STRATEGIC LEARNING

By

FIDAN BOYLU

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2006

To my mother, Leyla Boylu

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Gary J. Koehler, for giving me the great honor and pleasure to work with him. He has been beyond an advisor to me and I have always admired and looked up to him. He has become the second most important person in my life after my mom over the years I spent in Gainesville. He has an extraordinary power of making research enjoyable with his surprising ideas, his unique way of thinking and amazing style of attacking a problem. He has given me the most valuable skills, tools and lessons that will last a lifetime. I would like to thank him especially for his endless help and support at the times of stress. I would also wish to thank my cochair, Dr. Haldun Aytug, for always encouraging and challenging me throughout the process. It would not have been so much fun without his brilliance and smart ideas. I also would like to thank Dr. David Sappington and Dr. Praveen Pathak for their time and interest.

Special thanks go to my friend Bilge Gokhan Celik for making Gainesville a more bearable place by always having the time to listen to my pointless and endless monologues and also for making me feel better by proving that there is at least one other person around that could worry about anything and everything more than I could. I also wish to thank Arzu Erenguc for being the best friend ever.

Last but certainly not least, I am forever indebted to my mother, Leyla Boylu, for her influence and support in realizing my career goals and for teaching me how to be tough and not to give up no matter how difficult things get.

TABLE OF CONTENTS

page

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

STRATEGIC LEARNING

By

Fidan Boylu

August, 2006

Chair: Gary J. Koehler
Cochair: Haldun Aytug
Major Department: Decision and Information Sciences

It is reasonable to anticipate that rational agents who are subject to classification by a principal using a discriminant function might attempt to alter their true attribute values so as to achieve a positive classification. In this study, we explore this potential strategic gaming and develop inference methods for the principal to determine discriminant functions in the presence of strategic behavior by agents and show that this strategic behavior results in an alteration of the usual learning rule.

Although induction methods differ from each other in various aspects, there is one essential issue that is common to all: the assumption that there is no strategic behavior inherent in the sample data generation process. In that respect, the main purpose of this study is to research the question, "What if the observed attributes will be deliberately modified by the acts of some self-interested agents who will gain a preferred classification by engaging in such behavior."? Hence, we investigate the need for anticipating this kind of strategic behavior and incorporate it into the learning process.

Since classical learning approaches do not consider the existence of such behavior, we aim to contribute by using rational expectations theory to determine optimal classifiers to correctly classify instances when such instances are strategic decision making agents. We carry out our analysis for a powerful induction method known as support vector machines.

First, we define the framework of Strategic Learning. For separable data sets, we characterize an optimal strategy for the principal that fully anticipates agent behavior in a setting where agents have fixed reservation costs. For non-separable data sets, we provide an MIP formulation and apply it to a credit-risk evaluation setting. Then, we modify our framework by considering a setting where agent costs and reservations are both unknown by the principal. Later, we develop a Genetic Algorithm for Strategic Learning to solve larger versions of the problem. Finally, we investigate the situation where there is a need to enforce constraints on agent behavior in the context of Strategic Learning and thus we extend the concept of Strategic Learning to constrained agents.

CHAPTER 1
INTRODUCTION

In many situations a principal gathers a data sample containing positive and negative examples of a concept to induce a classification rule using a machine learning algorithm. Although learning algorithms differ from each other in various aspects, there is one essential issue that is common to all: the assumption that there is no strategic behavior inherent in the sample data generation process. In that respect, we ask the question "What if the observed attributes will be deliberately modified by the acts of some self-interested agents who will gain a preferred classification by engaging in such behavior."? Therein for such cases there is a need for anticipating this kind of strategic behavior and incorporating it into the learning process. Classical learning approaches do not consider the existence of such behavior. In this dissertation we study this paradigm.

This dissertation is organized as a collection of articles, each of which covers one of several aspects of the entire study. Each chapter corresponds to an article which is complete within itself. Due to this self-contained style of preparation, we allowed for redundancies across chapters. This chapter is intended to give an outline of this dissertation.

In both Chapters 2 and 3, we provide an overview of Strategic Learning with the exception that we intend to reach different type of readers. In Chapter 4, we give a comprehensive and intricate study of Strategic Learning and provide the details of the model. In Chapter 5, we develop a Genetic Algorithm for Strategic Learning to solve larger versions of the problem. In Chapter 6, we extend the Strategic Learning model to

handle more complex agent behaviors and in Chapter 7, we conclude with a discussion of results and future work.

CHAPTER 2
STRATEGIC LEARNING

Machine Learning and Data Mining Paradigm–Supervised Learning

Today's highly computerized environment makes it possible for researchers and practitioners to collect and store any kind or amount of information easily in electronic form. As a result, an enormous amount of data in many different formats is available for analysis. This increase in the availability and easy access of data enables many companies to constantly look for ways to make use of their vast data collections to create competitive advantage and keep pace with the rapidly changing needs of their customers. This strong demand for utilizing the available data has created a recent interest in applying machine learning algorithms to analyze large amounts of corporate and scientific data, a practice which is called "data mining." Here we use the terms data mining and machine learning interchangeably.

An important type of machine learning commonly used in data mining tasks is called "supervised learning" which is performed by making use of the information collected from a set of examples called the "training set." The training set usually takes the form $S = \left(\left(x_1, y_1 \right), ..., \left(x_\ell, y_\ell \right) \right)$ where ℓ is the total number of available examples. Each example (also called an instance) is denoted by $x_i = (x_{i1}, ..., x_{in})$ which is the vector of n attributes for the example. The label of each example is denoted by y_i and is assumed to be known for each instance which is why supervised learning is sometimes referred to as "learning with a teacher." Given this setting, we are interested in choosing

an hypothesis that will be able to discriminate between classes of instances. A wide range of algorithms have been developed for this task including decision trees (Quinlan 1986), neural networks (Bishop 1995), association rules (Agrawal et al. 1993), discriminant functions (Fisher 1936), and support vector machines (Cristianini and Shawe-Taylor 2000). Particularly, throughout this chapter, we focus on binary classification where $y_i \in \{-1,1\}$. Informally, we are interested in classification of two classes of instances which we call the negative class and positive class respectively. We choose a collection of candidate functions as our hypothesis space. For example, if we are interested in a linear classifier, then the hypothesis space consists of functions of the form $w'x+b$. Under these circumstances, the goal is to learn a linear function $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$ such that $f(x) \geq 0$ if $x \in X$ belongs to the positive class and $f(x) < 0$ if it belongs to the negative class.

Recently, a new paradigm has evolved for the binary classification problem. We call it "Strategic Learning." One typical aspect common to all data mining methods is that they use training data without questioning the future usage of the learned function. More specifically, none of these algorithms take into account the possibility that any future observed attributes might be deliberately modified by their source when the source is a human or collection of humans. They fail to anticipate that people (and collections of people) might "game the system" and alter their attributes to attain a positive classification. As an example, consider the credit card approval scenario where certain data (such as age, martial status, checking account balance, number of existing credit cards, etc.) are collected from each applicant in order to be able to make an approval decision. There are hundreds of websites that purport to help applicants increase their

credit score by offering legal ways to manipulate their information prior to the credit application. Also, the case of a terrorist trying to get through airline security is another vivid example of how certain individuals might try to proactively act in order to stay undetected under certain classification systems where a decision maker determines functions such as f to be able to classify between classes of individuals. Throughout this chapter, we will speak collectively of these yet to be classified individuals as "agents" and the decision maker as the "principal."

Until now most researchers assume that the observed data are not "strategic" which implicitly assumes that the attributes of the agents are not subject to modification in response to the eventual decision rule. However, this type of strategic behavior is usually observed in many real world settings as suggested by the above examples. Thus, it is reasonable to think that individuals or companies might try to game systems and Strategic Learning aims to develop a model for this specific type of classification setting where the instances which are subject to classification are known to be self-interested, utility maximizing, intelligent decision making units.

In the Strategic Learning setting, each agent i has a utility function, a "true" vector of attributes x_i and a true group membership (i.e., label) y_i. For linear utility, an agent has a vector of costs for modifying attributes c_i and, for a given task, a reservation cost r_i. For the task of achieving a particular classification, the reservation cost can be viewed as the maximum effort that an agent is willing to exert in order to be classified as a positive example which we assume is desirable. On the principal's side, C_{y_i} is assumed to be the penalty associated with misclassifications of a true type y_i. In that respect, we develop a model within utility theory combined with cost sensitive learning.

Before going into the details of Strategic Learning we look at a related area of machine learning where the discovery process considers economic issues such as cost and utility. In utility-based data mining the principal is a utility maximizing entity who considers not just classification accuracy in learning but also various associated costs. We briefly review this area before discussing Strategic Learning.

Economic Machine Learning–Utility-Based Data Mining

Utility-based data mining (Provost 2005) is closely related to the problem of Strategic Learning. This area explores the notion of economic utility and its maximization for data mining problems as there has been a growing interest in addressing economical issues that arise throughout the data mining process. It has often been assumed that training data sets were freely available and thus many researchers focused on objectives like predictive accuracy. However, economical issues come into play in data mining since over time data acquisition may become costly. Utility-based methods for knowledge induction incorporate data acquisition costs and trade these with predictive accuracy so as to maximize the overall principal utility. Hence these methods become more meaningful and reflective of the real world usage.

Utility-based data mining is a broad topic that covers and incorporates aspects of economic utility in data mining and includes areas such as cost-sensitive learning, pattern extraction algorithms that incorporate economic utility, effects of misclassification costs on data purchase and types of economic factors. Simply, all machine learning applications that take into account the principal's utility considerations fall into this category of data-mining research. The researchers in this area are focused primarily on two main streams. One stream focuses on cost sensitive learning (i.e., cost assigned to misclassifications) (Arnt and Zilberstein 2005, Ciraco et al. 2005, Crone et al. 2005,

Holte and Drummond 2005, McCarthy et al. 2005 and Zadrozny 2005). The other stream focuses on the costs associated with the collection of data (i.e., data acquisition cost) (Kapoor and Greiner 2005, Melville et al. 2005, Morrison and Cohen 2005). In the following section we will look at these two streams of research.

Cost Sensitive Learning

The first type of utility-based data mining explores the problem of optimal learning when different misclassification errors incur different penalties. This area has been revisited many times (Elkan 2001). Cost sensitive classification has been a growing area of research and aims to minimize the expected cost incurred in misclassifying the future instances rather than focusing on improving the predictive accuracy which is usually measured by the number of correctly classified instances. This shift of focus from predictive accuracy to cost of misclassifications is maintained by assigning penalties for misclassified instances based on the actual label of the instance. For example, in medical diagnosis domains, identifying a sick patient as healthy is usually more costly than labeling a healthy patient as sick. Likewise, in the spam filtering domain, false misclassification of a non-spam email is significantly more costly than misclassifying a spam email. Arnt and Zilberstein (2005) examine a previously unexplored dimension of cost sensitive learning by pointing to the fact that it is impractical to measure all possible attributes for each instance when the final result has time-dependent utility and they call this problem time and cost sensitive classification.

Holte and Drummond (2005) review the classic technique of classifier performance visualization, the ROC (receiver operating characteristic) curve, which is a two dimensional plot of the false positive rate versus the true positive rate and argue that it is inadequate for the needs of researchers and practitioners as they do not allow any of the

questions to be answered such as what is the classifier's performance in terms of expected cost or for what misclassification costs does the classifier outperform others. They demonstrate the shortcomings of ROC curves and argue that the cost curves overcome these problems. In that respect, the authors point to the fact that cost-sensitive measurement of classifier performance should be utilized since misclassification costs should be an important part of classifier performance evaluation.

Data Acquisition Costs

The second area of utility-based data mining, cost of data acquisition, is an important area which has potential implications for real-world applications and thus is a topic that receives positive attention from industry as well as academia. For example, for large, real-world inductive learning problems, the number of training examples often must be limited due to the costs associated with procuring, preparing, and storing the training examples and/or the computational costs associated with learning from them (Weiss and Provost 2003). In many classification tasks, training data have missing values that can be acquired at a cost (Melville et al. 2005). For example, in the medical diagnosis domain, some of the patient's attributes may require an expensive test to be measured. To be able to build accurate predictive models, it is important to acquire these missing values. However, acquiring all the missing values may not always be possible due to economical or other type of constraints. A quick solution would be to acquire a random subset of values but this approach may not be most effective. Melville et al. (2005) propose a method called *active feature-value acquisition* which incrementally selects feature values that are most cost-effective for improving the model's accuracy. They represent two policies, Sampled Expected Utility and Expected Utility, that acquire feature values for inducing a classification model based on an estimation of the expected

improvement in model accuracy per unit cost. Other researchers investigate the same problem under a scenario where the number of feature values that can be purchased are limited by a budget (Kapoor and Greiner 2005).

Whereas utility-based data mining incorporates a principal's utility, Strategic Learning additionally considers the possibility that the objects of classification are self-interested, utility maximizing, intelligent decision making units. We believe that Strategic Learning considerations make a substantial contribution to utility-based data mining. However it should be emphasized that Strategic Learning should be considered as a totally new stream of utility-based data mining. Strategic Learning looks at the problems where different classes of instances with different misclassification costs and utility structures can act strategically when they are subject to discrimination. In the next section, we cover the details of Strategic Learning.

Strategic Learning

As was mentioned before, we look into a certain class of problems in which a decision maker needs to discover a classification rule to classify intelligent agents. The main aspect of this problem that distinguishes it from standard data mining problems is that we acknowledge the fact that the agents may engage in strategic behavior and try to alter their characteristics for a favorable classification. We call this set of learning problems "Strategic Learning." In this type of data mining, the key point is to anticipate agent strategic behavior in the induction process. This has not been addressed by any of the standard learning approaches.

Depending on the type of application, the agent can be thought of as any type of intelligent decision making unit which is capable of acting strategically to maximize its

individual utility function. The following are some examples of strategic agents and corresponding principals under different real world settings:

- a credit card company (the principal) decides which people (agents) get credit cards.
- an admission board at a university (the principal) decides which applicants (agents) get admitted.
- an auditing group (principal) tries to spot fraudulent or soon to be bankrupt companies (agent).
- an anti-spam package (the principal is the package creator) tries to correctly label and then screen spam (which is agent created).
- airport security guards (the principal) try to distinguish terrorists from normal passengers (agents).

Apparently, in each of these settings and in many others that were not mentioned here, if agents know or have some notion of the decision rule that the principal uses, they can try to modify their attributes to attain a positive classification by the principal. In most cases, the attributes used by a principal for classification are obvious and many people can discern which might be changed for their benefit. In the credit approval case, it is likely that an increase in one's checking account balance or getting a job will be beneficial. Thus, it is reasonable to anticipate that the agents will attempt to manipulate their attributes (either thorough deceit or not) whenever doing so is in their best interest. This gaming situation between the agents and the principal leads to a need for anticipating this kind of strategic behavior and incorporating it to the standard learning approaches. Furthermore, if one uses classical learning methods to classify individuals when they are strategic decision making units, then it might be possible for some agents to be able to eventually game the system.

To date, very few learning methods fall in the Strategic Learning paradigm. The closest is called adversarial classification which we review below. Another related area is called reinforcement learning. This area has some aspects of Strategic Learning that we discuss below also.

Adversarial Classification

Dalvi et al. (2004) acknowledge the fact that classification should be viewed as a game between the classifier (which we call the principal) and the adversary (which we call the agent) for all the reasons that we have discussed so far. They emphasize the fact that the problem is observed in many domains such as spam detection, intrusion detection, fraud detection, surveillance and counter-terrorism. In their setting, the adversary actively manipulates data to find ways to make the classifier produce a false decision. They argue that the adversary can learn ways to defeat the classifier which would result in a degrading of its performance as the classifier needs to modify its decision rule every time agents react by manipulating their behaviors. Clearly, this leads to an arms race between the classifier and the adversary resulting in a never ending game of modifications on both sides since the adversary will react to classifier's strategy in every period and classifier will need to adjust accordingly in the next period. This poses an economical problem as well since in every period more human effort and cost are incurred to be able to modify the classifier to adapt to the latest strategy of the adversary.

