number of columns) of the puncture code book $\mathbf{C}^{(n,0)}_{d_{H_{th}}}$ (colored in green in Figure 7), retaining only those codewords of length $(n-p)$ whose hamming weights are in the range of 0 to $d_{H_{th}}$.



Universal code book $\{-1,+1\}^n$
$\{\mathbf{x} \in \{-1,+1\}^n : \mathbf{e}^T\mathbf{x} > 0\}$
$\mathbf{M}^{(n)}_{+}$ $:n\times 2^{n-1}$
$\mathbf{M}_{+}^{n,p}$
$\{-1,+1\}^{(n-p)\times 2^{n-1}}$
-1
+1
punctured rows
σ_{th}=0
$d_{H_{th}}$=(n-1)/2
punctured codebook
$\mathbf{C}^{(n,p)}_{+} = \frac{\mathbf{1}-\mathbf{M}^{(n,p)}_{+}}{2}$
$\{0,1\}^{(n-p)\times 2^{n-1}}$
0
1
$d_{H_{th}} = \left\lfloor \frac{n-\sigma_{th}}{2} \right\rfloor$
punctured rows
$\mathbf{C}^{(n,0)}_{+}$ $:n\times 2^{n-1}$
Universal code book $\{-1,+1\}^n$
$\mathbf{M}^{(n)}_{d_{H_{th}}}$ $:\mathbf{e}^T\mathbf{x} > \sigma_{th}$
$\mathbf{M}_{d_{H_{th}}}^{n,p}$
punctured rows
0
σ_{th}<0
σ_{th}>0
$d_{H_{th}} = \left\lfloor \frac{n-\sigma_{th}}{2} \right\rfloor$
$d_{H_{th}}$>(n-1)/2
punctured codebook
$\mathbf{C}_{d_{H_{th}}}^{n,0}$
$\mathbf{C}^{(n,p)}_{d_{H_{th}}} = \frac{\mathbf{1}-\mathbf{M}_{d_{H_{th}}}^{n,p}}{2}$
punctured rows
$\left\{\mathbf{b} \in \{0,1\}^n : d_H(\mathbf{b}) < d_{H_{th}} = \left\lfloor \frac{n-\sigma_{th}}{2} \right\rfloor\right\}$

Figure 7: Visualization of the codebook matrices in the code-puncturing approach

Now, let us define $\mathfrak{a}_{d_{H_{th}}}(n,p,i)$ as the generalization of the function $\mathfrak{a}_{+}(n,p,i)$, that evaluates the number of codewords of hamming weight i in this codebook $\mathbf{C}^{(n,p)}_{d_{H_{th}}}$ in lieu of $\mathbf{C}^{(n,p)}_{+}$. Harnessing the same ideas as in Theorem B.1, we see that the closed form expression for $\mathfrak{a}_{d_{H_{th}}}(n,p,i)$

simply entails changing the limit in the summation in order to incorporate the revised threshold hamming weight, $d_{H_{th}}$, being set to value other than $(n-1)/2$, leading to,

$$\mathfrak{a}_{d_{H_{th}}}(n,p,i) = \left(\begin{array}{c} n-p \\ i \end{array} \right) \sum_{k=0}^{d_{H_{th}}-i} \left(\begin{array}{c} p \\ k \end{array} \right). \tag{59}$$

Now, computing the generalized version of $\omega(n,p)$, denoted by $\omega_{\sigma_{th}}(n,p)$, is still the difference in the number of binary vectors (*codewords*) in the p-punctured codebook $\mathbf{C}_{d_{H_{th}}}^{(n,p)}$ of even weight and the number of binary vectors of odd weight. That is, using (59), we get,

$$\omega_{\sigma_{th}}(n,p) = \sum_{i=0}^{\left\lfloor \frac{n-\sigma_{th}}{2} \right\rfloor} \left\{ (-1)^i \left(\begin{array}{c} n-p \\ i \end{array} \right) \sum_{k=0}^{\left\lfloor \frac{n-\sigma_{th}}{2} \right\rfloor - i} \left(\begin{array}{c} p \\ k \end{array} \right) \right\},$$

which completes the proof.

