ROBUST AND ADVERSARIAL DATA MINING

A thesis submitted in fulfilment of the requirements for the
degree of Doctor of Philosophy in the School of Information Technologies at
The University of Sydney

Fei Wang
August 2015

Abstract

In the domain of data mining and machine learning, researchers have made significant contributions in developing algorithms handling clustering and classification problems. We develop algorithms under assumptions that are not met by previous works. (i) In adversarial learning, which is the study of machine learning techniques deployed in non-benign environments. We design an algorithm to show how a classifier should be designed to be robust against sparse adversarial attacks. Our main insight is that sparse feature attacks are best defended by designing classifiers which use ℓ_1 regularizers. (ii) The different properties between ℓ_1 (Lasso) and ℓ_2 (Tikhonov or Ridge) regularization has been studied extensively. However, given a data set, principle to follow in terms of choosing the suitable regularizer is yet to be developed. We use mathematical properties of the two regularization methods followed by detailed experimentation to understand their impact based on four characteristics. (iii) The identification of anomalies is an inherent component of knowledge discovery. In lots of cases, the number of features of a data set can be traced to a much smaller set of features. We claim that algorithms applied in a latent space are more robust. This can lead to more accurate results, and potentially provide a natural medium to explain and describe outliers. (iv) We also apply data mining techniques on health care industry. In a lot cases, health insurance companies cover unnecessary costs carried out by healthcare providers. The potential adversarial behaviors of surgeon physicians are addressed. We describe a specific context of private healthcare in Australia and describe our social network based approach (applied to health insurance claims) to understand the nature of collaboration among doctors treating hospital inpatients and explore the impact of collaboration on cost and quality of care. (v) We further develop models that predict the behaviors of orthopedic surgeons in regard to surgery type and use of prosthetic device. An important feature of these models is that they can not only predict the behaviors of surgeons but also provide explanation for the predictions.

Statement of Originality

I hereby certify that this thesis contains no material that has been accepted for the award of any other degree in any university or other institution.

____-

Fei Wang
March 31, 2015

Acknowledgements

Firstly, I would like to give thanks to my dear supervisor, Professor Sanjay Chawla, for his essential support and guidance during my research . The achievements during my three years research has been an outcome by his invaluable and indispensable expertise, stimulation and encouragement. I would also like to thank my industry supervisor Uma Srinivasan and Bavani Arunasalam for their invaluable advice and provision of domain knowledge, and my associate supervisor, Dr. Anastasios Viglas, for his purport to my comfortable research environment.

I would like to thank Capital Markets CRC for providing me with a PhD scholarship.

Finally, I want to thank my parents for their infinite support throughout my life overseas.

Publications

This thesis has led to the following publications:

1. Wang, Fei, Sanjay Chawla, and Didi Surian. “Latent outlier detection and the low precision problem.” Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description. ACM, 2013.

2. Wang, Fei, Sanjay Chawla, and Wei Liu. “Tikhonov or Lasso Regularization: Which Is Better and When.” Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on. IEEE, 2013.

3. Wang, Fei, Uma Srinivasan, Shahadat Uddin, and Sanjay Chawla. “Application of Network Analysis on Healthcare.” Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on. IEEE, 2014.

4. Wang, Fei, Wei Liu, and Sanjay Chawla. “On Sparse Feature Attacks in Adversarial Learning.” IEEE International Conference on Data Mining series (ICDM), 2014.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the technological advances during recent years, data scientists are able to obtain data that are significantly larger, complex and in real time with relatively minimal effort. Thus, the era of big data has arrived. Besides the need to modify traditional data mining techniques in order to be scalable to the ever larger data sets, there is a requirement to relax traditional assumptions associated with data mining tasks. For example, in many situations practical large scale systems which deploy classifiers, e.g., spam filters are subject to adversarial reaction. Thus there is a need to design algorithms which are robust against adversarial attack. The classification problem is one of the most intensively studied problem in machine learning and data mining. A fundamental assumption underlying most classification problems is that the training and the test data are generated from the same underlying probability distribution. This assumption underpins both the research and applied "prediction industry."

However there are at least two scenarios where the assumption does not hold in practice.

- Concept drift: In some scenarios, data naturally evolves with time. For example, suppose a credit card scoring model was built during "good" economic times. Then it is natural to expect that the performance of this model is likely to deteriorate during a recession.

- Adversarial attack: In some other situations an adversarial attack has been observed against the classifier. For example, spam filters (which are classifiers) routinely have to be retrained as an adversarial reaction causes their performance to deteriorate.

Furthermore, the role of data mining and machine learning is not just inference but also discovery and the outlier detection methods can also be an important tool for discovering potentially new and useful patterns in data. In fact it has been often stated that new scientific paradigms are often triggered by the need to explain outliers [47]. The availability of large and ever increasing data sets, across a wide spectrum of domains, provides an opportunity to actively identify outliers with the hope of making new discoveries. The obvious dilemma in outlier detection is whether the discovered outliers are an artifact of the measurement device or indicative of something more fundamental. Thus the need is not only to design algorithms to identify complex outliers but also provide a framework where they can be described and explained.

The adversarial effect can also be found in healthcare domain. To be able to identify and explain the potential adversary of healthcare providers could save insurance companies billions of dolors. Social network analysis is commonly used to study relationships between individuals and communities as they interact with each other. Analysing Facebook connections is one such classic example. The textbook by Easley and Kleinberg [27] offers deep insight into the complexity of a connected world. More interesting and novel applications of network theory are reported in specialised domains [1, 2]. In the healthcare domain, social network analysis has been used in different settings, for example to study collaboration among healthcare professionals in specific healthcare environments, to understand the impact of team structure on quality of care [80, 48, 10]. In this dissertation we describe our approach of applying social network analysis in the domain of health insurance claims. In particular, we use data from health insurance claims to design network-based models of collaboration among medical providers and analyse the impact of social networks and their underlying network structures, to discover provider communities and analyse the topology of the emerging community structure (of surgeons, anaesthetists and assistant surgeons) on treatment outcomes for patients who undergo specific category of surgeries, for example knee surgeries.

The increased demand for high quality and cost-effective delivery of healthcare services, brings the entire healthcare sector under close scrutiny. Medical organizations such as the American College of Physicians [63] have started evaluating the feasibility of medical interventions for clinicians in terms of long term benefits, potential harms and monetary considerations. Decisions related to the adoption or discontinuation of different types of medical interventions are often a collaborative process. Providers employed by the same hospitals, who share common patients as well as the working environment, may influence each other, and social relationships can become a powerful driver of learning and innovation, as often assumed in social learning theory [7].

1.1 Contributions of this Thesis

In this thesis, we present our recent research on designing robust methods for handling adversarial situations. Specifically, our main contributions are:

1. We claim in adversarial learning the aim of an adversary is not just to subvert a classifier but carry out data transformation in a way such that spam continues to appear like spam to the user as much as possible. We demonstrate that an adversary achieves this objective by carrying out a sparse feature attack. We design an algorithm to show how a classifier should be designed to be robust against sparse adversarial attacks. Our main insight is that sparse feature attacks are best defended by designing classifiers which use ℓ_1 regularizers.

2. We use mathematical properties of the ℓ_1 (Lasso) and ℓ_2 (Tikhonov or Ridge) regularization methods followed by detailed experimentation to understand their impact based on four characteristics: non-stationarity of the data generating process; level of noise in the data sensing mechanism; degree of correlation between dependent and independent variables and the shape of the data set. The practical outcome of our research is that it can serve as a guide for practitioners of large scale data mining and machine learning tools in their day-to-day practice.

3. We claim that algorithms for discovery of outliers in a latent space will not only lead to more accurate results but potentially provide a natural medium to explain and describe outliers. Specifically, we propose combining Non-Negative Matrix Factorization (NMF) with subspace analysis to discover and interpret outliers. We report on preliminary work towards such an approach.

4. We describe a specific context of private healthcare in Australia and describe our social network analysis (SNA) based approach (applied to health insurance claims) to understand the nature of collaboration among doctors treating hospital inpatients and explore the impact of collaboration on cost and quality of care. In particular, we use network analysis to (a) design collaboration models among

surgeons, anaesthetists and assistants who work together while treating patients admitted for specific types of treatments (b) identify and extract specific types of network topologies that indicate the way doctors collaborate while treating patients and (c) analyse the impact of these topologies on cost and quality of care provided to those patients.

5. We develop models that predict the behaviors of orthopedic surgeons in regard to surgery type and use of prosthetic device. The models utilize data on past practicing behaviours and take in account the social relationships existing among surgeons, anaesthetists and assistants. We refer to the models as the Social Relationship Model (SRM) and Positive Social Relationship Model (P-SRM). An important feature of these models is that they can not only predict the behaviors of surgeons but they can also provide an explanation for the predictions. Experimental results on both artificial and real hospital data sets show that our proposed models outperform the baseline model Online Majority Vote (OMV).

1.2 Organization

The remainder of this thesis is organized as follows. In Chapter 2 we review related literature, key concepts and evaluation metrics used in this thesis. Chapter 3, 4 and 5 respectively depict how we can build more robust models based on existing classification, regularization, and outlier detection algorithms etc. Chapter 6 describe our social network based approach (applied to health insurance claims) to understand the nature of collaboration among doctors treating hospital inpatients and explore the impact of collaboration on cost and quality of care. In chapter 7, we develop models that predict the behaviors of orthopedic surgeons in regard to surgery type and use of prosthetic devices. We conclude in Chapter 8 with directions for future work.

Chapter 2

Background

In this chapter, we present related literature, key concepts and evaluation metrics. The evaluation metrics introduced in this chapter are used throughout the thesis.

2.1 Related Literature

We review related work from two perspectives. We first overview the relevant algorithmic literature on adversarial learning and robust classification. As one of the important application domain is health-care analysis, we review important parts of the domain literature to put our work in an appropriate context.

2.1.1 Algorithm Oriented

Adversarial Classification

Dalvi et al. [24] modelled the interaction between a data miner and an adversary as a game between two cost sensitive players. The authors made an assumption that both adversary and data miner have full information of each other. This perfect information model is not realistic in many online settings . Lowd et al. [53] relaxed the perfect information assumption and derived an approach known as adversarial classifier reverse engineering (ACRE) to study the possible attacks the adversary may carry out. While this framework can help a learner to identify its vulnerability, no solution was proposed to learn a more robust classifier. Globerson et al. [35] formalized the interaction between the two players as a minimax game, in which both players know the strategy space of each other. They made the assumption that the effect of the adversary will be deletions of features at application time. This feature deletion assumption, however, fails to capture the scenarios where the adversary is capable of arbitrarily changing the features.

Liu et al. [52] formulated the interaction between a data miner and an adversary as a *Zero-sum game*, where the adversary is the leader and the data miner is the follower. However, the *Zero-sum game* indicates that the model assumes the adversary is being antagonistic against the data miner. The model also assumes the adversary is able to manipulate the entire feature space. We believe the two assumptions stated will lead to an overestimation of the adversarial's malicious behaviour. For example, in the case of spam email, a classifier's loss is not necessarily the spammer's gain, and the number of features an adversary can manipulate is limited. The model also assumes the adversary can only temper the positive data samples, which is reasonable and applied in our study. Brückner et al. [15] modelled the adversarial learning scenario as a *Stackelberg game* between two players. However, the leader role is played by the data miner and the authors assume the payoff of the two players while in conflict, are not entirely antagonistic. Unlike Liu et al. [52], they made the assumption that the adversary can manipulate both positive and negative instances. This assumption may also be an exaggeration of adversary's influence since in real adversarial environment the behaviour of legitimate users barely changes. They formulate the game as a bi-level optimization problem, which, in general, is not amenable to an efficient solution. Moreover, Brückner et al. [15] also made the unstated (but unrealistic) assumption that the adversary has the ability to change all the features i.e., the adversary engages in a dense feature attack. Recently, Zhou et al. [90] introduced a model based on support vector machines that can tackle two kinds of attacks an adversary may carry out. However, the model is only evaluated on synthetically generated data instead of real world evolved data under adversarial influence. In a subsequent paper, they enhanced their appraoch by combining hierarchical mixtures of experts (HME) [91], where more robust classifier are learned by training the model under adversarial influences. Xu et al. [85] find that solving lasso is equivalent to solving a robust regression problem. This robust property of lasso itself highlights the merits of using sparse modelling technique in the presence of potential adversaries.

Regularization

In terms of regularization, Tikhonov regularization was introduced to address the situation when the system of equation $Ax = b$ is ill-posed [78]. This can occur, for example, when the system can admit infinitely many solutions. To guide the search for solutions with appropriate properties, the following optimization solution has been proposed

$$||Ax-b|| + ||\Gamma x||^2 \tag{2.1}$$

When $\Gamma = \lambda I$, optimization problem is biased towards selecting a solution which has a small ℓ_2 norm. The λ controls the trade-off between how much freedom should be given to the data to dictate the solution versus the apriori constraint to have a solution with a small norm. From a machine learning and statistical perspective, models with small ℓ_2 norms have lower variance and better generalization properties.

As data sets with large number of features started becoming available, it was observed that ℓ_1 instead of ℓ_2 regularization can be used to elicit sparse solutions. Thus models with ℓ_1 regularizers can be used both for prediction and feature selection. ℓ_1 regularizers are called Lasso for “least absolute shrinkage and selection operator” [77, 51, 88]. The literature on both Tikhonov and Lasso is immense. Some recent and notable book level treatments include [56, 16].

Anomaly Detection

In the domain of anomaly detection, the task of extracting genuine and meaningful outliers has been extensively investigated in Data Mining, Machine Learning, Database Management and Statistics [19, 11]. Much of the focus, so far, has been on designing algorithms for outlier detection. However the trend moving forward seems to be on detection and interpretation. While the definition of what constitutes an outlier is application dependent, there are two methods which gained fairly wide traction. These are distance-based outlier techniques which are useful for discovering *global* outliers and density-based approaches for *local* outliers [45, 13]. Recently there has been a growing interest in applying matrix factorization in many different areas, *e.g.* [39],[46]. To the best of our knowledge, probably the most closest work to ours is by Xiong *et al.* [84]. Xiong *et al.* have proposed a method called Direct Robust Matrix Factorization (DRMF) which is based on matrix factorization. DRMF is conceptually based on Singular Value Decomposition (SVD) and error thresholding.

2.1.2 Healthcare Oriented

In the healthcare sector, collaboration among healthcare professionals has been studied from several perspectives. Cunningham et al. (2012) [23] have conducted an orderly review of 26 studies of professionals' network structures and analysed factors connected with network effectiveness and sustainability specifically in relation to the quality of care and patient safety. They discovered that the more cohesive and collaborative of the networks among health professionals, the higher the quality and safety of care they can provide.

For instance, in a classic study, Knaus and his team distinguished a compelling relationship between mortality rate of patient in intensive care units and the degree of collaboration among nurse-physician (Knaus et al., 1986) [43]. Based on their study of $5,030$ intensive care unit admissions, the treatment and outcome indicated that hospitals where nurse-physician collaboration is widespread indicate a lower mortality rate compared to the predicted number of patient deaths. On the other hand, hospitals which exceed their predicted number of patient deaths, usually corresponds to insufficient communication among healthcare professionals. Based on a two group quasi-experiment on $1,207$ general medicine patients, Cowan et al. (2006)[22] observed average hospital length of stay, total hospitalization cost and hospital readmission rate are considerably lower for patients in the experimental group than the control group (5 versus 6 days, $p < .0001$) which contributes a 'backfill profit' of USD1,591 per patient to hospitals. Sommers et al. (2000) [72] examined the impact of an interdisciplinary and collaborative practice intervention involving a principal care physician, a nurse and a social worker for community-dwelling seniors with chronic diseases. The study carried out is controlled cohort and based on 543 patients in 18 private office practices of primary care physicians. The intervention group received care from their primary care physician working with a registered nurse and a social worker, while the control group received care as usual from primary care physicians. They noticed that the intervention group produced better results in relation to readmission rates and average office visits to all physicians. Moreover, the patients in the intervention group also reported an increase in social activities compared with the control group. The studies which focus on collaboration among different professional disciplines related to effectiveness of patient outcomes are also relevant to our study. Another study, by Netting and Williams (1996) [60], based on data collected from 105 interviews (with 40 physician, 32 case managers, 23 physician office staff, 8 administrators and 2 case assistants), showed that there is a growing demand to cooperate and communicate across professional lines rather than make hypothesises between single professional sector and patient outcomes, professional satisfaction and hospital performance. There are other studies that analyse networked collaboration across healthcare specialists to explore different aspects of professional behaviour and quality of patient care. For example, Fattore et al. (2009) [28] evaluate the effects of GP network organisation on their prescribing behavior and (Meltzer et al., 2010) [54] develop a selection criteria of group members in order to enhance the efficacy of team-based approach to patient care. Other studies include physician-pharmacist collaboration (Hunt et al., 2008) [42], physician-patient collaboration (Arbuthnott & Sharpe, 2009) [5], hospital-physician collaboration (Burns & Muller, 2008) [17], and inter-professional, interdisciplinary collaboration (Gaboury et al., 2009) [32].

A common framework for studying how professionals influence and learn from each other is social learning theory [9, 55, 8]. According to social learning theory people learn and modify their behaviors not only in response to direct reinforcement but more generally by observing and responding to stimuli derived from the social context they live in. An important tenet of social learning theory is that the learner, the behavior, and the environment can influence each other. Therefore people's behaviors are influenced by the behaviors of their peers, their environments and by cognitive, biological and other personal factors. These notions are well formalized in social network analysis[44, 18], that uses concepts from network theory to analyze social relationships among a set of actors.

2.2 Key Concepts and Evaluation Metrics

In this section, we briefly review the elementary and commonly used evaluation metrics.

2.2.1 Non Zero-sum Game

We model the interaction between a classifier and an adversary in a game-theoretic setting. We assume that we are given a training data set $(\mathbf{x}_i, y_i)_{i=1}^n$, where $\mathbf{x}_i$ is a feature vector and $y_i \in \{-1, 1\}$ is a binary class label. In a standard classification problem the objective is defined as:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \mathbf{w}, \mathbf{x}_i) + \lambda_{\mathbf{w}} \|\mathbf{w}\|_p \tag{2.2}$$

Table 2.1: Commonly used loss functions for the two players.

	$\ell(y_i, \mathbf{w}, \mathbf{x}_i)$
Square	$\frac{1}{2}\|\|y_i - \mathbf{w}^T\mathbf{x}_i\|\|_2$
Logistic	$\log(1+\exp(-y_i\mathbf{w}^T\mathbf{x}_i))$
Hinge	$(1-y_i(\mathbf{w}^T\mathbf{x}_i))_+$

Here, ℓ is a suitable (convex) loss function, w is the weight vector, $\lambda_{\mathbf{w}}$ is a regularization parameter and $\|.\|_p$ is ℓ_p-norm to encourage generalization. When $p = 1,2$, the regularization is referred to as ℓ_1 regularizer and ℓ_2 regularizer respectively. Some examples of loss functions include square, logistic and hinge loss are shown in Table 4.1.

Now, we bring in an adversary whose objective is to distort the behaviour of the classifier to meet a pre-defined objective. For example, in a spam setting, the adversary would like a spam email to be classified as non-spam by the spam-filtering classifier. We model the adversary action as it controlling a vector α with which it modifies the training data $\mathbf{x}$. However, it is important to note that the adversary would only like to change the spam data (which is $y = 1$) and not the non-spam data. This setting is a non-zero sum game:, i.e., the gain for a classifier is not necessarily the loss for the adversary.

In order to formalize the objective of the adversary we separate the data into positive and negative parts, where the positive data is indexed as $(\mathbf{x}_i, 1)_{i=1}^{npos}$ and the negative data is indexed as $(\mathbf{x}_i, -1)_{i=npos+1}^{n}$.

After the adversary transforms positive data $(\mathbf{x}_i, 1)_{i=1}^{npos}$ to $(\mathbf{x}_i + \alpha, 1)_{i=1}^{npos}$, the classifier aims to re-build the optimal $\mathbf{w}$ denoted by $\mathbf{w}^*$:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{npos}\sum_{i=1}^{npos} \ell(1, \mathbf{w}, \mathbf{x}_i + \alpha^*) + \frac{1}{n-npos}\sum_{i=npos+1}^{n} \ell(-1, \mathbf{w}, \mathbf{x}_i) + \lambda_{\mathbf{w}}\|\mathbf{w}\|_p$$

subject to the constraint that α^* is given by

$$\alpha^* = \arg\min_{\alpha} \frac{1}{npos}\sum_{i=1}^{npos} \ell(-1, \mathbf{w}, \mathbf{x}_i + \alpha) + \lambda_\alpha\|\alpha\|_p \tag{2.3}$$

where $\ell(y_i, \mathbf{w}, \mathbf{x}_i)$ can be any of the three loss functions given in Table 4.1. There are several points that are worth noting about the above model:

1. The adversary is assumed to apply the vector α to the original positive samples by minimizing the same loss function, but *with a negative label.* Actual data does not exist in this form but this is precisely what the adversary would do: change feature vectors of the *positive* data to make it appear as non-spam to the classifier. Since our objective is to design a classifier which is robust against adversarial manipulations, we model the behaviour of the adversary in this particular form.

2. Note that we are normalizing the two terms of the classifier's objective function by the number of positive samples $(npos)$ and the number of negative sample $(n - npos)$. The advantage of this particular normalization is that it will automatically account for any imbalance in the data. If the number of positive sample $npos$ is small, then effectively there will be higher loss for misclassification.

3. The problem as stated above is an example of a bi-level optimization [81] because the constraint is a separate (but coupled) optimization problem in its own right.

2.2.2 Logistic regression

Logistic regression are universally favored for its generalization property. Here we show how the two different regularized logistic functions are derived.

Logistic Loss Function with Gaussian Prior

The logistic loss function is defined as:

$$\sum_{i}^{N} log(1 + e^{-y_i(\mathbf{w}^t \mathbf{x_i} + b)})$$

Where $y_i \in \{-1, 1\}$ is the actual class which a data point $\mathbf{x_i}$ belongs to, $\mathbf{w}$ is the feature weights and b is the bias. Normally, people intuitively add another term $\mathbf{w}^T \mathbf{w}$ to prevent over-fitting. Here we give the mathematical explanations of where ℓ_2 norm come from. Normally we assume the values of the elements in a feature vector could be any real number when we design a loss function, and that is why there exists over-fitting. In the process of fitting the model, the feature vector will only be changed to best classify the training data, so it could be formidably big in terms of the element value. People intuitively add a $||\mathbf{w}||^2$ to constrain the weights from deviate too much from zero. This is equivalent to assume each $\mathbf{w}_j$ follows a Gaussian distribution with mean 0 and variance τ_j [33]:

$$p(\mathbf{w}_j \mid \tau_j) = N(0, \tau_j) = \frac{1}{\sqrt{2\pi\tau_j}} \exp(\frac{-\mathbf{w}_i^2}{2\tau_j}), \ \ j = 1, ..., d. \tag{2.4}$$

Here a small value of τ_j means $\mathbf{w}_j$ is close zero, while a bigger τ_j means $\mathbf{w}_j$ will be further from zero. The maximum likelihood maximization of logistic regression in this case will be written as:

$$\prod_i^N \frac{1}{1+e^{y_i(\mathbf{w}^t\mathbf{x_i}+b)}} \prod_j^M \frac{1}{\sqrt{2\pi\tau_j}} \exp(\frac{-\mathbf{w}_j^2}{2\tau_j}),$$

Which M represent the number of $\mathbf{w}_j$. For each $\mathbf{w}_j$, we assume τ_j is equal to τ:

$$\prod_i^N \frac{1}{1+e^{y_i(\mathbf{w}^t\mathbf{x_i}+b)}} \prod_j^M \frac{1}{\sqrt{2\pi\tau}} \exp(\frac{-\mathbf{w}_j^2}{2\tau}),$$

Now we take the negative log likelihood, the above equation becomes:

$$L(\mathbf{w}) = \sum_i^N log(1+e^{-y_i(\mathbf{w}^t\mathbf{x_i}+b)}) + \sum_i^M \frac{\mathbf{w}_j^2}{2\tau} + \sum_i^M (\ln\sqrt{\tau} + \frac{\ln 2\pi}{2}) \tag{2.5}$$

The last part of the above equation is a constant which can be thrown away and we get the final equation:

$$L(\mathbf{w}) = \sum_i^N log(1+e^{-y_i(\mathbf{w}^t\mathbf{x_i}+b)}) + \frac{1}{2\tau}||\mathbf{w}||^2$$

Although, adding this regularizer will make the $\mathbf{w}_j$ close to zero, but does not favor $\mathbf{w}_j$ being exactly zero. In many application problems, it is better to get a feature vector with a lot zeros in it (i.e. a sparse solution). To achieve this, we have to assume another type of distribution for $\mathbf{w}_j$.

Logistic Loss Function with Laplace Prior

Similarly, the mathematical explanation of ℓ_1 norm is that besides we assume each $\mathbf{w}_j$ follows a Gaussian distribution with mean 0 and variance τ_j , we further assume each τ_j follows a exponential distribution with parameter γ:

$$p(\tau_{j|\gamma}) = \frac{\gamma}{2}\exp(-\frac{\gamma_j}{2}\tau_j), \ \ \gamma > 0. \tag{2.6}$$

If we combine Equation (2.4) and the above equation we will get Laplace distribution:

$$p(\mathbf{w}_i|\gamma_j) = \frac{\lambda_j}{2}\exp(-\lambda_j|\mathbf{w}_j|), \tag{2.7}$$

Again we assume each λ_j equals λ we will get:

$$\prod_i^N \frac{1}{1+e^{y_i(\mathbf{w}^t\mathbf{x_i}+b)}} \prod_j^M \frac{\lambda}{2}\exp(-\lambda|\mathbf{w}_j|), \tag{2.8}$$

And the log loss will be:

$$L(\mathbf{w}) = \sum_i^N log(1+e^{-y_i(\mathbf{w}^t\mathbf{x_i}+b)}) + \sum_i^M \lambda|\mathbf{w}_j| + \sum_i^M(\ln 2 + \ln\lambda) \tag{2.9}$$

Again, we thrown away the last part and get the final equation:

$$L(\mathbf{w}) = \sum_i^N log(1+e^{-y_i(\mathbf{w}^t\mathbf{x_i}+b)}) + \lambda||\mathbf{w}|| \tag{2.10a}$$

2.2.3 Regularizations

The main insight to distinguish between ℓ_1 and ℓ_2 regularization can be obtained by considering the one-dimensional linear regression problem solved using the least squares method. Extensions to higher dimensions and when features are correlated adds to symbol complexity but will be discussed wherever necessary.

Suppose we are given n data points $(y_i, x_i)_{i=1}^n$ and are interested in solving the linear regression problem. We assume a Gaussian error model which reduces to solving the least square problem. There are at least three scenarios:

Ordinary Least Square (OLS)

Here our aim is to select a w^{OLS} which minimizes

$$\sum_{i=1}^n (y_i - wx_i)^2 \tag{2.11}$$

We take derivative of the above equation and make it equal to zero:

$$\begin{aligned} 2(\mathbf{y} - w^T\mathbf{x})\mathbf{x} &= 0 \\ w &= \mathbf{y}^T\mathbf{x} \end{aligned}$$

It can be show that w^{ols} is dot product $\mathbf{y} \cdot \mathbf{x}$.

Tichonov or Ridge

Estimate a w^{ridge} which minimizes

$$\sum_{i=1}^{n} (y_i - wx_i)^2 + \lambda w^2, \tag{2.12}$$

where $\lambda > 0$. Again, it is straightforward to show that $w^{ridge} = \frac{\mathbf{y} \cdot \mathbf{x}}{1+\lambda}$. This clearly shows that as λ increases,the magnitude of the estimator w^{ridge} scales towards zero.

Lasso

Estimate w^{lasso} which minimizes

$$f(w) = \sum_{i=1}^{n} (y_i - wx_i)^2 + \lambda |w| \tag{2.13}$$

Now as $|w|$ is not differentiable we have to examine the sub-gradient(∂) and work through all the cases. The sub-gradient of $f(w)$ is given as

$$\partial(f(w)) = \begin{cases} w - \mathbf{y} \cdot \mathbf{x} - \lambda & \text{if } w < 0 \\ [-\mathbf{y} \cdot \mathbf{x} - \lambda, -\mathbf{y} \cdot \mathbf{x} + \lambda] & \text{if } w = 0 \\ w - \mathbf{y} \cdot \mathbf{x} + \lambda & \text{if } w > 0 \end{cases} \tag{2.14}$$

We now have to examine under what conditions will $0 \in \partial(w)$. This can happen under the following three scenarios which depend upon the strength and direction of the correlation $\mathbf{y}.\mathbf{x}$.

$$w^{lasso} = \begin{cases} \mathbf{y} \cdot \mathbf{x} + \lambda & \text{if } \mathbf{y} \cdot \mathbf{x} < -\lambda \\ 0 & \text{if } \mathbf{y} \cdot \mathbf{x} \in [-\lambda, \lambda] \\ \mathbf{y} \cdot \mathbf{x} - \lambda & \text{if } \mathbf{y} \cdot \mathbf{x} > \lambda \end{cases} \tag{2.15}$$

The above result clearly shows that w^{lasso} will be zero when $\mathbf{y}$ and $\mathbf{x}$ are weakly

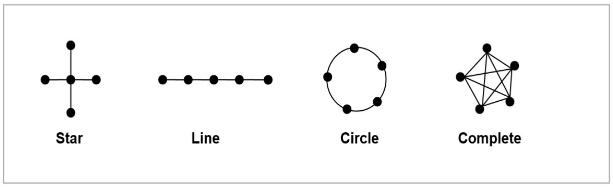


Figure 2.1: Illustration of a star, a line, a circle and a complete graph.

correlated relative to λ. In the multi-dimensional case, this observation has been generalized known as `SafeRule`, which is used for pruning variables whose weight in the solution vector will be zero. In particular, assume that the data is given in the form $(\mathbf{y}, \mathbf{X})$, where $\mathbf{X}$ is matrix representing the independent variables. Then the `SafeRule` [34] asserts that ${\mathbf{w_i}}^{lasso} = 0$ if

$$|\mathbf{X}_i^T \mathbf{y}| < \lambda - \|\mathbf{X}_i\|_2 \|\mathbf{y}\|_2 \frac{\lambda_{\max} - \lambda}{\lambda_{\max}}, \tag{2.16}$$

where $\lambda_{\max} = \|\mathbf{X}^T \mathbf{y}\|_\infty$. This has been used to safely remove X_i from the data set as it will not have any impact on the model.