They approach the Strategic Learning problem from a micro perspective by focusing on a single-shot version of the classification game where only one move by each player is considered. They start by assuming that the classifier initially decides on a classification rule when the data are not modified by the adversary. But, knowing that adversary will deploy an optimal plan against this classification rule, the classifier instead

uses an optimal decision rule which takes into account the adversary's optimal modifications. They focus on a Bayesian classification method.

Although their approach is quite explanatory, it is an initial effort since their goal was to be able to explain only one round of the game. However as they discuss, the ultimate solution is the one that solves the repeated version of this game. However, by viewing the problem as an infinite game played between two parties, they tend to encourage modifications rather then prevent these modifications. That leads to a key question that needs to be answered: is there an optimal strategy for the classifier which will possibly prevent an adversary from evolving against the classifier round after round when this strategic gaming is pursued infinitely? or is it possible to prevent an agent's actions by anticipating them before the fact and taking corrective action rather than reacting to the outcome after the game is played? These points are intensively investigated in Chapters 4 and 5 where we formulate the problem as the well-known principal-agent problem where the principal anticipates the actions of agents and uses that information to discover a fool-proof classifier which takes into account the possible strategic behavior. In that sense, their approach is more of a preventive one than a reactive one. Also, their model involves many strategic agents acting towards one principal as opposed to a two-player game setting.

Multi-Agent Reinforcement Learning

Another related area of research is reinforcement learning which involves learning through interactions (Samuel 1959 and Kaelbling et al. 1996). In this section, we will briefly discuss this area and point out its similarities and differences with Strategic Learning. Reinforcement learning is a field of machine learning in which agents learn by using the reward signals provided from the environment. Essentially, an agent

understands and updates its performance according to the interactions with its environment. In reinforcement learning theory, an agent is characterized by four main elements: a policy, a reward function, a value function, and a model of the environment. The agent learns by considering every unique configuration of the environment. An agent acts according to a policy that is essentially a function that tells the agent how to behave by taking in information sensed from the environment, and outputting an action to perform. Depending on the action performed, the agent can go from one state to another and the reward function assigns a value to each state the agent can be in. Reinforcement learning agents are fundamentally reward-driven and the ultimate goal of any reinforcement learning agent is to maximize its accumulated reward over time. A reward function outputs the immediate reward of a state while the value function specifies the long run reward of that state after taking into account the states that are likely to follow, and the rewards available in those states. This is generally achieved through a particular sequence of actions to be able to reach the states in the environment that offer the highest reward (Sutton and Barto 1998).

In that respect, a Strategic Learning agent is quite similar to reinforcement learning agent as the former also aims to maximize its utility while attempting to reach a preferred classification state. Also, the environment can be thought of as analogous to the principal who essentially decides on the reward function (the classifier).

However, the main difference of reinforcement learning from Strategic Learning is that learning is realized over time through interactions between the environment and agents, something that Strategic Learning essentially aims to avoid. The ultimate goal in Strategic Learning is to be able to anticipate the possible actions of the agents and take

preventive action. Strategic Learning attempts to avoid those interactions for the principal's sake by anticipating the agent's behavior since otherwise the principal is forced to modify its classifier after every interaction with the agents which is time consuming and costly in many data mining problems and, more importantly, causes degradation of the classifier's performance. In that respect, by using Strategic Learning model, a principal takes a more sophisticated approach and plans for his/her future course of action as opposed to doing simple reaction-based, trial-and-error exploration as in reinforcement learning. In addition, in reinforcement learning the interaction between the principal (the environment) and the agent is cooperative unlike in Strategic Learning where it is assumed adversarial.

A more specialized type of reinforcement learning which has more insight to the Strategic Learning is reinforcement learning in leader-follower multi-agent systems (Bhattacharyya and Tharakunnel 2005, Littman 1994). In leader-follower systems which have a number of applications such as monitoring and control of energy markets (Keyhani 2003), e-business supply chain contracting and coordination (Fan et al. 2003), modeling public policy formulation in pollution control, taxation etc. (Ehtamo et al. 2002), a leader decides on and announces an incentive to induce the followers to act in a way that maximizes the leader's utility, while the followers maximize their own utilities under the announced incentive scheme. This is similar in many ways to our principal-agent terminology as the leader acts as the principal and tries to identify and announce the ultimate decision rule that would maximize his/her own objective while the agents act as followers who seek to maximize their own utilities.

Bhattacharyya and Tharakunnel (2005) apply this kind of a sequential approach to propose a reinforcement based learning algorithm for repeated game leader-follower multi-agent systems. One of the interesting contributions of their work is the introduction of non-symmetric agents with varying roles to the existing multi-agent reinforcement learning research. This is analogous to our asymmetric setting where both the principal and agents are self-interested utility maximizing units with the exception that principal has a leading role in setting the classifier to which the agents react. This is similar to the interaction between leader and followers in their work.

However, a key difference between the two is the sequential nature of the decisions made by the leader and the followers. In the leader follower setting, the learning takes place progressively from period to period as leader and followers interact with each other according to the ideas of trial-and-error learning. Essentially, the leader announces his/her decision at specific points in time based on the aggregate information gained from the earlier rounds and the followers make their decisions according to the incentive announced at that period. In this scenario, the leader aims to learn an optimal incentive based on the cumulative information from the earlier periods while the followers try to learn optimal actions based on the announced incentive. Learning is achieved over successive rounds of decisions with information being carried from one round to the next.

Even though the leader-follower approach has similarities with Strategic Learning, there are fundamental differences. First, Strategic Learning is an anticipatory approach. In other words, in Strategic Learning, learning is achieved by anticipating strategic behavior by agents and incorporating this anticipation in the learning process rather than following an after the fact reactive approach. Second, Strategic Learning does not

involve periods based on the principles of principal-agent theory. Strategic Learning results show that a sequential approach will often yield suboptimal results.

Learning in the Presence of Self-Interested Agents

We approach to Strategic Learning more extensively in Chapters 3 and 4. Particularly, in Chapter 3, we explore the problem under the name of "Learning in the Presence of Self-interested Agents" and propose the framework for this type of learning that we briefly discuss in this section.

Refreshing the previously developed notation, let $X \subseteq \Re^n$ be the instance space which can be partitioned into two sets, a set consisting of positive cases consistent with some underlying but unknown concept and the remaining negative cases. For example, in Abdel Khalik and El-Sheshai (1980) the underlying concept is described by "firms that won't default or go bankrupt." Attributes such as the firm's ratio of retained earnings to total tangible assets, etc. were used in this study. One forms a training set by randomly drawing a sample of instances of size ℓ from X, and then determining the true label (-1 or 1) of each such instance. Using this sample, a machine learning algorithm infers a hypothesis. A key consideration is the choice of sample size. Whether it is large enough to control generalization error is of key importance.

Under our setting, the principal's goal is to determine the classification function f to select individuals (for example, select good credit applicants) or to spot negative cases (such as terrorists who try to get through as security line). However, each strategic agent's goal is to achieve positive classification (e.g., admission to university) regardless of their true label. For that, they act strategically and take actions in a way to alter their true attributes which may lead to an effectively altered instance space. In some cases it is

possible for the agents to infer their own rules about how the principal is making decisions under f and spot attributes that are likely to help produce positive classifications. Most classical learning methods operate using a sample from X but this space may be altered (call it $\overline{X}$) due to the ability of agents to infer their own rules about the classification rule. This possible change from X to $\overline{X}$ needs to be anticipated. Strategic Learning does this by incorporating rational expectations theory (Muth 1961) into the classical learning theory. Figure 2-1 outlines the Strategic Learning framework.

Figure 2-1 shows that when strategic behavior is applied to the sample space, X, it causes it to change from X to $\overline{X}$ (reflecting strategic behavior by the agents). Also, classical learning theory operates on X while Strategic Learning operates on all anticipated $\overline{X}$'s.

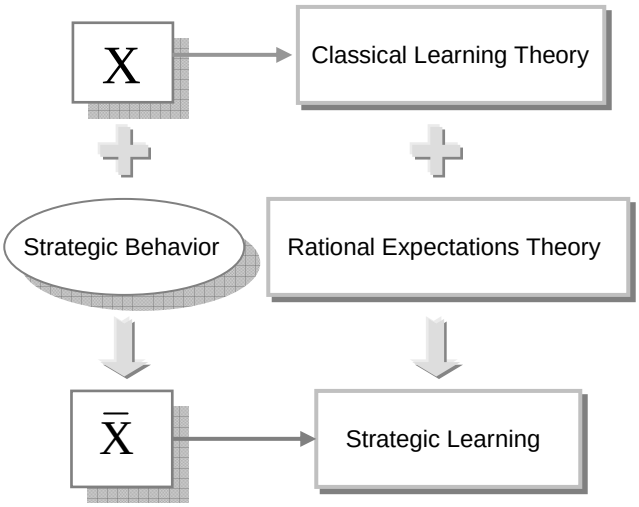


Figure 2-1. Strategic Learning framework.

Strategic Learning with Support Vector Machines

In Chapter 4 we consider Strategic Learning while inducing linear discriminant functions using Support Vector Machines (SVMs). SVM is an algorithm based on

Statistical Learning Theory (Vapnik 1998). In this section, we discuss Chapter 4 and some of our results.

SVMs are a computationally efficient way of learning linear discriminant functions as they can be applied easily to enormous sample data sets. In essence, SVMs are motivated to achieve better generalization by trading-off empirical error with generalization error. This translates to the simple goal of minimizing the margin of the decision boundary of the separating hyperplane with parameters (w, b). Thus, the problem reduces to minimizing the norm of the weight vector (w) while penalizing for any misclassification errors (Cristianini and Shawe-Taylor 2000). An optimal SVM classifier is called a maximum margin hyperplane. There are several SVM models and the first model is called the hard margin classifier which is applicable when the training set is linearly separable. This model determines linear discriminant functions by solving

$$\begin{aligned} &\underset{w,b}{Min}\ w'w \\ &s.t. \quad y_i(w'x_i + b) \geq 1 \quad i = 1, \ldots, \ell \end{aligned} \tag{1}$$

The above formulation produces a maximal margin hyperplane when no strategic behavior is present. To illustrate how strategic behavior alters the above model, we briefly look at the approach used in Chapter 4 of this dissertation to incorporate strategic behavior into the model. We start by introducing the agent's "strategic move problem" that shows how rational agents will alter their true attributes if they knew the principal's SVM classifier. If the principal's classification function ($f(x_i) = w'x_i + b - 1$) was known by rational agents, then each agent would solve what they call the strategic move problem to determine how they could achieve (or maintain) positive classification while

exerting minimal effort in terms of cost. This is captured in the objective function (cost minimization). Thus the agent's problem can be modeled as

$$\begin{aligned} &\min c_i' d_i \\ &st \quad w'\left[x_i + D(w) d_i\right] + b \geq 1 \\ &\quad d_i \geq 0 \end{aligned}$$

where $D(w)$ is a diagonal matrix defined by

$$D(w)_{j,j} = \begin{cases} 1 & w_j \geq 0 \\ -1 & w_j < 0 \end{cases}.$$

This problem finds a minimal cost change of attributes, $D(w) d_i$, if feasible. This is the amount of modification that an agent needs to make on his/her attributes to be classified as a positive case. This would be undertaken if this cost doesn't exceed the agent's reservation cost, r_i. Let $d_i^*(w,b)$ be an optimal solution for the agent's strategic move problem ($d_i^*(w,b)$ is zero if the strategic move problem is infeasible or if the agent lacks enough reservation). Then the principal's strategic problem becomes the following

$$\begin{aligned} &\min_{w,b} w'w \\ &s.t. \quad y_i \left\{ w'\left[x_i + D(w) d_i^*(w,b)\right] + b \right\} \geq 1 \qquad i = 1,...,\ell \end{aligned} \tag{2}$$

When compared with the non-strategic SVM model, the difference is the term $D(w) d_i^*(w,b)$ which depends on the agent's problem. Basically, this term represents the principal's anticipation of a modification of attributes by agent i. By incorporating this term into the principal's problem, this formulation makes it possible to prevent some misclassifications by taking corrective action before the fact (i.e., before the principal determines a classification rule and incurs misclassification cost as agents make modifications). The essential idea is to anticipate an agent's optimal strategic move and

use that information to infer a classification rule that will offset the agent's possible strategic behavior.

In Chapter 4, we derive a complete characterization for the solution of the principal's strategic problem under the base setting where all agents have the same reservation and change costs (i.e., $r_i = r$ and $c_i = c$) and $S = \left(\left(x_1, y_1 \right), ..., \left(x_\ell, y_\ell \right) \right)$ is linearly separable. Theorem 1 in Chapter 4 states the following: $\left(w^*, b^* \right)$ solves (1) if and only if $\left(\frac{2}{2 + t^*} w^*, \frac{2b^* - t^*}{2 + t^*} \right)$ solves (2) where t^* is given by $t^* = r \max_k \left| w_k^* \right| / c_k$ and $k = \arg \min_{j, w_j \neq 0} \frac{\left(c_i \right)_j \max\left(0, 1 - \left(b + w' x_i \right) \right)}{\left| w_j \right|}$.

In essence, Theorem 1 states that a principal anticipating strategic behavior of agents all having the same utilities and cost structures will use a classifier that is parallel to the non-strategic SVM solution, $\left(w^*, b^* \right)$. The solution to the strategic SVM is a scaled (by $2 / \left(2 + t^* \right)$) and shifted form of $\left(w^*, b^* \right)$. The margin of the strategic SVM solution hyperplane is greater than the non-strategic SVM solution and thus the probability of better generalization is greater. This scaling factor depends on the cost structure for altering attribute values, the reservation cost for being labeled as a positive case, and $\left(w^*, b^* \right)$.

Figure 2-2 (a) shows a completely separable training set with a hyperplane using the non-strategic SVM classifier (solid line) and corresponding positive and negative margins (dotted lines) along with data points in two-dimensional space. In Theorem 1, we show that the "negative" agents will try to achieve a positive labeling by changing

their true attributes if the cost of doing so doesn't exceed their reservation cost. This is indicated in Figure 2-2 (a) with the horizontal arrows pointing out from some the points for negative agents towards the positive side of the hyperplane. Clearly, these are the agents whose reservation costs are sufficiently high and are willing to engage in strategic behavior to move to the positive side of the hyperplane. However, the principal, anticipating such strategic behavior, shifts and scales the hyperplane such that no true negative agent will benefit from engaging in such behavior. Figure 2-2 (b) shows the sample space and the resulting strategic classifier. However, as Figure 2-2 (b) shows, the negative agents would have no incentive to change their true attributes since they will not be able to move to the positive margin any more and, hence, would not exert any effort. Apparently, this shift may leave some marginally positive agents in danger of being classified as negative agents. Since they too anticipate that the principal will alter the classification function so as to cancel the effects of the expected strategic behavior of the negative labeled agents, they might undertake changes. In other words, they are forced to move as indicated by the arrows in Figure 2-2 (b). Thus, the ones who are "penalized" for engaging in a strategic behavior (i.e., must exert effort to attain a positive classification) are not the negative agents but rather the marginal positive agents. Moreover, the resulting classifier has a greater margin and better generalization capability compared to the non-strategic SVM learning results.

In Figure 2-2 (c), the new strategic classifier with wider margins on each side and the resulting altered instances due to strategic behavior is pictured. Notice that the margin of the resulting hyperplane is wider and is in fact a scaled and a shifted version of the hyperplane in Figure 2-2 (a) and thus differs from non-strategic SVM results.

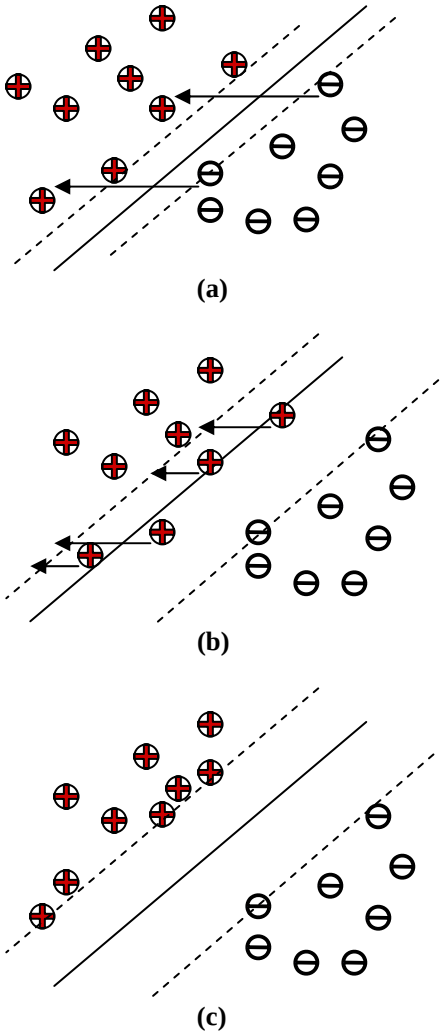


Figure 2-2. Wider margins.(a) Separable training set with a non-strategic SVM classifier (b) Sample space as the result of strategic behavior (c) Strategic SVM classifier.