Bibliography

[1] C. Levallois, “Umigon: sentiment analysis for tweets based on lexicons and heuristics,” in *Proceedings of the International Workshop on Semantic Evaluation, SemEval*, vol. 13, 2013.

[2] A. Sathi, *Big Data analytics: disruptive technologies for changing the game*. Mc Press, 2012.

[3] D. Bollier and C. M. Firestone, *The promise and peril of big data*. Aspen Institute, Communications and Society Program Washington, DC, USA, 2010.

[4] B. Brown, M. Chui, and J. Manyika, “Are you ready for the era of big data,” *McKinsey Quarterly*, vol. 4, pp. 24–35, 2011.

[5] [Online]. Available: https://www.facebook.com/

[6] [Online]. Available: https://twitter.com/

[7] [Online]. Available: http://disqus.com/

[8] [Online]. Available: www.wikipedia.org/

[9] [Online]. Available: citeseerx.ist.psu.edu/

[10] [Online]. Available: https://data.ny.gov/

[11] [Online]. Available: http://www.data.gov/

[12] M. Janssen, Y. Charalabidis, and A. Zuiderwijk, “Benefits, adoption barriers and myths of open data and open government,” *Information Systems Management*, vol. 29, no. 4, pp. 258–268, 2012.

[13] M. Newman, *Networks: an introduction*. Oxford University Press, 2010.

[14] T. G. Lewis, *Network science: Theory and applications*. John Wiley & Sons, 2011.

[15] D. Easley and J. Kleinberg, *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.

[16] [Online]. Available: http://dailycaller.com/2012/03/23/conservatives-hijack-ilikeobamacare-hashtag-on-twitter/

[17] [Online]. Available: http://www.gpo.gov/fdsys/pkg/PLAW-111publ148/html/PLAW-111publ148.htm

[18] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, pp. 415–444, 2001.

[19] D. Easley and J. Kleinberg, *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.

[20] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, no. 1-2, pp. 1–305, 2008.

[21] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[22] J. M. Moura, J. Lu, and M. Kleiner, "Intelligent sensor fusion: a graphical model approach." in *ICASSP (6)*, 2003, pp. 733–736.

[23] S. L. Lauritzen, *Graphical models*. Oxford University Press, 1996.

[24] B. A. Cipra, "An introduction to the ising model," *American Mathematical Monthly*, vol. 94, no. 10, pp. 937–959, 1987.

[25] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou, and P. Li, "User-level sentiment analysis incorporating social networks," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 1397–1405.

[26] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, 2010, pp. 36–44.

[27] H. Saif, Y. He, and H. Alani, “Alleviating data sparsity for twitter sentiment analysis,” in *The 2nd Workshop on Making Sense of Microposts*, 2012.

[28] V. Prabhu, R. Negi, and M. Rodrigues, “Bipartisan cloture roll call vote prediction using the joint press release network in us senate,” in *ICML workshop on Structured Learning: (SLG 2013)*, 2013.

[29] ——, “Community-based network aided lung and bronchus cancer classification,” 2013, poster presented at Workshop on Statistics for Complex Networks: Theory and Applications , January 30 - February 1, 2013 Eindhoven, The Netherlands.

[30] [Online]. Available: ftp://ftp.fu-berlin.de/pub/misc/movies/database/

[31] G. Salton and M. J. McGill, “Introduction to modern information retrieval,” 1983.

[32] [Online]. Available: http://www.census.gov/main/www/access.html

[33] L. Sandoval and I. D. P. Franca, “Correlation of financial markets in times of crisis,” *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 1, pp. 187–208, 2012.

[34] M. J. Bayarri and J. O. Berger, “The interplay of bayesian and frequentist analysis,” *Statistical Science*, pp. 58–80, 2004.

[35] A. Gelman and C. R. Shalizi, “Philosophy and the practice of bayesian statistics,” *British Journal of Mathematical and Statistical Psychology*, vol. 66, no. 1, pp. 8–38, 2013.

[36] L. Breiman *et al.*, “Statistical modeling: The two cultures (with comments and a rejoinder by the author),” *Statistical Science*, vol. 16, no. 3, pp. 199–231, 2001.

[37] J. Marroquin, S. Mitter, and T. Poggio, “Probabilistic solution of ill-posed problems in computational vision,” *Journal of the American Statistical Association*, vol. 82, no. 397, pp. 76–89, 1987.