2.2.4 Network concepts

Degree Centralisation

To explain degree centralisation, we need to first define degree centrality. Being one of the basic measures of network centrality, degree centrality captures the percentage of nodes that are connected to a particular node in a network. It highlights the node with the most connections to other actors in the network, and can be defined by the following equation for the actor (or node) i in a network carrying N actors (Wasserman and Faust 2003) [74]:

$$C_D^{'} = \frac{d(n_i)}{N-1}$$

The subscript D for ‘degree’ and $d(n_i)$ indicates the amount of actors with whom actor i is adjacent. The maximum value for $C_D^{'}$ reaches 1 as actor i is linked with everyone else in the network. Network degree centralisation is measured based on the set of degree centralities, which represents the collection of degree indices of N actors in a network. Formally, degree centralisation can be summarised by the following equation (Freeman et al. 1979) [31]:

$$C_D = \frac{\sum_{i=1}^{N}[C_D(n^*) - C_D(n_i)]}{[(N-1)][(N-2)]}$$

Where, $\{C_D(n_i)\}$ are the degree indices of N actors and $C_D(n^*)$ is the largest observed value in the degree indices. For a network, degree centralisation (i.e. the index C_D) reaches its maximum value of 1 when one actor chooses all other $(N-1)$ actors and the other actors interact only with this one (i.e. the situation in a star graph as illustrated in Figure 2.1). On the other hand, C_D attains its minimum value of 0 when all degrees are equal (As portrayed in Figure 2.1, i.e. the setting in a circle graph). Thus, regarding to both a star and circle graph, C_D signifies varying amounts of centralisation of degrees.

Closeness Centralisation

Likewise, closeness centrality needs to be defined before we make clear closeness centralisation. Being another aspect of actor centrality based on closeness, closeness centrality focuses on how ‘close’ an actor is to all the other actors in a network (Freeman et al. 1979) [31]. The idea is that if an actor can instantly interact with all other actors in a network, then it is of central stand. In the context of a communication relation, actors with central place need not rely on other actors for the relaying of information. For an individual actor, it can be represented as a function of shortest distances between that actor and all other remaining actors in the network. The following equation represents the closeness centrality for a node i in a network having N actors (Freeman et al. 1979; Wasserman and Faust 2003) [31, 74]:

$$C_C^{'}(n_i) = \frac{N-1}{\sum_{j=1}^{N} d(n_i, n_j)}$$

Where, the subscript C for ‘closeness’, $d(n_i, n_j)$ is the number of lines in the shortest path between actor i and actor j, and the sum is taken over all $i \neq j$. A higher value of $C_C^{'}(n_i)$ indicates that actor i is closer to other actors of the network, and will be 1 when actor i has direct links with all other actors of the network. The set of closeness centralities, which represents the collection of closeness indices of N actors in a network, can be summarised by the following equation to measure network closeness centralisation (Freeman et al. 1979) [31]:

$$C_C = \frac{\sum_{i=1}^{N}[C_C^{'}(n^*) - C_C^{'}(n_i)]}{[N-1][N-2]/[2N-3]}$$

Where, $\{C_C^{'}(n_i)\}$ are the closeness indices of N actors and $C_C^{'}(n_i)$ is the largest recognized value in closeness indices. For a network, closeness centralisation (i.e. the index C_C) reaches its maximum value of unity when one actor chooses all other $(N-1)$ actors and the other actors have shortest distances (i.e. geodesics) of length 2 to the remaining $(N-2)$ actors (i.e. the situation in a star graph as illustrated in Figure 2.1). This index (i.e. C_C) can attain its minimum value of 0 when the lengths of shortest distances (i.e. geodesics) are all equal (i.e. the situation in a complete graph and circle graph as illustrated in Figure 2.1). Thus, indicates varying amounts of centralisation of closeness compared to star, circle and complete graph.

Betweenness Centralisation

Betweenness centrality will be defined first before explaining betweenness centralisation. Betweenness centrality is obtained by deciding the frequency of a particular node being on the shortest path between any pair of actors (or nodes) in the network. It views an actor as being in a favoured position to the extent that the actor falls on the shortest paths between other pairs of actors in the network. That is, nodes that occur on many shortest paths between other pairs of nodes have higher betweenness centrality than those that do not (Freeman 1978) [30]. Therefore, it can be regarded as a measure of strategic advantage and information control. In a network of size n, the betweenness centrality for an actor (or node) i can be defined by the following equation (Wasserman and Faust 2003) [74]:

$$C_B^{'}(n_i) = \frac{\sum_{j<k} \frac{g_{ij}(n_i)}{g_{jk}}}{[(N-1)(N-2)]/2}$$

Where, $i \neq j \neq k$; $g_{jk}(n_i)$ represents the number of shortest paths linking the two actors that contain actor i; and g_{jk} is the number of shortest paths linking actor j and k. From the set of betweenness centralities of N actors in a network betweenness centralisation can be defined by the following equation:

$$C_B = \frac{\sum_{i=1}^{N}[C_B(n^*) - C_B(n_i)]}{N-1}$$

Where, $\{C_B^{'}(n_i)\}$ are the betweenness indices of N actors and $C_B(n^*)$ is the largest observed value in betweenness indices. For a network, betweenness centralisation (i.e. the index C_B) reaches its maximum value of unity when one actor chooses all other $(N-1)$ actors and the other actors have shortest distances (i.e. geodesics) of length

2 to the remaining $(N-2)$ actors (i.e. the situation in a star graph as illustrated in Figure 2.1). This index (i.e. C_B) can attain its minimum value of 0 when all actors have exactly the same actor betweenness index (i.e. the situation in a line graph as illustrated in Figure 2.1). Thus, C_B indicates varying amounts of centralisation of betweenness compared to both star and line graph.

Density

Density measures the connectivity of a graph. For example, if a graph G has N nodes, V edges then the density D_G of the graph G is calculated as :

$$D_G = \frac{2*V}{N(N-1)}$$

D_G reaches maximum value as 1 when the graph is fully connected, and reaches minimum value as 0 when there is no edge.

2.2.5 Confusion Matrix

Confusion matrix, also known as contingency matrix, is a popular method for visualizing performance. Various metrics can be derived from the values in the matrix. We take binary prediction as an example.

Table 2.2: Representation of Classification Results via a Confusion Matrix

		True Label	
		Negative	Positive
Predicted Label	Negative	**True Negative (TN)**	**False Negative (FN)**
	Postive	**False Positive (FP)**	**True Positive (TP)**

As shown in Table 2.2, the classification results are grouped into four subsets: true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN). Evaluation metrics are calculated based on the four subset values.

The classification *accuracy* is calculated as $\frac{TP+TN}{TP+FP+FN+TN}$. Other classification metrics include: (*i*) *true positive rate* (also known as *sensitivity*, or *recall*) which is defined as $\frac{TP}{TP+FN}$; (*ii*) *false positive rate* defined as $\frac{FP}{FP+TN}$; (*iii*) *true negative rate* (also known as *specificity*) defined as $\frac{TN}{TN+FP}$; and (*iv*) *precision* defined as $\frac{TP}{TP+FP}$.

The metric of *F1–measure* (harmonic mean of precision and recall) is also widely used. *F1-measure* is defined as the harmonic mean of precision and recall:

$$F1 - measure = \frac{2 \times precision \times recall}{precision + recall} \tag{2.17}$$

2.2.6 ROC

Receiver operating characteristic (ROC) is a curve for measuring binary classification performance. The term “receiver operating characteristic” first came from tests of the ability of World War II radar operators to determine whether a blip on the radar screen represented an object (signal) or noise. Provided the predicting result is a probability between 0 and 1. Each point in the curve is plotted as true positive rate to false positive rate as one varying the discrimination threshold. A rational behaving classification algorithm would have a corresponding curve above the $y = x$ line, which is the random guess classifier. The better a classifier performs, the closer the corresponding curve reaches the top left corner. Visually comparing different ROC outcomes can be subjective and time consuming, Thus the area under the curve (AUC) is a quantitative measurement of how good the curve is. It is measured as the percentage of the area under the ROC curve. A perfect prediction would have a AUC value of 1 and a random guess with a AUC value around 0.5.

Chapter 3

On Sparse Feature Attacks in Adversarial Learning

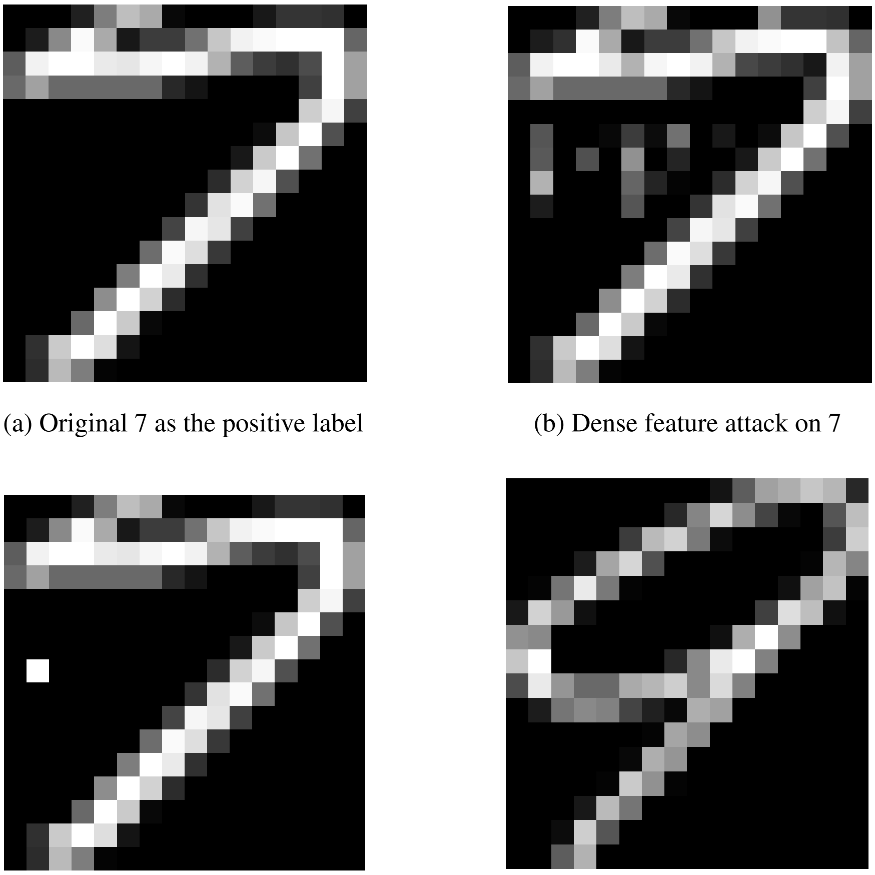
This chapter is based on the following publication:

Wang, Fei, Wei Liu, and Sanjay Chawla. On Sparse Feature Attacks in Adversarial Learning. IEEE International Conference on Data Mining series (ICDM), 2014.

3.1 Introduction

The focus of this chapter is to take the adversary into account during the design of classifier. Most existing work on adversarial learning make the assumption that all features of the training data will be simultaneously attacked (manipulated) by an adversary (which we call "dense feature attacks"). Here we propose and investigate a model where an adversary will only choose to manipulate a subset of the features in order to minimize its manipulation cost (which we call "sparse feature attacks"). But more importantly *this is because the adversary wants to construct spam so that it looks like non-spam to the classifier and the reader actually consume the spam.*

Fig. 3.1 illustrates the difference between dense and sparse feature attacks. This result is obtained from our experiment results on the famous hand-written digit data, where we use digits "7" and "9" as positive and negative class labels respectively. While dense feature attacks (Fig. 1(b)) transforms many of the original pixels (which are the features) of the original "7" to make it mis-classified as "9", sparse feature attacks (Fig. 1(c)) only need to transform one pixel to achieve the same misclassification. Therefore



(a) Original 7 as the positive label

(b) Dense feature attack on 7

(c) Sparse feature attack on 7

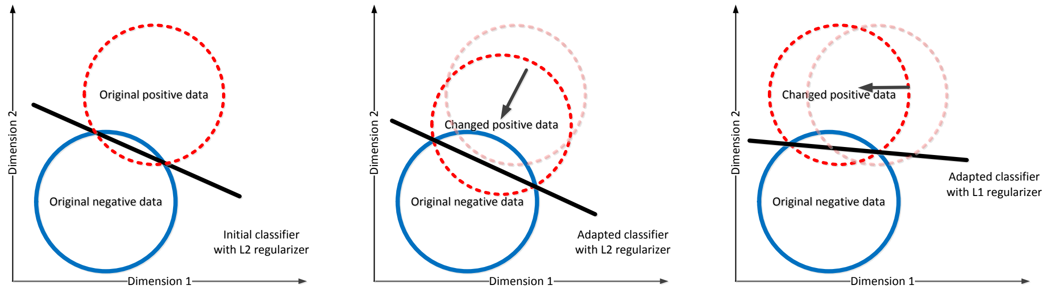
(d) Original 9 as the negative label

Figure 3.1: Sparse feature attack identifies the most significant feature that distinguishes "7" and "9", and keeps most original pixels intact after the attack.

a rational adversary is likely to select sparse feature attacks to significantly reduce the cost of the data transformation and simultaneously make the spam continue to look like non-spam to the classifier.

How are dense and sparse attacks modelled?

With the help of an example we demonstrate the game theoretic aspects of the two types of attacks. We use a two-class classification problem in two dimensional feature space as an example:
(**Step 1**) The data miner uses an acquired labelled data set to build a classifier (e.g., a spam filter). Figure 3.2a depicts the distributions of positive and negative data and the classification boundary.



(a) Classifier with ℓ_2 regularizer (b) Dense feature attack, changing both feature 1 and 2 (c) Sparse feature attack, changing only feature 2

Figure 3.2: Adversarial classification problems in two dimensional space. Each circle represents a group of data in the same category. Straight lines represent the classification boundaries.

(**Step 2**) An adversary (e.g., a spammer) deliberately transforms the positive data (e.g., spam email) towards the negative ones (e.g., legitimate emails) so as to cross the decision boundary. In Figure 3.2b, with dense feature attack, a spammer can manipulate the whole feature spaces, which may be infeasible. Moreover, the attack transforms the spam email to look more similar to non-spam email, which will decrease the advertising utility of the spammer. In Figure 3.2c, assuming sparse feature attack, an adversary can only transform positive data either horizontally or vertically (by manipulating features only along dimension 1 or 2). The sparse attack is reasonable since spammer can only have a limited budget to manipulate the features. More importantly, by moving the spams toward the side of the non-spam emails region, adversary keeps a reasonable advertising utility by keeping the spam distinctive from non-spam and consumed by users as spams.

(**Step 3**) Data miner responds by rebuilding the classifier. Under dense feature attack a classifier with an ℓ_2 regularizer moves its boundary towards negative data to adapt to the new situation, and thus suffers a substantial loss in classification accuracy. However, under sparse feature attacks, the classifier with an ℓ_1 regularizer adapts itself by changing the slope of the boundary, i.e., the boundary becomes flatter as it is a sparse vector.

In an adversarial environment with high dimensionality, the three-step game given here is played repeatedly in time. We claim when both players are utilizing sparse strategies, a more robust classifier be designed. In summary, we make the following *contributions*:

- We derive a new game-theoretic model which formulates the interactions between the data miner and the adversary as a *non zero-sum game*.

- We propose regularized loss functions so that the game is cast into two convex optimization problems, and propose an algorithm to solve the game.

- We investigate the use and robustness of the ℓ_1 and ℓ_2 regularizers (both for the data miner and the adversary) to examine the advantages of sparse models.

- We conduct experiments on two real email spam data sets and a hand-written digit data set which confirm the superiority of the sparse models against adversarial manipulation.

The outline of this chapter is as follows. We elaborate on the approach to solve the game in Section 3.2. Finally, in Section 3.3 we conduct the experiments together with analysis. Section 3.4 contains conclusions and future work.

3.2 Solving The Non Zero-sum Game

Our sparse model differs from the previous game theoretical models as summarized in Table 3.1. We note that in the column 'Adversarial Modification', only Brückner et al. [15] has a different formulation, which modifies all the data point in the union of classification boundary w, where τ_i is a scalar associated with each data point. In the column 'Adversarial attack type', Liu et al. [52] assume a zero-sum model Zhou et al. [90] make the assumption that classification boundary w is not disclosed to the adversary. In the column 'Adversarial budget' and 'attack type', only our *sparse model* incorporates a pre-defined the budget and sparse feature attack.

Table 3.1: Comparisons of our sparse model with previous game theoretic models.

	Zero-sum Game	Non Zero Sum Game		
Paper	Liu et al. [52]	Brückner et al. [15]	Zhou et al. [90]	our sparse model
Adversarial Modification	$\mathbf{x}_i^{npos} = \mathbf{x}_i^{npos} + \alpha$	$\mathbf{x}_i^n = \mathbf{x}_i^n + \tau_i * \mathbf{w}$	$\mathbf{x}_i^{npos} = \mathbf{x}_i^{npos} + \alpha$	$\mathbf{x}_i^{npos} = \mathbf{x}_i^{npos} + \alpha$
Adversarial attack type	Antagonistic,aware of w	Conflict, aware of w	Conflict, not aware of w	Conflict, aware of w
Adversarial Budget	Not defined	Not defined	Not defined	Defined
Attack type	Dense feature attack	Dense feature attack	Dense feature attack	Sparse feature attack

The above *non zero-sum game* in the case of adversarial classification can be solved as follows:

Data miner chooses strategy $\mathbf{w}_0$ based on the observed samples drawn from the sample space by minimizing its loss function:

$\mathbf{w}_0 = arg\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mathbf{w}^T\mathbf{x}_i) + \lambda_{\mathbf{w}}\|\mathbf{w}\|_p$.

In the following steps, $k = 1, \ldots + \infty$;

1. Adversary chooses strategy α_k, which is the manipulation vector, with the knowledge of data miner's strategy $\mathbf{w}_{k-1}$,

   i.e., $\alpha_k = arg\min_{\alpha} \frac{1}{npos}\sum_{i=1}^{npos} \ell(-1, \mathbf{w}_{k-1}^T(\mathbf{x}_i + \alpha)) + \lambda_{\alpha}\|\alpha\|_p$.

   The manipulation is then applied to the sample space $\mathbf{x}_i^{*npos} = \mathbf{x}_i^{npos} + \alpha_k$.

2. Data miner chooses strategy $\mathbf{w}_k$ based on the samples drawn from the manipulated sample space.

   $\mathbf{w}_k = arg\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mathbf{w}^T\mathbf{x}_i^*) + \lambda_{\mathbf{w}}\|\mathbf{w}\|_p$.

The above two steps are repeated sequentially and the game terminate with the accumulated change applied by the adversary reaches a predefined budget $MB > \sum_{k=1,2,\ldots,n}\|\alpha_k\|_1$ (which will be defined in Section 3.2.2). The pseudo-code of this procedure is described in Algorithm 1.

The goal of the data miner is to determine a decision boundary based on continuously manipulated training data in each step. For the adversary, the goal is to determine a manipulating vector based on a given budget in each step. What we are interested is an adapted classifier, i.e., feature weights $\mathbf{w}_k$, which has the potential of being more robust on future adversarially influenced data set. A classifier learnt from the game is designed to be more robust to future manipulations as shown in Figure 3.3.

3.2.1 Lasso and robust regression

A learning algorithm is robust if the model it is resistant to bounded perturbations in the data. Robust learning algorithms is an active area of research and the robust linear regression problem is defined as

$$\min_{w\in R^d}\{\max_{|z|\leq\lambda}\|y-(x+z)w\|_2\}. \tag{3.1}$$

Algorithm 1 Non Zero-sum Game

Input: Training data $D = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$, minimum budget MB, $\lambda_{\mathbf{w}}$, λ_{α} and Norm p
Output: $\mathbf{w}$ and α

```
// Build the initial classifier using original training data:
$\mathbf{w} = \arg\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mathbf{w}, \mathbf{x}_i) + \lambda_{\mathbf{w}} \|\mathbf{w}\|_p$
$Cost \leftarrow 0, \alpha_{Sum} \leftarrow 0$
while $Cost <= MB$ do
    // Step 1: Adversary attack (see explanation in Section III)
    // Learn $\alpha$ by assigning negative label to positive samples.
    $\alpha = \arg\min_{\alpha} \frac{1}{npos}\sum_{i=1}^{npos} \ell(-1, \mathbf{w}, (\mathbf{x}_i + \alpha)) + \lambda_{\alpha} \|\alpha\|_p$
    for positive data : $\mathbf{x}_i^{*npos} = \mathbf{x}_i^{npos} + \alpha$
    // Step 2: Data miner responds
    $\mathbf{w} = \arg\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mathbf{w}, \mathbf{x}_i^*) + \lambda_{\mathbf{w}} \|\mathbf{w}\|_p$
    // Calculate accumulated cost
    $Cost += \|\alpha\|_1$
end while
return $\mathbf{w}$ generated
```

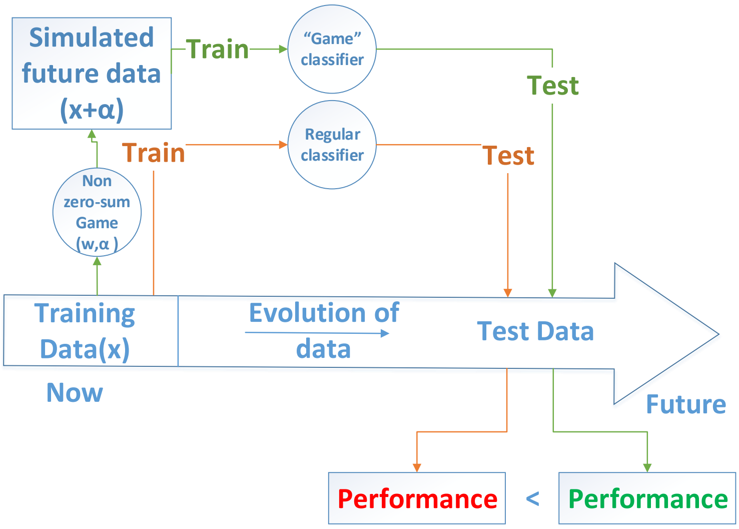


Figure 3.3: The graph demonstrating the merits of using a game-theoretic classifier.

The key insight about robust regression as defined in [85] can be derived from considering the one-dimensional case [82], i.e. we assume $d = 1$. For example, we first notice that

$$\max_{|z| \le \lambda} |y - (x+z)w| \quad \le |y - xw| + z|w|.$$

Now consider a specific $z^* = -\lambda sgn(w)sgn(y-xw)$. One can observed that $|z^*| \leq \lambda$.

$$\begin{aligned}
\max_{|z|\leq\lambda} |y-(x+z)w| &\geq |y-(x+z^*)w| \\
&= |y-xw| + |\lambda sgn(w)sgn(y-xw)w| \\
&= |y-xw| + \lambda|w|,
\end{aligned}$$

thus

$$\max_{|z|\leq\lambda} |y-(x+z)w| = |(y-xw)| + \lambda|w|. \tag{3.2}$$

This generalizes to

$$\min_{w\in\mathbb{R}^d} \{\max_{\mathbf{z}\in\mu} ||y-(\mathbf{x}+\mathbf{z})\mathbf{w}||_2\} = \min_{w\in\mathbb{R}^d} ||y-\mathbf{xw}||_2 + \sum_{i=1}^{d} \lambda_i ||\mathbf{w}_i||_1, \tag{3.3}$$

where $\mathbf{z} \in \mu$ is the worst case disturbance of noise and μ has

$$\mu \triangleq \{(\delta_1,...,\delta_d)|||\delta_i||_2 \leq \lambda_i, i=1,...,d\},$$

where δ is the range of disturbance.

This shows solving an ℓ_1 regularized least square problem is equivalent to solving a worst case linear square problem with noise $\mathbf{z}$. In other words, we are assuming the existence of an adversary with a perturbation matrix of $\mathbf{z}$. More importantly, this robust regression equivalence provides us a way for setting a reasonable budget for the adversary.

3.2.2 Robust regression and minimum budget of adversary

Since we know that by adding ℓ_1 regularizer, we are practically assuming there is an adversary that is adding noise to both positive and negative data to maximize the loss of the classifier according to the classification boundary. The largest perturbation $\mathbf{z}$ can achieve is $||\mathbf{z}_i||_2 = ||\delta_i||_2 = \lambda_i$, where $\mathbf{z}_i$ is the i-th column of $\mathbf{z}$. Assume $\lambda_i = \lambda, \forall i$, then $||\mathbf{z}||_2 = \sqrt{d}\lambda$. Suppose $\lambda^*_{(n,d)}$ is tuned with cross validation, where the training and test portions are ordered in time, not randomly divided. Then the intuition is that real adversary should have the ability to exert manipulation on data at least as much as the corresponding perturbation $\|\mathbf{z}^*\|_2 = \sqrt{d}\lambda^*_{(n,d)}$. Assume

$$\left\|\begin{matrix}\alpha\\ \vdots\\ \alpha\end{matrix}\right\|_2 = \|\mathbf{z}^*\|_2 = \sqrt{d}\lambda^*_{(n,d)}$$

then we have $\|\alpha\|_2 = \frac{\sqrt{d}\lambda^*_{(n,d)}}{\sqrt{n}}$. Since, the budget we defined is ℓ_1 norm of vector α, thus here we assume each element in α is the same, then we will have $\|\alpha\|_1$=$\sqrt{d}\|\alpha\|_2 = \frac{d\lambda^*_{(n,d)}}{\sqrt{n}}$. Thus we have the *MB* as:

$$MB \;=\; \frac{d\lambda^*_{(n,d)}}{\sqrt{n}}$$

Notice that the *MB* is a conservative estimation of an adversary's ability to influence the data.

3.2.3 Evaluation of regularizer

One can notice that we have four possible models by combining the two players' loss function with different regularizers. $\|\mathbf{w}\|_p$ and $\|\alpha\|_p$ can either be ℓ_1 or ℓ_2 norm. In the case of $\|\mathbf{w}\|_p = \|\mathbf{w}\|_2$ and $\|\alpha\|_p = \|\alpha\|_2$, we denote this model as $Game(\ell_2^d\ \ell_2^a)$. Similarly we denote the other three sparse models as $Game(\ell_2^d\ \ell_1^a)$, $Game(\ell_1^d\ \ell_2^a)$ and $Game(\ell_1^d\ \ell_1^a)$. We denote a regular classifier with ℓ_2 and ℓ_1 regularizer as *Regular*-ℓ_2 and *Regular*-ℓ_1 respectively. Experiments of the non zero-sum game are reported in Section 3.3.4.

3.3 Experiments

We now report on the experiments carried out to evaluate the effectiveness of the proposed model in adversarial settings. Our main focus is to compare the effectiveness of our game-theoertic model with ℓ_1 and ℓ_2 regularizers under both sparse and dense feature attacks. We use the BMRM [75] solver for logistic loss and CVX [37] for square and hinge loss. All the data and code is available for result replication[1].

[1] `https://www.dropbox.com/sh/tq8gbzzh59d0nu2/AAAB9RlkrKRaufyI2DzNo90ja`

3.3.1 Data set

3.3.1.1 (USPS) Digit Image data set

The US Postal Service (USPS) data set [38] consists of gray scaled images of handwritten digits from 0 to 9. Each image is of the size 16×16 (dimension 256). In the experiments, we pick pairs of digits to illustrate how an adversary can manipulate one digit (positive class) to look like another (negative class).

3.3.1.2 (Malinglist) Mailinglist data set

The data set is a collection of 128,117 emails arranged in a chronological order. The emails are extracted from a publicly available *mailing lists* and are augmented with spam emails from Bruce Guenter's spam trap of the same time period (01/04/1999 – 31/05/2006). This data set has been used in previous adversarial learning research [15]. The data set obtained is an inverted table of all the words and symbols in the original spam emails. We carried out feature selection by applying kernel-PCA map [71, 14] which is defined as:

$$\phi PCA : \mathbf{x} \mapsto \Lambda^{\frac{1}{2}^{+}} V^T [k(\mathbf{x}_1, \mathbf{x}), ..., k(\mathbf{x}_n, \mathbf{x})]^T. \tag{3.4}$$

Here V is the column matrix of eigenvectors of kernel matrix K, where K is the dot product of data points $k(\mathbf{x}, \mathbf{x}) = \mathbf{x}^T \mathbf{x}$. We use the first 2000 instances (in chronological order) to formulate the 2000×2000 kernel matrix. Λ is the diagonal eigenvalue matrix of K such that $K = V \Lambda V^T$, and $\Lambda^{\frac{1}{2}^{+}}$ represents the pseudo-inverse of $\Lambda^{\frac{1}{2}}$.

We use kernel PCA to reduce the dimensionality of the feature space from 266,378 to 50 dimensions. The 50 features are significant enough to fit a regular logistic classifier with an F-measure score of 0.967 on the training data. The model was trained with the original imbalanced data in order to reflect the true class distribution.

3.3.1.3 Spambase data set

We also used the popular "Spambase" data set [79] to test the robustness of different classifiers [29]. The data set has 4601 both spam and non-spam emails and has 57 features, out of which 48 are the frequencies of key words, 6 are percentage of key words.

3.3.2 Why sparse feature attack for the adversary?

In this section we illustrate three main insights with experimental validations of why a rational adversary will apply sparse feature attack.

3.3.2.1 A more realistic behavior

With the help of the USPS digit data set we demonstrate why a rational adversary is likely to mount a sparse attack. We select digit 7 and 9 to be the binary classification data set. Now we assume an adversary is able to control and manipulate 7 so that it will be misclassified. We first train a regular classifier with ℓ_2 regularizer, then the two types of attacks (ℓ_1 and ℓ_2) are performed on the data set. We show the misclassified 7 from each of the two types of attacks in Figure 3.1.

An example of the original digit 7 and 9 is shown in Figures 3.1(a,d). Under a sparse feature attack only one pixel is modified (Figure 3.1c) and that is sufficient to misclassify the image. It turns out that the pixel manipulated by the adversary is the most important feature to distinguish between 7 an 9. On the other hand a dense feature attack is shown in (Figure 3.1b) results in several pixels being modified but with a lower intensity. Thus with a sparse feature an adversary is able to make minimal observablechanges to the "spam" and circumvent the spam filter.

3.3.2.2 Leads to better classifier

Here we study the relationship between the performance of a game-theoretic classifier with varying attack strength, i.e. *Cost* as defined in the model. When we train a classifier on an tempered data set, in our case manipulated positive samples, it is necessary to ask how much will the new classifier differ from a regular classifier trained on original data set? To answer this question, we evaluate the classifiers using the false negative to false positive rate. In the Non-Zero sum game model, we have assumed that an adversary will manipulate the positive data so that it will go across the classification boundary. Therefore, the classifier learned in this case will move the classification boundary backward to the direction of negative samples. Thus, we would expect a classifier with lower false negative rate, at the same time, a higher false positive rate. The attack strength decides how far the boundary will move towards the negative samples. Thus we can vary the attack strength and examine the false negative rate and false positive rate. The model with a lower false negative rate is preferred. In this experiment we test the Non-Zero sum game with classifier using ℓ_2 norm, while the adversary uses ℓ_2 or ℓ_1 norm. We select the first 400 samples from the Malinglist data set as training data, which represent the older emails. For test data, we select the last 4000 samples from the data set. As shown in Figure 3.4, with the same false positive rate, classifier with ℓ_1 regularizer has a lower false negative rate. The experiment illustrates that by modeling an adversary with ℓ_1 regularizer we get overall better performance.