For non-separable datasets, there are no results comparable to Theorem 1. However, for that case where S is not linearly separable and all agents have their own reservation and change costs (i.e., r_i and c_i), we. derive mixed integer programming models for the solution of the principal's strategic problem. We apply our results on a credit card application setting and our results show that the strategic formulation performs better than its non-strategic counterpart.

Chapter 5 is an extension of our work which considers the cases when it is not realistic to let each attribute be modified unboundedly without posing any constraints on how much they can actually be modified. In other words, agents are constrained on how much and in which way they can modify their attributes. Towards that end we look at a spam categorization problem where spammers (negative agents) are only allowed to make modifications in the form of addition or deletion of words with an upper limit on the number of modifications allowed. Essentially, we formulate the problem by allowing only binary modifications which is an interesting constraint on the agent behavior. Clearly, agents can be constrained in many other ways such as upper and lower bounds on the modifications or the modifications may need to belong to a certain set of moves (like in checkers or chess).

Interestingly, for the spam categorization problem, we point out that not all agents are strategic and in fact only the negative agents (spammers in our case) act strategically since it is not reasonable for legitimate internet users to engage in strategic behavior and change the content of their emails. This is quite distinguishing as it is a Strategic Learning model for an environment where non-strategic and strategic agents coexist.

Strategic Learning–Future Research

The form of Strategic Learning problem discussed in this chapter assumes that the only costs are misclassification costs and there is no cost associated with making the true positive agents alter their behavior. Including these costs would create a formulation that is equivalent to the social welfare concept common in economics literature as the principal may need to trade-off the misclassification cost with the disutility caused by forcing the true positive agents to move. In that way, the principal would be able to maximize his/her own utility while minimizing the disutility of positive agents. Also,

Theorem 1 is only applicable to separable datasets and an important contribution would be to develop a similar theoretical result for non-separable datasets.

One of the most interesting angles for future research is to be able to remove the assumption that both the principal and the agents know each other's parameters. In practice this assumption rarely holds and it's usually the case that principal and agents will try to somehow roughly predict each others parameters. We provide several formulations in Chapter 4 for cases where the agent's utility parameters are not known with certainty. More work is needed in this area.

It is possible that the classifier developed when the agents do not collude may be suboptimal when the agents cooperate and act seemingly in an irrational way. For example, determining what happens in a scenario where agents collude and offer incentives to other agents to make sub-optimal changes in their attributes to confuse the principal would make this problem more realistic.

It is possible to reverse the Strategic Learning problem and use these ideas to create a classifier (or a policy) that promotes certain actions (rather than avoid) or use these ideas as a what-if tool to test the implications of certain policies. For example, board of directors could develop executive compensation policies to promote long term value generation by anticipating how the CEOs can game the system to their advantage (which usually causes short term gains at the cost of long term value).

All discussion so far has focused on using SVMs as a classifier even though learning theory and rational expectations theory are independent of implementation details. It would be very useful to determine the validity of these results independent of implementation and introduce the Strategic Learning problem to the other classifiers like

decision trees, nearest neighbor, neural networks, etc. Essentially, Strategic Learning is a general problem that will arise in any learning situation involving intelligent agents so it should be applied to other learning algorithms.

Another key area of future research is the application of domain knowledge to Strategic Learning. Kernel methods accomplish this by using a nonlinear, higher dimensional mapping of attributes to features to make the classes linearly separable. It may be possible to compute an appropriate kernel that can anticipate and cancel the effects of strategic behavior. Such a possible kernel can be developed using agents' utility functions and cost structures which are a form of domain specific knowledge.

The current research on Strategic Learning so far only addresses static situations. However, it is possible that some exogenous factors like the environment or the parameters being used are changing over time. For example, it might be possible that over time some new attributes may be added to the data set or conversely some may become obsolete. This type of a dynamic situation might need to be modeled in a way to accommodate the possible changes to be able to determine classifiers that will adapt efficiently.

There is yet another angle to approach the problem which is the game theoretical point of view which has been partially addressed by Dalvi et al. (2004). However, further investigation of this angle is an interesting and relevant task for future research in the area.

An important area of research in Strategic Learning is to find better algorithmic methods for solving the Strategic Learning problem. While mixed integer formulations exist, solution methods currently do not scale-up like the non-strategic counterparts.

CHAPTER 3
LEARNING IN THE PRESENCE OF SELF-INTERESTED AGENTS[1]

In many situations a principal gathers a data sample containing positive and negative examples of a concept to induce a classification rule using a machine learning algorithm. Although learning algorithms differ from each other in various aspects, there is one essential issue that is common to all: the assumption that there is no strategic behavior inherent in the sample data generation process. In that respect, we ask the question "What if the observed attributes will be deliberately modified by the acts of some self-interested agents who will gain a preferred classification by engaging in such behavior."? Therein for such cases there is a need for anticipating this kind of strategic behavior and incorporating it into the learning process. Classical learning approaches do not consider the existence of such behavior. In this chapter we study the need for this paradigm and outline related research issues.

Introduction

Machine learning research has made great progress in many areas and applications over the past decade. Many machine learning algorithms have evolved to the point that they can be used in typical commercial data mining tasks including credit approval (Chapter 4), spam detection (Fawcett 2003), fraud detection (Fawcet and Provost 1997), text categorization (Dumais et al. 1998), etc.

[1] An earlier version of this chapter was published in the Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06), Track 7, p. 158b.

One of the most common types of data mining task is supervised learning. Here the learner acquires a training sample that consists of ℓ previously labeled cases and applies a machine learning algorithm that uses the sample to choose a "best fit" hypothesis from a designated hypothesis space. This chosen concept will be used to classify (i.e., label) unseen cases later. We can think of an example as a vector of attribute values (an instance) plus a class label. The set of all instances is called the instance space.

Suppose there are n attributes. Then a vector of attributes is $x \in \Re^n$. For example, in a credit approval application, attributes might be age, income, number of dependents, etc. The label can be binary, nominal or real valued. For example, in the credit approval problem, the labels might be "good" or "bad" or simply +1 or -1. Then an example would be the pair (x, y) where $x \in \Re^n$ and $y \in \{-1,1\}$. An hypothesis space consists of all the possible ways the learner wishes to consider representing the relationship between the attributes and the label. For example, an often used hypothesis space is the set of linear discriminant functions. A particular hypothesis would be (w, b) where $w \in \Re^n$. Then $(w, b): \Re^n \to \{-1,1\}$ meaning if $w'x + b \geq 0$ then the label is +1, otherwise it is -1. A learning algorithm selects a particular hypothesis (e.g., a particular (w, b)) that best satisfies some induction principle. The "supervised" in supervised learning means that the training sample contains correct labels for each instance.

There are a plethora of learning methods. Some of the most popular methods include decision tree methods (Quinlan 1986), neural network methods (Rosenblatt 1958), Bayesian methods (Duda and Hart 1973), Support Vector Machines (SVMs) (Cristianini and Shawe-Taylor 2000), etc. Although many of the algorithms differ from

each other in many aspects, there is one essential issue that is common to all: the implicit assumption that there is no strategic behavior inherent in the sample data generation process. For example, in supervised learning where a decision maker uses a training set to infer an underlying concept, the training examples are taken "as is." In that respect, we ask the question "What if the observed attributes will be deliberately modified by the acts of some self-interested agents who will gain a preferred classification by engaging in such behavior."?

If these agents know the true classification rule, they can easily discern how changing their attributes could lead to a positive classification and, assuming the cost to change these attributes is acceptable, then proceed to make the changes. If the classification rule is not known by the agents, either the agents can attempt to discover what the important attributes might be or, more likely, use common sense to alter obvious attributes.

For example, poor credit risk individuals interested in obtaining credit might proactively try to manipulate their attributes (either through deceit or not) so as to obtain a positive rating (e.g., there are many websites that purport to help people change their credit ratings).

A more extreme example is the terrorist who tries to appear "normal" so as to gain access to potential target sites. Less extreme are spammers who continuously try to break through screening rules by changing their e-mail messages and titles.

So there is a need for anticipating this kind of strategic behavior and incorporating such considerations into the classical learning approaches. Currently none consider the existence of such behavior.

In this chapter, we outline the need for this kind of a paradigm and present many related emerging research issues. Many types of learning situations potentially fall in this setting. In fact, whenever the instances represent adaptive agents, the risk of strategic behavior is possible. We frame this type of learning in a principal-agent setting where there are many rational agents who act as autonomous decision making units working to maximize their individual utilities and a principal who needs to classify these agents as belonging to one class or the other. For example, a credit card company (the principal) decides which people (agents) get credit cards. An admission board at a university (the principal) decides which applicants (agents) get admitted. An auditing group (principal) tries to spot fraud or detect imminent bankruptcy in companies (agent). An anti-spam package (the principal is the package creator) tries to correctly label and then screen spam (agent created). In each of these examples, we assume that the agents can anticipate what the resulting classifier is and modify their attributes accordingly. Knowing this the principal will create a classifier to cancel the effects of such behavior. Such anticipation is based on the assumption that each party acts rationally and each knows the other's parameters.

We start with a discussion of related literature in the next section. Only two recent papers directly address Strategic Learning. We also discuss a somewhat similar scenario that arises in leader-follower learning. Later, we illustrate the ideas of Strategic Learning and outline areas of possible research. We end with a summary in the last section.

Related Literature

Since the agents can learn, anticipate and react to the principal's classification method by altering their behavior, the principal in turn needs to adapt his/her strategy in order to be able to cancel the effects of agents' strategic efforts. This arms race between

the two parties (principal as opposed to agents) is the essential motivation of Strategic Learning. However, as was pointed out in Dalvi et al. (2004) the key goal for the principal is to identify the ultimate decision rule initially especially when this strategic gaming could be pursued infinitely. That is, the principal rather than reacting to the agents' actions in an after the fact fashion, needs to anticipate their strategic behavior and identify an optimum strategy taking the agents' possible strategic behavior into account, an approach which is explored deeply in Chapter 4.

We note that even in learning situations where an optimal induction can be achieved, as we study in Chapter 4, the usual "arms race" approach may not discover this rule. That is, if a principal gathers data, induces a rule and starts using this rule, strategic agents will attempt to alter their attributes for positive classification. The instance space is now different forcing the principal to gather a new data set and induce a new rule. In Chapter 4 we give an example where this adaptive learning does not converge to an optimal rule. Indeed, in our setting, such convergence may be rare.

Dalvi et al. (2004) argue that in many domains such as spam detection, intrusion detection, fraud detection, surveillance and counter-terrorism, the data are being manipulated actively by an adversary ("agent" in our terminology) seeking to make the classifier ("principal" in our terminology) produce a false decision. They further argue that in these domains, the performance of a classifier can degrade rapidly after it is deployed, as the adversary learns to defeat it. We agree.

They view classification as a two period game between the classifier and the adversary, and produce a classifier that is optimal given the adversary's optimal strategy. They also show that a Nash equilibrium exists when the adversary incurs unit cost for

altering an observation. However, as they suggest, computation of a Nash equilibrium is hard since the computation time is exponential in the number of attributes. The experiments they do in the spam detection domain show that their approach can greatly outperform a standard classifier that does not take into consideration any strategic behavior by automatically adapting the classifier to the adversary's evolving manipulations.

Referring to their example, in the domain of e-mail spam detection, standard classifiers like naive Bayes were initially quite successful but spammers soon learned to fool them by inserting “non-spam" words into e-mails or breaking up “spam" ones with spurious punctuation, etc. As the spam filters were modified to detect these tricks, spammers started using new tricks (Fawcett 2003). Eventually spammers and filter designers are engaged in a never-ending game of modification as filters continually include new ways to detect spam and spammers continually invent new ways to avoid detection.

This kind of gaming is not unique to the spam detection domain and is found in many other domains such as computer intrusion detection, where the anti-virus programs are continuously updated as new attacks are experienced; fraud detection, where perpetrators change their tactics every time they get caught (Fawcett and Provost 1997); web search, where search engines constantly revise their ranking functions in order to cope with pages which are manipulated in order to have higher rankings; etc. As a result, the performance of principal can drop remarkably when an adversarial environment is present.

In Chapter 4 we develop methods to determine linear discriminant classifiers in the presence of strategic behavior by agents. We focus on a powerful induction method known as support vector machines, and for separable data sets, we characterize an optimal strategy for the principal that fully anticipates agent behavior. In our setting, agents have linear utility and have a fixed reservation costs for changing attributes. The principal anticipates the optimal agent behavior for a given classifier (i.e., he uses rational expectations) and chooses a classifier that can't be manipulated within the acceptable cost ranges for the agents. In this setting there is no possibility for a "cat and mouse" type scenario. No actions by the agents can alter the principal's classification rule.

Here, our first important result is that under specific conditions, an optimal linear discriminant solution with strategic behavior is a shifted and scaled version of the solution found by SVMs without strategic behavior. This result is striking since we prove that it is optimal. So far, this is the only Strategic Learning result that incorporates rational expectations theory into the classical learning approach and is proved to be optimum. Hence, the main contribution of our work is that rational expectations theory can be used to determine optimal classifiers to correctly classify instances when such instances are strategic decision making agents. This provides a new way of looking at the learning problem and thus opens up many research areas to investigation.

For non-separable data sets we give mixed integer programming formulations and apply those to a credit-risk evaluation setting. Our results show that discriminant analysis undertaken without taking into account the potential strategic behavior of agents could be misleading and can lead to unwanted results.

Although Strategic Learning approaches have not been used before, learning problems involving intelligent agents in a gaming situation have been investigated in other settings. For example, in leader-follower learning systems which have a number of applications such as monitoring and control of energy markets (Keyhani 2003), e-business supply chain contracting and coordination (Fan et al. 2003), modeling public policy formulation in pollution control, taxation etc. (Ehtamo et al. 2002), a leader decides on and announces an incentive to induce the followers to act in a way that maximizes the leader's utility, while the followers maximize their own utilities under the announced incentive scheme. This is analogous in some sense to our setting since the principal (the "leader" in their terminology) tries to identify (i.e., learn) and announce the ultimate decision rule that would maximize her own objective while the agents' ("followers" in their terminology) seek to maximize their own utilities. In both cases, it is possible to think of the situation as the principal (or leader) aiming to maximize some kind of a social welfare function given the self interested actions of the agents.

Specifically, these kinds of decision situations are termed incentive Stackelberg games (Von Stackelberg 1952) where the leader first determines an incentive function and announces it and the followers, after observing the announced incentive, make their own decisions. For example, in their work, Bhattacharyya et al. (2005) apply this kind of a sequential approach to propose a reinforcement based learning algorithm for repeated game leader-follower multi-agent systems. The key point here is the sequential nature of the decisions made by the leader and the followers. The learning takes place progressively as principal and agents interact with each other based on the principles of reinforcement learning (Kaelbling et al. 1996) which is centered around the idea of trial-

and-error learning. Specifically, the leader announces his decision at specific points in time based on the aggregate information gained from the earlier rounds, and the followers make their decisions according to the incentive announced at that period.

In this scenario, the leader aims to learn an optimal incentive based on the cumulative information from the earlier periods while the followers try to learn optimal actions based on the announced incentive. Learning is achieved over successive rounds of decisions with information being carried from one round to the next. This is where existing research and the framework proposed in this paper differ from each other. The ensuing analysis demonstrates that this sequential approach will often yield suboptimal results while the ultimate solution can only be found by anticipating and incorporating this anticipation in the learning process rather than following an after the fact reactive approach.