[38] [Online]. Available: http://www.people-press.org/2013/09/16/as-health-care-law-proceeds-opposition-and-uncertainty-persist/

[39] J. E. Besag, “On the statistical analysis of dirty pictures,” *Journal of the Royal Statistical Society, Series B*, vol. 48, pp. 259–302, 1986.

[40] P. Erdős and A. Rényi, "On random graphs," *Publicationes Mathematicae Debrecen*, vol. 6, pp. 290–297, 1959.

[41] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-worldnetworks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[42] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[43] D. Lusher, J. Koskinen, and G. Robins, *Exponential random graph models for social networks: Theory, methods, and applications*. Cambridge University Press, 2012.

[44] M. E. Newman, "Clustering and preferential attachment in growing networks," *Physical Review E*, vol. 64, no. 2, p. 025102, 2001.

[45] M. E. Newman, S. H. Strogatz, and D. J. Watts, "Random graphs with arbitrary degree distributions and their applications," *Physical Review E*, vol. 64, no. 2, p. 026118, 2001.

[46] M. E. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

[47] J. Kleinberg, "The small-world phenomenon: an algorithm perspective," in *Proceedings of the thirty-second annual ACM symposium on Theory of computing*. ACM, 2000, pp. 163–170.

[48] P. Bonacich, "Technique for analyzing overlapping memberships," *Sociological methodology*, vol. 4, pp. 176–185, 1972.

[49] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: bringing order to the web." 1999.

[50] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.

[51] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[52] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 44–54.

[53] E. M. Jin, M. Girvan, and M. E. Newman, "Structure of growing social networks," *Physical review E*, vol. 64, no. 4, p. 046132, 2001.

[54] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[55] R. R. Sarukkai, "Link prediction and path analysis using markov chains," *Computer Networks*, vol. 33, no. 1, pp. 377–386, 2000.

[56] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 2, 2007.

[57] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *The Journal of Machine Learning Research*, vol. 11, pp. 985–1042, 2010.

[58] F. Chierichetti, R. Kumar, S. Lattanzi, M. Mitzenmacher, A. Panconesi, and P. Raghavan, "On compressing social networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 219–228.

[59] H. Maserrat and J. Pei, "Neighbor query friendly compression of social networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 533–542.

[60] I. Tollis, P. Eades, G. Di Battista, and L. Tollis, *Graph drawing: algorithms for the visualization of graphs*. Prentice Hall New York, 1998, vol. 1.

[61] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Information processing letters*, vol. 31, no. 1, pp. 7–15, 1989.

[62] E. M. Reingold and J. S. Tilford, “Tidier drawings of trees,” *Software Engineering, IEEE Transactions on*, no. 2, pp. 223–228, 1981.

[63] T. M. Fruchterman and E. M. Reingold, “Graph drawing by force-directed placement,” *Software: Practice and experience*, vol. 21, no. 11, pp. 1129–1164, 1991.

[64] S. Kar and J. M. Moura, “Gossip and distributed kalman filtering: weak consensus under weak detectability,” *Signal Processing, IEEE Transactions on*, vol. 59, no. 4, pp. 1766–1784, 2011.

[65] A. G. Dimakis, S. Kar, J. M. Moura, M. G. Rabbat, and A. Scaglione, “Gossip algorithms for distributed signal processing,” *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.

[66] D. Shah, *Gossip algorithms*. Now Publishers Inc, 2009.

[67] A. Anandkumar, L. Tong, and A. Swami, “Detection of gauss–markov random fields with nearest-neighbor dependency,” *Information Theory, IEEE Transactions on*, vol. 55, no. 2, pp. 816–827, 2009.

[68] K. Binder, “Ising model,” in *Encyclopedia of Mathematics*. Springer Publishers, 2001, iSBN 9781-55608.

[69] E. Schneidman, M. J. Berry, R. Segev, and W. Bialek, “Weak pairwise correlations imply strongly correlated network states in a neural population,” *Nature*, vol. 440, no. 7087, pp. 1007–1012, 2006.

[70] G. Casella and R. L. Berger, *Statistical inference*. Duxbury Pacific Grove, CA, 2002, vol. 2.