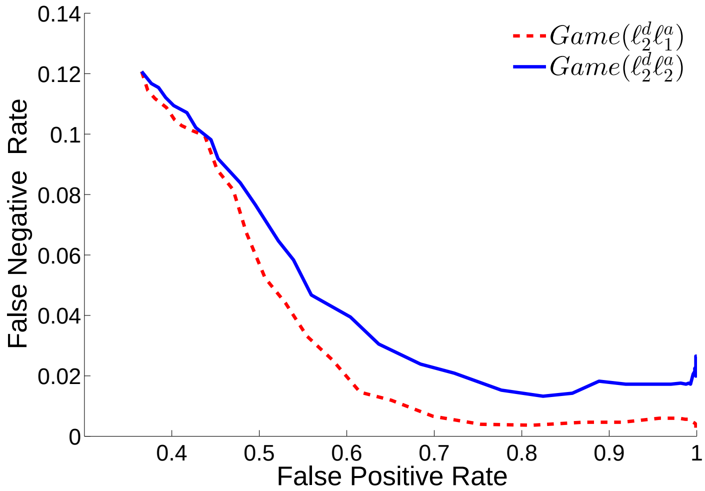


Figure 3.4: When adversary is assumed to apply sparse feature attack, the learned classifier has better performance.

3.3.2.3 The game converges faster with less cost and feature modifications

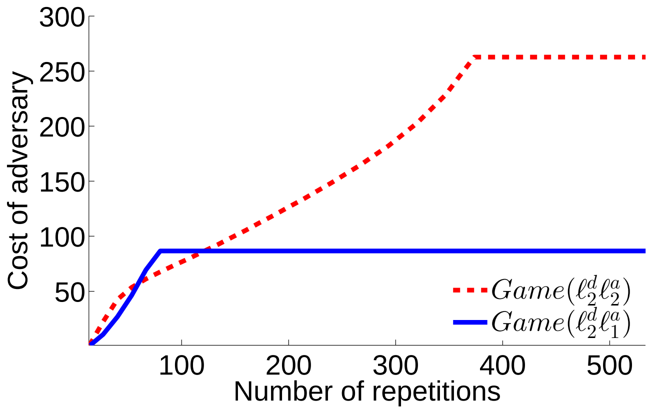


Figure 3.5: The game with sparse feature attack reaches a stable state much faster compared to a dense attack and is associated with lower cost.

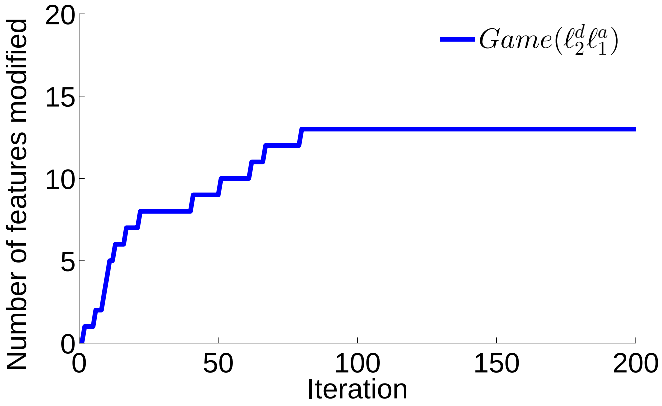


Figure 3.6: The game with sparse feature attack identify and modifies a limited number of features, in this case, 13 out of 50 features.

As one can anticipate that if we let the game play repeat indefinitely, it will reach a state where the positive and negative data almost overlap. However we would expect the classifier learned from such a game will have the least performance. In other words, this is an extreme case of overestimating the adversary. Still we conduct experiment to estimate the adversarial cost for reaching such a state. Since we have already concluded that a sparse attack is a better model for the adversary, we expect the adversary with sparse feature attack will achieve the overlap state with less cost compared to the one with dense feature attack.

We compare the the number of iterations required for *Game*$(\ell_2^d\ \ell_2^a)$ and *Game*$(\ell_2^d\ \ell_1^a)$ converge on the *mailinglist* data set . The *Cost* of the adversary is evaluated by accumulating the ℓ_1 norm of α in each step. We also evaluate the number of features modified under the sparse feature attack model *Game*$(\ell_2^d\ \ell_1^a)$ as a function of number of iterations. For *Game*$(\ell_2^d\ \ell_2^a)$, we know it is likely to modify all the features in each step.

Figure 3.5 shows the cumulative cost of the adversary as a function of the number of iterations in the game. It is clear that *Game*$(\ell_2^d\ \ell_1^a)$, i.e., the game where the adversary carries out a sparse feature attack converges faster to reach a stable state compared to the dense game *Game*$(\ell_2^d\ \ell_2^a)$. From Figure 3.6, we found the number of features being modified will also converge to a number far less than the total number of features. This again suggests that modeling an adversary using an ℓ_1 regularizer is a better reflection of reality.

3.3.3 Why ℓ_1 regularizer for data miner?

As reported in Section 3.2.1, the lasso problem is equivalent to a robust regression problem. Using a regularizer has always been considered as a method to penalize the weights to achieve better generalization. Here we find that ℓ_1 regularizer is not only a technique to prevent overfitting, but also a method for building more robust classification boundaries. This property itself indicates that classifier learnt with an ℓ_1 regularizer is more robust in the presence of certain data manipulations. Therefore, in the case of an adversarial environment, a classifier with ℓ_1 regularizer is preferred.

Existing literatures [68, 4] indicates that in a classical classification environment, when ℓ_1 regularizer is applied, there will always be a trade-off between the enforced sparsity and the accuracy obtained. However, depending on the data set itself, the overall performance of an ℓ_1 regularized classifier can sometimes beat an ℓ_2 regularized classifier [65] in terms of both bias and variance. A deeper discussion of this issue is beyond the scope of this research. Here, we compare the performance of the two classifier under the condition that the distribution of the test data is altered by an adversary.

To investigate the influence of the adversarial manipulation, we start by looking at how the loss of a data miner is increased by the adversary according to the loss function:

$$L(\mathbf{w}) = \sum_{i}^{n} log(1+e^{-y_i\langle \mathbf{w}^T,\mathbf{x}_i+\alpha\rangle}) + \lambda_{\mathbf{w}}\|\mathbf{w}\|_p = \sum_{i}^{n} log(1+e^{-y_i\langle \mathbf{w}^T,\mathbf{x}_i\rangle+\langle \mathbf{w}^T,\alpha\rangle}) + \lambda_{\mathbf{w}}\|\mathbf{w}\|_p$$

As we can see from this equation, the adversarial influence is captured only in the factor $\langle \mathbf{w}^T, \alpha\rangle$. This is the dot product of the feature weight vector and the manipulation vector, which can be expressed as: $w_1\alpha_1 + w_2\alpha_2 + ... + w_d\alpha_d$. One should notice that when both of the two vectors are dense vectors, $\langle \mathbf{w}^T, \alpha\rangle$ will always be non-zero. On the other hand, when both the vectors are sparse vectors, $\langle \mathbf{w}^T, \alpha\rangle$ will have a high probability of being zero. When this factor is zero, the influence of adversary also disappears. Thus, we can conclude that when adversary is modeled to use a sparse feature attack, data miner should apply ℓ_1 regularization to reduce the adversary's effect. This analysis can be easily generalized into higher dimensions. We can also hypothesise that the adversarial influence of the sparse classifier has a negative correlation with its sparsity. To test the hypothesis we conduct experiments to investigate whether classifier with ℓ_1 regularizer have better results under the sparse feature attack. We generate a data set which is adversarially transformed by an adversary who can only manipulate a limited

Algorithm 2 Robustness evaluation under sparse feature attack

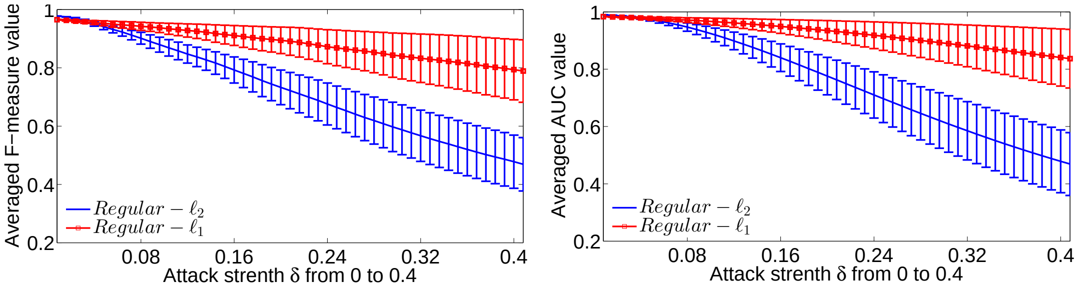
Input: Original positive data set $\{\mathbf{x}_i, y_i\}_{i=1}^{npos}$, Feature weights $\mathbf{w} \in \mathbb{R}^{d+1}$, Number of features to be changed $\{c \in \mathbb{N} | (0 < c \leq d)\}$. Attack strength $\{\delta \in \mathbb{R} | (0 < \delta < 1)\}$
Output: Accuracy of classifier on $\{\mathbf{x}_i^*, y_i\}_{i=1}^{npos}$

```
// Randomly select $c$ features of the data and index in vector $\alpha = \{0 < a_k \leq d\}_{k=1}^{c}$
for $i = 1 = i : npos$ do
    $\mathbf{x}_i^* \leftarrow \mathbf{x}_i, k \leftarrow 1$
    while $k <= c$ do
        if $w_{a_k} > 0$ then
            $x_{a_k}^* = x_{a_k}^*(1-\delta)$
        else
            if $w_k == 0$ then
                do nothing
            else
                $x_{a_k}^* = x_{a_k}^*(1+\delta)$
            end if
        end if
        $k = k+1$
    end while
end for
Evaluate classifier (using $\mathbf{w}$) on changed data set $\{\mathbf{x}_i^*, y_i\}_{i=1}^{npos}$
```

number of features. We then test the two initial classifiers with different regularizers on the transformed data set. The detailed procedure is described in Algorithm 2.

Now we report on experiments to compare the performance of ℓ_1 and ℓ_2 classifiers in a non-game setting where the classifiers were subject to feature attacks but without having an opportunity to respond. We use Spambase data set for this experiment. To simulate the attack, we randomly select twenty percent of the number of features to be changed, i.e. we set $c = 20\% \times d$. We vary the attack strength δ from 0 to 40% with step size of 0.2%.

As shown in Figure 3.7, both the classifiers start (in terms of accuracy) at nearly the same place but the classifier with ℓ_1 regularizer deteriorates at a much slower rate compared to the classifier with an ℓ_2 regularizer. This clearly demonstrates that the ℓ_1 classifier is more robust and is consistent with the theoretical observation that the ℓ_1 classifier is equivalent to robust classification.



(a) F-measure comparison.

(b) AUC value comparison.

Figure 3.7: *Regular*-ℓ_1 is more robust in both F-measure and AUC value compared to *Regular*-ℓ_2.

Table 3.2: The classifier achieves best performance when the *Cost* is close to $1 \times MB$.

Step	**F measure on future test data**						**Average**	**$Cost$**
	30/04/2000- 31/05/2001	30/04/2001- 31/05/2002	30/04/2002- 31/05/2003	30/04/2003- 31/05/2004	30/04/2004- 31/05/2005	30/04/2005- 31/05/2006		
1	**0.973**	**0.975**	0.966	0.942	0.945	0.931	0.955	0.000 $\times MB$
2	0.972	**0.975**	0.967	0.944	0.945	0.932	0.956	0.137 $\times MB$
3	0.971	0.974	0.966	0.938	0.940	0.929	0.953	0.300 $\times MB$
4	0.972	0.974	0.969	0.944	0.946	0.932	0.956	0.444 $\times MB$
5	0.969	**0.975**	0.970	0.951	0.952	0.935	**0.959**	0.586 $\times MB$
6	0.971	0.974	**0.972**	0.948	**0.955**	0.936	**0.959**	0.759 $\times MB$
7	0.970	**0.975**	0.971	0.949	0.951	0.936	**0.959**	**0.951** $\times MB$
8	0.970	0.973	0.970	**0.954**	0.952	**0.938**	**0.960**	**1.115** $\times MB$
9	0.968	0.973	0.967	0.952	0.945	0.933	0.956	1.321 $\times MB$
10	0.969	0.973	0.967	**0.954**	0.952	0.932	0.958	1.495 $\times MB$

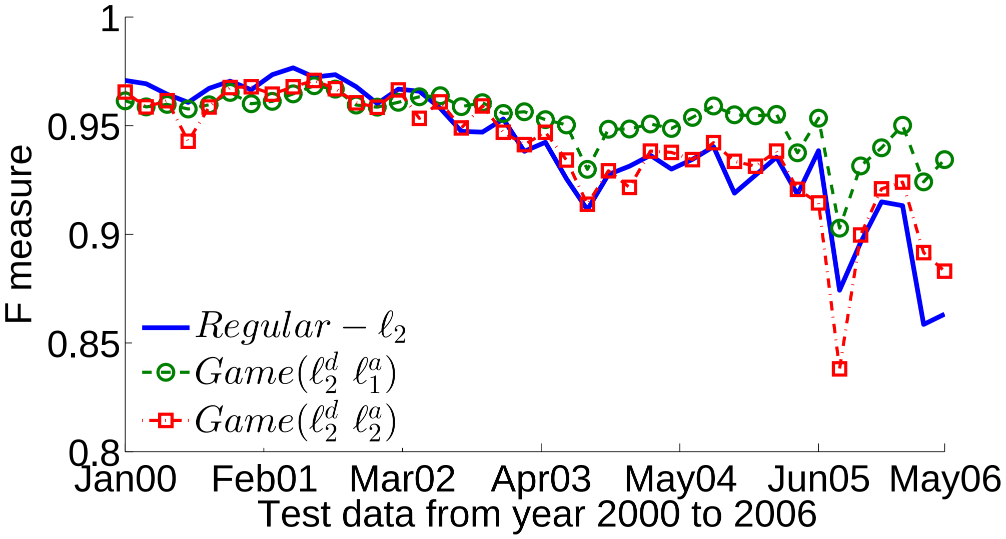
3.3.4 Evaluation of the budget *MB*

The budget *MB* of an adversary should be lower bounded by the value of the regularizer obtained using cross-validation. This is because, the value obtained from cross-validation is assumed to give best generalization performance. We can interpret the test data as a data set generated by an adversary who is restricted to use the same underlying probability distribution that generated the training data. As we noted in Section 3.2.1, using ℓ_1 regularization is equivalent to solving the robust regression problem.

We evaluate Algorithm 1 with the sparse model $Game(\ell_1^d \ell_1^a)$. To study performance of the learned classifier as a function of *Cost*, we repeatedly ran the game model until the *Cost* had substantially exceeded *MB*. We show the performance of the classifiers learned from the first 10 steps in Table 3.2. We first observe, at step 1, the regular classifier learnt with 0 adversarial *Cost* has the best performance on the near future data. While classifier learned with large adversarial *Cost* has better performance on data sets further in the future. We also notice when the *Cost* is close to *MB*, the average F-measure is close to its highest value. Thus we can conclude empirically that *MB* value derived in 3.2.2 is an appropriate budget for the adversary.

3.3.5 Evaluating logistic loss with ℓ_2^d

Here we compare the game-theoretic and regular classifier when the data miner (d) uses ℓ_2^d regularizer under logistic loss. More specifically, the three models are denoted as: *Regular*-ℓ_2, *Game*$(\ell_2^d\ \ell_2^a)$ and *Game*$(\ell_2^d\ \ell_1^a)$. We evaluate all the three models in terms of F-measure and AUC-value. The results are shown in Figure 3.8. We have further summarized the results in Table 3.3. We can conclude that game-theoretic classifiers deteriorate at a much slower rate on future data than the regular classifier. On the near future data, the regular classifier have a slightly better performance and this can be expected as the game-theoretic classifier are generalizing for better performance on future data. On the other hand, the AUC which measures the overall performance (without considering the time dimension), is nearly equal for all the three approaches.

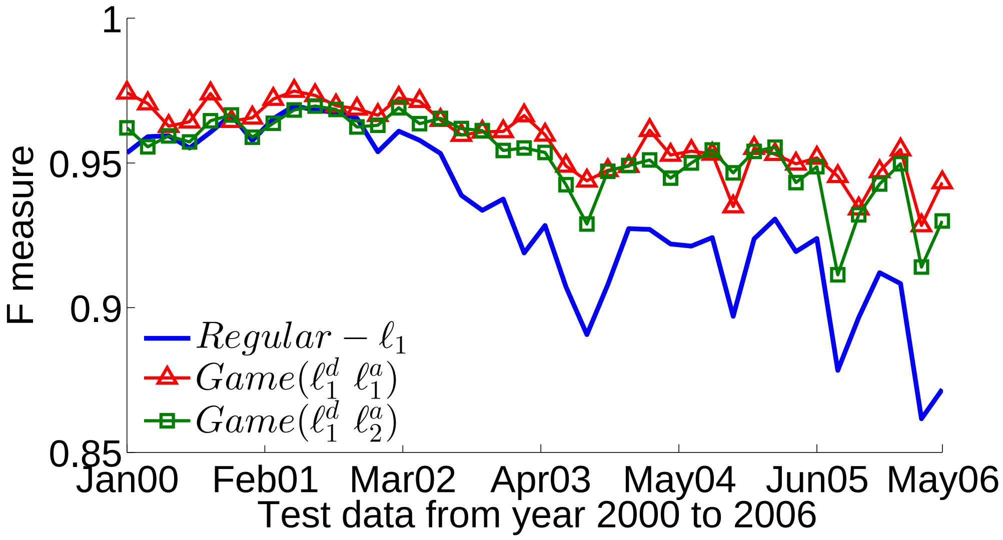


(a) F-measure on mailinglist.

Figure 3.8: *Game*$(\ell_2^d\ \ell_1^a)$ model outperforms *Regular*-ℓ_2 classifier in terms of F-measure data in further into the future.

Table 3.3: Data miner modeled with ℓ_2 regularizer. Game-theoretic classifier perform better on data further into the future.

	AUC	F-measure	
		Jan 00 - April 03	**April 03 - May 06**
Regular-ℓ_2	**0.96**	**0.968**	0.936
Game$(\ell_2^d \ell_2^a)$	0.95	0.962	0.939
Game$(\ell_2^d \ell_1^a)$	**0.96**	0.962	**0.953**



(a) F-measure on mailinglist.

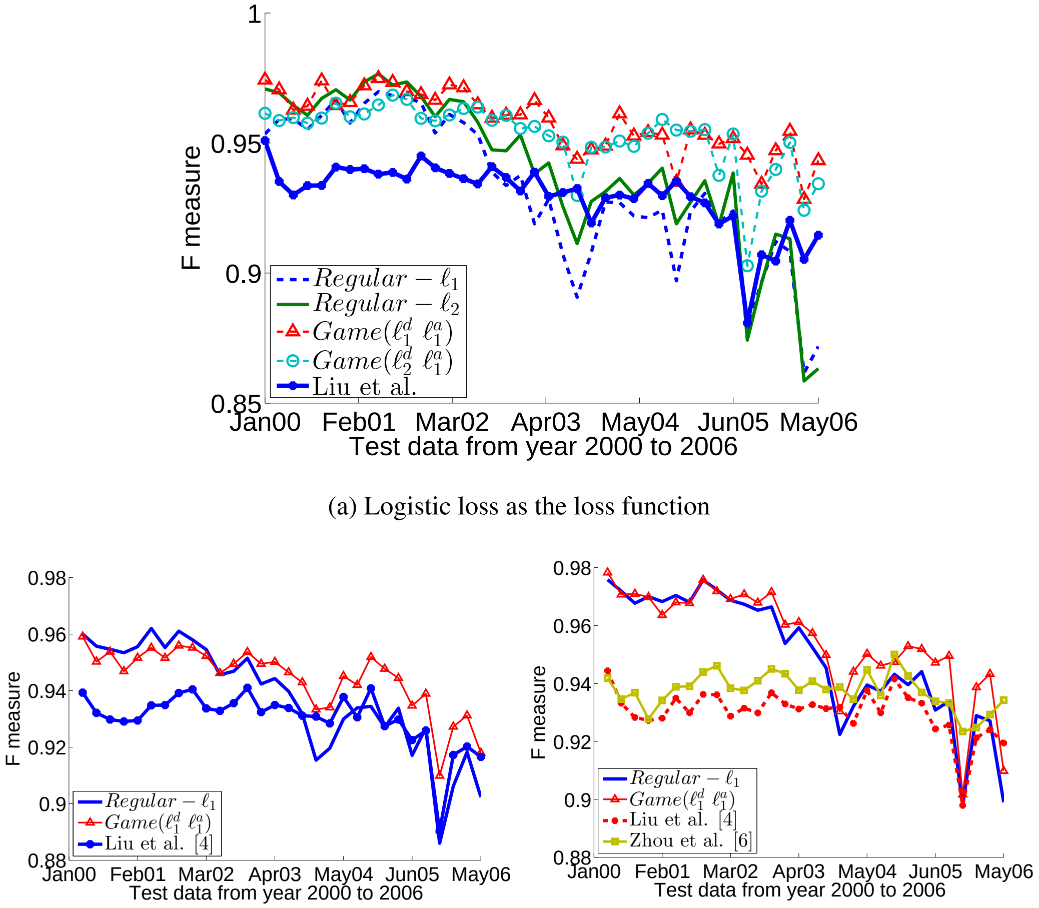
Figure 3.9: *Game*$(\ell_1^d\ \ell_2^a)$ and *Game*$(\ell_1^d\ \ell_1^a)$ both outperform the initial classifier with ℓ_1 regularizer in terms of both F-measure.

Table 3.4: Sparse model *Game*$(\ell_1^d\ \ell_1^a)$ achieves the best performance.

	AUC	F-measure	
		Jan 00 to April 03	**April 03 to May 06**
Regular-ℓ_1	0.95	0.961	0.922
Game$(\ell_1^d \ell_2^a)$	**0.96**	0.964	0.951
Game$(\ell_1^d \ell_1^a)$	**0.96**	**0.970**	**0.955**

3.3.6 Evaluating logistic loss with ℓ_1^d

Here we compare the game-theoretic and regular classifier when the data miner (d) uses ℓ_1^d regularizer under logistic loss. More specifically, the three models are denoted as: *Regular*-ℓ_1, *Game*$(\ell_1^d\ \ell_2^a)$ and *Game*$(\ell_1^d\ \ell_1^a)$. The results are shown in Figure 3.9 and Table 3.4. Surprisingly the F-measure results of the two game-theoretic classifiers outperform the regular classifier for almost the whole time span. In other words, the game-theoretic classifiers are both robust to near and far future test data. Furthermore, it is worth noting that the game-theoretic classifier *Game*$(\ell_1^d\ \ell_1^a)$ has the best performance. A similar results hold for the AUC-value as shown in Table 3.4.



(a) Logistic loss as the loss function

(b) Square loss as the loss function

(c) Hinge loss as the loss function

Figure 3.10: Sparse model *Game*$(\ell_1^d\ \ell_1^a)$ has the best F-measure results on data further into the future.

3.3.7 Comparison of ℓ_1^d and ℓ_2^d on logistic loss

Here compare ℓ_1^d and ℓ_2^d both on game-theoretic and regular classifiers. We also compare with Liu et al. [52]. From Figure 3.10a, it is clear that the use of ℓ_1^d regularizer significantly outperforms ℓ_2^d on data both near and further into the future.

3.3.8 Comparison with other methods

Table 3.5: Square loss as the loss function, $Game(\ell_1^d\ \ell_1^a)$ performs best on data further into the future.

	F-measure	
	Jan 00 to April 03	**April 03 to May 06**
$Regular\text{-}\ell_1$	**0.953**	0.922
$Game(\ell_1^d\ \ell_1^a)$	0.952	**0.937**
Liu et al	0.935	0.926

Table 3.6: Hinge loss as the loss function, $Game(\ell_1^d\ \ell_1^a)$ performs best on all the future data.

	F-measure	
	Jan 00 to April 03	**April 03 to May 06**
$Regular\text{-}\ell_1$	0.968	0.932
$Game(\ell_1^d\ \ell_1^a)$	**0.969**	**0.941**
Liu et al	0.933	0.927
Zhou et al	0.939	0.936

We now compare the two proposed game-theoretic classifier but trained using square and hinge loss functions. The results are shown in Figure 3.10b and 3.10c respectively. In both cases, we also compare with Liu et al[52]. For hinge loss, we also compare with free range attack model from Zhou et al. [90], where the adversary is allowed to manipulate the data with the budget MB.

We first notice that classifier learnt from model $Game(\ell_1^d\ \ell_1^a)$ is significantly more robust to future data in for both the two loss functions. The model of Liu et al[52] and Zhou et al. [90] exhibits similar phenomenon: they both have reasonable performance on data further into the future than the regular classifier but suffer significantly on near future data.

3.4 Summary

In many prediction environments including spam email and fraud detection, it has been observed that an adversarial phenomenon causes the prediction performance to deteriorate over time. This has resulted in a new class of machine learning methods, known as adversarial learning, which are robust in such settings. In this chapter we posit that a rational adversary is likely to employ a sparse feature attack, i.e., selectively change the features of the spam, in order to circumvent the classifier. Such an approach will not only cost less but will result in high spam utility, i.e., minimal changes are made on the spam in order to beat the spam detector. We model sparse feature attacks using an ℓ_1 regularizer. Our results clearly demonstrate that modeling an adversary as engaging in a sparse feature attack can be used to design more robust classifiers.

Chapter 4

Tikhonov or Lasso Regularization: Which is Better and When

This chapter is based on the following publication:

Wang, Fei, Sanjay Chawla, and Wei Liu. Tikhonov or Lasso Regularization: Which Is Better and When. Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on. IEEE, 2013.

4.1 Motivation

Crucial properties of a model like robustness and convergence are often related to the type of regualrizer the model use. One interesting problem addressed in this thesis is that practitioners of machine learning and data mining are confronted with the following situation. They receive or are given a large data set with millions of records and thousands of features and are interested in carrying out some form of regression, classification or ranking. For example, researchers working in the consumer internet space may be asked to predict the click through rates in the context of user, publisher and advertiser features [40]. In bio-medical settings, researchers want to predict whether MRI images can be mapped to various forms of disorder [87]. In a health insurance setting the task is to estimate the claim cost given information about patients and providers [57]. In all the above examples a practitioner is likely to conduct the following steps: (i) load and clean the data (ii) use a learning package to carry out some preliminary analysis and check the accuracy metrics; (iii) start iterating by tweaking the features

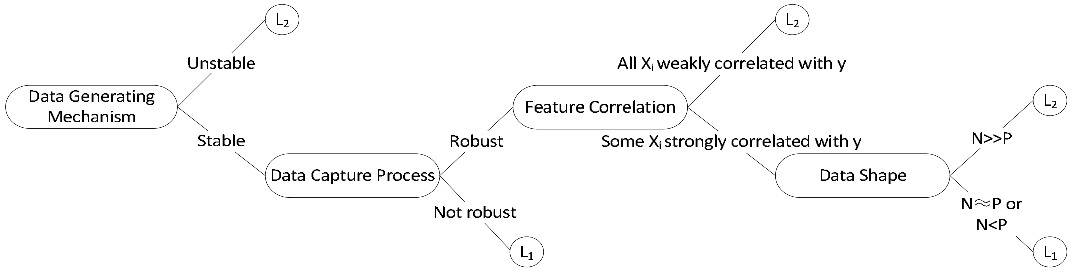


Figure 4.1: A map [left to right] to guide practitioners in their choice of a regularizer for supervised learning tasks

and the different types of regularizers available in the learning package. In this chapter, we study the different properties of the regularizer and show the principle of choosing it properly.

4.2 Contributions

While many of these results are known and scattered in the literature. Our contributions are summarized as follows:

- We first give insights of the difference between ℓ_1 and ℓ_2 regularizer through a simple analysis and thus show that why $\ell 1$ regularizer can result in sparse results.
- We provide a more complete analysis of algorithmic stability based on Xu et al. [85] Further we verify this with sufficient experiments.
- We provide a concise proof of algorithmic robustness of ℓ_1 regularization. We also verify the resulting claim with ample experiments.
- Then, we show how the existence of the Safe Rule could damage the fitting accuracy of a ℓ_1 regularized model. This is also supported by a wide range of experiments.
- Last but not least, we bring them together under a common mathematical framework and organize them using the decision map shown in Figure 4.1.

Several, somewhat surprising, conclusions can be drawn:

- If the data generating process is non-stationary (not stable), the ℓ_2 regularizer will result in a more stable solution compared to the ℓ_1 regularizer.

- The ℓ_1 regularizer is substantially more robust to noise in the data sensing or capturing mechanism.
- If most of the features in the data have weak correlation with the dependent variable then the use of ℓ_2 regularizer will result in superior prediction accuracy compared to ℓ_1.
- Finally, there is a strong relationship between the shape of the data, which relates to the volume and the dimension, and the choice of the regularizer. The shape of the data can be characterized as the ratio of number of data points to number of features

The rest of the chapter is organised as follows: In Section 4.3 we introduce the notation and specify the scope of the problem. Related work is presented in Section 4.4. We describe the "Decision Map" in detail in Section 4.5. We extensively evaluate the "Decision Map" in Section 4.6. Finally we conclude the chaper in Section 4.7 with a summary and directions for future work.

4.3 Notation and Setup

A generic supervised learning problem has the following formulation

$$f(\lambda) = \min_{w} \sum_{i=1}^{n} l(y_i, w^T x_i) + \lambda\Omega(w), \tag{4.1}$$

where $l()$ stands for the loss function, $w \in \mathbb{R}^d$ is the feature weight vector learned from the training data $x_i \in \mathbb{R}^d, i = 1,...,n$ and $\lambda\Omega()$ is the regularizer with parameter λ. For simplicity, we can either subsume the intercept term by appending a unit feature, i.e., $x \equiv [1,x]$ or by assuming all variables are centered. Common forms of loss functions (and their conjugate functions) $l(y_i, w^T x_i)$ are given in Table 4.1. Common regularizers can be ℓ_1 or ℓ_2 which corresponds to $||w||_1$ and $||w||_2^2$ respectively. In this chapter we will focus on the square loss for understanding the mathematical properties of the solutions while in the experiments we will use logistic loss to validate our results. However, since our focus is on how the choice of the regularizer impacts the resulting solution, our results are completely general and can be framed at a more abstract level by using the loss function and its corresponding conjugate function as shown in Table 4.1.

	$\ell(y_i, w^T x_i)$	$\ell^*(y_i, u)$
Square	$\frac{1}{2}(y_i - w^T x_i)^2$	$\frac{1}{2}u^2 + uy_i$
Logistic	$\log(1+\exp(-y_i w^T x_i))$	$(1+uy_i)\log(1+uy_i)$ $-uy_i \log(-uy_i)$
Hinge	$(1-y_i w^T x_i)_+$	$uy_i \times 1_{-uy_i \in [0,1]}$

Table 4.1: Commonly used loss functions and their conjugate functions. Since our focus is on the impact of the regularization, our results are general and are applicable to other loss functions by matching them with their appropriate conjugate function.