Illustration

To further illustrate the Strategic Learning framework discussed in this chapter, let $X \subseteq \Re^n$ be the instance space. X is partitioned into two sets, a set consisting of positive examples consistent with some underlying but unknown concept and the remaining negative examples. For example, in Messier and Hansen (1988) the underlying concept is described by "firms that won't default or go bankrupt." In their study, the attributes consists of values such as the firm's ratio of retained earnings to total tangible assets, etc. A training set is formed by randomly drawing a sample of instances of size ℓ from X, and then determining the true label (-1 or 1) of each such instance. From this sample, one uses a machine learning algorithm to infer a hypothesis from these examples. The choice of a sample size that is large enough to control generalization error is of key importance.

Under this setting, the principal's desired goal is to determine a classification function f (i.e., an hypothesis) to select individuals (for example, select college applicants likely to succeed) or to spot negative cases (such as which firms are likely to file for bankruptcy or commit fraud, etc.). However, the agents acting strategically may take actions in a way to improve their situation in hopes of achieving positive classification (e.g., admission to university). This may lead to an effectively altered instance space. After f is discovered by the principal, agents may take the classification rule into account even if f has not been announced. For example, agents can infer their own rules about how the principal is making decisions under f and spot attributes that are likely to help produce positive classifications. Most classical learning methods will operate using a sample from X but this space may be altered by subsequent agent actions (call it $\overline{X}$). This change needs to be anticipated. One powerful way of doing this is by incorporating rational expectations theory to the classical learning theory. Figure 2-1 of Chapter 2 outlines this new proposed framework.

Figure 2-1 shows that when strategic behavior is applied to the sample space, X, it causes it to change from X to $\overline{X}$ and parallel to that rational expectations theory applied to classical learning theory transforms it to what we suggest as Strategic earning. .Notice that while classical learning theory operates on X, strategic learning operates on all anticipated $\overline{X}$'s.

Learning theory assumes that the principal (learner) has a loss function that usually measures the error associated with the classification task. In general, the principal has to worry about two types of errors: the empirical error and the generalization error. The empirical error is easily estimated by using the sample. However, the generalization

error, a measure of how well the function will predict unseen examples,depends on what functional form is assumed for the target function (the hypothesis), the size of the training set and the algorithm used to discover it. Once the principal has chosen a representation for the function (a language for representing an hypothesis such as a decision tree, neural network, or linear discriminant function, etc.), learning theory dictates that the generalization error is bounded assuming that the true function can be represented using this language. For example, linear discriminant analysis assumes that the true function is linear.

The success of each algorithm operating on this language depends on how well it uses the information presented by the sample and how well it trades off the estimate of the generalization error with the empirical error. For example, decision tree algorithms achieve this balance by first minimizing the empirical error and then intelligently pruning leaves until it reaches some balance between the two types of errors. A very successful linear discriminant technique called the Support Vector Machines (Cristianini and Shawe-Taylor 2000), achieves this balance by trading off the margin of separation between two classes with respect to the separating hyperplane with a measure of empirical error. It is a well known result that the maximum margin hyperplane, which is the hyperplane far from both classes, minimizes the generalization error (Vapnik 1998). It is however not clear whether these results carry over when the input space is altered by strategic positioning of the agents.

To appreciate the impact of strategic behavior in classification, consider the setting in Chapter 4. We show that “negative” agents try to achieve a positive labeling by changing their true attributes if the cost of doing so doesn’t exceed their reservation cost.

However, the principal, anticipating such strategic behavior, shifts and scales the classification function (a maximum margin hyperplane) such that no true negative agent will benefit from engaging in such behavior. Thus, in practice, the negative agents have no incentive to change their true attributes and, hence, do not exert any effort. However, this shift may leave some marginally positive agents in danger of being classified as negative. Since they too anticipate that the principal will alter the classification function so as to cancel the effects of the expected strategic behavior of the negative labeled agents, they will undertake changes. Thus, the ones who are "penalized" for engaging in a strategic behavior are not the negative agents but rather the marginal positive agents. Moreover, the resulting classifier has a greater margin and better generalization capability (Vapnik 1998) compared to the SVM learning results when we assume X is static. Hence, normal SVM methods can never discover an optimal strategic classifier, even under repeated applications. Such observations have implications not only for induction of classifiers but also for tasks such as policy setting. Figure 2-2 of Chapter 2 illustrates the resulting classifier under strategic behavior. Notice that the margin of the resulting hyperplane is wider and is in fact a scaled and a shifted version of the hyperplane in Figure 2-2 (a) and thus differs from normal SVM results without strategic behavior.

Figure 2-2 (a) shows the sample space without strategic behavior and the corresponding SVM classifier (the continuous line). Given these conditions some negative agents can engage in strategic behavior as indicated by arrows. Figure 2-2 (b) shows the sample space and the resulting optimum classifier given the strategic behavior of agents. Notice that the marginally positive agents are the ones that are forced to move shown by the arrows while the negative agents prefer not to alter their attributes since

they no longer have any incentive of doing so. It costs them too much. Finally, in Figure 2-2 (c), the new classifier with wider margins on each side and the resulting altered instances due to strategic behavior is pictured.

The fact that most of the learning takes place in dynamic and changing environments where there is interaction between agents and the principal or among agents themselves leads us to question the fundamentals of the supervised learning algorithms which mainly operate on fixed but unknown distributions. This static approach to supervised learning, that does not take into account any possible strategic activity in data generation, does not seem to be realistic given the vast array of examples where active manipulation of attributes by actions of agents is realized. Although learning algorithms exist that consider actions of agents (Kaelbling et al.1996), their main concern is to research how agents learn from such interactions so, in that sense, the research framework proposed here differs from those. Here the main concern is not to learn from the mistakes but to prevent mistakes from happening and to improve learning by anticipating behavior that would otherwise cause faulty decisions.

Research Areas

We believe there is an important need to explore various aspects of this new paradigm. In the following we identify some issues that have not been explored yet.

An obvious area of future research is to generalize and extend the methods shown in Chapter 4 and Dalvi et al. (2004) to the other classifiers like decision trees, nearest neighbor, neural networks, etc. Essentially, the problem can be extended appropriately and applied to many other learning algorithms since it is beneficial to take strategic behavior into account since not doing so might be misleading as the current research in the area suggests.

One interesting research angle would be to approach the problem from a game theoretical point of view and to be able to answer questions like "What is the best (optimum) strategy for the principal given the agents' strategic behavior?" Are there any conditions other than those identified by Dalvi et al. (2004) under which these kinds of problems have Nash equilibria? If so, what form do they take, and are there cases where they can be computed efficiently? Under what conditions do repeated games converge to these equilibria?

Since strategic behavior alters the input space it is not clear to what extent the results on learning bounds (Cristianini and Shawe-Taylor 2000, Vapnik 1998) from statistical learning theory apply even when the classifier anticipates this behavior. The results in Chapter 4 suggest that the results may not carry over as is.

One issue of great importance that has not been explored yet is the application of domain knowledge. Each learning technique incorporates domain knowledge differently. So-called kernel methods (such as those used with SVMs) use a nonlinear, higher dimensional mapping of attributes to features to make the classes linearly separable. It can be shown that such a task can be carried-out by kernel functions (hence the name) and the kernel itself can be seen as a similarity measure that is meaningful in that domain. It may be possible to anticipate and cancel the effects of strategic behavior by applying an appropriate kernel mapping. It may be possible to develop such a kernel using agents' utility functions and cost structures since that knowledge is one form of domain specific knowledge.

Existing research in Strategic Learning (i.e., Chapter 4 and Dalvi et al. (2004)) assumes that all agents have the same cost of changing an attribute. Also, it assumes that

agent's have linear utilities. Relaxing these assumptions would be interesting for future research.

Additional areas of future research include the following. It is possible to apply the proposed ideas to cooperative multi-agent systems. For example, if agents can collude and offer side payments to other agents to make sub-optimal moves to confuse and thwart the principal, can this be anticipated in the induction process? Are there conditions under which collusion by the agents will beat a classifier that does not explicitly consider it?

Human beings are often quite good at the level of adaptation (either as a principal or agent) in which we are interested. Good sports players will carefully watch how their opponents play and change strategy based on their opponents' actions. Inspired by such human behavior, we would like to apply Strategic Learning to competitive multi-agent settings where multiple principals/agents interact and try to learn while competing.

Research so far focuses on static situations. If the instance space changes over time (due to some exogenous factors), is it possible to dynamically model user behavior and determine classifiers that will adapt efficiently? For example, allowing new features to be added as time progresses could be a good way to model such a dynamic and interactive environment. The goal would be to make this adaptation as easy and productive as possible.

Current research assumes that principal and agent know each others' parameters. In other words, the principal is well informed about the costs and the problem that the agent faces. When the principal and the agent do not know each other's parameters, how would that affect the optimal strategies and what would be the additional learning needs? This is the issue on which economists focus their analysis of the "principal - agent"

problem where they consider cases in which the principal is less omniscient (Laffont and Martimort 2002).

If the computation of an optimal strategy is too expensive in some cases, would approximate solutions and weaker notions of optimality become sufficient in real-world scenarios? It would be valuable to derive approximately optimal solutions to the Strategic Learning problem.

This framework can also be used to encourage real change rather than preventing negative behavior. This might have applications in public policy problems.

Summary

In this chapter we outline the need for machine learning methods that anticipate strategic actions by the agents over which a principal is inducting a classification rule. For example, credit approval, fraud detection, college admission, bankruptcy detection, spam detection, etc. are all cases involving strategic agents who might try to achieve positive classifications by altering their attributes.

We reviewed two recent studies initializing results on this problem. Dalvi et al. (2004) use a two stage model to produce a superior principal classifier. In our study (Chapter 4), we go a step further. Ideally, using rational expectation theory, one might be able to fully anticipate agent actions and incorporate this in a machine learning induction process to determine dominant classifiers as done in Chapter 4.

After outlining and illustrating the new paradigm, we discussed many potential research areas and questions.

CHAPTER 4
DISCRIMINATION WITH STRATEGIC BEHAVIOR

Introduction and Preliminaries

Strategic Learning

We study the problem where a decision maker needs to discover a classification rule to classify intelligent agents. Agents may engage in strategic behavior and try to alter their characteristics for a favorable classification. We show how the decision maker can induce a classification rule that fully anticipates such behavior. We call this "learning in the presence of self-interested agents" or simply "Strategic Learning."

Suppose $X \subseteq \Re^n$ contains vectors whose n observable components represent values of attributes for rational agents that will be classified by a decision maker (for example, as good credit risks or bad ones). X is partitioned into two sets, a set consisting of cases consistent with some underlying but unknown concept (called the positive examples of the concept) and the remaining cases (a set of negative examples of the concept). For example, in Messier and Hansen (1988) the underlying concept is described by "firms that won't default or go bankrupt" and attributes consists of values such as the firm's ratio of retained earnings to total tangible assets, the firm's ratio of total earnings to total tangible assets, etc. A desired goal is to sample X, determine the true label of each example and infer the concept from these examples. This is a typical task in data mining, machine learning, pattern recognition, etc.

In such situations, the concept of interest is assumed fixed but unknown. The observable relevant attributes of interest are assumed given for our problem. During the

inference process, a decision maker observes instances drawn randomly with replacement from X each having a label -1 or 1 denoting whether it is a negative or positive example of the concept. These labels are not normally observable, but during the inference process we assume that such labels are available, perhaps through extensive study or from past outcomes. The collection of these examples forms the training "set" $S = \left(\left(x_1, y_1 \right), ..., \left(x_\ell, y_\ell \right) \right)$ of ℓ observations where $y_i \in \{-1,1\}$ identifies the label. We assume there are at least two elements of S having opposite labels. (Strictly speaking, S is not a set since there may be duplicate entries).

The decision maker uses the training set to determine an instance of a representation of the concept that we embody in a function $f : X \to \Re$ where $f(x) \geq 0$ if $x \in X$ belongs to the positive class and $f(x) < 0$ if it belongs to the negative class. In the language of learning theory (Vapnik 1988) we are performing "supervised learning" when we infer f from a sample S, each example of which has a known label. The decision maker must choose a general form for f (e.g., a decision tree, a linear function, a neural network, etc.). Depending on the representation chosen for the target concept, one may use inference methods that produce the desired output. For example, for representations such as neural networks (Tam and Kiang 1992), decision trees (Quinlan 1986), discriminant functions (Hand 1981), support vector machines (Cristianini and Shawe-Taylor 2000), etc. many methods have been developed to determine an actual f given a sample S. The representation choice sets the induction bias and the methodology choice determines the quality of the final f found.

It is often the case that a decision maker, heretofore the principal, determines functions such as f to select individuals (as for college entrance, credit approval, etc.) or

to spot early signs of problems in companies (such as bankruptcy, fraud, etc.). We will speak collectively of these yet to be classified individuals or companies as "agents" where agent i is represented by his/her true vector of attributes x_i and true label y_i.

Although the exact nature of f might be unknown to these agents, it is typically the case that the direction of change in an attribute that increases the likelihood of a positive classification is known. For example, it is generally believed that better grades positively influence admission to most universities. Hence, taking actions to improve one's grades will help in achieving a positive labeling by a principal in charge of admission decisions.

Hence, it is reasonable to anticipate that such agents who are subject to classification by a principal using f might attempt to alter superficially their true attribute values so as to achieve a positive classification under f when they may actually belong to the negative class or be only marginally positive. This is not to say that agents need to lie or engage in deceit, although these behaviors could certainly be used to alter the true attribute values. Instead, they might proactively try to change their attribute values prior to their classification by a decision maker. For example, in a college entrance decision, one attribute often used to discriminate is the level of participation in extracurricular activities. An individual could discern this and make an effort to join clubs, etc. merely to increase his/her chance of a positive entrance decision. There are even websites that purport to help one increase his/her scores (i.e., the value that f gives for them). For example, http://www.consumercreditbuilder.com advertised they have ways one can learn how to raise his/her credit scores, even as high as 30% or better. http://www.testprep.com claims "Increases of 100 points on the SAT and 3 to 4 on the ACT have been common, with higher scores often achieved."

http://www.brightonedge.org/ even offers a college admissions camp aimed at increasing the chances of being admitted to college.

We explore this potential strategic gaming and develop inference methods to determine f realizing that strategic behavior may be important in using f. That is, we anticipate strategic behavior in the induction process leading to f.

Suppose the principal can assess the costs to an agent needed to change an attribute value. Some attributes may have a very high cost to change and some a relatively low cost. For example, a potential college applicant might note that the cost of participating in extracurricular activities might be relatively low compared to studying harder to change a grade point average. The latter in the short run may be impossible to alter. The costs of getting caught with lying or deceit might be very high, for example, in fraud cases. Let $c_i > 0$ be the vector of costs to agent i for changing a true vector $x_i \in \Re^n$ to $x_i + Dd_i$ for $d_i \geq 0$ (diagonal matrix D, with diagonal components of +1 and -1, merely orients the moves to reflect directions where increases lead to better scores under f). The cost of such a change to the agent is then $c_i' d_i$. We assume that the reservation cost of being labeled a positive example is r_i for agent i. We further assume a rational agent will engage in strategic behavior if

$$r_i \geq \min_{d_i \geq 0} c_i' d_i \qquad \text{s.t.} \quad f\left(x_i + Dd_i\right) \geq 0 .$$

Thus, we envision a situation where the original instance space, X, is possibly perturbed after f is discovered by the principal, even if f is kept secret. Most induction methods will operate using a sample from X. However, we contend that strategic

behavior will result in a change $X \xrightarrow{f} \bar{X}$ from which future instances will be sampled. This needs to be anticipated.

In this chapter we develop methods to determine linear discriminant classifiers in the presence of strategic behavior by agents. We focus on a powerful induction method known as support vector machines. For separable data sets, we characterize (Theorem 1) an optimal principal induction method that anticipates agent behavior. For non-separable data sets (i.e., data sets where no linear separator exists that can separate the negative from positive examples) we give a general approach. Then, we apply these approaches to a credit-risk evaluation setting. Later, we extend the general approach to a stochastic version of the problem where agent parameters such as c_i and r_i are not known with certainty by the principal. We conclude with a discussion and possible future research in the last section.