[71] E. L. Lehmann and J. P. Romano, *Testing statistical hypotheses*. Springer Science & Business Media, 2006.

[72] S. M. Kay, *Fundamentals of statistical signal processing, Volume III: practical algorithm development*. Pearson Education, 2013, vol. 3.

[73] N. N. Schraudolph and D. Kamenetsky, “Efficient exact inference in planar ising models,” *arXiv:0810.4401*, 2008.

[74] P. W. Kasteleyn, "Dimer statistics and phase transitions," *Journal of Mathematical Physics*, vol. 4, p. 287, 1963.

[75] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "A new class of upper bounds on the log partition function," *Information Theory, IEEE Transactions on*, vol. 51, no. 7, pp. 2313–2335, 2005.

[76] ——, "Tree-reweighted belief propagation algorithms and approximate ml estimation by pseudo-moment matching."

[77] "Factfinder2 - census.gov," 2014. [Online]. Available: http://factfinder2.census.gov/faces/nav/jsf/pages/index.xhtml

[78] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83–85, 2005.

[79] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[80] L. Getoor and B. Taskar, *Introduction to statistical relational learning*, 2007.

[81] S. Chakrabarti, B. Dom, and P. Indyk, "Enhanced hypertext categorization using hyperlinks," in *ACM SIGMOD Record*, vol. 27, no. 2. ACM, 1998, pp. 307–318.

[82] D. Jensen, J. Neville, and B. Gallagher, "Why collective inference improves relational classification," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 593–598.

[83] J. Neville and D. Jensen, "Iterative classification in relational data," in *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, 2000, pp. 13–20.

[84] B. Taskar, P. Abbeel, and D. Koller, "Discriminative probabilistic models for relational data," in *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2002, pp. 485–492.

[85] Q. Lu and L. Getoor, "Link-based classification," in *International Conference on Machine Learning (ICML)*, 2003, p. 496503.

[86] S. A. Macskassy and F. Provost, “Classification in networked data: A toolkit and a univariate case study,” *The Journal of Machine Learning Research*, vol. 8, pp. 935–983, 2007.

[87] Z. L. Stan, “Markov random field modeling in computer vision,” 1995.

[88] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, “Collective classification in network data,” *AI magazine*, vol. 29, no. 3, p. 93, 2008.

[89] Z. Zhou, R. Leahy, and J. Qi, “Approximate maximum likelihood hyperparameter estimation for gibbs priors,” *Image Processing, IEEE Trans. on*, vol. 6, no. 6, pp. 844–861, 1997.

[90] M. Mechelke and M. Habeck, “Calibration of boltzmann distribution priors in bayesian data analysis,” *Physical Review E*, vol. 86, no. 6, p. 066705, 2012.

[91] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992, pp. 144–152.

[92] D. F. Specht, “Probabilistic neural networks,” *Neural networks*, vol. 3, no. 1, pp. 109–118, 1990.

[93] S. E. Shimony, “Finding maps for belief networks is np-hard,” *Artificial Intelligence*, vol. 68, no. 2, pp. 399–410, 1994.

[94] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, “Collective classification in network data,” *AI magazine*, vol. 29, no. 3, p. 93, 2008.

[95] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge university press Cambridge, 2008, vol. 1.

[96] S. A. Macskassy and F. Provost, “Classification in networked data: A toolkit and a univariate case study,” *The Journal of Machine Learning Research*, vol. 8, pp. 935–983, 2007.

[97] [Online]. Available: http://www.acm.org/about/class/ccs98-html

[98] [Online]. Available: http://www2.census.gov/geo/docs/maps-data/maps/reg_div.txt

[99] S. A. Binder and S. S. Smith, *Politics or Principle?: Filibustering in the United States Senate*. Brookings Institution Press, 1997.

[100] "Precedence of motions (rule xxii), United States Senate," January 2010.

[101] R. A. Arenberg and R. B. Dove, *Defending the Filibuster: The Soul of the Senate*. Indiana University Press, 2012.

[102] [Online]. Available: http://www.senate.gov/

[103] H. Nishimori, *Statistical physics of spin glasses and information processing: an introduction*. Clarendon Press, 2001, vol. 111.

[104] M. Schmidt, "Ugm: A matlab toolbox for probabilistic undirected graphical models," 2011.