4.4 Related Work

The algorithmic overhead of solving ℓ_1 regularized systems is larger than for ℓ_2 because of the non-differentiability of the ℓ_1 norm. By analyzing the dual of a least-square regression problem with an ℓ_1 regularizer, El Ghaoui et al. [34] proposed a `SafeRule` to prune features which will, for a given value of λ have zero weights. A contribution of our work is to show the consequences of this property on the relative accuracy of the ℓ_1 and ℓ_2 solutions. Using `SafeRule` as a motivation, Tibishirani et al. [76] has proposed Strong Rules which are more aggressive in pruning features. However, they are known to lead to inconsistent models at least in theoretically constructed cases.

Andrew Ng [61] has compared ℓ_1 and ℓ_2 regularizers in the context of number of irrelevant features. He has shown that with ℓ_1 regularization, the number of training samples required for learning a good model grows logarithmically in the number of irrelevant features. With ℓ_2 regularization, on the other hand, the size of the training data needs to grow linearly in the number of irrelevant features. Thus if only a few number of features are relevant and the size of training data is relatively small, ℓ_1 regularization is preferred. Later developments in compressed sensing also show that only a logarithmic number of sensing measurements are required to recover a signal using ℓ_1 minimization.

Robustness and stability are different properties related to data capturing process, the data generating mechanism and also algorithms. In a theoretical work, Xu et al. [85] have recently demonstrated that solving a ℓ_1 regularized least square problem is equivalent to solving a least square problem with a worst case bounded perturbation in the training data. This suggests that ℓ_1 regularization should be the method of choice when the data capturing process generates data which has a low signal to noise ratio. An example is the data generated by EEG probes [69] for understanding the functionality of the brain. Xu et al. [85] also prove a "no free lunch" theorem showing that the ℓ_1 regularized models are not stable. In another work by Zhang et al. [89] showed that collaborative representation (CR) instead of ℓ_1 regularizer truly improves the classification accuracy.

Finally, it is important to note that there has been work which combines the use of both ℓ_1 and ℓ_2 regularizers. For example, the elastic net regularizer [92] can be used to apply sparsity at the level of groups of features. Consider microarray applications [67], genes often work together to regulate proteins and their expression levels tend to be highly correlated. However, for a given task only few groups of genes are relevant. Thus the elastic net method tends to sparsify at the group level but reduces the norm of the feature weights inside groups.

4.5 Decision Map

In this section, we will first investigate properties of the ℓ_1 regularization in terms of Stability and Robustness. Then we look into the properties of the data in terms of correlation and shape. These analysis will form the basis of the decision map shown in Figure 1.

4.5.1 Algorithmic stability

algorithmic stability is a well studied problem and has been used to re-derive classical generalization bounds [12, 56]. Intuitively a learning algorithm is stable, when if it is trained on two similar data sets, the output models should be similar. The formal definition of uniform stability is defined as [12, 85]:

Definition 1: An algorithm $\mathbb{L}$ has uniform stability bound of β_n with respect to the loss function l if the following holds

$$\forall D \in \mathscr{X}^n, \forall_i \in \{1, \cdots, n\}, \|l(\mathbb{L}_D, \cdot) - l(\mathbb{L}_{D^{\setminus i}}, \cdot)\|_\infty \leq \beta_n. \tag{4.2}$$

Here $\mathscr{X}^n$ is the sample space, D is a given training sample, $\mathbb{L}_{D^{\setminus i}}$ stands for the learned solution with the i^{th} sample removed from D, $l(\mathbb{L}_{D^{\setminus i}}, \cdot)$ stands for the loss of the solution on any given test data.

We provide a simplified analysis of the results presented by Xu et al. [85]. Again consider the one-dimensional case. Assume

$$w^* = \arg\min_{w} \sum_{i=1}^{n} l(y_i, w \cdot x_i) + |w|, \tag{4.3}$$

now consider the case where the feature x is replicated, for example, an data point (x_i, y_i) becomes $((x_i, x_i), y_i)$. the loss function will become

$$(w_1^*, w_2^*) = \arg\min_{w_1, w_2} \sum_{i=1}^{n} l(y_i, w_1 \cdot x_i + w_2 \cdot x_i) + |w_1| + |w_2|. \tag{4.4}$$

Now we show that $w_1^* + w_2^* = w^*$. The key is to notice that the data corresponding to both w_1 and w_2 are identical. According to the stationarity condition of the Lagrangian of lasso [36]

$$\mathbf{X}_i^T u^* \in \begin{cases} \{\lambda\} & \text{if } w_i^* > 0 \\ \{-\lambda\} & \text{if } w_i^* < 0 \\ [-\lambda, \lambda] & \text{if } w_i^* = 0 \end{cases}, i = 1, ...d, \tag{4.5}$$

we see that w_1 and w_2 have the same sign. Equation 4.4 can be expressed as

$$(w_1^*, w_2^*) = \arg\min_{w_1, w_2} \sum_{i=1}^{n} l(y_i, w_1 \cdot x_i + w_2 \cdot x_i) + |w_1 + w_2|. \tag{4.6}$$

Now consider a new data point $((0, z), 0)$ which is added to D. This leads to a new term in the loss function.

$$\sum_{i=1}^{n} l(y_i, w_1 \cdot x_i + w_2 \cdot x_i) + l(0, 0 + w_2 \cdot z) + |w_1 + w_2| \tag{4.7}$$

Now, $(w^*, 0)$ is the optimal solution of Equation 4.7. However, when the point $(0, 0, z)$ is removed, $(0, w^*)$ is an optimal solution. Thus we have showed that for ℓ_1 regularizer, $\beta_n \geq w^* z$ and the lower bound can increase arbitrarily.

4.5.2 Robustness

A learning algorithm is robust if the model it generates is resistant to bounded perturbations in the data. Robust learning algorithms is an active area of research and the robust linear regression problem is defined as

$$\min_{w \in R^d} \{ \max_{|z| \leq \lambda} \|y - (x+z)w\|_2 \}. \tag{4.8}$$

The key insight about robust regression as defined in [85] can be derived from considering the one-dimensional case. For example, we first notice that

$$\max_{|z| \leq \lambda} |y - (x+z)w| \quad \leq |y - xw| + c|w|.$$

Now consider a specific $z^* = -\lambda sgn(w) sgn(y - xw)$. Clearly $|z^*| \leq \lambda$. Furthemore

$$\begin{aligned} \max_{|z| \leq \lambda} |y - (x+z)w| &\geq |y - (x+z^*)w| \\ &= |y - xw| + |\lambda sgn(w) sgn(y - xw) w| \\ &= |y - xw| + \lambda |w|, \end{aligned}$$

thus

$$\max_{|z| \leq \lambda} |y - (x+z)w| = |(y - xw)| + \lambda |w|. \tag{4.9}$$

This generalizes to

$$\min_{w \in R^d} \{ \max_{|z| \leq \lambda} \|y - (\mathbf{x} + \mathbf{z})\mathbf{w}\|_2 \} = \min_{w \in R^d} \|y - \mathbf{x}\mathbf{w}\|_2 + \lambda \|\mathbf{w}\|_1 \tag{4.10}$$

This means solving a ℓ_1 regularized least square problem is equivalent to solving a worst case linear square problem with noise $|z| \leq \lambda$.

4.5.3 Correlation

`SafeRule` itself has only been considered in terms of feature pruning technique, here, we discuss its influence on the accuracy of the learned model. One can observe that Equation 2.16 of `SafeRule` only depends on the correlation of the independent and dependent variable. Loosely speaking, for a given $\mathbf{X}_i$, the lower the correlation, the higher the possibility this variable will be pruned. This property makes it undesirable in the case where all the features are weakly correlated with the dependent variable. i.e. $|y^T \mathbf{X}_i| \leq \tau$ for $i = 1, ...d$, where τ is a small value. In this scenario all the weak features should be utilized to learn a model, however because of the `SafeRule`, all the features are likely to be pruned with ℓ_1 regularizer. The ℓ_2 regularizer on the other hand will be able to combine all the features to produce potentially more accurate model.

Suppose we have a sample set $D = (y, \mathbf{X})$, where $\mathbf{X}_i$ are all weakly correlated with y. Further suppose $w^*_{\ell_2}$ is the optimal fit with ℓ_2 regularization and we define y_r as the residual on training data.

$$y_r = y - \mathbf{X}^T w^*_{\ell_2}$$

We can then add a new feature α which has no correlation with X_i while still weakly correlated with y. Define the new sample set as $D^* = (y_r, \alpha)$, and we have the following minimization problem:

$$f(w_\alpha) = \min_{w_\alpha} ||y_r - \alpha^T w_\alpha||_2^2 + \frac{\lambda}{2} ||w_\alpha||_2^2.$$

One should notice α is correlated with y_r, i.e. $\alpha^T y_r \neq 0$. Suppose w^*_α is the optimal fit, we want to prove that

$$||y_r - \alpha^T w^*_\alpha||_2^2 < ||y_r||_2^2. \tag{4.11}$$

The proof is as follow: We first notice that $f(w^*_\alpha)$ is upper bounded by $||y_r||_2^2$, i.e.,

$$f(w^*_\alpha) \leq ||y_r||_2^2 = f(0).$$

As $f(w_\alpha)$ is a convex minimization problem, we have :

$$\begin{aligned} f(w^*_\alpha) < ||y_r||_2^2 = f(0) \text{ if: } f^{'}(0) \neq 0 \\ \text{Now: } f^{'}(0) = 2(y_r - \alpha^T 0)\alpha + \lambda 0 = \alpha^T y_r \neq 0. \end{aligned}$$

Thus:

$$\begin{aligned} f(w^*_\alpha) = \frac{\lambda}{2} ||w^*_\alpha||_2^2 + ||y_r - \alpha^T w^*_\alpha||_2^2 \quad &< ||y_r||_2^2 \\ ||y_r - \alpha^T w^*_\alpha||_2^2 \quad &< ||y_r||_2^2. \end{aligned}$$

Here we can conclude that the new weak feature α improved the fit when using L_2 regularization.

In the case of L_1 regularization, since the new feature α is weakly correlated with y, it is quite possible that the `SafeRule`, in this case $|\mathbf{X}_i^T y| < \lambda$, will ignore that feature. Thus $w_\alpha^* = 0$, and we have

$$||y_r - \alpha^T w_\alpha^*||_2^2 = ||y_r||_2^2,$$

which does not lead to any improvement at all.

In the experiment section we formulate artificial data set 'AllWeak' with all features weakly correlated with dependent variable and show that in this case ℓ_2 achieves better accuracy.

4.5.4 Shape

[61] studied the properties of ℓ_1 and ℓ_2 regularizer in the presence of irrelevant features, which is similar to few features are highly correlated with dependent variable. As it indicated that in this case, a well learned ℓ_2 regularizer needs far more training samples. This also indicates that for ℓ_1 regularizer, if the number of training data is sufficient, for a given λ, the number of zero feature weights also achieves maximum. This will be clearly visible in the experiment section.

4.5.5 Decision map

Based on the four factors discussed in the previous four sections, we propose the "decision tree" map as depicted in Figure 4.1. Generally speaking, a practitioner should first consider the data generating process. When the data source is not stable, then one should stop analysis and choose ℓ_2 regularizer. When the data is stable, we then consider the data capture process. Data captured through instruments like sensors are intrinsically not robust and in this case we should use ℓ_1 regularizer to compensate the offset by the date. Then, if the data is also robust, we consider the correlation between dependent and independent variables. Due to the existence of the `SafeRule`, we know that ℓ_2 regularizer will be better for data sets composed of features all weakly correlated with dependent variable. Then, when data sets have some strongly correlated feature dependent variable pairs, we consider the shape of the data set. As indicated by the study of [61], for large number of training data, ℓ_2 and ℓ_1 can both learn well. However, the `SafeRule` will still come into offset. Thus, in the case of $N \gg P$, where N is the number of training sample P is the dimension of the data, One should apply ℓ_2 regularizer. For the case $N \simeq P$ and $N < P$, the number of training sample will be far from enough to training a ℓ_2 regularized model well. Thus we should apply ℓ_1 regularizer in this case.

4.6 Experiments

We present our experimental setup and results to evaluate the decision map shown in Figure 4.1 here. Note that all experiments used the binary classification problem as a prototypical data mining task where the use of regularization is important. However, all our conclusions should hold for other tasks including regression and ranking. Again, our conclusions hold for other loss functions including the hinge loss. After some careful deliberation we selected the Vowpal Wabbit (VW) [49] package for both ℓ_1 and ℓ_2 regularization. Our decision was primarily motivated by the fact that, as data set size increases, VW scales gracefully in both ℓ_1 and ℓ_2 situations. However, VW has several tuning parameters and understanding the full implications of different parameter settings can lead to a combinatorial explosion of the experimental space. Finally, all experiments were carried out on a platform with Intel core i5 processor and 4GB RAM.

4.6.1 Data Sets

We constructed three synthetic data sets and used four real data sets:

- ‘AUSUSD’ is the currency exchange rate between Australian dollar and US dollar during the time period (01/03/2007 – 01/03/2013) [6].

- ‘AllWeak’ is a synthetically constructed data set. It has one hundred features which are all weakly correlated with the dependent variable y. Also the features are partitioned into 50 pairs where features with a pair a strongly correlated with each other.

- ‘FewStrong’ is a synthetically created data set composed of hundred features with five of them have a correlation with dependent variable y as 0.1 and the rest are again weakly correlated with y.

- ‘MajorityIrrelevant’ is a synthetic data set with one hundred features with only ten features that are correlated with dependent variable y.

- ‘Mailinglist’ is a real data set which consists of publicly available emails from mailing lists and spam emails from Bruce Guenter’s spam trap during the time period $(01/04/1999 - 31/05/2006)$. The transformed features are not correlated with each other and each of them is strongly correlated with the dependent variable.

- ‘Spambase’ is also a spam email data set [29]. Spam e-mails came from their postmaster and individuals who had filed spam and non-spam e-mails from work and personal e-mails. Most of the features (48 out of 57) are frequency of key words. In this data set some of the features are highly correlated with the dependent variable.

- ‘WEBSPAM’ consists of link-based features computed from the web graph [86]. The features capture network properties including in-degree, out-degree, page rank, edge reciprocity.

A summary of the data set is shown in Table 4.2. All features of the data sets are standardized to have mean zero and standard deviation one.

Data Set Name	Samples	Features	Type
WEBSPAM-UK2007	114,529	40	Real
Mailinglist	128,117	50	Real
Spambase	4,601	57	Real
AllWeak	2000	100	Synthetic
FewStrong	2000	100	Synthetic
MajorityIrrelevant	2000	100	Synthetic
AUDUSD	2000	100	Real

Table 4.2: Four real and three synthetically constructed data sets were used for the experiments. More detail about the data sets is in the text.

4.6.2 Stability and Robustness

To evaluate the impact of stability and robustness on the regularizers we proceed as follows. We first describe the stability experiment. Suppose X is a training data set.

Let the model learnt by using the ℓ_1 and ℓ_2 regularizer on X be denoted as w^1 and w^2 respectively. We randomly remove one data point from X and reconstructed the model. Denote the model from the i-th run as w^1_{-i} and w^2_{-i}. For example, w^1_{-i} are the weights of classifier when the i-th point is removed and the model is trained using the ℓ_1 regularizer.

Now form two sets DS_1 and DS_2 as

$$DS_1 = \{\|w^1_{-i} - w^1\|_2 \mid i = 1\dots100\} \tag{4.12}$$

$$DS_2 = \{\|w^2_{-i} - w^2\|_2 \mid i = 1\dots100\}. \tag{4.13}$$

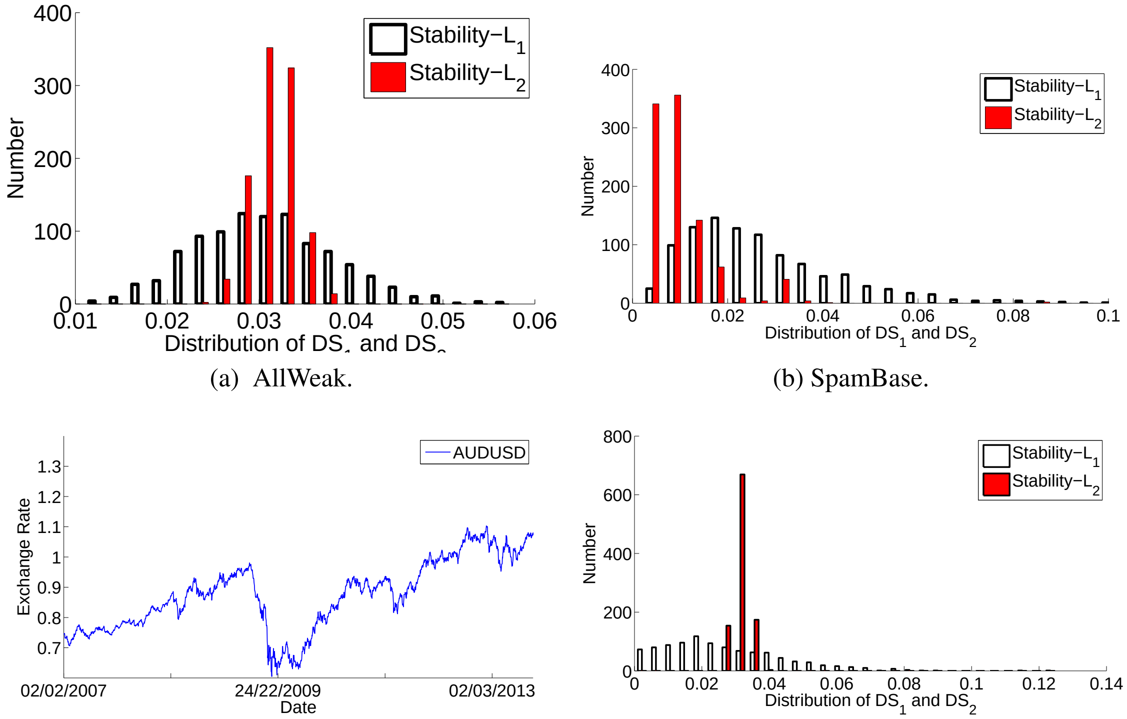
We have plotted the distribution of both DS_1 and DS_2 using three data sets: AllWeak, SpamBase and AUDUSD. and the results are shown in Figure 4.2. It is clear from the plots that the values of DS_2 are tightly concentrated while those of DS_1 are more spread out. This is especially obvious for the financial data AUDUSD, which has an unstable data generating process. This confirms our hypothesis that ℓ_2 regularizers result in more stable models than ℓ_1.

To test for robustness we need to proceed in a different fashion. We again begin with a data set X. On each separate run i we add bounded noise ΔX_i. Let w_1 be the model on X using the ℓ_1 regularizer and w_2 using the ℓ_2 regularizer. For each i we compute the AUC for $X + \Delta X_i$ for each of the regularizers. Thus $f_1^\Delta(i) = AUC(X + \Delta X_i | w_1)$ and likewise for $f_2(i)$. Let $f_2^\Delta(i) = AUC(X + \Delta X_i | w_2)$, i.e., the accuracy of the model trained on the data set $X + \Delta X_i$. Like in the case of stability, we form the sets

$$DR_1 = \{f_1^\Delta(i) - f_1(i) \mid i = 1\dots1000\} \tag{4.14}$$

$$DR_2 = \{f_2^\Delta(i) - f_2(i) \mid i = 1\dots1000\}. \tag{4.15}$$

The results of the distribution of both DR_1 and DR_2 are shown in Figure 4.3 and clearly show that the data points of ℓ_1 are more tightly concentrated compared to ℓ_2. This again confirms that when the data capturing process is noisy, then ℓ_1 leads to more robust classifiers compared to ℓ_2.



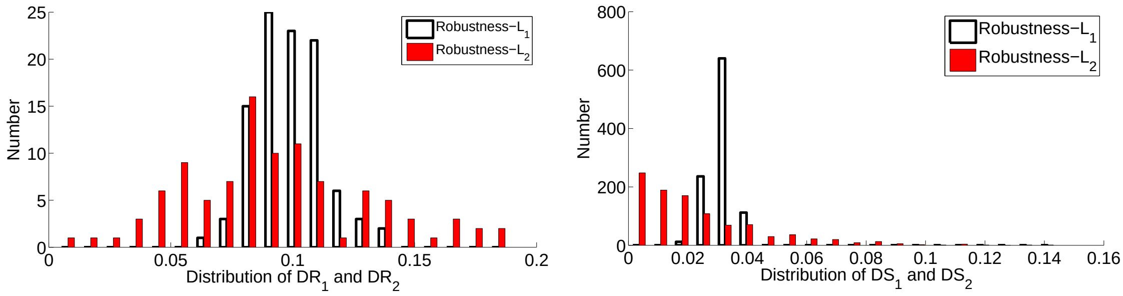
(a) AllWeak.

(b) SpamBase.

(c) Exchange Rate of AUDUSD.

(d) Stability test of AUDUSD.

Figure 4.2: (c) shows the Exchange rate of AUDUSD in six years. One can notice that the data is rather not stable. (a),(b),(d) clearly show that distribution results of ℓ_2 regularizer has lower standard deviation and thus more stable,



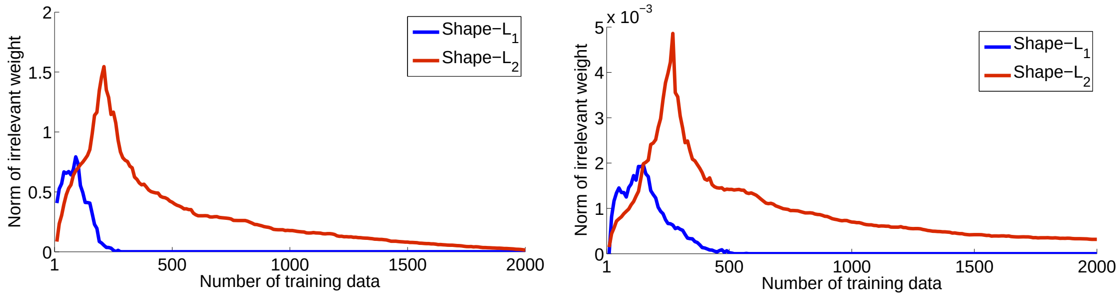
(a) AllWeak.

(b) SpamBase.

Figure 4.3: Robustness: The figures indicate that distribution results of ℓ_1 regularizer has lower standard deviation and thus more robust.

4.6.3 Shape

This experiment evaluate how many samples are required before the learning algorithm can determine if the features in a given data set are irrelevant. Figure 4.4 shows that the norm of the irrelevant features using ℓ_1 regularizer converges much faster to 0 as a function of the number of training points compared to ℓ_2.



(a) MajorityIrrelevant.

(b) Mailinglist.

Figure 4.4: Shape: The figures indicate that ℓ_1 regularization ignores the irrelevant features with much fewer training samples.

4.6.4 Correlation and Shape

In this section we investigate the influence of correlation and again the shape of the data on the choice of regularizers. Here we use all the four data sets and all the experiments are averaged over 20 runs. We only consider the accuracy of the model (as measured by AUC) and omit factors like running time. VW has almost similar time complexity for both ℓ_1 and ℓ_2 regularization.

In our experiments we vary the size of the training data and the regularizer value λ for the two different models. In all cases we start with just twenty instances and then in each step increase the number of instances by a step size of thirty. The reason we start from small data sets is to understand the impact of the regularizer when data is limited. This is partly because the ℓ_1 and ℓ_2 regularizers can be interpreted as Laplacian and Gaussian priors. For large data sets, the prior effect is often (but not always) subsumed by the (likelihood term) of the model. We also vary λ between a small range $[10^{-4}, 60^{-4}]$. Again this choice was determined by the fact that we wanted to gauge the influence of the regularizer in a transition zone.

Before we describe the results in details we want to provide a small guide to understand Figure 4.5 which contains four plots organized as four by four table. Each row corresponds to one data set. In all the plots the y-axis is always the AUC value. In the first two columns we measure the impact of different values of λ's as we increase the training data size. In column three and four, the roles of λ and the training data size are reversed. In the x-axis we vary λ and each line in the plot corresponds to a different training data set size.

4.6.4.1 AllWeak

We start by looking at the first row of Figure 4.5 which is the result of 'AllWeak' data set. From Figure 4.5a we first notice that for small value of λ, AUC value increases as training data size increases. However, for large value of λ, AUC value decreases when training data size increases to around 100, which is the number of features. This is because with more data and thus more information, the `SafeRule` is taking effect with large value of λ and potentially informative predictors are being pruned. Since `SafeRule` only applies to ℓ_1, the effect is only visible in Figure 4.5a. Figure 4.5c indicates that for ℓ_1 regularizer, AUC value is sensitive to the value of λ. This can also be explained by the effect of the `SafeRule`. For ℓ_2 regularizer, Figure 4.5b and 4.5d indicates AUC value increases in the number of training data and is stable in the value of λ. This is because all the features are weakly correlated with the dependent variable and thus all the predictors are supposed to be small. Thus a large value of λ has no impact. Perhaps the most important conclusion we can draw from the first row of Figure 4.5 is that ℓ_2 regularizer achieves better AUC value when the number of training data size increases. Again, this is because for ℓ_1 regularizer, `SafeRule` is taking effect and informative features are being pruned. This experiment validates the third node in the 'Decision Map', where when all the features are weakly correlated with the dependent variable, a practitioner should use ℓ_2 regularizer.

4.6.4.2 FewStrong

Now we look at the second row of Figure 4.5 which is the results of 'Few Strong' data set. For ℓ_1 regularizer, from Figure 4.5e and 4.5g we observe behavior as in the 'AllWeak' data set. The difference is that in this case the AUC value is also sensitive to the size of the training data. This is because only a few features are highly correlated with the dependent variable, thus more data is needed to learn a model well. Figure 4.5f and 4.5h convey similar information as in 'AllWeak' data set. Comparing ℓ_1 and ℓ_2, one can observe that the ℓ_1 regularizer achieves better AUC value at some small values of λ. However for small λ, the model is not likely to be sparse and thus defeating the one of the strong reasons for using ℓ_1. For large values of λ sparsity is achieved but the AUC value of ℓ_1 regularizer goes down training data size increases. Thus, we can conclude from this experiments that when the training number is large compared with the number of features i.e. $N \gg P$, a practitioner should use ℓ_2 regularizer.

4.6.4.3 SpamBase

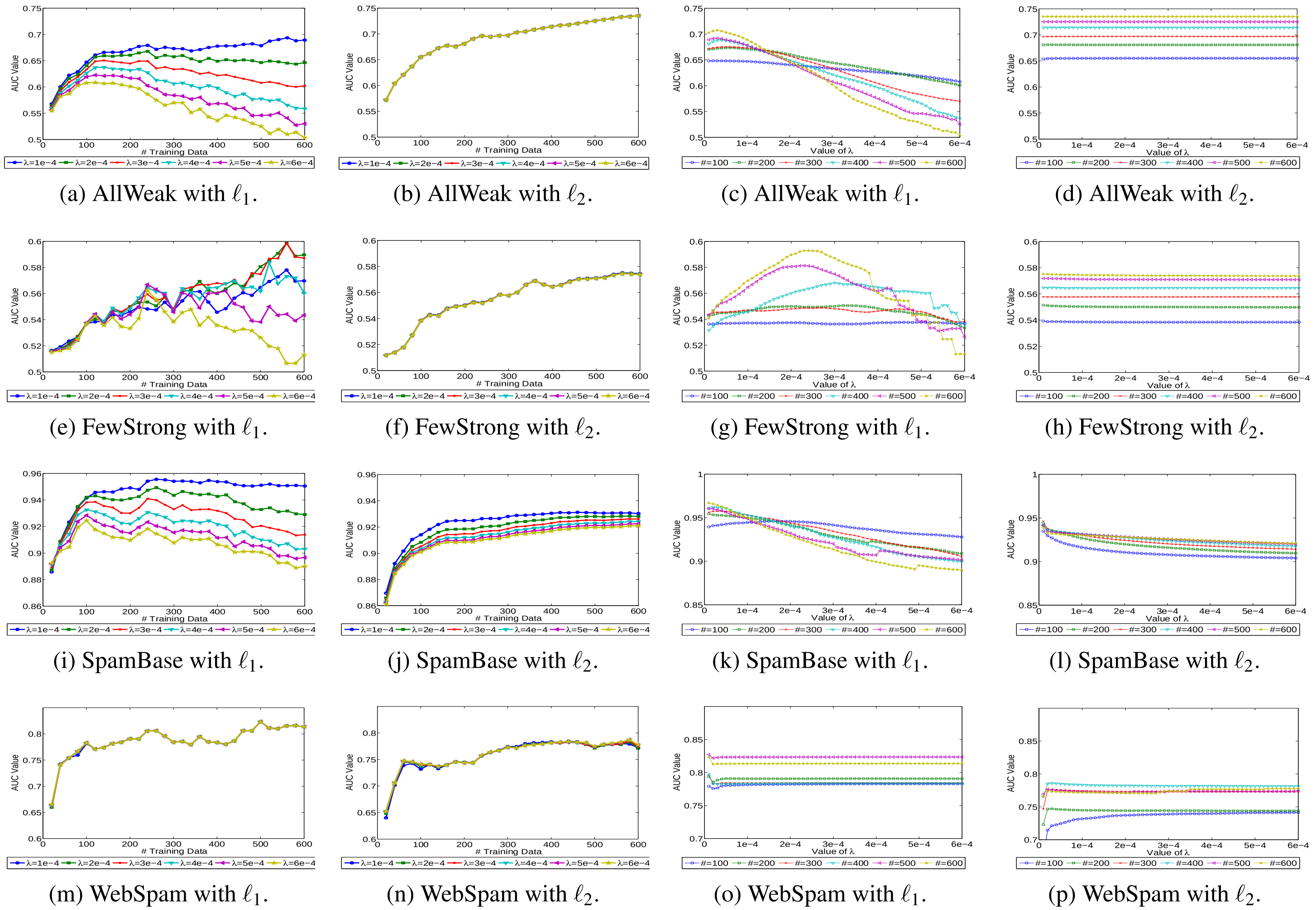
The results on the ‘SpamBase’ data set are described in the third row of Figure 4.5 The features of ‘SpamBase’ are frequency of key words and thus some words can be much more informative than others. Thus this data set is similar to the ‘FewStrong’ synthetic data set. When the training data is small , i.e. $P \approx N$, Figure 4.5i and 4.5j indicate that ℓ_1 regularizer has better performance while for large number of training data sets, ℓ_2 regularizer has better performance.

4.6.4.4 WebSpam

Now we look at the fourth row of Figure 4.5 which is the results of ‘WebSpam’ data set. Note that the features of this data set are derived features related to network properties like in-degree, page rank and edge reciprocity. It is well known that features like page rank can be used to distinguish between spam and non-spam web pages. Thus these are strongly correlated features with the dependent variable. Now for large values of λ, the ℓ_2 regularizer will pull the feature weights closer to zero thus reducing the AUC value while in the case of ℓ_1, the `SafeRule` will not come into effect.