Related Literature

Around the time of our first draft of this study, Dalvi et al. (2004) described a scenario where an adversary alters the data of the negative class subject to known costs and utility. They formulate the problem as a game with two players, one named Adversary and the other Classifier, in which the Adversary tries to alter true negative points to mislead the Classifier into classifying them as positive. In our terminology, the Classifier is our principal. Their Adversary is able to control the agent attributes. The authors show that if the Adversary incurs a unit cost for altering an observation then there exists a Nash equilibrium solution for this game. Since, finding a Nash equilibrium is prohibitive in the general case they focus on a one-step game in which they assume that the Classifier publishes a classifier (C_0) before the game and the Adversary modifies the

sample points to fool C_0 but knowing that the Adversary will engage in such behavior the Classifier actually uses a new classifier C_1. The authors focus on the Naïve Bayes Classifier (NBC) and show that updating NBC based on the expectation that the Adversary will try to alter the observations yields better results. They provide an algorithm and an application to the spam filtering problem where spammers (agents) quickly adapt their e-mail tactics (they alter their attribute vector) to circumvent spam filters.

While our general task is somewhat similar to that studied in Dalvi et al. (2004), we outline some major differences between our approaches. In general, we focus on a case where agent attributes belonging to either class can be altered. Dalvi et al.(2004) only consider negative cases. For example, if a marginal positive credit applicant would be labeled as a negative instance by the principal, the agent may use some of the techniques at http://www.consumercreditbuilder.com to increase the chance of a positive labeling. This is a major difference. As we show in Theorem 1 only marginal positive cases must change attributes. Secondly, we use support vector machines as our learning algorithm since it has many nice theoretical properties relating to induction risk and optimization which we discuss below. The linear classifier induced by support vector machines is also very easy to interpret as a scoring function, unlike the Naïve Bayes Classifier. Finally, our formulation is a version of the basic principal/agent problem formulation which inherently represents a two player game with infinite steps. Dalvi et al. only consider a one step game. Since we assume that our observations are actual agents engaging in strategic behavior, our formulation inherently models a multi-agent game in which one principal, many negative agents and many positive agents may modify their behavior.

With the exception of the Dalvi et al. (2004) and our earlier drafts, Strategic Learning approaches have not been used before. However, learning problems involving intelligent agents in a gaming situation have been investigated in other settings. For example, in leader-follower systems a leader (i.e., our principal) decides on and announces an incentive to induce followers (i.e., our agents) to act in a way that maximizes the leader's utility, while the followers maximize their own utilities under the announced incentive scheme. This is analogous in some sense to our setting since the leader tries to identify and announce the ultimate decision rule that would maximize his/her own objective while the followers seek to maximize their own utilities. In both cases, this can be viewed as the leader trying to maximize some kind of a social welfare function given the self interested actions of the followers.

These kinds of decision are termed incentive Stackelberg games (Von Stackelberg 1952) where the leader first determines an incentive function and announces it and the followers, after observing the announced incentive, make their own decisions. For example, Bhattacharyya at al. (2005) apply this kind of a sequential approach to propose a reinforcement based learning algorithm for repeated game leader-follower multi-agent systems. A key point here is the sequential nature of the decisions. Learning takes place progressively as principal and agents interact with each other based on the principles of reinforcement learning (Sutton and Barto 1998) which uses the idea of trial-and-error learning.

In this scenario, the leader tries to learn an optimal incentive based on the cumulative information from the earlier periods while the followers try to learn optimal actions based on the announced incentive. Learning is achieved over successive rounds

with information being carried from one round to the next. This differs from our method in the sense that this sequential approach will often yield suboptimal results while the ultimate solution can only be found by anticipating and incorporating this anticipation in the learning process itself rather than following an after the fact reactive approach.

One other line of research that is closely related to our problem is utility-based data mining (Provost 2005). Due to recent growing demand for solving economical problems that arise during the data mining process, there has been an interest among researchers to explore the notion of economic utility and its maximization for data mining. So far the focus has been on objectives like predictive accuracy or minimization of misclassification costs assuming that training data sets were freely available. However, over time, it may become costly to acquire and maintain data causing economical problems in data mining. Utility-based data mining trades off these acquisition costs with predictive accuracy to maximize the overall utility of the principal. While utility-based data mining is concerned with the principal's utility, Strategic Learning additionally considers the possibility that the objects of classification are self-interested, utility maximizing, intelligent decision making units.

Linear Discriminant Functions

Among the many possible classification methods, linear discriminant functions are the most widely used since they are simple to apply, easy to interpret and provide good results for a wide range of problems (Hand 1981). In this study we restrict the class of functions, f, over which the principal searches, to linear discriminant functions (LDFs). As we will show, this is not restrictive since kernel mappings can be applied for non-linear domains. Many methods exist for finding linear discriminant functions. However, a powerful methodology, support vector machines, has been developed over the last 10

years that builds on statistical learning theory ideas. We discuss the importance of this below. A brief review LDFs is provided first.

Linear Discriminant Methods

Linear discriminant analysis for binary classification is usually performed by first determining a non-zero vector $w \in \Re^n$ and a scalar b such that the hyperplane $w'x + b = 0$ partitions the n-dimensional Euclidian space into two half-spaces. Then, an observed vector x_i is assigned to the positive class if it satisfies $w'x_i + b \geq 0$. Otherwise it is assigned to the negative class. That is, $(w, b): \Re^n \rightarrow \{-1, +1\}$ where +1 denotes the positive class (points giving $w'x_i + b \geq 0$) and -1 denotes the negative class.

Fisher (1936) was the first to introduce linear discriminant analysis seeking a linear combination of variables that maximizes the distance between the means of the two classes while minimizing the variance within each class. He developed methods for finding w and b from a training set already classified by a supervisor. The results of Fisher were followed by many other approaches to determine (w,b) which mainly differ on the criterion used to make the decision of choice between a number of candidate functions. For instance, some of the statistical approaches focused on making different assumptions about the underlying distribution. For example, logit analysis (Cooley and Lohnes 1971) which is a type of regression, uses a dummy dependent variable which can only have the values 1 or 0 and considers the maximum likelihood methods to estimate w and b. Bayesian methods (Duda and Hart 1973) on the other hand seek the optimum decision rule that minimizes the probability of error.

In determining LDFs, numerous mathematical programming methods have been studied. These distribution free methods were first introduced in (Mangasarian 1965) and

then more actively explored in the 1980's (Freed and Glover 1981a, 1981b, 1982, 1986a, 1986b, Glover 1990, Koehler and Erenguc 1990b). In general these methods attempt to find w and b that optimize directly (or some proxy for) the number of misclassifications. For instance, Freed and Glover (1981) maximized the minimum deviation of any data point from the separating hyperplane. Also, Freed and Glover (1986) focus only on the observations that ended on the wrong side of the hyperplane. They determined a (w,b) that minimized the maximum exterior deviation. Most of these methods exhibited undesirable properties (Markowski and Markowski 1985, Koehler 1989a, 1989b). Different approaches like non-linear (Stam and Joachimsthaler 1989) and mixed integer programming (MIP) (Koehler and Erenguc 1990a) have also been studied. Moreover, models that combine objectives such as minimizing the cost of misclassification along with the number of misclassifications have also been developed (Bajgier and Hill 1982). Heuristic optimization has also been used. For example, Koehler (1991) studied Genetic Algorithms to determine linear discriminant functions.

Often mathematical programs that directly minimize the number of misclassifications resort to some type of combinatorial search method and are computationally expensive and usually impractical. An important exception was given by Mangasarian (1994) who gave three non-linear formulations for solving the misclassification problem using bi-linear constraints to "count" the number of misclassifications. Good experimental results have been observed (Bennett and Bredensteiner 1997).

However, the problem with these and other learning approaches is that they have focused and depended on just the training data set such that the hypothesis correctly

classified the data on the training set but made essentially poor predictions on the unseen data. That is, they typically would over-fit during the induction process at the later expense of generalization. This is primarily due to the fact that these approaches were based on procedures that directly minimize the classification error (or a proxy for classification error) over a training data set.

Statistical learning theory (Vapnik 1998, 1999) attempts to overcome this problem by trading-off this potential over-fitting in training with generalization ability. This fact makes the theory a powerful tool for theoretical analysis and also a strong basis for practical algorithms for estimating multidimensional discriminant functions. We build our Strategic Learning model along these lines. In the following section, we provide a brief overview of these statistical learning theory results which form the basis for support vector machines. We follow this with a brief discussion on support vector machines.

Statistical Learning Theory

Statistical learning theory (Vapnik 1998, 1999) provides a solid mathematical framework for studying some common pitfalls in machine learning such as over-fitting. Assuming that x is an instance generated by sampling randomly and independently from an unknown but fixed probability distribution, the learning problem then consists of minimizing a risk functional represented by the expected loss over the entire set of instances. Since the sampling distribution is unknown the expected loss cannot be evaluated directly and some induction principle must be used. However, a training set of instances is available.

Many approaches use the empirical risk minimization principle which infers a function using the training set by minimizing the empirical risk (which is usually measured as the number of misclassifications in the training data set). The empirical risk

minimization principle often leads to over-fitting of data. That is, it often discovers a function that nicely discriminates on the training set but can not do better than chance (or even worse) on as yet unseen points outside the training set. This has been observed in many studies. For example, Eisenbeis (1987) critiques studies based on such over-fitting.

Statistical learning theory approaches this problem by using the structural risk minimization principle (Vapnik 1999, Schölkopf and Smola 2001). It has been shown that, given S, for any target function with a probability at least $1-\eta$ the risk functional can be bounded by the sum of the empirical risk and a term largely capturing what is called the structural risk (see Vapnik (1999) for details). The structural risk is a function of the number of training points, ℓ, the target confidence level, η, and the capacity, h, of the target function. The capacity, h, measures the expressiveness of the target class of functions. In particular, for binary classification, h is the maximal number of points (k) that can be separated into two classes in all possible 2^k ways using functions in the target class of functions. This measure is called the VC-dimension and the size of the training set is required to be proportional with this quantity to ensure good generalization. For linear discriminant functions, without additional assumptions, the VC-dimension is $h = n + 1$ (Cristianini and Shawe-Taylor 2000,Vapnik and Chervonenkis 1981). A common assumption added for certain learning situations is that $x \in X \subset \Re^n$ implies $\|x\| \le R$. This is called the boundedness assumption. A class of linear discriminant functions of the form $y(x)(w'x + b - \Delta) \ge 0$ with $\|w\| = 1$ is termed $\Delta - \text{margin}$ LDFs. Under the boundedness assumption, the VC-dimension, h, for $\Delta - \text{margin}$ LDFs, is bounded above by $1 + \min\left(n, \lceil R^2/\Delta^2 \rceil\right)$ and may be much smaller than n+1. Support

Vector Machines determine an LDF by directly minimizing the theoretical bound on the risk functional. The bound is improved by decreasing the VC-dimension, so that for $\Delta - \text{margin}$ LDFs one can focus on minimizing R^2 / Δ^2. The following section contains a brief review of this methodology.

Support Vector Machines

Support Vector Machines (SVMs) offer a powerful method for discovering linear discriminant functions by directly minimizing a theoretical bound on the risk functional. Having its primary motivation of minimizing a bound on the generalization error distinguishes SVM approaches from other popular methods such as neural networks which use heuristic methods to find parameters that best generalize. In addition, SVM learning is theoretically guaranteed to find an optimum concept (since the induction problem reduces to a quadratic, convex minimization problem) which marks a distinction between this system and most other learning methods. Neural networks, decision trees, etc. do not carry this guarantee often leading to local minima and an accompanying plethora of heuristic approaches to find acceptable results. For example, motivated by the Ockham's razor principle, most decision tree induction and pruning algorithms try to create the smallest tree that produces an acceptable training error in the hopes that smaller trees generalize better (Quinlan 1996). Unfortunately, there is no guarantee that the tree produced by such heuristics minimize generalization error. SVM algorithms also scale-up to very large data sets and have been applied to problems involving text data, pictures, etc.

There are several SVM models. The first model is the so-called maximal margin classifier model. When the training set is linearly separable, this model determines linear discriminant functions by solving

$$\underset{w,b}{Min}\ w'w$$
$$s.t.\quad y_i(w'x_i + b) \geq 1 \qquad i = 1,...,\ell$$

As discussed below, this formulation produces a maximal margin hyperplane with a geometric margin equal to $\Delta = 1/\|w\|_2$. (Note, we show the 2-norm being used here. Other norms can also be used as we discuss later.)

In general, for many real world problems the sample space may not be linearly separable. When the data are not separable the SVM problem can be formulated with the introduction of margin slack variables as follows:

$$\underset{\substack{\xi_i \geq 0 \\ w,b}}{Min}\ w'w + C\sum_{i=1}^{\ell}\xi_i$$
$$s.t.\quad y_i(w'x_i + b) \geq 1 - \xi_i \qquad i = 1,...,\ell$$

where ξ_i is the margin slack variable measuring the shortfall of a point in its margin from the hyperplane and C is a positive parameter. C is chosen to trade-off between margin maximization and training error minimization. This formulation is termed the soft-margin SVM (Cristianini and Shawe-Taylor 2000).

In the first model, where the data are separable, the objective function minimizes the square of the norm of the weight vector, w. It can be shown that this is equivalent to maximizing the geometric margin when the functional margin of the hyperplane is fixed to 1 (Cristianini and Shawe-Taylor 2000.) as follows. For some x^+ and x^- the (linearly separable) SVM is guaranteed to find an optimal weight vector which will satisfy

$w'x^{+} + b = 1$ and $w'x^{-} + b = -1$

so the margin is then the half of the distance between x^{+} and x^{-} which is

$$\Delta = \frac{1}{2}\left(\frac{w'x^{+}}{\|w\|} - \frac{w'x^{-}}{\|w\|} \right) = \frac{1}{\sqrt{w'w}}.$$

Thus, minimizing $w'w$ is the same as maximizing the margin which, in turn, minimizes the $\lceil R^2 / \Delta^2 \rceil$ term bounding the VC-dimension. The non-separable case trades-off the margin with the margin shortfall. By minimizing a quadratic objective function with linear inequality constraints, SVMs manage to escape the problem of local optima faced by other learning methods since the problem becomes a convex minimization problem having a global optimal solution.

It is also possible to employ kernel mappings in conjunction with SVMs to learn non-linear functions. A kernel is an implicit mapping of input data onto a potentially higher dimensional feature space. The higher dimension feature space improves the computational power of the learning machine by implicitly allowing combinations and functions of the original input variables. These combinations and functions of original input variables are usually called features while the original input variables are called attributes. For example, consider some financial attributes like total debt and total assets. If a properly chosen kernel is used, it will allow a debt ratio (total debt to total assets) to be examined as well as the total debt and total assets and, potentially, has more informational power than total debt and total assets variables used alone. Thus, a kernel allows many different relationships between variables to be simultaneously examined. SVM finds a linear discriminant function in the feature space, which is usually then

nonlinear in the attribute space. It sometimes happens that a kernel mapping can map data not linearly separable in the attribute space to a linearly separable feature space.

There are many classes of generic kernels (Genton 2001) but also kernels are developed for specific application areas such as text recognition. For the purposes of this dissertation, we assume a kernel has already been applied to the initial data so that we may treat these features as primitive attributes. We thus focus our research on finding LDFs under strategic behavior leaving the related complications introduced by kernels to future research. In the next section we characterize the induction process for finding an optimal SVM solution to the strategic LDF problem.

Learning while Anticipating Strategic Behavior: The Base Case

In this section we start by introducing the agent's "strategic move problem" that shows how a rational agent will alter his/her true attributes if he/she knew the principal's classification LDF. We then turn to the simplest version of the Strategic Learning problem and derive a complete characterization for the principal's strategic LDF. The base problem is generalized in subsequent sections.