[105] [Online]. Available: http://www.cdc.gov/

[106] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[107] P. N. Howard and M. M. Hussain, *Democracy's Fourth Wave?: Digital Media and the Arab Spring*. Oxford Univ. Press, 2013.

[108] [Online]. Available: http://www.statisticbrain.com/twitter-statistics/

[109] H. Saif, Y. He, and H. Alani, "Semantic sentiment analysis of twitter," in *The Semantic Web–ISWC 2012*. Springer, 2012, pp. 508–524.

[110] M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge, "Twitter polarity classification with label propagation over lexical links and the follower graph," in *Proceedings of the First workshop on Unsupervised Learning in NLP*. Association for Computational Linguistics, 2011, pp. 53–63.

[111] B. A. Cipra, "The ising model is np-complete," *SIAM News*, vol. 33, no. 6, pp. 07–00, 2000.

[112] M. Kochmański, T. Paszkiewicz, and S. Wolski, "Curie–weiss magneta simple model of phase transition," *European Journal of Physics*, vol. 34, no. 6, p. 1555, 2013.

[113] M. Mezard and A. Montanari, *Information, physics, and computation*. Oxford University Press, 2009.

[114] N. N. Schraudolph and D. Kamenetsky, "Efficient exact inference in planar ising models," in *Advances in Neural Information Processing Systems*, 2009, pp. 1417–1424.

[115] B. Flach, "A class of random fields on complete graphs with tractable partition function," *arXiv:1212.2136*, 2012.

[116] H. Cantril, "Gauging public opinion," 1944.

[117] W. Lippmann, *Public opinion*. Transaction Publishers, 1946.

[118] J. Manza, F. L. Cook, and B. I. Page, *Navigating public opinion: Polls, policy, and the future of American democracy*. Oxford University Press, 2002.

[119] M. A. Smith, *American business and political power: public opinion, elections, and democracy*. University of Chicago Press, 2000.

[120] C. J. Neely, "Technical analysis in the foreign exchange market: a layman's guide," *Federal Reserve Bank of St. Louis Review*, vol. 79, no. September/October 1997, 1997.

[121] J. D. Clinton and A. Meirowitz, "Integrating voting theory and roll call analysis: a framework," *Political Analysis*, vol. 11, no. 4, pp. 381–396, 2003.

[122] B. Liu, *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012, vol. 16.

[123] T. M. Liggett, *Interacting particle systems*. Springer Science & Business Media, 2006.

[124] F. P. Preparata, G. Metze, and R. T. Chien, "On the connection assignment problem of diagnosable systems," *Electronic Computers, IEEE Transactions on*, no. 6, pp. 848–854, 1967.

[125] E. Mossel, J. Neeman, and O. Tamuz, "Majority dynamics and aggregation of information in social networks," *Autonomous Agents and Multi-Agent Systems*, vol. 28, no. 3, pp. 408–429, 2014.

[126] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.

[127] G. B. Mertzios, S. E. Nikoletseas, C. L. Raptopoulos, and P. G. Spirakis, “Determining majority in networks with local interactions and very small local memory,” in *Automata, Languages, and Programming*. Springer, 2014, pp. 871–882.

[128] B. Golub and M. O. Jackson, “Naive learning in social networks and the wisdom of crowds,” *American Economic Journal: Microeconomics*, pp. 112–149, 2010.

[129] M. Banton, “Social capital as a source of majority sentiment,” *Human Figurations*, vol. 2, no. 2, 2013.

[130] M. Baker and J. Wurgler, “Investor sentiment in the stock market,” 2007.

[131] B. Canes-Wrone, T. S. Clark, and J. P. Kelly, “Judicial selection and death penalty decisions,” *American Political Science Review*, vol. 108, no. 01, pp. 23–39, 2014.

[132] [Online]. Available: http://www.pewresearch.org

[133] [Online]. Available: www.rand.org/

[134] [Online]. Available: http://www.brookings.edu/

[135] B. A. Huberman, D. M. Romero, and F. Wu, “Social networks that matter: Twitter under the microscope,” *Available at SSRN 1313405*, 2008.

[136] Y. Takhteyev, A. Gruzd, and B. Wellman, “Geography of twitter networks,” *Social networks*, vol. 34, no. 1, pp. 73–81, 2012.