4.7 Summary

The main contribution of the chapter is to construct a decision map which compares the performance of ℓ_1 and ℓ_2 regularizer based on four characteristics of data: stability, robustness, correlation between independent and dependent variable, and the shape. Future work will focus on a deeper mathematical analysis of the regularizer and the evaluation of the decision map on other situations.



(a) AllWeak with ℓ_1.

(b) AllWeak with ℓ_2.

(c) AllWeak with ℓ_1.

(d) AllWeak with ℓ_2.

(e) FewStrong with ℓ_1.

(f) FewStrong with ℓ_2.

(g) FewStrong with ℓ_1.

(h) FewStrong with ℓ_2.

(i) SpamBase with ℓ_1.

(j) SpamBase with ℓ_2.

(k) SpamBase with ℓ_1.

(l) SpamBase with ℓ_2.

(m) WebSpam with ℓ_1.

(n) WebSpam with ℓ_2.

(o) WebSpam with ℓ_1.

(p) WebSpam with ℓ_2.

Figure 4.5: AUC value of four data sets. For each data set we have four plots, ℓ_1 and ℓ_2 regularization and one is with x-axis as the number of training samples and one is with x-axis as the value of λ.

Chapter 5

Latent Outlier Detection and the Low Precision Problem

This chapter is based on the following publication:

Wang, Fei, Sanjay Chawla, and Didi Surian. Latent outlier detection and the low precision problem. Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description. ACM, 2013.

5.1 Introduction

It is well known that new scientific discoveries or "paradigm shifts" are often triggered by the need to explain outliers [47]. The availability of large and ever increasing data sets, across a wide spectrum of domains, provides an opportunity to actively identify outliers with the hope of making new discoveries.

The obvious dilemma in outlier detection is whether the discovered outliers are an artifact of the measurement device or indicative of something more fundamental. Thus the need is not only to design algorithms to identify complex outliers but also provide a framework where they can be described and explained. Sometimes it is easy to explain outliers. For example, we applied the recently introduced *k*-means-- algorithm [20] on the 2012 season NBA player data set[1]. *k*-means-- extends the standard k-means algorithm to simultaneously identify clusters and outliers. The result of the Top-5 outliers are shown in Table 5.9 and matches with the top players in the NBA "All Star" team.

[1] www.basketball-reference.com

An NBA star is an outlier and given the highly competitive nature of NBA, *an outlier is most likely a star.* Or in other words there are no bad players in the NBA but some players are very good! However, in many other applications it is not at all clear how to proceed to explain outliers. This can be termed as the "Low Precision Problem (*LPP*)" of outlier detection.

Table 5.1: Given the highly competitive nature of the NBA, not only are stars outliers, but outliers are stars! All the top five outliers are well known leading players of NBA.

Outlier Rank	Player Name	All Star Team (Y/N)
1	Kevin Durant	Y
2	Kobe Bryant	Y
3	LeBron James	Y
4	Kevin Love	N
5	Russell Westbrook	Y

Problem 1 *The Low Precision Problem (LPP) in outlier detection is that*

$$P(\textit{genuine outlier}|\textit{predicted outlier}) \approx \textit{low} \tag{5.1}$$

LPP occurs because it is hard to disambiguate genuine outliers from errors occurring in the measurement device.

The main algorithm proposed in this chapter extends the work on k-means-- proposed in et al. [20] which unifies clustering and outlier detection. Furthermore we have taken inspiration from a body of work on multiple subspace outlier detection to distinguish between genuine and accidental outliers [59].

5.2 The multiple subspace view

A starting point towards addressing *LPP* and explaining and sifting genuine outliers from measurement errors is to view data from multiple perspectives [59]. In the context where data entities are described by a vector of features, examining an entity in all possible feature subspaces can potentially lead to isolating genuine outliers. This is especially true in high dimensional settings. For example assume that each entity is described by a feature vector of size m. Furthermore, assume that the probability of each feature being recorded incorrectly is p and is independent of other features. Then if m is large, the probability that at least one feature value has been recorded incorrectly is $1-(1-p)^m$ and this can be close to 1 when m is large. Thus having at least one feature value which is corrupted due to measurement error is high. However if we can view the data in multiple subspaces then a genuine outliers will consistently stand out.

A limitation of the multiple subspace approach is that there are exponentially many subspaces leading to intractable algorithms. However the problem can be ameliorated if we notice that in real data sets, the *intrinsic dimensionality* (which is the minimum number of variable need to represent the data) of the data is much lower than the *ambient dimensionality* (which is the actual number of dimension we perceived of the data) as we now explain.

5.3 High-Dimensional Anomalies

It is now part of the data mining folklore that in real data sets, the "degrees of freedom" which actually generate the data is small, albeit unknown. This can be illustrated using examples from computer vision. For example, consider a subset of the Yale Face data shown in Figure 5.1. Each image is very high-dimensional ($64 \times 64 = 4,096$), however the set of images together live on a three dimensional manifold where the degree of freedom are governed by the rotation of the camera and the lighting. The bottom right hand image (transpose of the top left image) is an outlier as it lives outside the manifold [25].

Thus given a high-dimensional space, if we can project data into a lower-dimension space which preserves the intrinsic structure of the data, then not only can we identify outliers efficiently but potentially explain the discovered outliers. An example of manifold-preserving projection are the family of random projections which preserve pairwise distances with high probability [25]. However, while random projections can lead to improvements in efficiency, by their very nature they make it nearly impossible to interpret the outliers. Thus we need a set of projections to which we can also ascribe some meaning. We next describe matrix factorization methods which are projections of data into lower dimensional space where each dimension aggregates a group of correlated features.

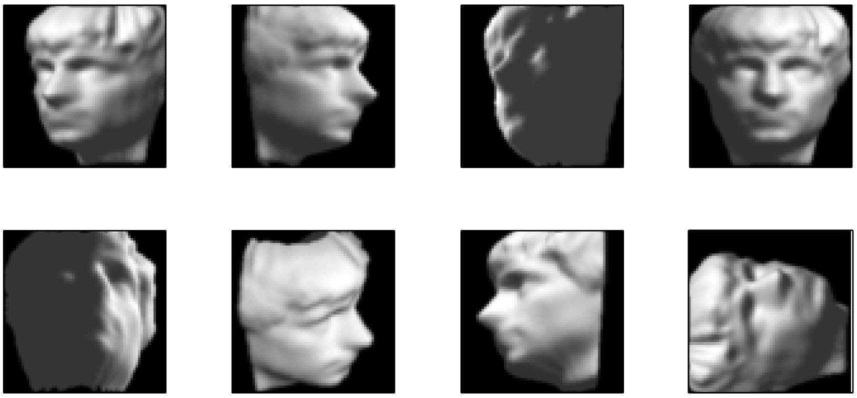


Figure 5.1: An example to explain the difference between intrinsic and ambient dimension. Samples from the 698-image Yale face data. Each 64 x 64 is a point in a 4,096 dimensional space. However the set of images live in a three dimension set. The bottom right image is added as the transpose of the top left image and is an outlier.

5.4 Matrix Factorization

As we have noted, the challenge in outlier detection is the difficulty to separate true outliers from those data points that are caused because of measurement errors. We have also noted that in high-dimensional space most of the features tend to be correlated. Thus if a data point is a true outlier that fact should be visible in several features. Thus if we take a subspace approach then a genuine outlier will show up as an outlier in more subspaces than an accidental outlier. The challenge in pursuing a subspace approach is that the *space of subspaces* is exponential in the number of features and thus intractable to explore for most practical problems.

One way to address the intractability is to reduce the dimensionality of the original space. This can be carried out using matrix factorization approaches. Factorization is a principled approach of simultaneously aggregating correlated features into a reduced number of “meta-features” which in turn can be imbued with semantics related to the application domain. While Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) have been around for a long time, the recent surge in new methods like Non-Negative Matrix Factorization (NMF) and Bayesian factorization have enhanced the reach of these methods [70]. The key advantage of NMF, say over SVD, is the enhanced interpretation that these methods afford. For example, if X is non-negative document-word matrix or data from a micro-array experiment and

$X = UV$ is a non-negative factorization (i.e., both U and V are also non-negative) then the factors can be ascribed a meaning as shown in Table 5.2.

Table 5.2: Non-Negative Factorization provides enhanced interpretation of the meta-features. In text processing, the meta-features can be interpreted as topics, while in micro-array analysis, the meta-features are group of correlated genes.

X	U	V
Document-Word	Document-Topic	Topic-Word
Exp-Gene	(Exp,Functional Group)	(Functional Group, Gene)

5.4.1 The impact of Projections

Outliers can potentially be impacted in different ways depending upon the nature of outliers. For example, consider the projection shown in Figure 5.2. The projection shown will have no impact on data point 1, will force data point 3 into a cluster and data point 2 will continue to remain an outlier even though it is far away from the projection plane. Now, which one of these points are genuine outliers is potentially application dependent. However, if we take a subspace perspective, then data point 1 is more likely a genuine outlier. This is because it preserves the correlation between its components but each component is moved far from the main cluster.

5.4.2 Sensitivity to Outliers

While techniques like NMF provide a promising way to address the combinatorial explosion problem associated with multiple subspace viewing, like SVD, they are highly sensitive to outliers. Thus if our aim is to find outliers, then our method of discovering outliers should not in turn be affected by them. For example, it is well known that both mean and the variance-covariance matrix are extremely sensitive to the presence of even one extreme value and their use for outlier detection will often mask the discovery of genuine outliers. Thus we first have to modify NMF to make them more robust against outliers. Thus we define the following problem:

Problem 2 *[NMF(k,ℓ)] Given a non-negative matrix* $\mathbf{X} \in \mathbb{R}_{+}^{m \times n}$*, fixed integers k and* ℓ*, find matrices* $\mathbf{U} \in \mathbb{R}_{+}^{m \times k}$*,* $\mathbf{V} \in \mathbb{R}_{+}^{k \times n}$ *and a subset* $L \subset N$*,* $|L| = \ell$*, which minimizes* $||X_{-\ell} - UV_{-\ell}||_F$*, where* $X_{-\ell}$ *is a submatrix consisting of all columns except those from the set L.*

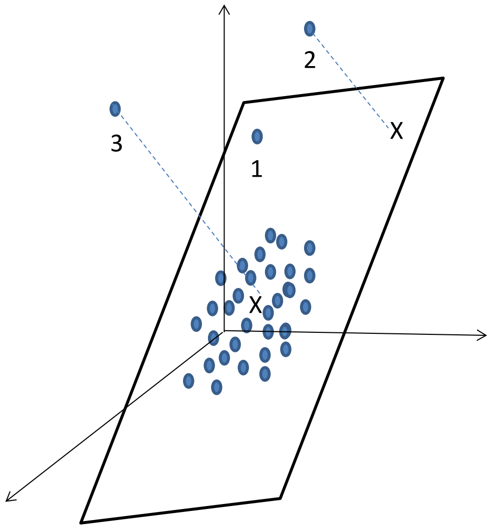


Figure 5.2: The figure shows the impact of projections of outliers in a lower dimensional space. Data points 1 and 2 remain outliers after projection, while data point 3 is mixed with normal after the projection [41].

To solve the $NMF(k,\ell)$ problem we present the R-NMF algorithm shown in Algorithm 3. The algorithm belong to the class of alternating minimization methods and is very similar to the standard NMF algorithm except for a few caveats. We begin by initializing U in Line 1. In Line 4, we solve for V which minimizes the Frobenius norm of $\|X - U^{i-1}V\|_F$. In Line 5, we compute the residual between X and the current estimate of the product $U^{i-1}V$. In Line 6, we rank the residuals based on the norm of their column values, and L is the index vector of the ranking. We then generate new matrices $X_{-\ell}$ and $V_{-\ell}$ by removing the first ℓ values of the set X and V in Line 7 and 8. In Line 9, we estimate U by minimizing the Frobenius norm of $X_{-\ell}$ and $UV^{i}_{-\ell}$. We iterate until the convergence criterion is met.

The R-NMF algorithm is an analogous extension of the recently proposed k-means-- algorithm [20]. We should note that another extension for NMF to find outliers has been proposed by Xiong et al. [84] introduced the method of Direct Robust Matrix Factorization (DMRF). The DMRF method first assumes the existence of a small outlier set S and then infers the low-rank factorization UV by removing S from the data set. It then updates S by using the inferred factorization. In the experiment section we will compare R-NMF with DNMF.

Algorithm 3 [R-NMF Algorithm]

```
Require: A matrix $X$ of size $m \times n$, $m$ number of features, $n$ number of samples
    $k$ the size of the latent space
Ensure: An $m \times k$ matrix $U$ and $k \times n$ matrix V
    $R \approx UV$
$U^0 \leftarrow$ random $m \times k$ matrix
$i \leftarrow 1$
while (no convergence achieved) do
    $V^i = \arg\min_V \|X - U^{i-1}V\|_F$
    $R = X - U^{i-1}V^i$        \\$R$ is a residual matrix
    Let $L = \{1, 2, \ldots, n\}$ be a new ordering of the columns of $R$ such
    $\|R(:,1)\| \geq \|R(:,2)\| \ldots \geq \|R(:,n)\|$
    $X_{-\ell} \leftarrow X(:, L \setminus L(1:\ell))$
    $V_{-\ell} \leftarrow V(:, L \setminus L(1:\ell))$
    $U^i = \arg\min_U \|X_{-\ell} - UV^i_{-\ell}\|$
    $i \leftarrow i + 1$
end while
```

The R-NMF algorithm forms the kernel of the subspace algorithm, SR-NMF shown in Algorithm 4 which combines subspace enumeration with R-NMF. Note we only take subspace of the "meta-features." The intuition is that genuine outliers will emerge as outliers in the latent subspaces.

Here we design algorithm that incorporate both the concept of multi subspace view and matrix factorization. As we mentioned before the shortage in [59] is that due to the high dimensionality nature in most of the data set, one simply can not brute force and traversal each and every subspaces. We solve this problem by investigate the problem in a latent space where data are confined in a much small dimensionality.

5.5 Experiments and Results

In this section we evalute both R-NMF,DRMF from [84] and SR-NMF on several data sets. Our ultimate objective is to verify if SR-NMF can be used to address the **LPP** problem. All our experiments were carried out on a PC with following configurations. Intel(R) Core(TM) i5-2400 CPU @3.1GHz 4GB RAM running on 64-bit Microsoft Windows 7 Enterprise Edition.

Algorithm 4 [SR-NMF]

```
Require: A matrix $X$ of size $m \times n$, $m$ number of features, $n$ number of samples, $k$ the size of the latent space, $\ell$ number of outliers
Ensure: A vector $R$ represent the ranking of anomalies with a score in descending order
Using $R-NMF$ algorithm we get $U$ and $V$ such that $X \approx UV$
$(U,V) = R-NMF(k,\ell)$
$j \leftarrow 0; RANKS \leftarrow$ empty matrix;
STEP1 generate ranks for each subspace
for $i = 1 \rightarrow k$ do
    generate all set of combinations $AS$ from ($k$ choose $i$)
    for each $S \in$AS do
        $Residual = X - U(:,S)V(S,:)$
        $RNorm = columnNorm(Residual)$
        $[-,RANK] = sort(RNorm, \text{'descend'})$
        $RANKS = [RANKS; RANK]$
        $j++$
    end for
end for
STEP2 merge ranks into one rank
$R \leftarrow$ vector of size $n$;
for $i = 1 \rightarrow j$ do
    for $p = 1 \rightarrow n$ do
        $R(RANKS(i,p)) = R(RANKS(i,p)) + i$
    end for
end for
sort $R$ in descending order
$[-,R] = sort(R, \text{'descend'})$ (Note: Matlab Notation)
```

5.5.1 Data Sets

We used three data sets from different application domains which we now describe.

NBA 2012

The NBA 2012 data set consists of 483 players and 20 features. The features are values related to metrics used to evaluate performances of the players in a season. For example, features like 3PAr (3 point Attempt Rate), TOV (Turnover Percentage) and MP(Minutes Played) etc.

Abstract

The Abstract data set is a collection of abstracts from Physics and Science papers. The data is formatted into document-words matrix with 1000 samples from each group and 3894 features which are the most frequent words. In the experiments we compose two types of data sets from this data set. One is composed of 1000 Physics and 100 Science abstracts and we label Physics abstract as anomalies. The other one is 100 Physics and 1000 Science abstracts.

Spambase

'Spambase' is a spam email data set [29] consisting of 4,601 emails out of which 1,813 (39%) are spam. The spam e-mails came from their postmaster and individuals who had filed spam and non-spam e-mails from work and personal e-mails. Most of the features (48 out of 57) are frequency of key words.

Research Abstracts

We took around one thousand computer science paper titles from DBLP and also a thousand physics research paper abstracts. We created two data sets. In the first we kept the thousand CS titles and merged them with one hundred physics abstracts. For the second data set, we kept the thousand physics abstracts and merged them with a random subset of one hundred computer science titles. We call the former CSet and the latter PSet.

5.5.2 Results

We report results on robustness, convergence, runtime and accuracy on the three aforementioned data sets.

Results:Robustness of R-NMF

Here we report on results about the sensitivity of the R-NMF against the classical NMF algorithm, which we denote as O-NMF. We applied both R-NMF and O-NMF algorithm on the NBA 2012 data set but modified one entry in the matrix as a multiple of the mean value. This is shown on the x-axis of Figure 5.3. For each different value on the x-axis we computed the U matrix and computed the difference in the norm of the new U matrix and the original U matrix. The U matrix is the base matrix and stores the meta-features in terms of the original features.

Figure 5.3 shows that R-NMF is more robust against perturbations while the U matrix using O-NMF increases without bound. This clearly demonstrates that the traditional NMF algorithm should not be used for any serious applications as it is extremely sensitive to data perturbations.

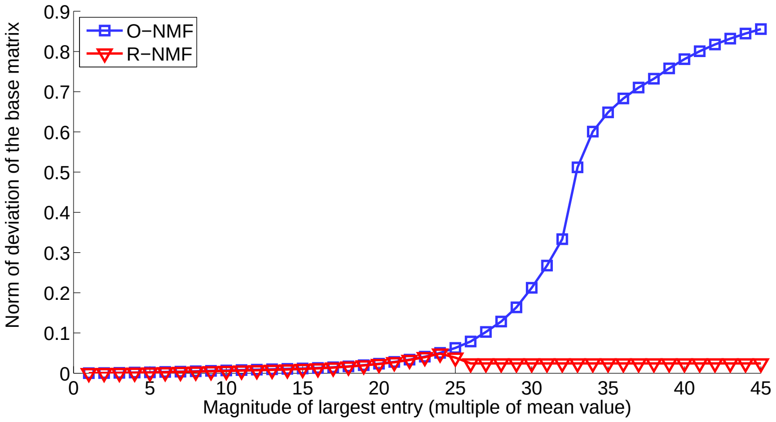


Figure 5.3: R-NMF is substantially more robust against the presence of outliers in the data compared to standard O-NMF.

Results:Convergence Analysis

Here we investigate the convergence properties of the R-NMF algorithms. From Algorithm 3 we know that for each iteration R-NMF will reconstruct U with a given number of outliers excluded. However, each iteration the algorithm may exclude different data points as outliers, this could potentially make the algorithm unstable. Thus, it is necessary to study whether this new algorithm will converge properly.

We conduct the experiments as follows. We use the Spambase data set, and set the number of outliers for R-NMF as the number of spam emails. We vary k and present the results for k=9,12,15, and 18.

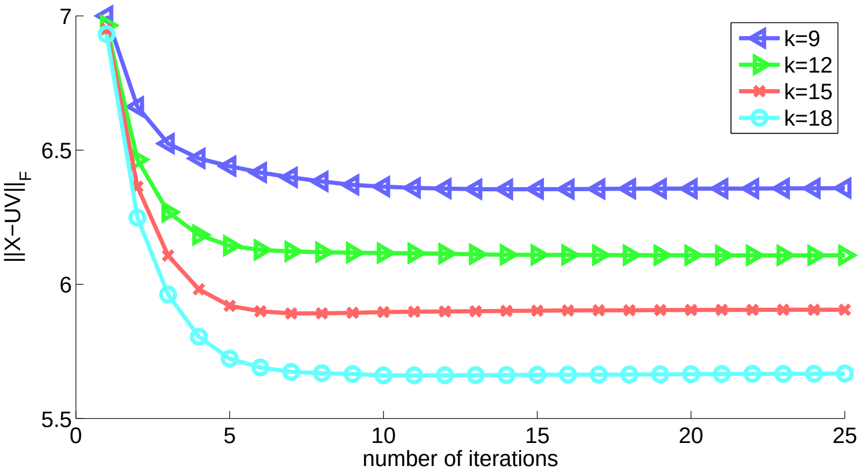


Figure 5.4: R-NMF converges with all given settings of k. As the dimension of the subspace (k) increases, residual of R-NMF algorithm goes down.

As can be seen from Figure 5.4, the first thing one can notice is that with bigger k, the residual of the algorithm goes down. This is because with bigger k, the decomposed matrices UV can better reconstruct the original X. Most importantly, the algorithm converge at all given settings of k within 20 repetitions.

Results:Runtime

We present the run time results of R-NMF algorithm for the Spambase data sets in Figure 5.5 respectively. As expected, we observe that the run time of R-NMF decreases as the number of outliers is increased. This trend follows the intuition of R-NMF algorithm that the construction of base matrix U is based on the data X without the anomalous points (Algorithm 3 line 5-8).

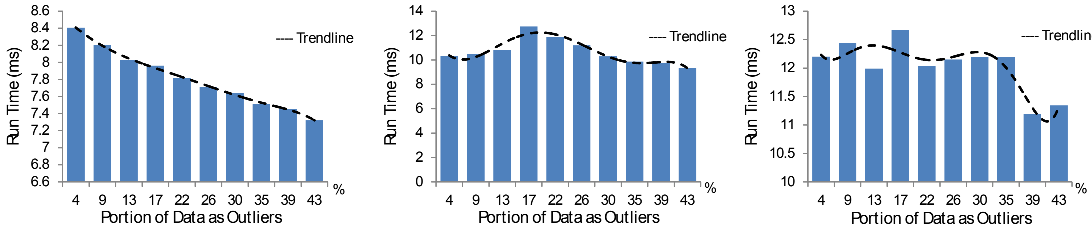


Figure 5.5: Average Run time R-NMF on Spambase data set: (**Left**) k = 1, (**Middle**) k = 2, (**Right**) k = 3. As the number of outliers increases, the run time for R-NMF decreases. The values here are the average values for all iterations.

Results:Precision and Recall

We compute precision and recall on the Spambase, PSet and the CSet data sets. The outliers are considered as *positives*. The experiments are conducted as follows. We vary the two variables: k and ℓ, We compared the two proposed algorithms: R-NMF and SR-NMF against the Direct Robust Matrix Factorization (DMRF) approach proposed by [84]. The results for different values of k and different sizes of the outliers specified are show from Table 3-7. At the moment it is hard to draw conclusions from the results. Futher work is required to analyse the results and determine the root cause of the outliers.

The experiments are conducted as follows. We vary two variables: k and ℓ, and take precision and recall as the metrics. For this data set, we define the **precision** as the number of true positive divided by the number of predicted positive (Equation 5.2), while the **recall** is defined as the number of true positive divided by the number of positive (Equation 5.3). Note that here we refer the *predicted positive* as the number of outliers detected by our algorithms, *positive* as the above-mentioned records that have complications, and *true positive* as the number of records detected by our algorithms which are part of records that have complications. Both R-NMF, SR-NMF and DRMF are applied with the same settings. Since the data set is already selected with fine granularity, thus the rank of the original matrix is low, we set small values for k, from 1 to 3 precisely, while ℓ is varied from 200 to $2,000$.

$$Precision = \frac{\texttt{number of true positive}}{\texttt{number of predicted positive}} \tag{5.2}$$

$$Recall = \frac{\texttt{number of true positive}}{\texttt{number of positive}} \tag{5.3}$$

Spambase data set.

'Spambase' is a spam email data set [29] with a total number of 4,601 email, out of which 1,813 (39%) are spam emails. The spam e-mails came from their postmaster and individuals who had filed spam and non-spam e-mails from work and personal e-mails. Most of the features (48 out of 57) are frequency of key words.

For the Spambase data set, we perform the same steps as previously. In this experiment, we vary the variables k from 6, 9 and 12, while ℓ is varied from 100 to 500. We use the same equations (Equation 5.2 and 5.3) to compute the precision and recall, however, in this case the we consider the spam emails as the positive data. We present the results for precision and recall on Spambase data set in Table 5.3 and 5.4 respectively. The tables show that in some configurations, R-NMF gives competitive results, however, in general SR-NMF outperforms RNMF and DRMF in both precision and recall.

Table 5.3: Precision on Spambase: DRMF, SR-NMF and R-NMF. Best values are highlighted.

k	Portion of data as outliers								
	7%			10%			13%		
	DRMF	SR-NMF	R-NMF	DRMF	SR-NMF	R-NMF	DRMF	SR-NMF	R-NMF
6	0.27	**0.30**	0.29	**0.32**	0.26	0.29	**0.37**	0.32	0.36
9	0.26	0.26	**0.30**	0.28	**0.31**	0.28	0.31	**0.35**	**0.35**
12	0.25	**0.32**	0.30	0.30	**0.33**	0.29	0.30	0.32	**0.36**

Table 5.4: Recall on Spambase: DRMF, SR-NMF and R-NMF. Best values are highlighted.

k	Portion of data as outliers								
	7%			10%			13%		
	DRMF	SR-NMF	R-NMF	DRMF	SR-NMF	R-NMF	DRMF	SR-NMF	R-NMF
6	0.05	**0.06**	0.05	**0.08**	0.07	0.07	**0.12**	0.10	**0.12**
9	0.05	0.05	**0.06**	0.07	**0.08**	0.07	0.10	**0.12**	**0.12**
12	0.04	**0.06**	0.05	**0.08**	**0.08**	0.07	0.10	0.10	**0.12**

Table 5.5: Precision on Abstract: DRMF, SR-NMF and R-NMF. PhysicsAnomaly.

k	Portion of data as outliers								
	35%			40%			45%		
	DRMF	SR-NMF	R-NMF	DRMF	SR-NMF	R-NMF	DRMF	SR-NMF	R-NMF
6	0.10	**0.13**	**0.13**	0.10	**0.12**	**0.12**	0.09	**0.11**	**0.11**
9	0.09	0.12	**0.14**	0.10	**0.12**	**0.12**	0.09	**0.11**	**0.11**
12	0.10	**0.12**	**0.12**	0.09	0.12	**0.13**	0.09	**0.11**	**0.11**

Table 5.6: Recall on Abstract: DRMF, SR-NMF and R-NMF. PhysicsAnomaly.

k	Portion of data as outliers								
	35%			40%			45%		
	DRMF	SR-NMF	R-NMF	DRMF	SR-NMF	R-NMF	DRMF	SR-NMF	R-NMF
6	0.39	0.49	**0.50**	0.45	0.51	**0.54**	0.47	**0.56**	0.55
9	0.36	0.47	**0.52**	0.45	0.52	**0.54**	0.46	**0.56**	**0.56**
12	0.39	**0.48**	0.47	0.40	0.53	**0.55**	0.45	**0.56**	0.52

Table 5.7: Recision on Abstract: DRMF, SR-NMF and R-NMF. ScienceAnomaly.

k	Portion of data as outliers								
	35%			40%			45%		
	DRMF	SR-NMF	R-NMF	DRMF	SR-NMF	R-NMF	DRMF	SR-NMF	R-NMF
6	0.13	**0.15**	0.15	**0.16**	0.15	**0.16**	**0.15**	**0.15**	0.14
9	0.16	0.16	**0.18**	0.16	0.16	0.16	0.15	0.15	0.15
12	0.17	0.16	**0.18**	0.16	0.16	0.16	0.15	0.15	0.15

Table 5.8: Recall on Abstract: DRMF, SR-NMF and R-NMF. ScienceAnomaly.

k	Portion of data as outliers								
	35%			40%			45%		
	DRMF	SR-NMF	R-NMF	DRMF	SR-NMF	R-NMF	DRMF	SR-NMF	R-NMF
6	0.49	0.56	**0.58**	**0.72**	0.66	0.70	**0.72**	0.72	0.70
9	0.60	0.60	**0.69**	0.70	0.69	0.70	0.72	**0.73**	0.72
12	0.65	0.60	**0.69**	0.70	**0.72**	0.70	0.72	0.72	**0.73**

Abstract data set.

Here we compare the three algorithms on two sets of data sets as we described in the Data Sets section. We first generate a data set composed of 1000 Sciences sample and 100 Physics samples. We label the Physics samples as anomalies and denote this data set as PhysicsAnomaly. Then similarly we generate another data set composed of 1000 Physics samples and 100 Science samples. We label the Science samples as anomalies and denote this data set as ScienceAnomaly.

The settings are as follows. For each experiment, we set k as 6,9 and 12. l is set as 35%, 40% and 45% of the full sample space. Results are presented in Table 5.5,5.6 , 5.7 and 5.8

We can observe that overall, R-NMF has the best performance in both precision and recall. We can also learn from this experiment that simply identifying minorities as anomalies can give reasonable result.

Basketball data set.

Here we compare the NMF-Subspace algorithm an k-means--on the Basketball data set. k is set as 10 for both the two aoglrithms. In terms of appearance, there is only one difference in the first five anomalies compared with k-means--algorithm. 'Dwight

Howard' instead of 'Russell Westbrook' indentified as anomaly. This is a better result since 'Dwight Howard' is a more famous player.

Table 5.9: Given the highly competitive nature of the NBA, not only are stars outliers, but outliers are stars! All the top five outliers are well known leading players of NBA.

Outlier Rank	Player Name	All Star Team (Y/N)
1	Kevin Durant	Y
2	Dwight Howard	Y
3	Kevin Love	N
4	LeBron James	Y
5	Kobe Bryant	Y

5.6 Summary

Outlier Detection is a core task in data mining. In fact as the size and complexity of data sets increases the need to identify meaningful and genuine outliers will only grow. Almost all major applications ranging from health analytic to network data management to bio-informatics require analytical tools which can identify and explain genuine outliers.

The core challenge in outlier detection is to distinguish between genuine and noise outliers. The former are indicative of a new, previously unknown process while the latter is often a result of error in the measurement device. The difficulty to distinguish between genuine and noise outliers leads to the Low Precision Problem (*LPP*). Our claim is that *LPP* is the fundamental problem in outlier detection and algorithmic approaches to solve *LPP* are urgently needed.

One approach to distinguish between genuine and noise outliers is to take a multiple subspace viewpoint. A genuine outlier will stand out in multiple subspaces while a noise outlier will be separated from the core data in much fewer subspaces. However the problem in subspace exploration is that current methods are unlikely to scale to high dimensions.

Matrix factorization methods provide a balanced compromise between full subspace exploration in the feature space versus exploration in the meta-feature or latent space. The advantage of working in the latent space is that many of the features are aggregated into a correlated meta-feature. Often these features in the latent space can be imbued with a semantic meaning relevant to the problem domain. For example, in the case of text mining, the features correspond to words while meta-features correspond to topics.