The Agent Problem

If the principal's classification function ($w'x_i + b - 1$) was known to the rational agents, they would solve what we call the strategic move problem to determine how to achieve (or maintain) positive classification under the principal's LDF at minimal cost to themselves. The problem is

$$\begin{aligned} &\min c_i' d_i \\ &s.t. \quad w'\left[x_i + D(w) d_i\right] + b \geq 1 \\ &\qquad d_i \geq 0 \end{aligned}$$

where $D(w)$ is a diagonal matrix defined by

$$D(w)_{j,j} = \begin{cases} 1 & w_j \geq 0 \\ -1 & w_j < 0 \end{cases}.$$

If feasible, this problem determines a minimal cost change of attributes, $D(w)d_i$, needed to be classified as a positive case. This would be undertaken if this cost doesn't exceed the agent's reservation cost, r_i. Since this optimization problem has only one constraint (other than non-negativity constraints), the following can be determined. For non-zero w, let k satisfy

$$j^* = \arg \min_{j, w_j \neq 0} \frac{(c_i)_j \max\left(0, 1-\left(b + w'x_i\right)\right)}{\left|w_j\right|}$$

then for

$$z_i^*(w,b) = \frac{\max\left(0, 1-\left(b + w'x_i\right)\right)}{\left|w_{j^*}\right|}$$

we have

$$d_i^*(w,b) = \begin{cases} z_i^*(w,b)1_{j^*} & \text{if } (c_i)_{j^*} z_i^*(w,b) \leq r_i \\ 0 & \text{otherwise} \end{cases}$$

$z_i^*(w,b)$ can be interpreted as the projection of the amount of modification that the agent needs to make on the $j^{*\text{th}}$ attribute to achieve positive classification with respect to the (w,b) that the principal chooses. For w equal to zero or infeasible strategic move problems, set $d_i^*(w,b) = 0$. An infeasible move occurs when $c_{j^*} z_i^*(w,b) > r_i$. Notice that if the ratio $(c_i)_{j^*} / \left|w_{j^*}\right|$ is the same for different values of j^*, the agent problem has alternate optimal solutions. That is, the agent will be indifferent between multiple j^*

values corresponding to moving in different optimal directions (or convex combinations thereof).

Notice that it might be possible for some attributes to be correlated with each other. For instance, some of the variables of the agent's optimization problem can be linearly dependent. This can be formulated by expressing dependent variables in terms of independent variables. For example, the agent's problem would be

$$\begin{aligned} &\min c_i' d_i \\ &st \quad w'\left[x_i + D(w) A_i d_i\right] + b \geq 1 \\ &\quad\quad d_i \geq 0 \end{aligned}$$

where $A_i d_i$ captures the linear relationships. Solving gives

$$A_i d_i^*(w,b) = \begin{cases} Z_i(w,b) 1_k & if\ (c_i)_k\, Z_i(w,b) \leq r_i \\ 0 & otherwise \end{cases}$$

Consequently, this effects $d_i^*(w,b)$ such that it might be a combination of movements in different directions since changing one attribute may cause others to change. This merely complicates the presentation but does not substantively affect the results. For this reason, we assume linear independence between the attributes for simplicity.

The Base Case

We start by studying the simplest version of the principal's Strategic Learning problem. We assume:

- all agents have the same reservation and change costs (i.e., $r_i = r$ and $c_i = c$).
- $S = \left((x_1, y_1), ..., (x_\ell, y_\ell)\right)$ is linearly separable

These assumptions are removed in the next section. Using SVM while ignoring strategic behavior would be accomplished by solving

$$P1: \quad \min_{w,b} w'w$$
$$s.t. \quad y_i \{w'x_i + b\} \geq 1 \qquad i = 1,...,\ell$$

Under Strategic Learning the principal anticipates any possible agent actions and solves the following.

$$P2: \quad \min_{w,b} w'w$$
$$s.t. \quad y_i \left\{w'\left[x_i + D(w)d_i^*(w,b)\right] + b\right\} \geq 1 \qquad i = 1,...,\ell$$

This problem is no longer a nice convex optimization problem with linear constraints. The constraints are not even piecewise linear convex and/or concave (as we show in the next section).

Nonetheless, the following result characterizes an optimal solution to the principal's problems under strategic behavior by agents. For ease of presentation the proof is in the Appendix A.

Theorem 1

(w^*, b^*) solves P1 if and only if $\left(\frac{2}{2+t^*}w^*, \frac{2b^* - t^*}{2+t^*}\right)$ solves P2 where t^* is given by $t^* = r \max_j |w_j^*| / c_j$.

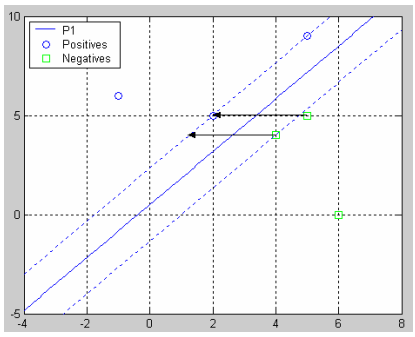
Theorem 1 states that a principal anticipating strategic behavior of agents all having the same utilities and cost structures will use a classifier that is parallel to the SVM-LDF (w^*, b^*) determined without taking into consideration strategic behavior. This function is a scaled (by $2/(2+t^*)$) and shifted form of the original so the objective of P2 is strictly

smaller than P1's meaning that the margin is greater and that the probability of better generalization is greater. The scaling and shift of the hyperplane depend on the cost structure for altering attribute values, the reservation cost for being labeled as a positive case, and $\left(w^*, b^*\right)$. For example, suppose we have the following training set:

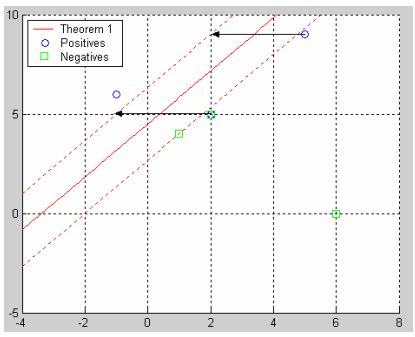
Positive cases ($y_i = 1$): $x_1 = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$, $x_2 = \begin{bmatrix} 5 \\ 9 \end{bmatrix}$, $x_3 = \begin{bmatrix} -1 \\ 6 \end{bmatrix}$,

Negative cases ($y_i = -1$): $x_4 = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$, $x_5 = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$ and $x_6 = \begin{bmatrix} 6 \\ 0 \end{bmatrix}$

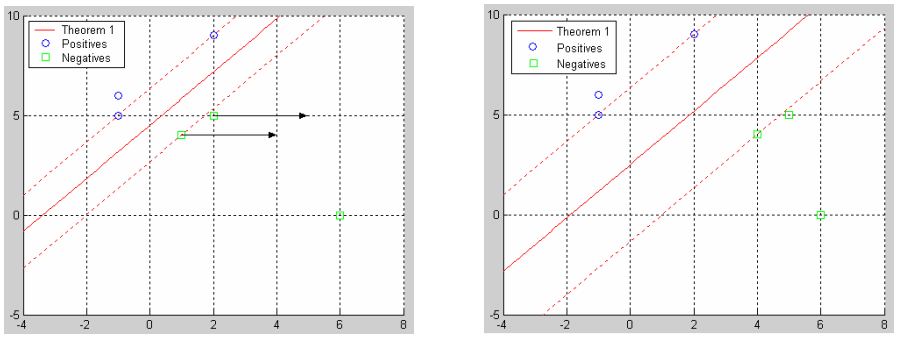
and that $c = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and $\mathrm{r} = 3$. Solving P1 gives $w^* = \begin{bmatrix} -0.7272 \\ 0.5454 \end{bmatrix}$ and $b^* = -0.272727$. Theorem 1 shows that the optimal LDF under strategic behavior is $w = \begin{bmatrix} -0.34783 \\ 0.26087 \end{bmatrix}$ and $b = -0.65217$. Figure 4-1 shows this, the margins and the moved points.



(a)



(b)



(c) (d)

Figure 4-1. Theorem 1. (a) A normal SVM LDF would result in two negative points changing an attribute enough to be classified as positive. (b) Theorem 1 shifts the LDF causing two positive points to move to stay classified as positive. (c) The two negative points gain nothing by moving so would not do so (thus figuratively returning to their original positions). (d) The final LDF with wider margins reflects these anticipated steps.

In our setting, the true "negative" agents try to achieve a positive labeling by changing their true attributes if the cost of doing so doesn't exceed their reservation cost. However, an astute principal, anticipating such strategic behavior, shifts the hyperplane such that no true negative agent will benefit from engaging in such behavior. Thus, in practice, the negative agents have no incentive to change their true attributes and, hence, will not exert any effort. However, the agents who are marginally positive are now in danger of being classified as negative. Since they too anticipate that the principal will alter the discriminant function so as to cancel the effects of the expected strategic behavior of the negative labeled agents, they will undertake changes. Thus, roughly speaking, the ones who are "penalized" for engaging in a strategic behavior are not the negative agents but rather the marginal positive agents. So, the final discriminant function leads to a bigger gap between the two classes of points and produces a separation. Figure 4-1 illustrates these points.

In fact, Theorem 1 shows that the principal is pushing marginal positive agents to alter their attributes in part to gain better generalization results. The new margin need not

be this large merely to keep negative agents from altering their attributes to gain a positive classification. So the principal is left with a trade-off between forcing marginal positive agents to make large changes and possibly increasing the generalization error of the induced LDF. More specifically, the new margin is

$$\frac{1}{\|w^*\|}\left(\frac{t^*}{2}+1\right).$$

Any margin greater than $t^*/2$ will label negative strategic agents as negative. So we can choose parallel hyperplanes giving geometric margins between $\frac{1}{\|w^*\|}\left(\frac{t^*}{2},\frac{t^*}{2}+1\right]$. Theorem 1, giving an optimal solution to P2, yields the largest margin. A principal might elect to choose a smaller margin to spare marginal positive cases the extra effort needed to still be labeled positive.

A separate line of thought might argue for the maximal margin used in Theorem 1. Since positive agents are unaware of the locations of negative agents, they may act to move maximally to ensure their positive labeling.

What if Theorem 1 (or P2) isn't used but rather iterated forms of P1. Interestingly, if a normal, non-strategic SVM were applied to the new instance space, $\bar{\mathrm{X}}$ (i.e., resulting from $\mathrm{X}\underset{\mathrm{f}}{\rightarrow}\bar{\mathrm{X}}$), it may not produce the same result with the altered versions of the original sample as P2. Suppose that instead of using the shifted classifier that incorporates the effects of strategic behavior, the principal chooses to use one that is found by the normal SVM algorithm and updates the classifier every round after obtaining observed attributes (now reflecting strategic behavior). For example, solving $P1$ for the sample formed by

the above points yields $w^* = \begin{bmatrix} -0.7272 \\ 0.5454 \end{bmatrix}$ and $b^* = -0.272727$. Realizing that the principal will use the hyperplane (w^*, b^*) in the first round, agents will adjust their attribute vectors accordingly. Calculating $d_i^*\left(w^*, b^*\right)$ for these points we get

$d_1(w^*, b^*) = 0$, $d_2(w^*, b^*) = 0$, $d_3(w^*, b^*) = 0$,

$d_4(w^*, b^*) = \begin{bmatrix} 2.7502 \\ 0 \end{bmatrix}$, $d_5(w^*, b^*) = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$ and $d_6(w^*, b^*) = 0$.

As a result of strategic behavior, the principal will observe the following perturbed sample after the first round

$$x_1 = \begin{bmatrix} 2 \\ 5 \end{bmatrix},\ x_2 = \begin{bmatrix} 5 \\ 9 \end{bmatrix},\ x_3 = \begin{bmatrix} -1 \\ 6 \end{bmatrix},\ x_4 = \begin{bmatrix} 1.2498 \\ 4 \end{bmatrix},\ x_5 = \begin{bmatrix} 2 \\ 5 \end{bmatrix} \text{ and } x_6 = \begin{bmatrix} 6 \\ 0 \end{bmatrix}.$$

Note that x_1 and x_5 are the same point so this set is not separable. In the second round the principal will adjust the hyperplane using the normal SVM solution with the perturbed sample observed after the first round. However, note that the sample observed after the first round is no longer linearly separable so the principal will use the soft margin SVM (say with $C_1 = 1$ and $C_{-1} = 5$ reflecting the fact that there is a greater penalty for mislabeling a negative case). Solving for (w^*, b^*) for the second round yields

$w^* = \begin{bmatrix} -0.4 \\ 0.8 \end{bmatrix}$ and $b^* = -4.2$.

Notice that $c_j / \left|w_j\right|$ is the same for $j = 1, 2$, so the agent problem has alternate optima. That is, the agent can choose to alter any of the two attributes since they give the same objective value $\min c' d_i = 5$. For simplicity, we assume that all agents will choose to move in the smallest indexed attribute in case of alternate optima.

The agents who engaged in strategic behavior in the first round had incurred some cost causing a reduction in their reservation values. If we calculate the residual reservation cost left for each agent after the first round we get $r_1 = 3$, $r_2 = 3$, $r_3 = 3$, $r_4 = 0.2498$, $r_5 = 0$ and $r_6 = 3$.

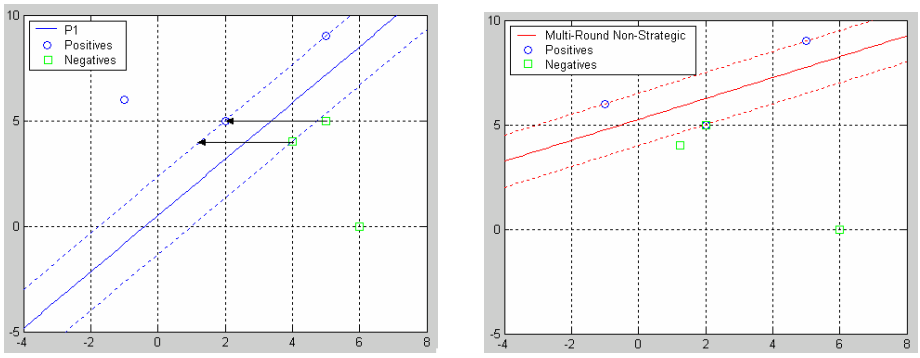
Calculating $d_i^*(w^*, b^*)$ for these points for the second round, we get

$d_1(w^*, b^*) = 0$, $d_2(w^*, b^*) = 0$, $d_3(w^*, b^*) = 0$,

$d_4(w^*, b^*) = 0$, $d_5(w^*, b^*) = 0$ and $d_6(w^*, b^*) = 0$.

Thus, the sample after the second round stays the same and solving for (w^*, b^*) for the second round yields the same hyperplane. None of the points in the sample satisfy the constraint $c_k z_i^*(w, b) \le r$ so the strategic behavior ends in the second round.

Figure 4-2 displays the hyperplane that solves P1 and the final hyperplane of the iterative approach. The optimal hyperplane of Theorem 1 is superior to the resulting hyperplane of the iterative approach in the sense that it prevents the movements of all negative instances while the hyperplane of the iterative approach still can't prevent misclassifications to occur as a result of strategic behavior.



(a) (b)

Figure 4-2. Multi-round non-strategic SVM. (a) shows the movement of negative points after the SVM LDF is announced. (b) shows a new SVM based on these moved points.

Remark: Theorem 1 is similar to Theorem 2.1 in Dalvi et al. (2004). Theorem 1 shows that if we knew the true class membership of each point before the game started, it is possible to create a classifier that has the same performance when strategic behavior (i.e., gaming) is present. Moreover, Theorem 1 characterizes the optimal strategy of each type of agent as well as the principal; something that has been omitted in Dalvi et al. (2004). It also shows that under rational expectations the positive instances also have to change their behavior. In essence, they are the only ones that end up modifying their behavior.

Learning while Anticipating Strategic Behavior: The General Case

In this section we assume that agents may have their own reservation and change costs and that the data set may not be separable. Refreshing the notation in the Strategic Learning setting, each agent i has a true vector of attributes x_i, a true label y_i, reservation cost r_i and a vector of costs for modifying attributes c_i. Reservation cost can be viewed as the maximum effort that an agent is willing to exert in order to be classified as a positive agent. On the principal's side, C_{y_i} is the penalty associated with the margin shortfall of an agent of true type y_i. (Other schemes can be used to price the margin shortfall of sub-categories of these cases, but we forego the extra notational burden to show this). Towards that end, in the absence of Strategic Learning, we would solve the soft-margin form of SVM. The straightforward modification of the soft-margin SVM to handle Strategic Learning is as follows.