[137] M. S. Granovetter, “The strength of weak ties,” *American journal of sociology*, pp. 1360–1380, 1973.

[138] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, “Sentiment analysis of twitter data,” in *Proceedings of the Workshop on Languages in Social Media*. Association for Computational Linguistics, 2011, pp. 30–38.

[139] R. Negi, V. Prabhu, and M. Rodrigues, “Latent sentiment detection in online social networks: A communications-oriented view,” in *Communications (ICC), 2014 IEEE International Conference on*, June 2014, pp. 3758–3763.

[140] R. G. Gallager, *Stochastic processes: theory for applications*. Cambridge University Press, 2013.

[141] M. Jerrum and A. Sinclair, *Polynomial-time approximation algorithms for the Ising model*. Springer, 1990.

[142] L. Lovász, "Random walks on graphs: A survey."

[143] J. Jonasson, "Lollipop graphs are extremal for commute times."

[144] Y. Iwamasa and N. Masuda, "Networks maximizing the consensus time of voter models," *Physical Review E*, vol. 90, no. 1, p. 012816, 2014.

[145] M. Southworth and E. Ben-Joseph, *Streets and the Shaping of Towns and Cities*. Island Press, 2003.

[146] A. Cardillo, S. Scellato, V. Latora, and S. Porta, "Structural properties of planar graphs of urban street patterns," *Physical Review E*, vol. 73, no. 6, p. 066107, 2006.

[147] M. Schmidt, "Ugm: Matlab code for undirected graphical models." [Online]. Available: http://www.di.ens.fr/

[148] D. J. Hand, "Statistics and data mining: intersecting disciplines," *ACM SIGKDD Explorations Newsletter*, vol. 1, no. 1, pp. 16–19, 1999.

[149] A. G. T. Jaakkola, "Approximate inference using planar graph decomposition," in *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*, vol. 19. Mit Press, 2007, p. 473.

[150] R. Diestel, *Graph Theory: Springer Graduate Text GTM 173*. Reinhard Diestel, 2012, vol. 173.

[151] N. Streib, A. Streib, I. Beichl, and F. Sullivan, "A binomial approximation method for the ising model," *Journal of Statistical Physics*, pp. 1–13, 2014.

[152] V. I. Levenshtein, "Krawtchouk polynomials and universal bounds for codes and designs in hamming spaces," *Information Theory, IEEE Transactions on*, vol. 41, no. 5, pp. 1303–1321, 1995.

[153] E. Catalan, “Question 1135,” *Nouvelles annales de mathématiques: Journal des candidats aux écoles polytechniques et normales, Series*, vol. 2, no. 13, p. 207, 1874.

[154] E. Georgiadis, A. Munemasa, and H. Tanaka, “A note on super catalan numbers,” *Interdisciplinary Information Sciences*, vol. 18, no. 1, pp. 23–24, 2012.

[155] [Online]. Available: https://oeis.org/A008949

[156] E. Allen and I. Gheorghiciuc, “A weighted interpretation for the super catalan numbers,” *arXiv preprint arXiv:1403.5246*, 2014.

[157] X. Chen and J. Wang, “The super catalan numbers $s(m, m+s)$ for $s \leq 4$,” *arXiv preprint arXiv:1208.4196*, 2012.

[158] J. Pearl, “Reverend bayes on inference engines: A distributed hierarchical approach,” in *AAAI*, 1982, pp. 133–136.

[159] Y. Watanabe and K. Fukumizu, “New graph polynomials from the bethe approximation of the ising partition function,” *Combinatorics, Probability and Computing*, vol. 20, no. 02, pp. 299–320, 2011.

[160] M. Chertkov and V. Y. Chernyak, “Loop series for discrete statistical models on graphs,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2006, no. 06, p. P06009, 2006.

[161] R. Gallager, “A simple derivation of the coding theorem and some applications,” *Information Theory, IEEE Transactions on*, vol. 11, no. 1, pp. 3–18, 1965.

[162] I. Krasikov and S. Litsyn, “On integral zeros of krawtchouk polynomials,” *journal of combinatorial theory, Series A*, vol. 74, no. 1, pp. 71–99, 1996.