The challenge with matrix factorization methods is that they are highly sensitive to outliers. This can be a serious problem whenever there is a mismatch between the data and the proposed model. One way to ameliorate the problem is to use an alternate minimization approach to estimate both the matrix decomposition and the outlier set. This is the basis of the NMF(k,ℓ) problem and the R-NMF algorithm. Preliminary results show that R-NMF is substantially more robust compared to NMF in the presence of data noise. This opens up a promising avenue for further exploration and address the *LPP*.

Chapter 6

Network Analysis on Healthcare

This chapter is based on the following publication:

Wang, Fei, Uma Srinivasan, Shahadat Uddin, and Sanjay Chawla. Application of Network Analysis on Healthcare. Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on. IEEE, 2014.

6.1 Introduction

Previous work on health insurance analytics using data mining and predictive modelling [73] gives a good understanding of the semantics and the data available in a private health insurance (PHI) claim, and the claiming patterns of hospitals and medical providers. Within the context of Australian PHI there are two types of claims. A medical claim is sent by a doctor - also referred to as a provider - who performs a service to treat a patient who is a member of a particular private health insurer. The medical claim has information about the provider, the member, the hospital where the patient was treated, the details of the treatment and the cost of the services provided. A hospital claim is sent by a hospital's billing department and includes details of treatment, theatre charges, accommodation charges, prosthetics charges and charges for other services provided. Leveraging on that understanding, we have started using social network analysis techniques to model provider relationships, and analyse the impact of provider community structures on healthcare costs and quality of care.

We present two types of networks to explore collaboration among medical providers: (**i**) collaboration networks (CN) designed to capture the collaboration among surgeons, anaesthetists and assistant surgeons (**ii**) surgeon centric collaboration networks (SCCN) which explore an individual surgeon's connections.

In terms of the network representations used in this chapter, a node in the network represents a (medical) provider such as surgeon, anaesthetist, assistant surgeon; the node size indicates the total amount charged by that provider; the thickness of the edge (or tie strength) connecting two nodes represents the number of common hospital admissions between the two providers. In this chapter, an admission refers to a single episode of admitted patient care. The time interval between the date of admission and the date of discharge represents the length of stay for that admission.

In addition to size of nodes and tie strength, other network measures - closeness centrality and betweenness centrality that are related to the position of the node in the network, and centralization measures that indicate how central its most central node is in relation to how central other nodes are, can provide interesting insights about the influence of the node in the overall communication control capacity and the network. For example, the larger nodes with a more influential position in the network have the capacity to provide additional meaning within the context of the graph.

In the context of healthcare, the questions we are trying to answer are:

- Is there a team structure that emerges as providers work together on a number of shared admissions?

- What is the impact of an individual surgeon's network on cost and quality of care of the surgeries performed?

- What types network structures have positive or negative impact on cost and quality of care?

Our experiments indicate that betweenness centralisation in the SCCN network is the variable that has significant positive influence on Length of Stay (LoS), Complication rate and Medical cost. This gives an indication that nodes with high betweenness centrality are likely to be in more demand.

Our theoretical analysis combined with empirical investigation over a large data set also suggest that surgeons who collaborate with more number of teams appear to have a lower average LoS.

The rest of this chapter is organized as follows: Section 6.2 presents a brief review of collaboration models explored in the context of healthcare domain. Section 6.3 describes our research methodology to explore collaborations among providers using PHI claims data. Section 6.4 presents an analysis of our findings. And finally Section 6.5 presents some conclusions and future work.

6.2 Collaboration in health care

Our study of surgeon collaboration presented in this chapter offers a unique perspective as it combines theoretical analysis with empirical investigations of a PHI large data set. The design of the collaboration model presented in this chapter is influenced by the requirements of domain experts who wish to understand the nature of team structures that have an impact on cost as well as quality of care provided to patients for specific types of treatments. In order to keep it simple, in this chapter, we have used only knee procedures as the exemplar treatment group. We have designed similar models for other orthopaedic procedures as well other treatment groups such as cardiology and cardio-thoracic procedures. More details of the application scenario is explained in the following section.

The hospital and medical claims processed by an insurer contain data that specify the type of service provided during an admission, the length of stay for that admission, and the cost of that service. The service is specified as a Medicare Benefit Schedule (MBS) code [21], as stipulated by the Australian Government. The hospitals also send additional data related to an admission, once the patient is discharged. For any given hospital admission, we deal with three sets of data:

1. Medical claims - these show the provider-ID i.e. who performed a service, and the service is indicated by the MBS code for the specific type of treatment performed while the patient was in hospital;

2. Hospital claims - these are sent by the billing department of the hospital, and include MBS codes, accommodation cost, prosthetic costs, laboratory and radiology costs; and

3. Hospital discharge data that consolidates the patient's clinical care during that particular admission, and includes details such as length of stay, whether this was

an unplanned readmission, and additional diagnosis codes that indicate complications or infections that occurred during that admission. Therefore the discharge data provides us with valuable information pertaining to quality of care.

We use data from all three sources to design our network models. The network graph presented in this chapter represents the collaboration among three specific types of medical providers; the surgeons, the anaesthetists and assistant surgeons, as they perform knee-related surgical procedures.

6.3 Surgeon Collaboration network Design

In this section, we first report on the graphical models designed to capture the collaboration among medical providers. In addition we also explore a Surgeon-Centric Collaboration Network (SCCN) which explores an individual surgeon's connections. Finally, we provide network concepts that are related to our work.

6.3.1 Design of Collaboration Network (CN)

The objective of our initial design is to investigate the quality of care provided by a specific provider or a group of providers who collaborate while performing knee surgeries. The PHI domain experts are interested in understanding the impact of collaboration among the three types of providers: surgeons, anaesthetists, assistant surgeons (also refer to assistants in the rest of the chapter). This leads us to design a tripartite graph in which the nodes correspond to the three types of providers.

The data we have for any private health insurer include the three types of data sets specified in Section 6.2. Therefore, the information represented in the three sets of data offers us content-rich health information about each admission episode. The admissions are categorized by the treatment codes as specified in the MBS coding taxonomy. For example, knee surgeries are coded in the following hierarchy: 'Therapeutic $\rightarrow$ Surgical Operation $\rightarrow$ Orthopaedic $\rightarrow$ Knee'. The nodes represent the providers, and the edges as the number of common admissions shared by the two providers. We then associate the node size with the total medical charges of the corresponding provider. The three different types of providers are shown in three distinct colors: red for surgeons, blue for anaesthetists and light blue for assistants. A thicker edge indicates a higher number of shared admissions. Figure 6.1 shows an example of a collaboration graph.

Just by glancing at the graph, one can immediately identify the 'big' providers, i.e. providers with high medical charges, as well as highly connected providers. We can also see isolated cluster of providers. Often such isolated clusters indicate providers working in a specific geographic region. This network graph offers a powerful visualization to study collaboration among providers.

The primary focus of our investigation is to study the impact of the collaboration network structure on quality of care. To do this we consider all possible network features.

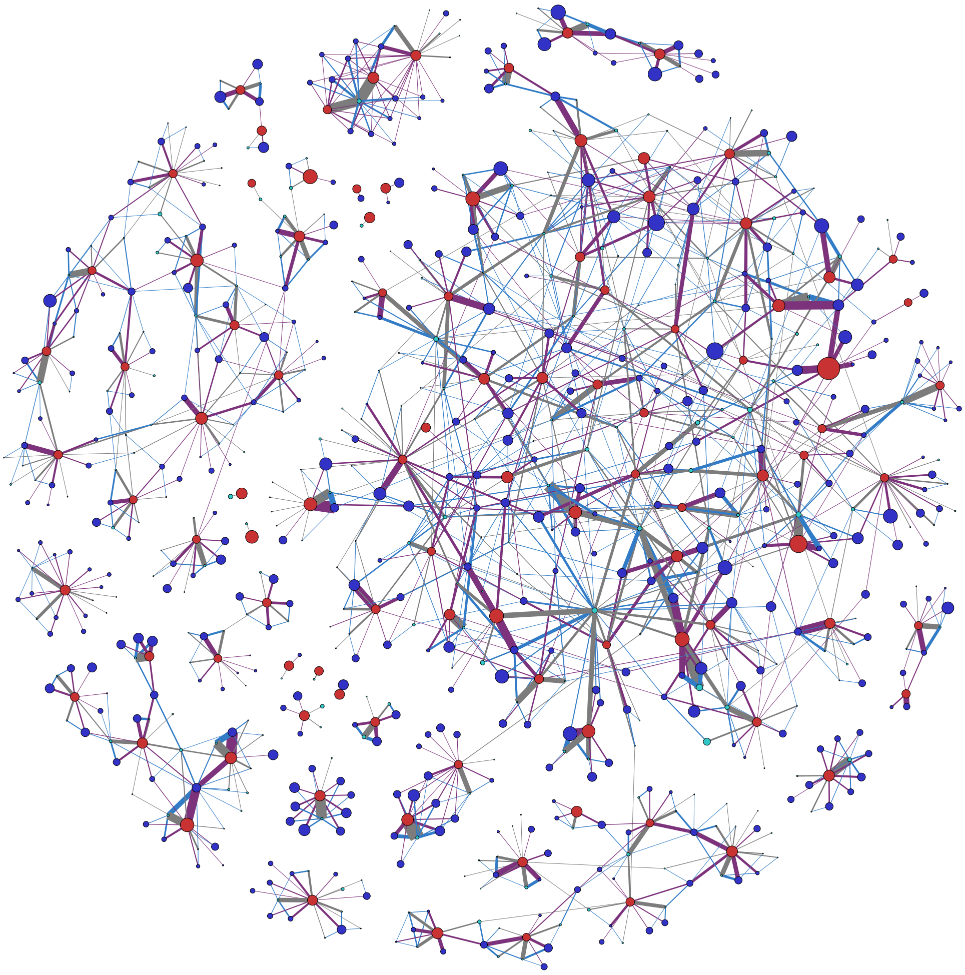


Figure 6.1: The tripartite graph represent the collaboration between three types of providers: surgeons (red), anaesthetists (blue) and assistants (light blue). The edge thickness is modeled as the number of collaborating claims by two types of providers. The size of the node is modeled as the medical charge of the provider.

6.3.2 Design of Surgeon Centric Collaboration Network (SCCN)

Since the focus is on surgeons, we investigate a specific surgeon node in the CN and build a lower level Surgeon-Centric Collaboration Network (SCCN). The SCCN is a network of a specific surgeon. It shows how a specific surgeon collaborates with the assistants and anaesthetists, and the hospital(s) in which they work together while performing knee surgeries. The individual surgeon node is not shown in the SCCN as all admissions (which are modeled as the edges) relate to a particular surgeon. Therefore, we only model two types of edges, one is the edge between assistants and hospitals, the other is between anesthetists and hospitals. Since it's a surgeon centric network, we have not shown the links between assistants and anesthetists. However such links are shown in the CN graph. The SCCN network also shows the hospitals where the surgeon performs knee surgeries. The hospital node is represented symbolically in the form of a building. The size of the building indicates the total medical cost. Edge thickness is modeled as the number of admissions of the specific surgeon with an anaesthetic or assistant in that hospital. Two SCCN graphs are shown in Figure 6.2. The graph on the left shows a surgeon who only works in one hospital and collaborates with nine anaesthetists or assistants. The graph on the right shows a surgeon who works in two hospitals and collaborates with anaesthetists or assistants who also work in those hospitals.

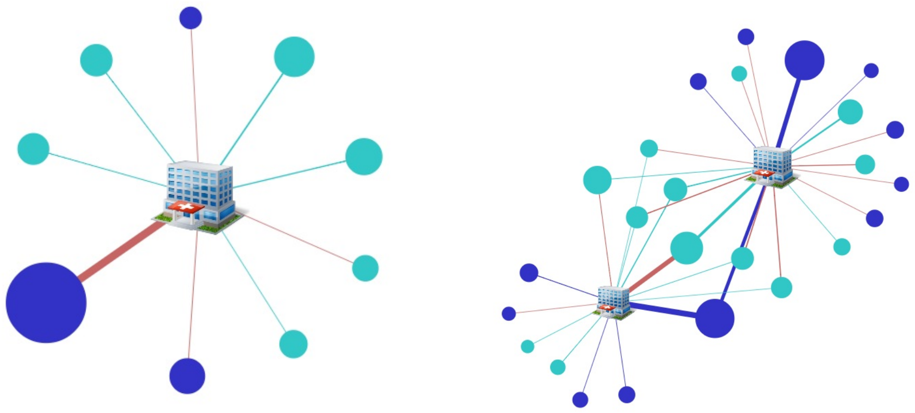


Figure 6.2: Two SCCN graphs with hospital represented by a building icon.

6.4 Data analysis

This section describes the experimental analysis staring with the data preparation, selection of network variables , selection of quality of care parameters, the regression model and finally an empirical investigation to compare the theoretical results within the context of the large PHI data corpus.

6.4.1 Data preparation

6.4.1.1 Selection of admission-related variables

In terms of non-network variables, we have identified four admission related features, which are shown in the top section of Table 6.1. The admission data shows all the medical providers who are involved in treating a patient during that admission. We consider four types of providers that includes: anaesthetists, assistants, pathologists and imaging providers. Typically a surgery has one principal surgeon and assistant and anaesthetist who work with the surgeon during the surgery. Specifically, we consider the number of distinct providers a surgeon collaborates with while performing a knee surgery. For the data analysis, we consider the percentage of distinct providers who collaborate with the surgeon rather than the absolute number of providers. The percentage is calculated with the denominator as the sum of distinct number of the four types of providers collaborating with the surgeon in knee procedure.

6.4.1.2 Selection of network variables

For network features, we first consider CN graph as depicted in Figure 6.1. We have three types of nodes in the graph, out of which, around 500 nodes are 'surgeon' nodes. Amongst several possible network variables, we have specifically selected five network features as shown in Table 6.1. We have chosen these variables as they have the potential to offer insights into the collaboration patterns among providers.

Clustering-coefficient: The local clustering coefficient of a vertex (node) in a graph quantifies how close its neighbors are to being a clique (complete graph). In our context, this represents the strength of the surgeon's network.

Number of triangles: The global clustering coefficient is based on triplets of nodes. A triplet consists of three nodes that are connected by either two (open triplet) or three (closed triplet) undirected ties. A triangle consists of three closed triplets, one centred

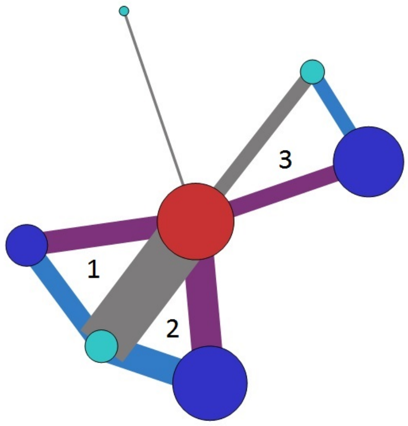


Figure 6.3: Surgeon node with Number of triangle as 3

on each of the nodes. In our context, a triangle shows the three types of providers i.e. surgeons, anaesthetists and assistants working together while performing knee surgeries. For example, in Figure 6.3, we depict a surgeon node with its surrounding assistants and anaesthetists. This is a small subset extracted from the CN graph shown in Figure 6.1. The surgeon node, which is represented as the red colour, has three triangles connected to it. This indicates that the surgeon performs knee surgeries frequently with three pairs of assistants and anaesthetists. . The features from the CN graph are all node-level features.

Next we consider network-level features. For each surgeon, we have one SCCN graph. We will consider the four network measures for the SCCN graph which are shown in the bottom section of Table 6.1.

6.4.1.3 Selection of Quality of Care parameters

As for the quality of care, we have chosen three parameters:

LoS - the average length of stay for all admissions of the surgeon. This information is available in hospital discharge data as explained in Section 6.2.

Medical cost - the average medical charge for all the knee related admissions teated by the surgeon.

Complication rate - calculated as the percentage of admissions with complications out of all the admissions of a surgeon.

Table 6.1: The table shows all the features we have extracted from the data and the network.

	Non-network features
1	%. of distinct anaesthetists
2	%. of distinct assistants
3	%. of distinct pathologists
4	%. of distinct imaging providers
	Network features of CN
1	Clustering coefficient
2	Number of triangles
3	Degree centrality
4	Closeness centrality
5	Betweenness centrality
	Network features of SCCN
1	Degree centralisation
2	Closeness centralisation
3	Betweenness centralisation
4	Density

6.4.1.4 Data cleansing and transformation

Our data set for knee surgeries includes a total of $59,256$ admissions performed by 870 surgeons. However, in order to make robust conclusions, we only considered surgeons who had more than ten claims. We further looked at the distribution of the surgeons according to each variable shown in Table 6.1 and removed surgeons who appeared as outliers. Our analysis was carried out on a set of 559 surgeons. For all the variables in Table 6.1, we applied z-score standardization. Thus each variable had a mean of zero and a standard deviation of one. In the simple regression analysis, that we will report on, this allow us to interpret the constant and the "slope" term appropriately.

6.4.2 Simple linear regression

The quality of care parameters introduced in Section 6.4.1.3 are the dependent variables in all the regression experiments. Since the independent variables are semantically distinct in the healthcare domain, they have been dealt with independently. Hence an individual linear model has been constructed for each variable. Although most of the linear models have a low R^2 value, our focus are the β values, which are significant.

Table 6.2: Table explores the impact of all non-network attributes on quality of cares (i.e. LoS, Medical cost)

Model	Dependent Variable	Independent Variable	R^2 value	β	Constant	Sig.
1	LoS	**%. of distinct anaesthetists**	**0.098**	**-0.438**	**3.506**	**0**
2		**%. of distinct assistants**	**0.023**	**-0.214**	**3.506**	**0**
3		%. of distinct pathologists	0.003	0.074	3.506	0.211
4		**%. of distinct imaging providers**	**0.179**	**0.592**	**3.506**	**0**
5	Medical cost	**%. of distinct anaesthetists**	**0.042**	**-58.127**	**1016.063**	**0**
6		**%. of distinct assistants**	**0.011**	**-29.481**	**1016.063**	**0.015**
7		%. of distinct pathologists	0.002	14.200	1016.063	0.240
8		**%. of distinct imaging providers**	**0.074**	**77.484**	**1016.063**	**0**

Table 6.3: The table explores the impact of the network structure around a specialist (based on SCCN) on quality of cares (i.e. LoS, Complication rate and Medical cost).

Model	Dependent Variable	Independent Variable	R^2 value	β	Constant	Sig.
1	LoS	**Degree centralization**	**0.014**	**0.164**	**3.506**	**0.005**
2		**Closeness centralization**	**0.023**	**0.212**	**3.506**	**0**
3		**Betweenness centralization**	**0.033**	**0.253**	**3.506**	**0**
4		Density	0	0.024	3.506	0.681
5	Complication rate	Degree centralization	0.002	0.002	0.047	0.343
6		Closeness centralization	0.001	0.001	0.047	0.496
7		**Betweenness centralization**	**0.014**	**0.006**	**0.047**	**0.005**
8		Density	0.001	-0.001	0.047	0.494
9	Medical cost	Degree centralization	0	-1.684	1016.063	0.889
10		**Closeness centralization**	**0.013**	**32.698**	**1016.063**	**0.007**
11		**Betweenness centralization**	**0.011**	**29.638**	**1016.063**	**0.014**
12		**Density**	**0.009**	**-27.014**	**1016.063**	**0.025**

Table 6.4: The table shows the impact of network position of individual specialist in the complete network (CN) on quality of cares (i.e. LoS, Complication rate).

Model	Dependent Variable	Independent Variable	R^2 value	β	Constant	Sig.
1	LoS	Clustering coefficient	0.001	0.052	3.506	0.384
2		**Number of triangles**	**0.005**	**-0.101**	**3.506**	**0.089**
3		Degree centrality	0.003	-0.080	3.506	0.179
4		Closeness centrality	0.004	-0.085	3.506	0.149
5		Betweeness centrality	0	-0.016	3.506	0.788
6	Complication rate	Clustering coefficient	0.002	-0.002	0.047	0.277
7		**Number of triangles**	**0.007**	**-0.004**	**0.047**	**0.048**
8		Degree centrality	0.002	-0.002	0.047	0.298
9		Closeness centrality	0.002	0.002	0.047	0.350
10		Betweeness centrality	0	0.001	0.047	0.712

6.4.2.1 Non network features

Table 6.2 explores the impact of all admission-related features on the dependent variables (i.e. LoS, and Medical cost). We can see that a higher percentage of anaesthetist and assistant indicates a lower LoS and Medical cost, while a higher percentage of pathologists and imaging providers indicates a higher LoS and Medical cost. This is intuitive since admissions with more imaging may be more severe situations and thus incur longer LoS and higher Medical cost.

6.4.2.2 Network features of SCCN

In Table 6.3, we observe that betweenness centralisation is the only variable that has significant positive influence on LoS, Complication rate and Medical cost. This can be interpreted as follows: From the perspective of a SCCN structure, a high betweenness centralisation indicates that the structure of the corresponding SCCN follows a star-like or centralized structure since betweenness centralisation reaches its highest value of 1 for a star network. A star-like or centralized network has few actors with higher betweenness centrality values and the rest actors have very low betweenness centrality values. In this type of network, only a small number of actors play major collaboration and communication role (Wasserman and Faust 2003). That indicates there is a presence of "network hubs" in this type of network. On the other side, if network actors have almost equal level of network connectivity (as like a line graph) then betweenness centralisation will be small and in such networks there does not present any "network hub". Therefore, SCCN, where participating actors have almost equal level of network connectivity, will produce lower LoS, Complication rate and Medical cost. In the context of health care domain, this offers an interesting insight. In their corresponding hospitals, healthcare managers or administrators could encourage a practice culture where each member will have equal level of network connectivity.

6.4.2.3 Network features of CN

Table 6.4 explores the impact of the network position of the individual specialist in the complete network (CN) on independent variables (i.e. LoS, Complication rate). We can observe that in the case of both LoS and Complication rate, the variable 'Number of triangles' has a negative correlation. That is, when a surgeon works with a large number of distinct groups, LoS and Complication rate are lower.

Intuitively, we have two assumptions with respect to the variable 'Number of triangles': (**i**) Surgeons who work with large number of distinct assistants or anaesthetists could be involved in more complicated surgeries and thus resulting longer Los and higher complication rate. (**ii**) Surgeons who consistently work with only a few distinct assistants or anaesthetists have a lower number of triangles. For these cases, our analysis shows a higher LoS. Our conjecture is that this limits external influence of other providers on the surgeon. The converse case where the number of triangles is higher clearly shows lower LoS. Thus, to figure out which assumption is true we investigated the different categories of knee surgeries and their impact on LoS in Section 6.4.3.

6.4.3 Treatment analysis

Table 6.5 shows the distribution of the different types of knee surgeries performed in the data set used for analysis in this chapter. The data set used includes about 59,256 knee surgeries performed by 559 surgeons over a period of 2 years. As per the MBS descriptions, there are four broad categories of knee surgeries with varying degrees of complexity. Accordingly, the average length of stay for each category of knee surgery varies. Column 3 of the table also shows the distribution of the four categories of knee surgery. We conducted an empirical investigation to analyze the performance of teams of surgeons indicated by the No, of triangles as shown in Table 6.4. Table 6.6 shows two groups of providers: Group A and group B. Group A represents the 200 surgeons having the least Number of triangles, and group B represent 200 surgeons with largest Number of triangles. The purpose of this analysis is to compare the Average LoS for each category of knee surgery for the two groups of providers.

Table 6.5: Average LoS and percentage of admissions of the four knee categories.

Treatment type	**Average LoS**	**Admissions %**
Knee arthroscopy	1.24	**58**
Knee Revision	4.40	1
Knee Reconstruction	2.15	9
Knee Replacement	7.66	28

Table 6.6: We can observe that, in terms of all the four treatment types, group B consistently has a lower Average LoS compared to group A and also the whole data set as shown in Table 6.5.

Treatment type	**Average LoS**		**Admissions %**	
Group	A	**B**	A	B
Knee arthroscopy	1.31	**1.21**	58	57
Knee Revision	7.69	**4.01**	1	1
Knee Reconstruction	2.38	**2.07**	7	10
Knee Replacement	7.67	**7.61**	31	28

Next we compare the Average LoS for each category of knee surgery for the two groups in Table 6.6 with the Average LoS of the complete data set summarised in Table 6.5. We can observe that in all four categories of knee surgeries, group B consistently has a lower Average LoS compared to group A, as well as the whole data set shown in Table 6.5. The empirical investigation implies that surgeons who work with a higher number of teams appear to have a lower length of stay. One could intuit that there is social learning that comes into play. However, further investigation is required to confirm the intuitive analysis. Since the category distribution for group A and B are almost identical. This makes assumption (**i**) in Section 6.4.2.3 invalid. Thus assumption (**ii**): Lower Number of triangles limits external influence of other providers on the surgeon, is a possible explanation.

6.5 Summary

In this chapter we have investigated the impact of network structure on the performance of surgeon teams with respect to efficiency metrics including Medical costs, Length of Stay (LoS) and Complication rate. Our data set was obtained from Australian PHI data and consists of both medical and hospital claims. To reduce the impact of confounding variables, we focused our analysis on “knee surgeries.” Our results provide a strong indication that network features like degree, betweenness and closeness centralization and number of triangles have a statistically significant impact on efficiency metrics. In particular, for surgeon centric networks, betweenness centralization is significant for all three metrics: Length of Stay, Complication rate and Medical cost. This observation can potentially be used by health care providers to reorganize surgical teams and improve the overall efficiency of health care delivery.

Chapter 7

Social Learning on Surgeons' Behaviour

7.1 Introduction

In order to understand the impact of social learning on the behavior of members in a medical team, we focus on orthopedic surgeons performing knee surgeries. Knee surgeries are of particular interest because there is a specific procedure, called knee arthroscopy, that is now considered to provide minimal health advantages, which begs the question of why its use has not been discontinued [58]. Within this context we are interested in the temporal patterns of two sets of variables: the use of specific types of surgeries and the use of specific prosthetic devices, as to specific behavior indicators.

In this chapter we adapt a logistic regression model for the prediction of changes in behavioral patterns of surgeons to incorporate dynamic social network structures. The social network structures and behavioral patterns are extracted from health insurance patient-level data that contains information about the typs of surgery performed, the type of prosthetic devices used and the medical personnel involved in the surgery. Due to the granularity and uniqueness of the data, we have the resources to construct three social networks that describe connections among surgeones arising from: (i) practicing at the same hospitals (ii) sharing the same assistants (iii) sharing the same anaesthetists.

Our primary objective is to enable private and public healthcare organizations to better understand how behavioral trends may influence the delivery of healthcare services. These organizations can use this valuable information for planning preventive health management strategies to improve the effectiveness of care and patient health outcomes.

Key contributions of this chapter are as follows:

- We incorporate network structures into logistic regression prediction for our proposed approaches, Social Relationship Model (SRM) and its variant model Positive SRM (P-SRM).

- We conducted experiments to validate and evaluate the models on artificial data and a real data set obtained from the health insurance industry.

- We verify that the social network connections have influence on the surgeons' behaviors, where the behavior is the change in the workload distribution of knee surgeries and change in the use of certain prosthetic devices used in knee surgeries.

The structure of the chapter is as follows: In Section 7.2 we describe related work and some concepts in social learning theory. Then we formally define the general problem of dynamic node behaviours in Section 7.3. Sections 7.4 describe the mathematical formulation which incorporate network structures into logistic regression prediction and introduce our proposed models, SRM and P-SRM respectively. The subsequent sections, contain a discussion of the results from artificial experiments and our application to hospital data as well as a summary of our contributions.

7.2 Background and Related Work

In this work the actors of interest are surgeons. We assume that surgeons form a social network and communicate their experiences with their other network members. This exchange of information can alter professional attitudes of surgeons, resulting in behavioral changes such as the adoption of a medical intervention (e.g. new equipment, surgeries or medication). In this setting, knowledge is strongly tied to medical practices, and learning occurs when a group of surgeons collaborate in order to achieve a common goal, that is to improve patients' well being [62, 26, 66].

In the context of surgeons, we concentrate on how surgeons are influenced by the attitudes and behaviors of other colleagues as well as their hospital environment. The influence of a surgeon is related to their knowledge, where an individual's level of experience within a group defines whether they are a sender or receiver of information [50]. Thus a knowledge-rich surgeon potentially has a greater influence on the professional practices of their fellow surgeons. This becomes an imitative behavior for a less experienced surgeon where they adjust their behaviors accordingly [55].

Similar problems relating to the influence of peers on their behaviors have been addressed by Wei et al. [64], who developed a computational model to predict the adoption of smart phone apps by analysing social network connections among the users. Experimentally the model showed superior results than generic models, although this outcome may not necessarily hold in the context of surgeon analysis.

Fei et al. [83] also investigated the potential impact of network relationships on the quality of care provided by surgeons. The main result showed that patterns in potential network connections can influence quality of healthcare services. This research relied on finding correlations between network features and thus developing metrics for quality of care.

7.3 Problem Definition

We consider a network with N nodes, corresponding to N "actors", where the network relationship among nodes is encoded in several adjacency matrices $\mathbf{A} = \{\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_Q\}$ with $\mathbf{A}_q \in \mathbb{R}^{N\times N}$. There are C binary labels associated with each node, describing the behavioral status of the node. This is captured by the matrix $\mathbf{Y} \in \mathbb{R}^{N\times C}$ where the nth row (nodes) and cth column (behaviors) can be denoted as $y_{n,c} \in \{-1, 1\}$ or $y_{n,c} \in \{0, 1\}$. In our surgeons data set the N actors are surgeons, and the Q adjacency matrices represent different ways in which surgeons are related. For example, one matrix may represent the sharing of the same hospital(s) and another matrix represnt the sharing of same anaesthetists or assistants. The C labels associated with each surgeon will denote behaviors, such as the performance of certain types of surgery or the use of certain prosthetic devices.

The evolving and dynamic nature of behaviors is considered by examining two time periods, t_1 and t_2 ($t_2 > t_1$). The behavioral status of the nodes at t_1 and t_2 are denoted as $\mathbf{Y}^{(1)}, \mathbf{Y}^2 \in \mathbb{R}^{N\times C}$. We make the key assumption that the relationships among the nodes influence the behavioral status of each node over time: this is central to our study as our primary objective is to predict the node's behavioral status in the future period t_2 based on the available data at t_1.

The (i, j) element of an adjacency matrix is a positive number that quantifies the strength of a social relationship between actor i and actor j if a relationship exists, and it is 0 otherwise. The diagonals of these adjacency matrices are set to zeros. As mentioned above, more than one set of network relationships may be defined among the same set of actors. Suppose we have a set of $N = 5$ nodes, and among the nodes, there are $Q = 2$ network relationships as shown in Figure 7.1. Then we can summarize the network relationships among the nodes with the corresponding adjacency matrices given in Table 7.1.