First, let $q_i(w,b)$ be the amount of bias that an agent i can introduce to the principal's classification function $w'x_i + b - 1$ by engaging in strategic behavior. As discussed in Section 3, a principal may want to trade-off some confidence in the

generalization error of the induced LDF for lowered effort needed by positive agents to stay positive. Later we argue that a reasonable way for the principle to implement this is to penalize agent effort by adding

$$\lambda \sum_{y_i=1} q_i$$

where $0 < \lambda < C_{+1}$ to the soft-margin objective. With this, the general model for Strategic Learning is:

$$P3: \quad \min_{w,b} w'w + \sum_{i=1}^{\ell} C_{y_i} \xi_i + \lambda \sum_{y_i=+1} q_i(w,b)$$
$$s.t. \quad y_i \left\{ w'x_i + q_i(w,b) + b \right\} \geq 1 - \xi_i \qquad i = 1,...,\ell$$

where

$$q_i(w,b) = \begin{cases} 0 & if \quad 1 - b - w'x_i < 0 \\ 0 & if \quad 1 - b - w'x_i > z_i \\ 1 - b - w'x_i & otherwise \end{cases}$$

and

$$z_i \equiv r_i \max_{j \ni (c_i)_j \neq 0} \frac{|w_j|}{(c_i)_j}$$

for $c_i \geq 0$ with at least one j satisfying $\infty > (c_i)_j > 0$.

In the next section, we study this problem and show it is not mathematically well-posed but may be modified to produce an epsilon-optimal formulation. Following that section, we develop a mixed integer quadratic program and a mixed integer linear program (for the 1-norm counterpart) for solving P3.

Properties of P3

Let

$$f(w,b) = m(\|w\|) + \sum_{i=1}^{\ell} C_{y_i}\xi_i(w,b) + \lambda \sum_{y_i=+1} q_i(w,b)$$

where

$$\xi_i(w,b) = \max\left(0, 1 - y_i\left[w'x_i + q_i(w,b) + b\right]\right)$$

and $\lambda \geq 0$, $C_{y_i} > 0$ with $\|w\|$ being a norm of w and $m(\ )$ an increasing function with $m(0) = 0$. We are interested in minimizing $f(w,b)$ which will be formulated as a mixed integer program when $m(\|w\|) = \sum_j |w_j|$ and a quadratic mixed integer program when $m(\|w\|) = w'w$. The following table illustrates the cases that result depending on an agent's functional distance to the positive margin $(1 - b - w'x_i)$.

Table 4-1. Possible cases depending on z_i

Case	y_i	$1 - b - w'x_i$	$q_i(w,b)$	$\xi_i(w,b)$
1	-1	< 0	0	$1 + b + w'x_i$
2	-1	$\in [0, z_i]$	$1 - b - w'x_i$	2
3a	-1	$> z_i$ and < 2	0	$1 + b + w'x_i$
3b	-1	$> z_i$ and ≥ 2	0	0
4	1	< 0	0	0
5	1	$\in [0, z_i]$	$1 - b - w'x_i$	0
6	1	$> z_i$	0	$1 - b - w'x_i$

Now, we define

$$f(b \mid w) = m(\|w\|) + \sum_{i=1}^{\ell} C_{y_i} \xi_i (b \mid w) + \lambda \sum_{y_i=+1} q_i (b \mid w)$$

which is the total cost function $f(w,b)$ when w is fixed. Let $b(w)$ be an optimal solution to $\min_b f(b \mid w)$. There are four cases involved with determining an optimal $b(w)$. The following Figures illustrate these cases for the following two points:

Positive case ($y_i = +1$): $x_p = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$,

Negative case ($y_i = -1$):, $x_n = \begin{bmatrix} -1 \\ 6 \end{bmatrix}$

for $w = \begin{bmatrix} -0.8 \\ 0.6 \end{bmatrix}$, $c = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $C_{+1} = 1$ and $C_{-1} = 5$.

Figure 4-3 shows how the costs $C_{+1} \max(0, 1 - b - q_i(w,b) - w' x_i) + \lambda q_i(w,b)$ vary with b for a positive agent. Figure 4-4 shows the same when $\lambda = 0$.

Figure 4-5 shows how the costs $C_{-1} \max(0, 1 + b + q_i(w,b) + w' x_i)$ vary for a negative agent.

Figure 4-6 shows this when the agent does not have a high enough reservation to cover all positions within the margin. Notice it has a similar (though reversed) graph as a positive agent with $\lambda > 0$.

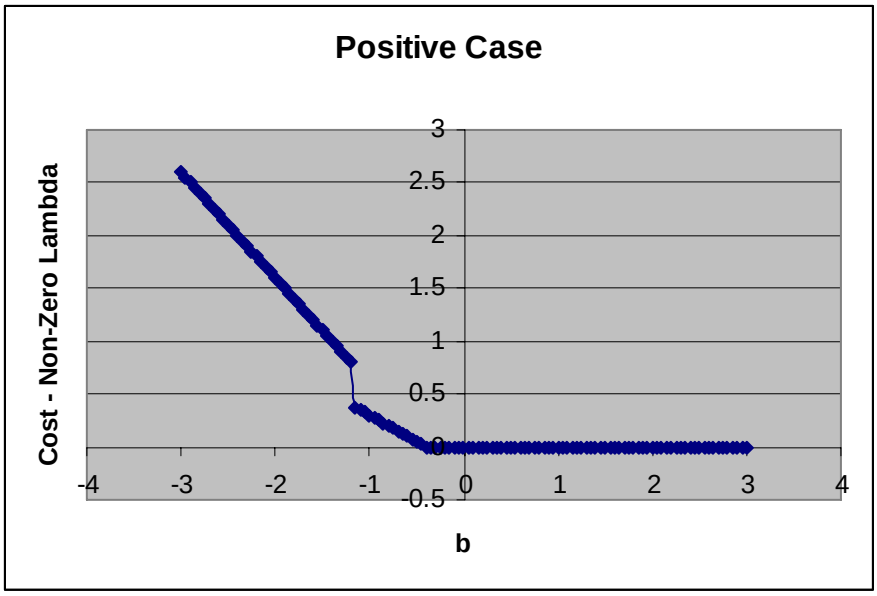


Figure 4-3. Positive case with $r_i = 1$ and $\lambda = 0.5$.

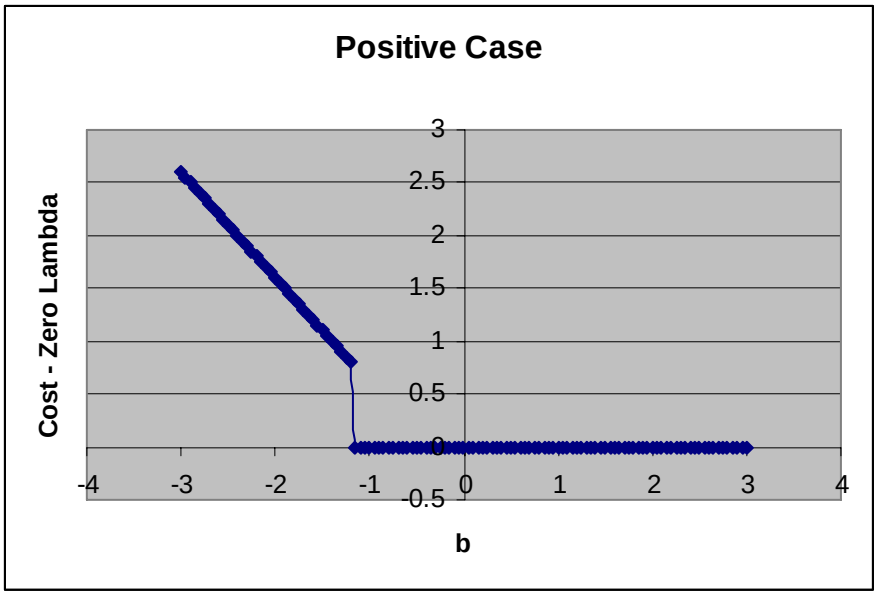


Figure 4-4. Positive case with $r_i = 1$ and $\lambda = 0$.

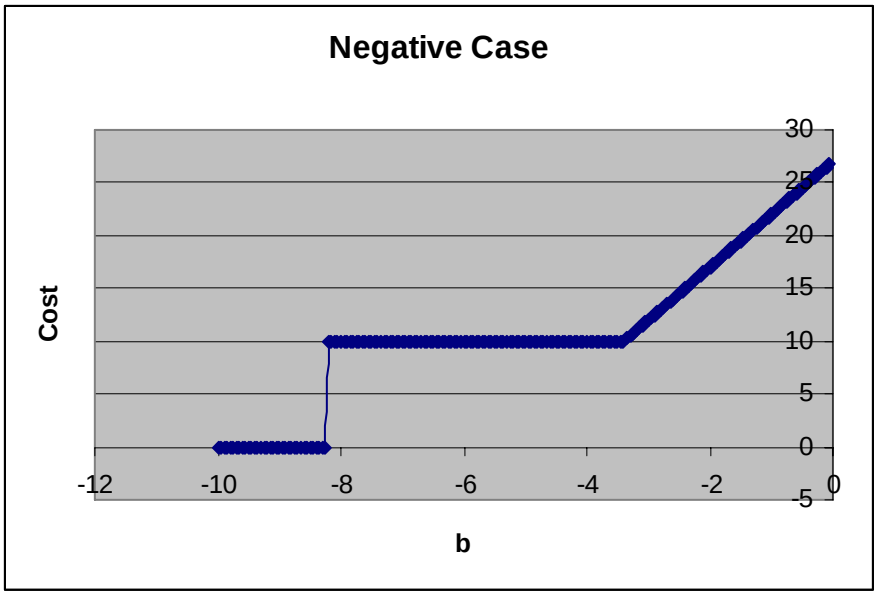


Figure 4-5. Negative case with $r_i = 6$.

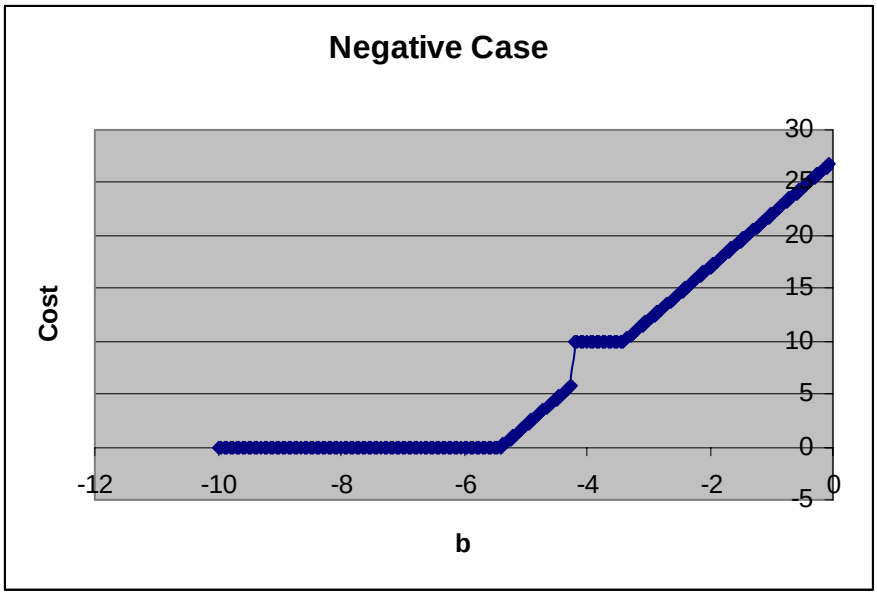


Figure 4-6. Negative case with $r_i = 1$.

Table 4-2 gives the regions shown in these cases.

Table 4-2. Different regions of costs

Positive Cases		Negative Cases	
$\lambda = 0$	$\lambda > 0$	$z_i < 2$	$z_i \geq 2$
$(-\infty, 1 - z_i - w'x_i)$	$(-\infty, 1 - z_i - w'x_i)$	$(-\infty, -1 - w'x_i]$	$(-\infty, 1 - z_i - w'x_i)$
$[1 - z_i - w'x_i, \infty)$	$[1 - z_i - w'x_i, 1 - w'x_i)$	$[-1 - w'x_i, 1 - z_i - w'x_i)$	
	$[1 - w'x_i, \infty)$	$[1 - z_i - w'x_i, 1 - w'x_i)$	$[1 - z_i - w'x_i, 1 - w'x_i)$
		$[1 - w'x_i, \infty)$	$[1 - w'x_i, \infty)$

As a typical graph of $f(b \mid w)$ where all negative and positive points are combined, we see graphs like Figure 4-7 which was produced using the following six points.

Positive cases ($y_i = +1$): $x_1 = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$, $x_2 = \begin{bmatrix} 5 \\ 9 \end{bmatrix}$, $x_3 = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$,

Negative cases ($y_i = -1$): $x_4 = \begin{bmatrix} -1 \\ 6 \end{bmatrix}$, $x_5 = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$ and $x_6 = \begin{bmatrix} 6 \\ 0 \end{bmatrix}$

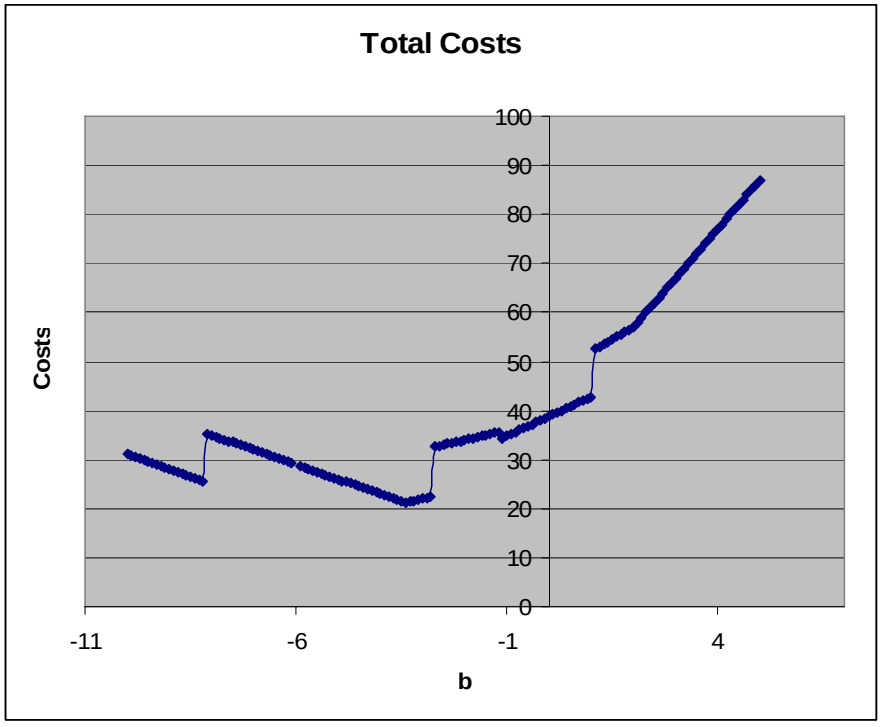


Figure 4-7. A typical graph of $f(b \mid w)$ for $w = \begin{bmatrix} -0.8 \\ 0.6 \end{bmatrix}$, $c = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $C_{+1} = 1$, $C_{-1} = 5$, $r_i = 1$ and $\lambda = 0.5$.

Clearly $f(b \mid w)$ is not any form of convexity (like quasiconvex). Furthermore, there are points of discontinuity where the function is upper semi-continuous. When trying to find a minimizing point for $f(b \mid w)$, these points of discontinuity pose a problem when $\limsup_{y \to b} f(y \mid w) < f(b \mid w)$ which is caused by negative agents. Figure 4-8 shows three cases that may arise with linear upper semi-continuous functions. Here $b = 2$ is a point of discontinuity and all three cases have an open interval adjoining it to the left. In the first case, $b - \varepsilon$ is actually an optimal point (because there are multiple optima and any point in $[1, 2)$ is optimal. The remaining two cases have no optimal point in $[1, 2)$ so $\min_b f(b \mid w)$ is not a well-posed problem.