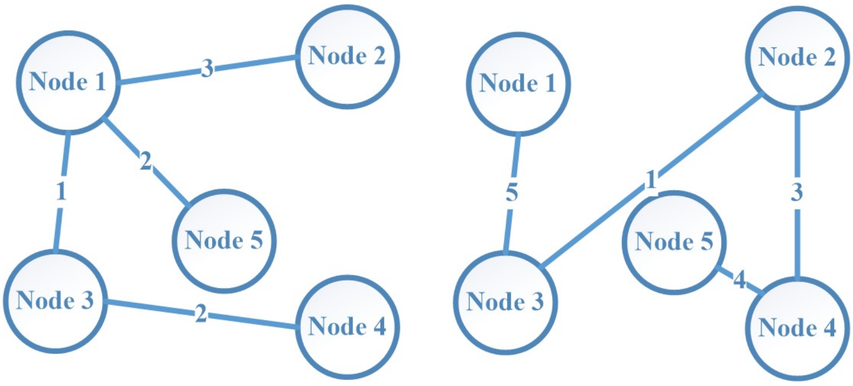


Figure 7.1: Networks defined in terms of two different relationships.

Table 7.1: Adjacency matrices based on networks in Figure 7.1.

	(a)					(b)				
	Node 1	Node 2	Node 3	Node 4	Node 5	Node 1	Node 2	Node 3	Node 4	Node 5
Node 1	0	3	1	0	2	0	0	5	0	0
Node 2	3	0	0	0	0	0	0	1	3	0
Node 3	1	0	0	2	0	5	1	0	0	0
Node 4	0	0	2	0	0	0	3	0	0	4
Node 5	2	0	0	0	0	0	0	0	4	0

7.4 Models

In sections 7.4.1 and 7.4.2 we introduce two models that we will compare to the standard majority vode model, briefly described in section 7.4.3.

7.4.1 Social Relationship Model (SRM)

Adjacency matrices are assumed to be symmetric because they capture the existence of a relationship among actors. However the influence of the behavior of actor i on the behavior of actor j is not necessarily symmetric, and it is important to build in the model the possibility that some actors are more influential than others (for example by being leaders in the adoption of new technologies). We formalize this by assuming that node n in a network q has an influential index $s_{q,n}$, where $\mathbf{S} \in \mathbb{R}^{Q \times N}$, representing the amount of influence of node n the other nodes in network q that are connected to it. In this model, we allow the influence to be both positive and negative.

The variables of interest are the C vectors $\mathbf{y}_c^{(2)}$ representing the behaviors of the nodes at time t_2, which are binary. Therefore we take a latent variable approach and assume that the behavior at time t_2 is driven by the latent variable N-vector $\hat{\mathbf{y}}_c^{(2)}$ defined below:

$$\hat{\mathbf{y}}_c^{(2)} = b_c \mathbf{1} + \sum_{q=1}^{Q} \left[\mathbf{A}_q \mathrm{diag}(\mathbf{s}_q)\right] \mathbf{y}_c^{(1)} \tag{7.1}$$

where b_c is a constant offset, $\mathbf{1}$ is an $N \times 1$ vector of ones, $\mathbf{s}_q$ is a column vector of $\mathbf{S}$, $\mathrm{diag}(\mathbf{s}_q)$ is the $N \times N$ matrix with $\mathbf{s}_q$ on the diagonal and zeros elsewhere. The latent variable model is then expressed as follows:

$$y_{n,c}^{(2)} = \begin{cases} 1 & \text{if } \hat{y}_{n,c}^{(2)} + \varepsilon_{n,c} > 0 \\ -1 & \text{otherwise} \end{cases} \tag{7.2}$$

where $\varepsilon_{n,c}$ are i.i.d. random variables with cumulative distribution $F(\cdot)$. For ease of computation we choose the distribution F to be the logistic function, and therefore arrive to the following model for the future behavior at node n:

$$P\big(y_{n,c}^{(2)} = 1 \big| \mathbf{A}, \mathbf{y}_c^{(1)}\big) = \frac{1}{1 + \exp\big(-\hat{y}_{n,c}^{(2)}\big)}$$

The unknows of the model are the vector of offsets $\mathbf{b} \equiv (b_1, \ldots, b_c)$ and the matrix $\mathbf{S}$ of influential indices. The negative log-likelihood $f(\mathbf{S}, \mathbf{b})$ for model 7.2 is easily derived as follows:

$$f(\mathbf{S},\mathbf{b}) = -\log \left[\prod_{c=1}^{C} \prod_{n:y_{n,c}^{(2)}=1} P\big(y_{n,c}^{(2)} = 1 \big| \mathbf{A}, \mathbf{y}_c^{(1)}\big) \right. \tag{7.3}$$

$$\left. \prod_{n:y_{n,c}^{(2)}=-1} P\big(y_{n,c}^{(2)} = -1 \big| \mathbf{A}, \mathbf{y}_c^{(1)}\big) \right] \tag{7.4}$$

$$= \sum_{c=1}^{C} \sum_{n=1}^{N} \log \Big(1 + \exp\big(-y_{n,c}^{(2)}\, \hat{y}_{n,c}^{(2)} \big) \Big) \tag{7.5}$$

We note that in our model the number of unknowns ($N \times Q + C$) is roughly equal to the number of samples ($N \times C$), and therefore a regularizing term is necessary in order to avoid overfitting [61, 82]. Since our application of interest will be the adoption or usage of certain technologies or procedure, we expect that only a relatively few number of surgeons will have a considerable effect on their peers. Therefore an ℓ_1 regularizing term that enforces sparsity seems better suited to this task than the standard ℓ_2 term. Hence we propose to estimate the unknowns of the model by minimizing the following cost function:

$$H(\mathbf{S},\mathbf{b}) = \sum_{c=1}^{C} \sum_{n=1}^{N} \log \Big(1 + \exp\big(-y_{n,c}^{(2)}\, \hat{y}_{n,c}^{(2)} \big) \Big) + \lambda \|\mathbf{S}\|_1$$

where λ is the regularization parameter that controls the sparsity of the matrix $\mathbf{S}$ and $\|\mathbf{S}\|_1$ is the sum of the absolute values of $\mathbf{S}$. Since the objective function $H(\mathbf{S}, \mathbf{b})$ is convex, global optimal solutions can be obtained. We implemented the model in Matlab using the popular convex solver CVX [37].

7.4.2 Positive Social Relationship Model (P-SRM)

In a number of situations it is sensible to assume that if two nodes are connected in a network then the influence that they have on each other can only be positive, that is it can only lead to a reinforcement of a behaviour. A similar hypothesis was made in the work by Wei et al. [64], where they studied the prediction of app adoption by smart phone users. In the context of our model this assumption is formalized by assuming that the behaviors are represented by binary values taking values in $\{0,1\}$, and by assuming that the unknown parameters **b** and **S** that enter the latent variable have all non-negative entries. We refer to this model as the Positive Social Relationship Model (P-SRM), and we will see shortly that it may have better performances than the unconstrained model in certain situations.

Since the latent variables are now all positive by constructions, model 7.2 needs to be revised so that the density $F(\cdot)$ is supported on the positive axis. We follow Wei et al. [64] in defining the conditional probability of a positive outcome as follows:

$$P\big(y_{n,c}^{(2)}=1\big|\mathbf{A},\mathbf{y}_c^{(1)}\big)=1-\exp\big(-\hat{y}_{n,c}^{(2)}\big)$$

The weighted negative log-likelihood associated to this model has the form:

$$f(\mathbf{S},\mathbf{b})=-\sum_{c=1}^{C}\sum_{n=1}^{N}W_{n,c}\log\Big(y_{n,c}^{(2)}+(1-2y_{n,c}^{(2)})\exp\big(-\hat{y}_{n,c}^{(2)}\big)\Big)$$

and therefore the parameters estimates for this model solve the following constrained optimization problem:

$$\begin{aligned}&(\hat{\mathbf{S}},\hat{\mathbf{b}})=\operatorname*{arg\,min}_{\mathbf{S},\mathbf{b}} f(\mathbf{S},\mathbf{b})+\lambda\|\mathbf{S}\|_1\\&\text{subject to: } \mathbf{S}\geq 0,\mathbf{b}\geq 0\end{aligned}$$

As with model 7.2 the optimization problem above is also convex and therefore global optimal solutions can be computed.

7.4.3 Baseline Model OMV

One of the simplest methods for network node prediction is the online majority vote (OMV) algorithm. When applied to our context the OMV algorithm predicts future behaviors at a node by taking a majority vote of the current behaviors of the neighbors of that node, weighted by the strength of the network relationship. Despite its simplicity this method proved to be highly effective and outperform other more complex methods on a variety of data sets [3]. In formulas the predicted behaviors have the following form:

$$\hat{\mathbf{y}}_c^{(2)} = \operatorname{sign}\left(\sum_{q=1}^{Q} \mathbf{A}_q \mathbf{y}_c^{(1)}\right)$$

In the OMV algorithm the influence of a node on another is fixed, and given by the elements of the adjacency matrix. Therefore this algorithm is unable to discover, based on the data, which nodes may have more influence in predicting future behaviors.

7.5 Artificial Data Experiments

In this section we compare the models described above (SRM, P-SRM and OMV) on two artificial data sets, generated according to the SRM and P-SRM models, respectively. The artificial data sets consist of three networks encoded by $Q = 3$ adjacency matrices. Each network has $N = 60$ nodes and each node is associated with $C = 10$ binary labels. In the log-likelihood all the observations are assumed to have the same weight. The data sets are generated as described below.

Artificial data: SRM

1. The adjacency matrices $\mathbf{A}_q$ are constructed by randomly deleting 95% of the edges of the fully connected graph.

2. The labels for the first period $\mathbf{Y}^{(1)}$, where $y_{n,c} \in \{-1, 1\}$ are sampled from a binomial distribution with balanced classes of labels.

3. The influential index $s_{q,n}$ of each node n in network q was generated according to a uniform distribution over an interval. For the training set the interval was set to $[-10, 10]$, while for the test set the interval was set to $[-\alpha, \alpha]$, with $\alpha \in \{0, 2, 4, 6, 8, 10\}$. Six different data sets were created, corresponding to the six values of α.

4. The dependent variable is determined using a simplified version of the SRM model 7.2, in which the noise is negligible. Therefore the dependent variables are simply obtained by taking the sign of the corresponding latent variables:

$$\hat{\mathbf{y}}_c^{(2)} = \operatorname{sign}\left(\sum_{q=1}^{Q} \left[\mathbf{A}_q \operatorname{diag}(\mathbf{s}_q)\right] \mathbf{y}_c^{(1)}\right)$$

Artificial data: P-SRM

1. The adjacency matrices $\mathbf{A}_q$ are constructed by randomly deleting 95% of the edges of the fully connected graph.

2. The labels for the first period $\mathbf{Y}^{(1)}$, where $y_{n,c} \in \{0,1\}$ are sampled from a binomial distribution with balanced classes of labels.

3. The influential index $s_{q,n}$ for n in network q was generated using the same procedure as for the SRM data, except that the interval was restricted to the positive numbers.

4. As with the SRM data the values of the dependent variables were assigned by a simple thresholding of the latent variable, where the threshold has been set to 0.5 to account for the positivity constraint:

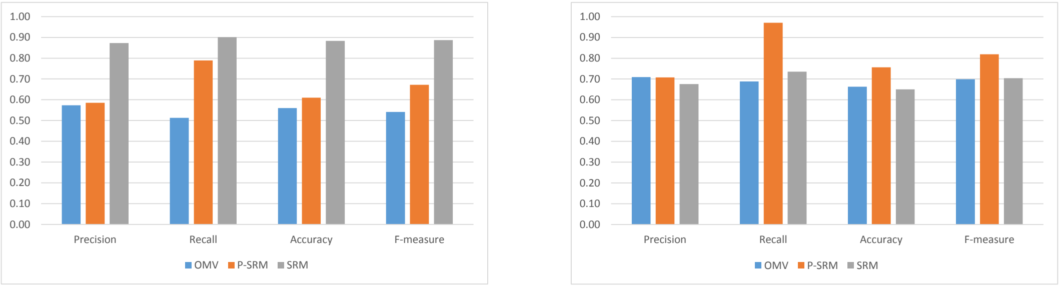
$$\hat{\mathbf{y}}_c^{(2)} = \text{sign}\left(\sum_{q=1}^{Q} \left[\mathbf{A}_q \text{diag}(\mathbf{s}_q)\right] \mathbf{y}_c^{(1)} - 0.5\right)$$

7.5.1 Performance tests

We compare the three models OMV, SRM and P-SRM on the two artificial data sets described above. The data sets are split into training and test sets with equal sample sizes and the performances of the algorithms are evaluated by the followig standard four metrics: precision, recall, accuracy and F-measure. The regularization parameter λ is set to 0.5 for all experiments, since we found that changing its value and attempting to optimize it did not affect performances to great extent.

Note that we expect the models to perform best when the training data consists of surgeons who have a significant influential index compared to the surgeons in the test set, or, alternatively, if surgeons in the test set have small influential index. Therefore in the generation of the baseline data sets we set the influential indices of test samples to zero. In the sensitivity analysis of section 7.5.2 we will then allow the influential indices in the test set to grow, by varying the parameter α introduced in the description of the data generation above. This allows to study how performances degarde as the data become increasingly difficult to predict.

In Figure 7.2 (a) and (b) we report the performance measures for the SRM and P-SRM data sets respectively. We observe that the SRM model and the P-SRM model outperform the other models on the SRM and P-SRM data sets, respectively. This result is expected and it is important for two reasons: 1) it is a sanity check that confirms that the models are internally consistent, and 2) it gives us an idea of how much better the SRM and P-SRM can perform, compared to OMV.



(a) SRM

(b) P-SRM

Figure 7.2: The four performance measures for the the three algorithms on the SRM data (a) and the P-SRM data (b). As expected the SRM model outperforms both the OMV and P-SRM when applied to the SRM data and the P-SRM outperforms OMV and the SRM when applied to the P-SRM data.

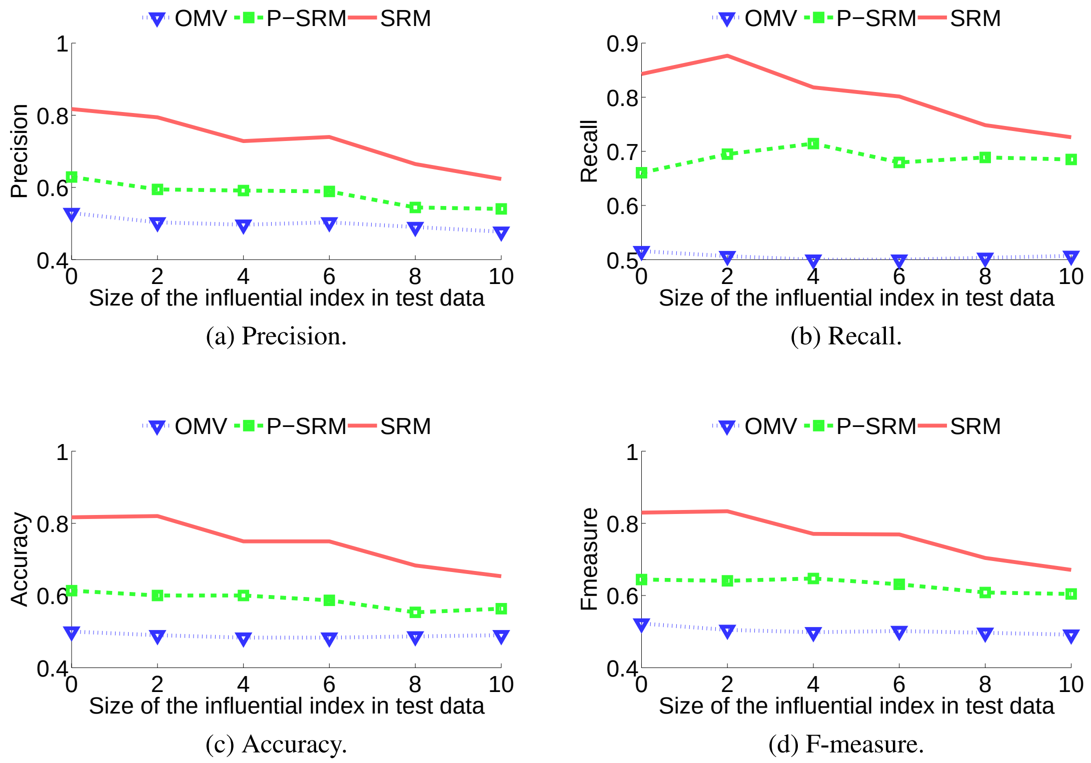
7.5.2 Sensitivity to influential index

In the baseline data of figure 7.2 the influential index for the surgeons in the test data was set to 0. This makes the problem easier, because it implies that their future behavior depends mostly on the behavior of the surgeons in the training set. Varying the scaling factor α from 0 to 1 we obtain data sets that are increasingly more difficult to predict, and it is therefore interesting to study how the performances of the three different method degrade.

The results of this experiment on the SRM data are shown in Figure 7.3. The key message emerging from this figure is that even if the performance of the SRM and P-SRM algorithm degrade as the complexity of problem increases they perform much bettter than OMV even in the hardest case.

7.5.3 Sensitivity to network sparsity

The sparsity of the network is defined as the proportion of edges that are not connected. In the baseline data the three networks have a sparsity equal to 95%. Since different



(a) Precision.

(b) Recall.

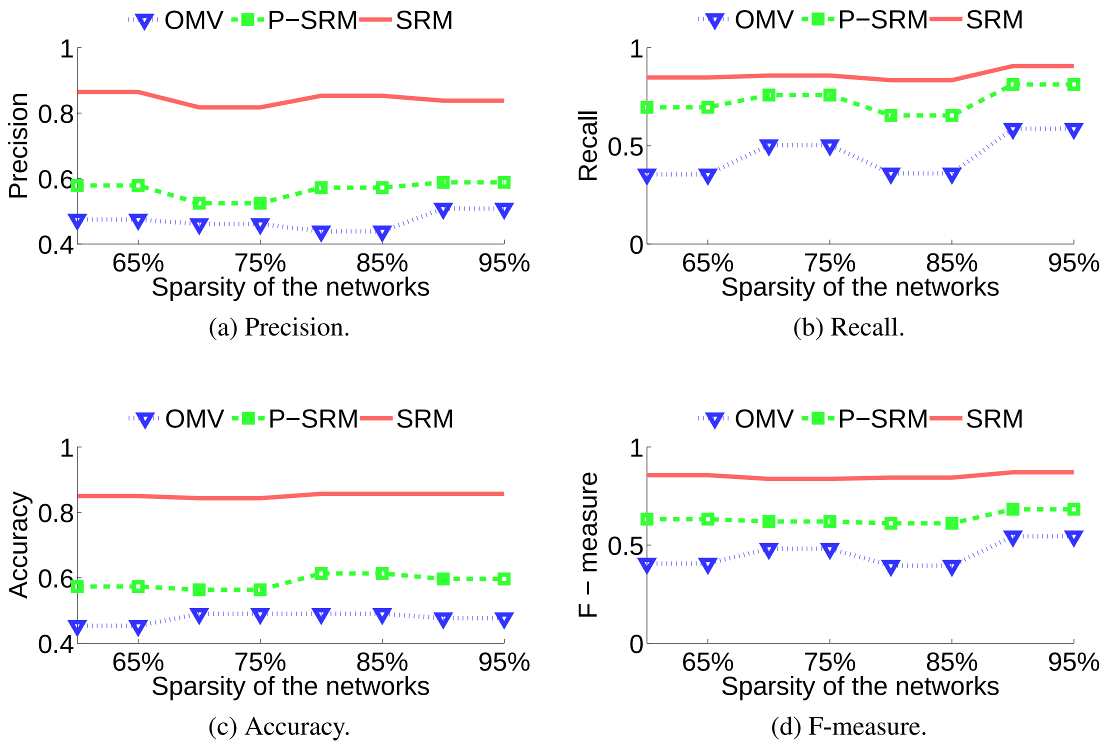
(c) Accuracy.

(d) F-measure.

Figure 7.3: As the influential index *S* of test samples increase, the performance of SRM and P-SRM decreases, however SRM and P-SRM still outperforms the baseline method OMV.

applications may have networks with different sparsity it is important to understand whether the performances of the algorithms change with the sparsity of the network. Intuitively the OMV algorithm is expected to perform better at a high sparsity level, with very localized interaction. Since the SRM and P-SRM algorithms learn networks effects from the data one would expect them to be reasonably insensitive to the sparsity level.

We repeated the experiments for levels of sparsity varying from 55% to 95%, in steps of 5%, and report the results in Figure 7.4. The key message of the figure is that the performances of the SRM algorithms are unaffected by the sparsity level. The P-SRM algorithm is somewhat more sensitive to the overall sparsity level, and tend to perform slightly better at higher sparsity. This is also true for the OMV model, which is the most sensitive of the three.



(a) Precision.

(b) Recall.

(c) Accuracy.

(d) F-measure.

Figure 7.4: SRM has the best performance, while OMV has the least scores when varying values of sparsity of networks. We also notice that SRM has the most stable and robust performance.

7.6 Hospital Data Set

7.6.1 Data Preparation

In order to study the problem in a reasonable granularity, we examines knee procedures as the exemplar treatment group, utilizing data from the health insurance industry. In the following sub-sections, we describe how the hospital data was prepared before applying our models.

7.6.1.1 Surgeon networks

We are interested in studying surgeon behavior. In particular we wish to predict the behavior of a surgeon at time t_2 given the behavior of his/her peers at time t_1. There are different ways in which surgeons can affect each other's behaviors, and each corresponds to a separate network. The simplest form of interaction between two surgeons arises from the fact that they practice at the same hospital. Therefore we construct a network that has one node for each surgeon in the data and one edge for each pair of surgeons practicing at the same hospitals. More precisely the element (i,j) of the adjacency matrix $\mathbf{A}^h$ is equal to the number of hospitals that surgeons i and j have in common.

However it is also possible that surgeons influence each other via the people who work closely with them. Our data of patient surgeries contains unique identification for surgeons, assistants and anaesthetists. Therefore we define other two networks, represented by the adjacency matrices $\mathbf{A}^{an}$ and $\mathbf{A}^{as}$, such that element (i,j) of these matrices is equal to the number of anaesthetists and assistants that surgeons i and j have previously worked with, respectively. The details of the adjacency matrices are summarized in Table 7.2.

Table 7.2: The three networks used in the hospital data experiments.

Notation	**Description**	**Nodes**	**Edge Weight**
$\mathbf{A}^h \in \mathbb{R}^{N\times N}$	Adjacency matrix of surgeons working in the same hospital	Surgeons	Number of common hospitals
$\mathbf{A}^{an} \in \mathbb{R}^{N\times N}$	Adjacency matrix of surgeons working with the same anaesthetists	Surgeons	Number of common anaesthetists
$\mathbf{A}^{as} \in \mathbb{R}^{N\times N}$	Adjacency matrix of surgeons working with the same assistants	Surgeons	Number of common assistants

7.6.1.2 Surgical procedures

The number and composition of the procedures performed by surgeons fluctuates over time in a non-random way. Surgeons data show clear trends where some procedures gets dropped while others are adopted. We hypothesize that one of the reasons behind these trends, other than demand fluctuations, is the fact that surgeons may influence each other in a number of ways. It is particularly interesting to study how trends propagate, and a natural question to ask is whether changes in behavior this year predict changes in behaviors next year.

We formalize this notion by introducing a set of binary variables $\mathbf{Y}^{(1)}, \mathbf{Y}^{(2)} \in \mathbb{R}^{N\times C}$, where N is the number of surgeons in the data, C is the number of surgical procedures we consider and the superscripts refer to period 1 and 2. When $y^t_{n,c}$ is equal to 1 it means that during period t ($t = 1,2$) surgeon n has increased his/her activity on procedure c.

In order to determine the value of $y^t_{n,c}$ for each surgeon and each procedure we proceed as follows. We divide period t (one year) in an even number L time windows, and for each window α we compute the number v_α of procedures of type c performed by surgeon n (we dropped the indices n and c from v_α in order to ease notation). The vector $\mathbf{v}$ with elements v_α captures the trend of usage of procedure over time. In order to convert it to a binary variable we apply a linear filter to it. The linear filter allocates higher weight closer to the endpoints and subtracts the second half of the vector from the first. The precise form of filtering is described in Algorithm 5 below. In our experiments we choose $L = 4$, so that there were 4 windows of three months each.

Algorithm 5 Surgeon Binary Labeling

```
Input: $\mathbf{v} \in \mathbb{R}^L$: Number of specific behaviors over $L$ continuous time windows.
Output: Labels $y \in \{-1, 1\}$ or $y \in \{0, 1\}$.

F = $\sum_{i=1}^{\frac{L}{2}} (2i-1)(v_{L-i+1} - v_i)$
if $F > 0$ then
    $y = 1$
else
    $y = -1$ for SRM, $y = 0$ for P-SRM,
end if
Return $y$
```

In our hospital data set we have identified 20 procedures related to knee surgery, and therefore $C = 20$. In Australian hospital data each type of surgery procedure is assigned a unique Medicare Benefit Schedule (MBS) code [21]. The distribution of the MBS codes is shown in Figure 7.5 and the MBS code descriptions are provided in Table 7.3. We observe there are three MBS codes performed much more frequently than the other procedures.

7.6.1.3 Prosthetic devices

In addition to surgical procedures we are also interested in investigating the behavioral change of surgeons related to the use of prosthetic devices. In knee procedures, surgeons can be influenced by other surgeons by either adopting or dropping a particular device. The hospital data allows to identify the specific prosthetic device used by each surgeon, and therefore the prediction problem is exactly as the one described in the previous section, with MBS codes replaced by codes for prosthetic devices. Thus, the algorithm to extract the binary labels for prosthetic devices is the same as in Algorithm 5.

The distribution of prosthetic devices is depicted in Figure 7.6 and a sample of the devices is shown in Table 7.4. Unlike with surgical procedure, where few MBS

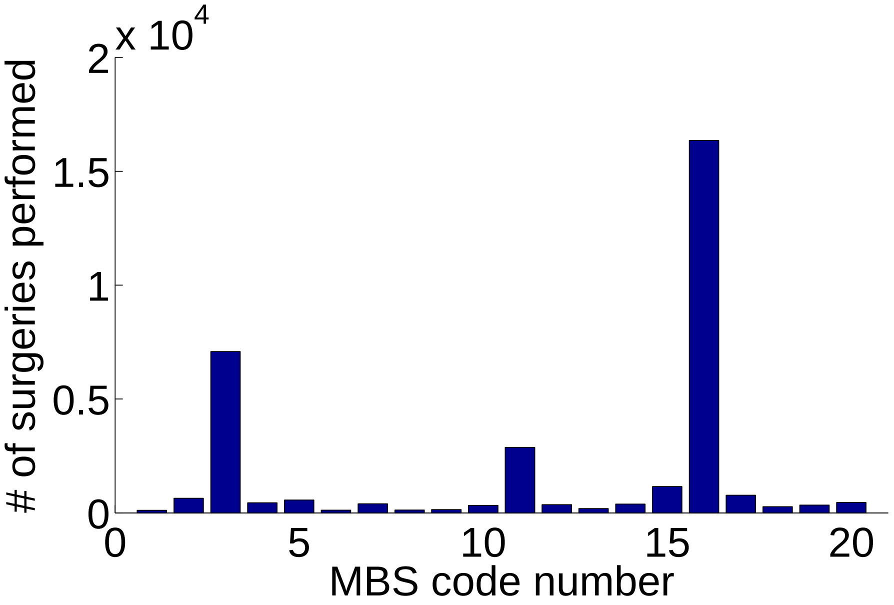


Figure 7.5: Distribution of surgical procedures performed in the hospital data.

codes dominate the distribution, it appears that surgeons use prosthetic devices in similar amounts, with no particular preference.

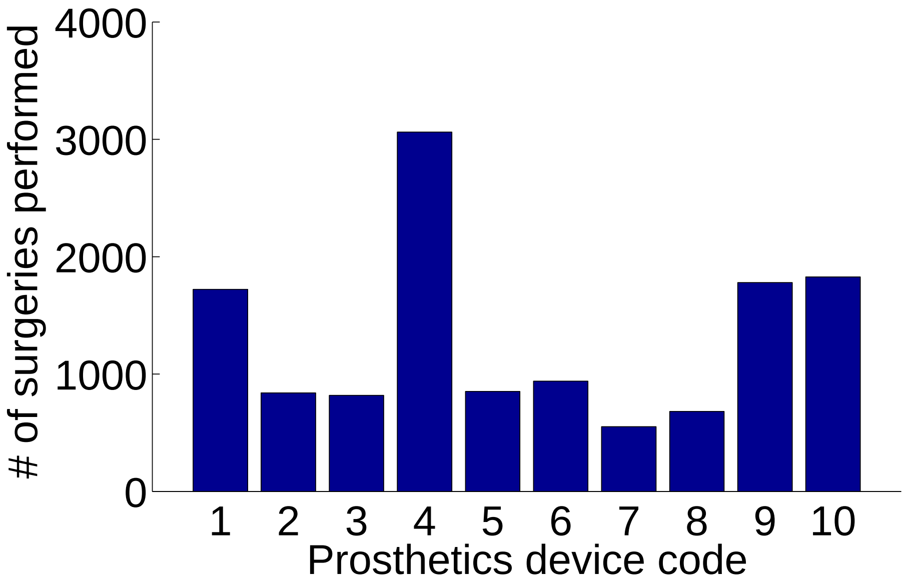


Figure 7.6: Distribution of prosthetic devices in the hospital data.

7.6.1.4 Observation weighting

Over the course of our experiments we found useful to assign weights to each observation appearing in the likelihood of equation 7.5, in order to account for the fact that there is a wide variation in the number of procedures performed by surgeons. Therefore for each surgeon n and procedure c we compute the total number $w_{n,c}$ of procedures performed. Rather than setting weights equal to $w_{n,c}$, that would allow few very active surgeons to dominate the likelihood we opted for a logarithmic weight of the form $\log(1+w_{n,c})$, which performed better than linear weighting in our experiments.

7.6.2 Experimental Results

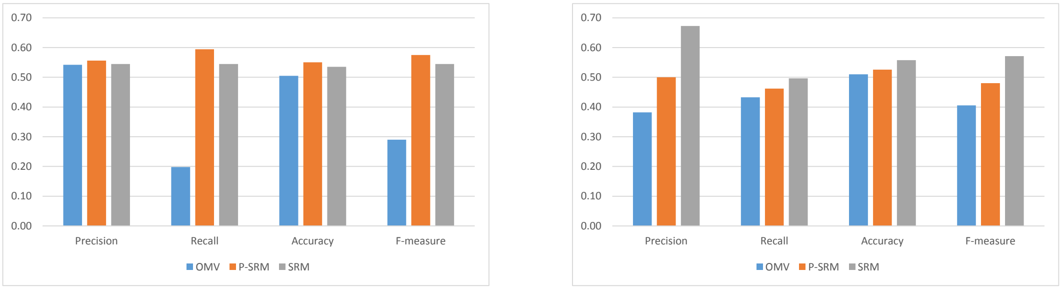
In this section we compare the three models, SRM, P-SRM and OMV on the hospital data. We consider two types of dependent variables: one is the behavior of surgeons regarding the performance of specific surgeries, identified by 20 different MBS codes, and the other is the behavior regarding the implant of specific prosthetic devices, identified by 10 different prosthetic codes.

7.6.2.1 Surgical procedures

In our hospital data we selected surgeons with at least 100 surgeries performed over a two year period, obtaining a sample size of 121 surgeons. The training and test sets were randomly selected using equal sample sizes. In order to avoid random variation due to choice of training and test data set each experiment was repeated 10 times over different training and test data, and the performance measures were averaged of the 10 results.

We calculate performance results in the test data for each MBS code separately as well as in the aggregate. Since the MBS code specific results are well reflected in the aggregate results, for ease of exposition we only report the aggregate results, which are shown in Figure 7.7(a).

What stands out from Figure 7.7(a) is that SRM and P-SRM have similar performances, with P-SRM having a small advantage over SRM. Both models seem to have a major advantage on the OMV model when it comes to recall. Therefore the two models outperform OMV by a large margin in the task of correctly identifying surgeons who will increase the use of specific procedures in the next period.



(a) MBS codes

(b) Prosthesis

Figure 7.7: The four performance measures for the the three algorithms on the MBS codes (a) and the prosthesis data (b).

An attractive feature of the SRM and P-SRM models is that it provides, for each surgeon, an estimate of the associated influential index. Therefore these models allow to identify, for example, who are the leaders of innovation in a surgeon network. We hypothesize that in our hospital data only a small number of surgeons would be "leaders". This hypothesis is easily tested by looking at the distribution of the values of the three groups of influential indices (one for each network) estimated by the SRM model, that we report in Figure 7.8. A close inspection of the figure would reveal that only 13% of the influential indices are significantly different from 0, and only few surgeons have influential indices of the order of three or four. The figure also shows that the vast majority of influential indices are positive. This may explain why the P-SRM model performs better than the SRM model, since it makes the positivity assumption from the beginning, and therefore the algorithm does not need to "discover" it from the data.

Studying the estimates of the influential indices can also provide other useful insight in the data. For example, we observe that the three set of influential indices, one for each type of network, are quite uncorrelated, with correlation coefficients ranging from 0.07 to 0.23. This finding support the notion that SRM and P-SRM models truly capture network effects, and not other features of the data, such as for example hospital effects. In fact, if that were the case, we would expect the influential indices to be quite similar and highly correlated, which is certainly not the case.

We further analyzed three distinct networks to compute the social influence of surgeons. The three networks analyzed are based on: (a) connections among surgeons who

Table 7.3: MBS code description

Code number	MBS code	MBS code description	# of surgeries
1	49509	Knee, total synovectomy or arthrodesis with ...	109
2	49517	Knee, hemiarthroplasty of (Anaes.) (Assist.)	643
3	49518	Knee, total replacement arthroplasty of...	7084
4	49519	Knee, total replacement arthroplasty of...	442
5	49521	Knee, total replacement arthroplasty of...	564
6	49524	Knee, total replacement arthroplasty of...	116
7	49527	Knee, total replacement arthroplasty of...	396
8	49533	Knee, total replacement arthroplasty of...	123
9	49536	Knee, repair or reconstruction of, for...	146
10	49539	Knee, reconstructive surgery of cruciate...	325
11	49542	Knee, reconstructive surgery of cruciate ...	2876
12	49551	Knee, revision of procedures to which item...	357
13	49557	Knee, diagnostic arthroscopy of ...	189
14	49558	Knee, arthroscopic surgery of, involving...	383
15	49560	Knee, arthroscopic surgery of, involving...	1153
16	49561	Knee, arthroscopic surgery of, involving...	16353
17	49562	Knee, arthroscopic surgery of, involving...	771
18	49563	knee, arthroscopic surgery of, involving...	270
19	49564	Knee, patello-femoral stabilisation of...	338
20	49566	Knee, arthroscopic total synovectomy of...	457

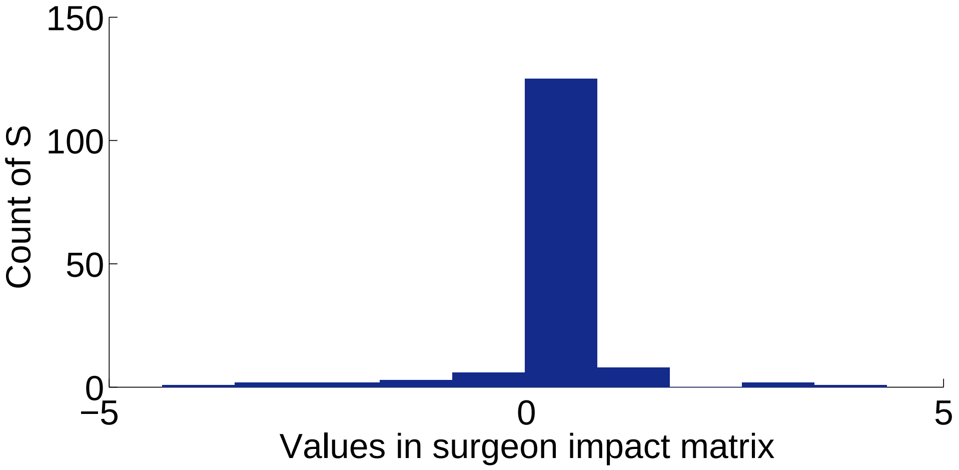


Figure 7.8: As expected the influential indices estimated by the SRM model show that the surgeon network contains a very small number of surgeon "leaders", with high influential indices. The overall sparsity of the indices is 13%, meaning that only 13% of the influential indices are significantly different from zero.

work at the same hospital (b) connections among surgeons who work with the same anaesthetists, and (c) connections among surgeons who work with the same assistant surgeons.

Figure 7.9 shows the comparative performance of most influential surgeons in the three networks, in regard to their adoption of certain types of knee procedures. A distinct influencer is evident in the anaesthetist-sharing network while performing CMBS

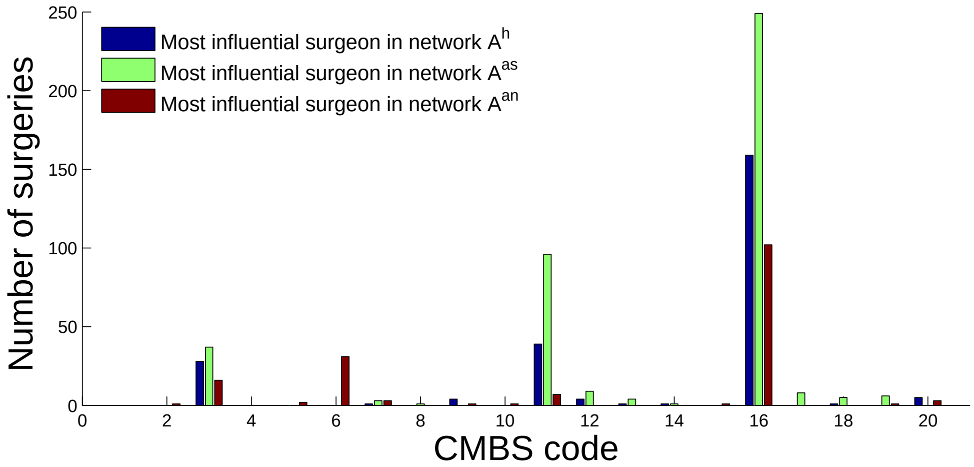


Figure 7.9: Comparative workload distribution of most influential surgeons in the three networks.

code 6, which indicates a specific type of knee replacement requiring bone grafting.

It also indicates that the anaesthetist network appears to have a higher influence in the surgeon behaviour in regard to this specific procedure, which is more complex as this requires bone grafting, and more anaesthetists time while performing knee replacement surgeries.

7.6.2.2 Prosthetic devices

Another variable of interest is whether surgeons increase or decrease the number of implants of specific prosthetic devices in the next period. The experimental results on the prosthetic devices data set shown in Figure 7.7(b). Similarly to the results for surgical procedure we find that both the SRM and P-SRM model outperform the OMV model. However, on this data set it is the SRM that outperforms P-SRM, which is the opposite of what we found on the surgical procedure data. We also notice that while in the surgical data most of the advantage over OVM was observed in recall, in this case most of the advantage is manifested in the precision performance measure. Therefore the performance measure on which we observe the highest difference between SRM, or P-SRM, and OVM appears to be a property of the data, rather than the methods themselves.

Table 7.4: Prosthesis code description

Code number	Prosthesis code	Prosthesis code description	# of surgeries
1	BX246	Infusor; Sterile infusor devices, spring or...	1081
2	DP107	CMW Bone Cement; CMW Bone Cement Various without Antibiotic	763
3	DP152	PFC Sigma Knee System patella component; Cemented, all ...	868
4	HK006	Palacos or Palamed Bone Cement with Gentamicin; Single Mix ...	3582
5	SK325	Triathlon Knee System Femoral Component ; Minimally ...	966
6	SK327	Triathlon Knee System Tibial Baseplate; Tibial ...	1051
7	SK419	Triathlon Knee System Patella Component; Patella ...	637
8	SN464	Bone Staple; Fixation - Short Leg	784
9	SN853	Endobutton; Fixation for ACL or PCL reconstruction	1708
10	SN857	Genesis II Knee System Tibial Baseplate; Tibial Baseplate...	1854

7.6.2.3 Comparing results

The experimental results on MBS codes are similar to the results for prosthetic devices in the sense that they both perform better than OMV. A key difference, however, is the performance of the P-SRM compared to SRM on the two data sets. Specifically, P-SRM outperforms SRM on surgical procedures, while SRM outperforms P-SRM on prosthetic devices.

The reason for this difference may lie with the fact that in terms of MBS codes, surgeons are more likely to assert a positive influence on other surgeons. In other words, a surgeon adopting a new procedure will probably cause the surgeons connected to him to do the same. Howecer, a surgeon dropping a procedure will not have any influence on the surgeons connected to him, and the P-SRM model is better suited to fit this assumption. However, in the case of prosthetic devices it is reasonable that surgeons can both positively and negatively influence the connected surgeons, and therefore this would make the SRM preferable.

7.7 Summary

In this chapter we have proposed a Social Relationship Model (SRM) to predict how a surgeon's choice of treatment is influenced by their peer networks. In a surgeon network, the nodes are surgeons and there is an edge between two nodes if they have operated in the same hospital or have worked with a common anaesthetist or assistant. SRM consists of an extension of the logistic regression model to incorporate network features. A unique contribution of this work is the application of the proposed model on an extremely fine-grained data set acquired from a health insurance company about the eco-system surrounding knee surgeons. SRM can be used to quantify the influence of a surgeon on their peers over time. While it is well known that peer interaction plays an important role in diffusion of knowledge and behavioral choices in a healthcare environment, our approach provides the first quantitative tool to actually measure the impact of social learning. SRM can be used by both practicing healthcare professional and management to shape the treatment environment in an organization and manage both the quality and cost of healthcare.

Chapter 8

Conclusions and Future Work

In this chapter we present a summary of the thesis, and highlight future research directions that could be extended from the research of the thesis.

8.1 Conclusions of the Thesis

- Adversarial learning is the study of machine learning techniques deployed in non-benign environments. Till now, the standard assumption about modeling adversarial behavior has been to empower an adversary to change *all* features of the classifiers at will. However, we claimed the aim of an adversary is not just to subvert a classifier but carry out data transformation in a way such that spam continues to appear like spam to the user as much as possible. In Chapter 3 we demonstrated that an adversary achieves this objective by carrying out a sparse feature attack. We designed an algorithm to show how a classifier should be designed to be robust against sparse adversarial attacks. We showed that sparse feature attacks are best defended by designing classifiers which use ℓ_1 regularizers.

- Chapter 4, We use mathematical properties of the two regularization methods, ℓ_1 (Lasso) and ℓ_2 (Tikhonov or Ridge), followed by detailed experimentation to understand their impact based on four characteristics: non-stationarity of the data generating process; level of noise in the data sensing mechanism; degree of correlation between dependent and independent variables and the shape of the data set. Thus, by considering the four characteristics, we developed a guide for practitioners of large scale data mining and machine learning tools in their day-to-day practice.

- In Chapter 5, we claim that *LPP* is the fundamental problem in outlier detection and algorithmic approaches to solve *LPP* are urgently needed. Matrix factorization methods provide a balanced compromise between full subspace exploration in the feature space versus exploration in the meta-feature or latent space. Results showed that our proposed model R-NMF is substantially more robust compared to NMF in the presence of data noise. This opens up a promising avenue for further exploration and address the *LPP*.

- Data analytic techniques such as data mining and predictive modelling are being used to gain new insights into health care costs, In chapter 6 we described a specific context of private healthcare in Australia and describe our SNA based approach (applied to health insurance claims) to understand the nature of collaboration among doctors treating hospital inpatients and explore the impact of collaboration on cost and quality of care. In particular, we use network analysis to (a) design collaboration models among surgeons, anaesthetists and assistants who work together while treating patients admitted for specific types of treatments (b) identify and extract specific types of network topologies that indicate the way doctors collaborate while treating patients and (c) analyse the impact of these topologies on cost and quality of care provided to those patients.

- In Chapter 7 we developed models that predict the behaviors of orthopedic surgeons in regard to surgery type and use of prosthetic device. The models utilize data on past practicing behaviours and take in account the social relationships existing among surgeons, anaesthetists and assistants. We refer to the models as the Social Relationship Model (SRM) and Positive Social Relationship Model (P-SRM). An important feature of these models is that they can not only predict the behaviors of surgeons but they can also provide an explanation for the predictions. Experimental results on both artificial and real hospital data sets show that our proposed models outperform the baseline model Online Majority Vote (OMV).

8.2 Recommendations for Future Work

We have presented a number of robust learning models in this thesis. We also developed models mining adversarial behaviors on healthcare data. Here we also list potential future research directions.

- In Chapter 3, we developed robust classification algorithms by assuming the adversary is carrying out sparse feature attacks. Similarly adversary can exist in anomaly detection and clustering problems. The need for a robust learning method under such problems is in need to be devised.

- For healthcare domain, in Chapter 7, we have proposed models that predict the potential adversarial behavior of surgeons. However, more modelling technique can be carried out based on entities such as hospitals, patients or even administrators working in the hospital. The reward for such adversarial behavior identification will not only reduce cost for insurance company, but also significantly increase the quality of care and even help improve the overall healthcare system.

Bibliography

[1] Theodore B Achacoso and William S Yamamoto. *AY's Neuroanatomy of C. elegans for Computation*. CRC Press, 1991. 1

[2] Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM, 2005. 1

[3] Omar Ali, Giovanni Zappella, Tijl De Bie, and Nello Cristianini. An empirical comparison of label prediction algorithms on automatically inferred networks. In *ICPRAM (2)*, pages 259–268, 2012. 7.4.3

[4] C.D. Aliprantis and S.K. Chakrabarti. *Games and decision making*. Oxford University Press Oxford, 2000. 3.3.3

[5] Alexis Arbuthnott and Donald Sharpe. The effect of physician–patient collaboration on patient adherence in non-psychiatric medicine. *Patient education and counseling*, 77(1):60–67, 2009. 2.1.2

[6] AUDUSD. *Forex Historical Data*, 2013 (accessed March 20, 2013). http://www.forexrate.co.uk/forexhistoricaldata.php. 4.6.1

[7] Albert Bandura. Psychotherapy based upon modeling principles. *Handbook of psychotherapy and behavior change*, 653:708, 1971. 1

[8] Albert Bandura. *Social foundations of thought and action : a social cognitive theory*. Prentice-Hall., 1986. 2.1.2

[9] Albert Bandura and David C McClelland. *Social learning theory*. Prentice-Hall Englewood Cliffs, NJ, 1977. 2.1.2

[10] Michael L Barnett, Nicholas A Christakis, A James OMalley, Jukka-Pekka Onnela, Nancy L Keating, and Bruce E Landon. Physician patient-sharing networks and the cost and intensity of care in us hospitals. *Medical care*, 50(2):152–160, 2012. 1

[11] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley and Sons, 1994. 2.1.1

[12] Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002. 4.5.1

[13] M.M. Breunig, H.P. Kriegel, R.T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In *IEEE International Conference on Data Mining (ICDM)*, 2000. 2.1.1

[14] M. Brückner, C. Kanzow, and T. Scheffer. Static prediction games for adversarial learning problems. 2011. 3.3.1.2

[15] M. Brückner and T. Scheffer. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 547–555. ACM, 2011. 2.1.1, 3.2, 3.1, 3.3.1.2

[16] Peter Buhlmann and Sara van de Geer. *Statistics for High-Dimensional Data*. Springer, 2011. 2.1.1

[17] Lawton Robert Burns and Ralph W Muller. Hospital-physician collaboration: Landscape of economic integration and impact on clinical integration. *Milbank Quarterly*, 86(3):375–434, 2008. 2.1.2

[18] Peter J Carrington, John Scott, and Stanley Wasserman. *Models and methods in social network analysis*, volume 28. Cambridge University Press, 2005. 2.1.2

[19] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), 2009. 2.1.1

[20] Sanjay Chawla and Aris Gionis. kmeans–: A unified approach to clustering and outlier detection. In *SIAM International Conference on Data Mining (SDM SIAM)*, 2013. 5.1, 5.1, 5.4.2

[21] cmbs. MBS online. http://www.mbsonline.gov.au/, 2014. [Online; accessed 8-April-2014]. 6.2, 7.6.1.2

[22] Marie J Cowan, Martin Shapiro, Ron D Hays, Abdelmonem Afifi, Sondra Vazirani, Cathy Rodgers Ward, and Susan L Ettner. The effect of a multidisciplinary hospitalist/physician and advanced practice nurse collaboration on hospital costs. *Journal of Nursing Administration*, 36(2):79–85, 2006. 2.1.2

[23] Frances C Cunningham, Geetha Ranmuthugala, Jennifer Plumb, Andrew Georgiou, Johanna I Westbrook, and Jeffrey Braithwaite. Health professional networks as a vector for improving healthcare quality and safety: a systematic review. *BMJ quality & safety*, 21(3):239–249, 2012. 2.1.2

[24] Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108, New York, NY, USA, 2004. ACM. 2.1.1

[25] Timothy de Vries, Sanjay Chawla, and Michael E. Houle. Density-preserving projections for large-scale local anomaly detection. *Knowledge and Information Systems (KAIS)*, 32(1):25–52, 2012. 5.3

[26] Nathan Eagle and Alex Pentland. Reality mining: sensing complex social systems. *Personal and ubiquitous computing*, 10(4):255–268, 2006. 7.2

[27] David Easley and Jon Kleinberg. Networks, crowds, and markets: Reasoning about a highly connected world, 2012. 1

[28] Giovanni Fattore, Francesca Frosini, Domenico Salvatore, and Valeria Tozzi. Social network analysis in primary care: The impact of interactions on prescribing behaviour. *Health Policy*, 92(2):141–148, 2009. 2.1.2

[29] A. Frank and A. Asuncion. UCI machine learning repository, 2010. 3.3.1.3, 4.6.1, 5.5.1, 5.5.2

[30] Linton C Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1979. 2.2.4

[31] Linton C Freeman, Douglas Roeder, and Robert R Mulholland. Centrality in social networks: Ii. experimental results. *Social networks*, 2(2):119–141, 1980. 2.2.4

[32] Isabelle Gaboury, Mathieu Bujold, Heather Boon, and David Moher. Interprofessional collaboration within canadian integrative healthcare clinics: Key components. *Social Science & Medicine*, 69(5):707–715, 2009. 2.1.2

[33] A. Genkin, D.D. Lewis, and D. Madigan. Large-scale bayesian logistic regression for text categorization. *Technometrics*, 49(3):291–304, 2007. 2.2.2

[34] L.E. Ghaoui, V. Viallon, and T. Rabbani. Safe feature elimination for the lasso and sparse supervised learning problems. *arXiv preprint arXiv:1009.4219*, 2010. 2.2.3, 4.4

[35] Amir Globerson and Sam Roweis. Nightmare at test time: robust learning by feature deletion. In *Proceedings of the 23rd international conference on Machine learning*, pages 353–360, New York, NY, USA, 2006. ACM. 2.1.1

[36] Geoff Gordon and Ryan Tibshirani. Duality correspondences. *Optimization*, 10(725/36):725, 2013. 4.5.1

[37] Michael Grant, Stephen Boyd, and Yinyu Ye. Cvx: Matlab software for disciplined convex programming, 2008. 3.3, 7.4.1

[38] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009. 3.3.1.1

[39] D. M. Hawkins, L. Liu, and S. S. Young. Robust singular value decomposition. *Technical Report, National Institute of Statistical Sciences*, 2001. 2.1.1

[40] Jian Hu, Hua-Jun Zeng, Hua Li, Cheng Niu, and Zheng Chen. Demographic prediction based on user's browsing behavior. In *Proceedings of the 16th international conference on World Wide Web*, pages 151–160. ACM, 2007. 4.1

[41] M. Hubert, Pj Rousseeuw, and Vanden. ROBPCA: a new approach to robust principal component analysis. *Technometrics*, 47:64–79, 2005. (document), 5.2

[42] Jacquelyn S Hunt, Joseph Siemienczuk, Ginger Pape, Yelena Rozenfeld, John MacKay, Benjamin H LeBlanc, and Daniel Touchette. A randomized controlled trial of team-based care: impact of physician-pharmacist collaboration on uncontrolled hypertension. *Journal of general internal medicine*, 23(12):1966–1972, 2008. 2.1.2

[43] William A Knaus, ELIZABETH A DRAPER, Douglas P Wagner, and Jack E Zimmerman. An evaluation of outcome from intensive care in major medical centers. *Annals of Internal Medicine*, 104(3):410–418, 1986. 2.1.2

[44] David Knoke and Song Yang. *Social Network Analysis*, volume 154 of *Quantitative Applications in the Social Sciences*. SAGE Publications, 2008. 2.1.2

[45] E.M. Knorr and R.T. Ng. Algorithms for mining distance-based outliers in large datasets. In *International Conference on Very Large Data Bases (VLDB)*, 1998. 2.1.1

[46] H.-P. Kriegel, P. Krogel, E. Schubert, and A. Zimek. A general framework for increasing the robustness of pca-based correlation clustering algorithm. In *Scientific and Statistical Database Management Conference (SSDBM)*, 2008. 2.1.1

[47] Thomas Kuhn. *The Structure of Scientific Revolutions*. University of Chicago, 1962. 1, 5.1

[48] Bruce E Landon, Nancy L Keating, Michael L Barnett, Jukka-Pekka Onnela, Sudeshna Paul, A James OMalley, Thomas Keegan, and Nicholas A Christakis. Variation in patient-sharing networks of physicians across the united states. *JAMA*, 308(3):265–273, 2012. 1

[49] John Langford, Lihong Li, and Tong Zhang. Sparse online learning via truncated gradient. *The Journal of Machine Learning Research*, 10:777–801, 2009. 4.6

[50] Lihui Lin, Xianjun Geng, and Andrew B Whinston. A sender-receiver framework for knowledge transfer. *MIS quarterly*, pages 197–219, 2005. 7.2

[51] Jun Liu, Lei Yuan, and Jieping Ye. An efficient algorithm for a class of fused lasso problems. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–332. ACM, 2010. 2.1.1

[52] W. Liu and S. Chawla. Mining adversarial patterns via regularized loss minimization. *Machine learning*, 81(1):69–83, 2010. 2.1.1, 3.2, 3.1, 3.3.7, 3.3.8

[53] Daniel Lowd and Christopher Meek. Adversarial learning. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 641–647, New York, NY, USA, 2005. ACM. 2.1.1

[54] David Meltzer, Jeanette Chung, Parham Khalili, Elizabeth Marlow, Vineet Arora, Glen Schumock, and Ron Burt. Exploring the use of social network methods in designing healthcare quality improvement teams. *Social science & medicine*, 71(6):1119–1130, 2010. 2.1.2

[55] Neal Miller and John Dollard. *Social Learning and Imitation.* New Haven, CT: Yale University Press., 1941. 2.1.2, 7.2

[56] M. Mohri, Rostamizadeh A., and A. Talwalkar. *Foundations of Machine Learning*. MIT Press, 2012. 2.1.1, 4.5.1

[57] Katharina Morik and Hanna Köpcke. Analysing customer churn in insurance data–a case study. In *Knowledge Discovery in Databases: PKDD 2004*, pages 325–336. Springer, 2004. 4.1

[58] J. Bruce Moseley, Kimberly O'Malley, Nancy J. Petersen, Terri J. Menke, Baruch A. Brody, David H. Kuykendall, John C. Hollingsworth, Carol M. Ashton, and Nelda P. Wray. A controlled trial of arthroscopic surgery for osteoarthritis of the knee. *New England Journal of Medicine*, 347(2):81–88, 2002. PMID: 12110735. 7.1

[59] Emmanuel Müller, Ira Assent, Patricia Iglesias, Yvonne Mülle, and Klemens Böhm. Outlier ranking via subspace analysis in multiple views of the data. In *IEEE International Conference on Data Mining*, pages 529–538, 2012. 5.1, 5.2, 5.4.2

[60] F Ellen Netting and Frank G Williams. Case manager-physician collaboration: Implications for professional identity, roles, and relationships. *Health & Social Work*, 21(3):216–224, 1996. 2.1.2

[61] Andrew Y Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78. ACM, 2004. 4.4, 4.5.4, 4.5.5, 7.4.1

[62] Niels Noorderhaven and Anne-Wil Harzing. Knowledge-sharing and social interaction within MNEs. *Journal of International Business Studies*, 40(5):719–741, 2009. 7.2

[63] Douglas K Owens, Amir Qaseem, Roger Chou, and Paul Shekelle. High-value, cost-conscious health care: concepts for clinicians to evaluate the benefits, harms, and costs of medical interventions. *Annals of internal medicine*, 154(3):174–180, 2011. 1

[64] Wei Pan, Nadav Aharony, and Alex Pentland. Composite social network for predicting mobile apps installation. In *AAAI*, 2011. 7.2, 7.4.2

[65] G.G. Pekhimenko. Penalizied logistic regression for classification. 3.3.3

[66] Bernice A. Pescosolido and Judith A. Levy. The role of social networks in health, illness, disease and healing: the accepting present, the forgotten past, and the dangerous potential for a complacent future. In *Social Networks and Health*, volume 8 of *Advances in Medical Sociology*, pages 3–25. Emerald Group Publishing Limited, 2002. 7.2

[67] Soumya Raychaudhuri, Joshua M Stuart, and Russ B Altman. Principal components analysis to summarize microarray experiments: application to sporulation time series. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, page 455. NIH Public Access, 2000. 4.4

[68] S. Shalev-Shwartz, N. Srebro, and T. Zhang. Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM Journal on Optimization*, 20(6), 2010. 3.3.3

[69] Gary W Sherwin. Low noise electroencephalographic probe wiring system, July 7 1987. US Patent 4,678,865. 4.4

[70] AjitP. Singh and GeoffreyJ. Gordon. A unified view of matrix factorization models. In *Machine Learning and Knowledge Discovery in Databases*, volume 5212

of *Lecture Notes in Computer Science*, pages 358–373. Springer Berlin Heidelberg, 2008. 5.4

[71] Alexander Johannes Smola and B Schölkopf. *Learning with kernels*. Citeseer, 1998. 3.3.1.2

[72] Lucia S Sommers, Keith I Marton, Joseph C Barbaccia, and Janeane Randolph. Physician, nurse, and social worker collaboration in primary care for chronically ill seniors. *Archives of Internal medicine*, 160(12):1825–1833, 2000. 2.1.2

[73] Uma Srinivasan and Bavani Arunasalam. Leveraging big data analytics to reduce healthcare costs. *IT Professional*, 2013. 6.1

[74] Wasserman Stanley and Katherine Faust. Social network analysis: methods and applications. *Cambridge: Cambridge University*, 2003. 2.2.4

[75] C.H. Teo, SVN Vishwanthan, A.J. Smola, and Q.V. Le. Bundle methods for regularized risk minimization. *Journal of Machine Learning Research*, 11:311–365, 2010. 3.3

[76] R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R.J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2012. 4.4

[77] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. 2.1.1

[78] A.N. Tikhonov and V.Y Arsenin. *Solution of Ill-posed problems*. Washington:Winston and Sons, 1977. 2.1.1

[79] UCI. Spambase data set. https://archive.ics.uci.edu/ml/datasets/Spambase, 2015. [Online; accessed 8-April-2018]. 3.3.1.3

[80] Shahadat Uddin, Liaquat Hossain, Jafar Hamra, and Ashraful Alam. A study of physician collaborations through social network and exponential random graph. *BMC health services research*, 13(1):234–247, 2013. 1

[81] Luis N Vicente and Paul H Calamai. Bilevel and multilevel programming: A bibliography review. *Journal of Global optimization*, 5(3):291–306, 1994. 3

[82] Fei Wang, Sanjay Chawla, and Wei Liu. Tikhonov or lasso regularization: Which is better and when. In *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pages 795–802. IEEE, 2013. 3.2.1, 7.4.1

[83] Fei Wang, Uma Srinivasan, Shahadat Uddin, and Sanjay Chawla. Application of network analysis on healthcare. In *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*. IEEE, 2014. 7.2

[84] Liang Xiong, Xi Chen, and Jeff G. Schneider. Direct robust matrix factorizatoin for anomaly detection. In *IEEE International Conference on Data Mining (ICDM)*, pages 844–853, 2011. 2.1.1, 5.4.2, 5.5, 5.5.2

[85] Huan Xu, Constantine Caramanis, and Shie Mannor. Robust regression and lasso. *Information Theory, IEEE Transactions on*, 56(7):3561–3574, 2010. 2.1.1, 3.2.1, 4.2, 4.4, 4.5.1, 4.5.1, 4.5.2

[86] Yahoo!Research. *Web Spam Collections*, 2013 (accessed March 20, 2013). http://barcelona.research.yahoo.net/webspam/datasets/. 4.6.1

[87] Jieping Ye, Kewei Chen, Teresa Wu, Jing Li, Zheng Zhao, Rinkal Patel, Min Bae, Ravi Janardan, Huan Liu, Gene Alexander, et al. Heterogeneous data fusion for alzheimer's disease study. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1025–1033. ACM, 2008. 4.1

[88] Guo-Xun Yuan, Chia-Hua Ho, and Chih-Jen Lin. An improved glmnet for l1-regularized logistic regression. *The Journal of Machine Learning Research*, 98888:1999–2030, 2012. 2.1.1

[89] Lei Zhang, Meng Yang, and Xiangchu Feng. Sparse representation or collaborative representation: Which helps face recognition? In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 471–478. IEEE, 2011. 4.4

[90] Y. Zhou, M. Kantarcioglu, B. Thuraisingham, and B. Xi. Adversarial support vector machine learning. In *Proceedings of the 18th ACM SIGKDD international*

conference on Knowledge discovery and data mining, pages 1059–1067. ACM, 2012. 2.1.1, 3.2, 3.1, 3.3.8

[91] Yan Zhou and Murat Kantarcioglu. Adversarial learning with bayesian hierarchical mixtures of experts. 2.1.1

[92] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005. 4.4