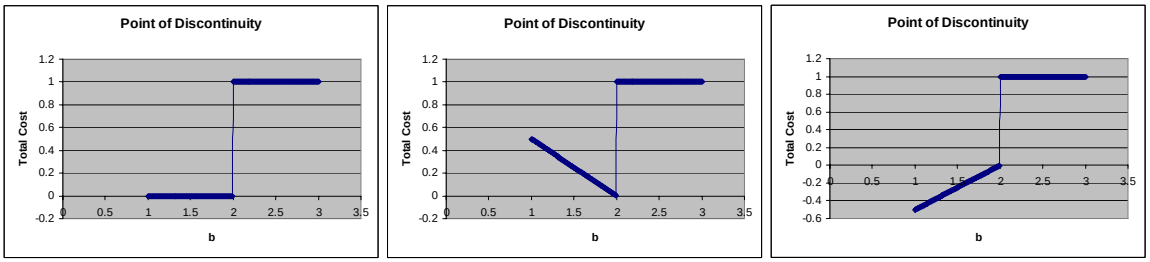


Figure 4-8. Possible cases for points of discontinuity of $f(b \mid w)$.

In the latter two cases we consider the point $b - \varepsilon$ as optimal (assuming it has a lower cost than at b) and call it an epsilon-optimal point of the neighborhood. We denote this problem as

$$\varepsilon - \min_b f(b \mid w)$$

and a solution as $b(w)$. In the next section we reformulate P3 to produce epsilon optimal solutions.

Strategic Learning Model

In this section we develop mixed integer models of the (epsilon) Strategic Learning model. To evaluate

$$r_i \max_j \frac{|w_j|}{(c_i)_j}$$

we introduce binary variables, H, to get

$$r_i \frac{|w_j|}{(c_i)_j} \le z_i \le r_i \frac{|w_j|}{(c_i)_j} + MH_{i,j} \qquad j = 1,...,n$$
$$\sum_j H_{i,j} = n-1$$

Here, for each i, one $H_{i,j}$ must be zero so $z_i \le r_i \frac{|w_j|}{(c_i)_j}$ for only one j. With this, the lower bound forces z_i to be the maximal value. These constraints need only appear for the K different cost vectors ($c_i = v_k$). That is, although there are ℓ agents there may only be K unique cost vectors (Theorem 1 assumes all have the same cost vector so K = 1).

Let q_i be a decision variable representing $w'D(w)d_i(w,b)$. The cost of moving to a positive agent is $(c_i)_{j^*} q_i / |w_{j^*}|$ where

$$j^* = \arg\max_j \frac{|w_j|}{(c_i)_j}.$$

This can be evaluated by

$$(c_i)_{j^*} q_i / |w_{j^*}|.$$

However, adding

$$\lambda \frac{\sum_{y_i=1} (c_i)_{j^*} q_i}{\left| w_{j^*} \right|}$$

to the SVM objective for a positive λ would yield a non-positive definite objective. So, instead, as discussed earlier, we use a proxy for the agent effort of

$$\lambda \sum_{y_i=1} q_i$$

where $0 < \lambda < C_{+1}$.

To evaluate absolute values of components of w the usual trick of finding absolute values by

$$\min_{s_k \geq 0} \sum_j s_j$$
$$s.t. \quad -s \leq w \leq s$$
other constraints

won't work because our objective function has terms whose values will be impacted by minimizing the sum of the s variables. Hence we introduce another vector of binary variables, I, and the following constraints to handle absolute values.

$$w = w^+ - w^-$$
$$MI_j \geq w_j^+ \geq 0 \qquad j = 1, \ldots, n$$
$$M - MI_j \geq w_j^- \geq 0 \qquad j = 1, \ldots, n$$

We need to determine the q_i variables. No agent will exert effort to adjust their attributes if the effort does not yield a positive classification. Conversely, if exerting effort (not exceeding the reservation limit) will result in a positive classification, then an agent who would otherwise be classified as negative will exert the effort. Consider the case of a negative agent. We replace

$$y_i\left\{w'\left[x_i + D(w)\,d_i^*(w,b)\right] + b\right\} \geq 1 - \xi_i$$

with the following:

$$y_i\left\{w'x_i + b\right\} \geq 1 - \xi_i$$
$$\xi_i \geq 2V_i$$
$$1 - w'x_i - b + MV_i \geq z_i + \varepsilon$$

where $V_i \in \{0,1\}$ and where $M > 0$ is a sufficiently large and $\varepsilon > 0$ sufficiently small. Table 4-3 includes the full implications of these constraints together with the objective function that minimizes ξ_i when it is otherwise unconstrained from above. Notice that q_i is not explicitly needed for the negative cases. The last case has $V_i = 0$ even though the constraints also allow $V_i = 1$ because the minimization process will force ξ_i (and hence V_i) to zero.

Table 4-3. Negative cases.

$1 - w'x_i - b$	V_i	ξ_i
< 0	1	$1 - y_i\left\{w'x_i + b\right\} > 2$
$\in [0, z_i]$	1	2
$> z_i$	0	0 for $z_i \geq 2$
		> 0 for $z_i < 2$

Consider the case of a positive agent. Here we replace

$$y_i\left\{w'\left[x_i + D(w)\,d_i^*(w,b)\right] + b\right\} \geq 1 - \xi_i$$

with the constraints

$$y_i\left\{w'x_i + b + q_i\right\} \geq 1 - \xi_i$$
$$M - MV_i \geq \xi_i \geq 0$$
$$MV_i \geq q_i \geq 0$$
$$z_i \geq q_i$$

where $V_i \in \{0,1\}$ and $M > 0$ is sufficiently large. Table 4-4 includes the full implications of these constraints together with the objective function that minimizes ξ_i over q_i when possible.

Table 4-4. Positive cases.

$1 - w'x_i - b$	V_i	q_i	ξ_i
< 0	0 or 1	0	0
$\in [0, z_i]$	1	$1 - w'x_i - b$	0
$> z_i$	0	0	$1 - w'x_i - b$

Collecting the above gives:

$$P4: \quad \min_{\xi_i \geq 0} w'w + \sum_{i=1}^{\ell} C_{y_i} \xi_i + \lambda \sum_{y_i = 1} q_i$$

s.t.

effort $(y_i = +1)$:

$$w'x_i + b + q_i \geq 1 - \xi_i$$
$$M - MV_i \geq \xi_i$$
$$MV_i \geq q_i \geq 0$$
$$z_i \geq q_i$$

effort $(y_i = -1)$:

$$-w'x_i - b \geq 1 - \xi_i$$
$$\xi_i \geq 2V_i$$
$$1 - w'x_i - b + MV_i \geq z_i + \varepsilon$$

max *adjustment*:

$$r_i \frac{\left(w^+ + w^-\right)_j}{\left(c_i\right)_j} \leq z_i \leq r_i \frac{\left(w^+ + w^-\right)_j}{\left(c_i\right)_j} + MH_{i,j} \qquad j = 1, \ldots, n, i = 1, \ldots, \ell$$
$$\sum_j H_{i,j} = n - 1 \qquad i = 1, \ldots, \ell$$

absolute value:

$$w = w^+ - w^-$$
$$MI_j \geq w_j^+ \geq 0 \qquad j = 1, \ldots, n$$
$$M - MI_j \geq w_j^- \geq 0 \qquad j = 1, \ldots, n$$

int *egrality* :

$$I_j, H_{i,j}, V_i \in \{0,1\}$$

We note that the SVM model typically uses a 2-norm measure (our w'w in the objective) but a 1-norm alternative is equally acceptable and many researchers focus on it (e.g., see Fung and Mangasarian (2002)). In such a case, the objective would be

$$\sum_j \left(w^+ + w^-\right)_j + \sum_{i=1}^{\ell} C_{y_i}\xi_i + \lambda \sum_{y_i=1} q_i$$

making the 1-norm version of P4 a Mixed Integer Linear program. This is fortuitous since the 1-norm problem is much easier to solve. In the next section, we use P4 to solve a strategic version of a credit-risk evaluation problem. We then look at stochastic versions of P4 where the principal may not know certain agent parameters.

Sample Application

In this section we apply our results to a credit-risk evaluation dataset which is publicly available at the UCI repository (http://www.ics.uci.edu/~mlearn/MLRepository.html) and referred to as German credit data. The original dataset consists of 1,000 instances with 20 attributes (7 numerical, 13 categorical).

For the purposes of our analysis, some of these categorical attributes, such as status of existing checking account (greater than zero, between zero and 200 DM, greater than 200 DM) were converted to numerical values (e.g., 0, 100, 300) others were replaced by binary dummy variables.

For the attributes that were converted, we assumed the value of the attribute to be the midpoint of the interval it lies within and for values that are outside the specified intervals we incremented with an amount reflecting the pattern of increase. Thus, the

resulting dataset has 52 attributes summarized in the Table 4-5. For numerical reasons we standardized the converted data set.

Table 4-5. Converted German credit data.

Attribute index (i)	Attribute name	Type	c_k
0	Checking Account Balance	Converted to Continuous	0.1
1	Duration	Continuous	100
2,3,4,5,6	Credit History	Converted to Binary	∞
7,8, 9,10,11,12, 13,14,15,16,17	Purpose	Converted to Binary	∞
18	Credit Amount	Continuous	10
19	Savings Account Balance	Converted to Continuous	0.01
20	Employment Since	Converted to Continuous	100
21	Instalment rate	Continuous	100
22,23, 24, 25, 26	Personal Status	Converted to Binary	∞
27, 28, 29	Other Parties	Converted to Binary	∞
30	Residence Since	Continuous	100
31,32, 33, 34	Property	Converted to Binary	∞
35	Age	Continuous	100
36, 37, 38	Other Instalment Plans	Converted to Binary	∞
39, 40, 41	Housing	Converted to Binary	∞
42	Number of Existing Credit Cards	Continuous	100
43, 44, 45, 46	Job	Converted to Binary	∞
47	Number of Dependents	Continuous	100
48, 49	Own Telephone	Converted to Binary	∞
50, 51	Foreign worker	Converted to Binary	∞

The categorical attributes contained in the original dataset (that we converted to binary variables) such as personal status (divorced male, married female and such) and sex are almost impossible to alter by an agent or not worth altering for the purposes of a credit application so we assigned their c value infinite cost.

This was operationalized in P4 by leaving out the inequalities

$$r_i \frac{\left(w^+ + w^-\right)_j}{\left(c_i\right)_j} \le z_i \le r_i \frac{\left(w^+ + w^-\right)_j}{\left(c_i\right)_j} + MH_{i,j}$$

corresponding to such j's and by reducing the right side of

$$\sum_j H_{i,j} = n - 1$$

by one for each such j.

Each attribute is assigned a different cost ranging between 0.01 and 100 reflecting our subjective assessment of the relative level of difficulty of changing that attribute's value. These costs are summarized in Table 4-5.

We solve this problem assuming K = 1 (i.e., all agents have the same utility structure) with $r = 15$. Further we set $\varepsilon = 10^{-7}$, $C_{+1} = 4$ and $C_{-1} = 5$.

Applying P4 to 100 point subset of the full 1,000 point, non-separable data set using the same reservation and misclassification cost structure, we got the results summarized in Tables 4-6 and 4-7.

Table 4-6 focuses on strategic solutions varying λ for the 1-norm version of P4 and Table 4-7 focuses on the 2-norm version of P4. Also the results of the non-strategic solutions are included in each table.

We used CPlex (ILOG 2005) to solve all the problems. In Tables 4-6 and 4-7, The Number of Positive and Negatives Moved are the number of cases that could move to a positive labeling with respect to the solution and the Total Reservation Cost Used is the total cost to the agents for these moves ($\sum_i (c_i)_{j^*} q_i / |w_{j^*}|$).

The rows of the tables listed under the heading of "Strategic Impact" such as number of points moved and misclassified, total misclassification cost ($\sum_{i=1}^{\ell} C_{y_i} \xi_i$), objective value with strategic moves etc. are the results of strategic behavior.

Notice that $\sum_{i=1}^{\ell} C_{y_i} \xi_i$ is reported in two different rows. For the strategic solutions, this result is the same in both rows for the obvious reason of P4 being modeled to anticipate and take into account the possible strategic behavior. However for non-strategic solutions, there is no such anticipation so the first row corresponds to the misclassification costs without strategic behavior and the second one is calculated taking into account the misclassification costs of points after they move with respect to the non-strategic solution.

Similarly, the variable q_i does not exist in any of the non-strategic formulations. Thus, the term $\lambda \sum_{y_i=1} q_i$ corresponding to the non-strategic solutions in all tables are calculated theoretically taking in account the movements of the positive points with respect to the solution (In Tables 4-6 and 4-7, $\lambda = 1$). Likewise, the moved positive and negative points and final misclassifications reflect after-solution moves.

j^* was found to be attribute 19 ("Savings Account Balance") for most of the cases. However in some of the cases we observe a tie between attribute 19 and attribute 0 ("Checking Account Balance") leaving the agents indifferent between making modifications on these two attributes. For those cases, j^* was chosen to be the attribute with the lower index number.

Table 4-6. 1-norm strategic SVM solutions (P4) for various values of λ vs non-strategic solution for 100 instances.

	1-Norm								
	Non-Strategic	Strategic							
Attribute name	$\lambda = 1$	$\lambda = 0$	$\lambda = 0.01$	$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 1$	$\lambda = 1.5$	$\lambda = 2$
0 - Checking Account Balance	0	0.000565	0.000565	0.000733	0.000711	-0.01333	-0.01046	0.013333	-0.01333
1 – Duration	-1.42275	-0.00827	-0.00833	-0.00867	-0.00868	0.060293	0	-0.86934	-1.15545
18 - Credit Amount	0	0	0	-0.00063	-0.00063	-0.7216	-0.84269	-0.32216	-0.43376
19 - Savings Account Balance	0.276264	0.001333	0.001333	0.001333	0.001333	0.001333	0.001333	0.001333	0.001333
20 - Employment Since	0.043407	0	0	0.000378	0.000494	0.255101	0.345983	0.151507	0.240438
21 - Instalment rate	-1.58332	-0.00527	-0.0053	-0.00543	-0.00547	-0.68458	-0.4795	-0.28348	-1.20469
30 - Residence Since	2.678459	0.010079	0.010143	0.010765	0.010761	1.312225	2.110051	2.06295	2.185361
35 - Age	2.073676	-0.00203	-0.00201	-0.00023	0	0.11225	0	0.216032	0.7117
42 - # of Existing Credit Cards	-0.59592	-0.00314	-0.00316	-0.00386	-0.004	-0.92055	-1.01822	-0.6245	-0.46409
47 - Number of Dependents	1.637194	0.003129	0.003154	0.003255	0.003241	1.559915	1.757889	3.144514	1.835584
b^*	1.36938	-0.99817	-0.99816	-0.99714	-0.85675	-0.24652	0.23245	0.61498	0.83092
$\|w\|$	23.55181	0.07271	0.072972	0.07898	1.10333	11.98734	15.60196	19.13669	20.81358
$\sum_{i=1}^{\ell} C_{y_i}\xi_i$	101.6607	8.0395	8.0392	8.0366	8.0367	10.00	20.00	31.00	57.5435
Objective Value	125.213	8.11226	9.48827	14.9906	21.4947	53.6396	75.3268	93.7098	105.838
	Strategic Impact								
# of Positives Moved	6	69	69	69	64	51	32	24	15
# of Negatives Moved	26	0	0	0	0	1	1	2	4
# of Pos. Misclassifications	0	1	1	1	1	0	0	0	1
# of Neg. Misclassifications	30	0	0	0	0	1	2	3	4
Total Reservation Cost Used	3.73468	1034.28	1032.00	1031.25	926.60	474.78	297.93	217.86	103.05
$\sum_{i=1}^{\ell} C_{y_i}\xi_i$ (with Strategic moves)	318.8339	8.0395	8.0392	8.0366	8.0367	10.00	20.00	31.00	57.5435
$\lambda \sum_{y_i=1} q_i$	103.17571	0	1.37600	6.8750	12.35469	31.65221	39.72485	43.57221	27.48069
Objective Value (with Strategic moves)	350.2729	8.1122	9.4882	14.9906	21.4947	53.6396	75.3268	93.7098	105.838
Seconds to solve (3.4 GHz Xeon proc.)	0.016	4.376	4.391	2.422	2.844	2.047	1.344	1.625	1.282

Table 4-7. 2- norm strategic SVM solutions (P4) for various values of λ vs non-strategic solution for 100 instances

	2-Norm								
	Non-Strategic	Strategic							
Attribute name	$\lambda = 1$	$\lambda = 0$	$\lambda = 0.01$	$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 1$	$\lambda = 1.5$	$\lambda = 2$
0 - Checking Account Balance	0.077049	0.000685	0.000128	0	0.010434	-0.01333	0	0.013333	-0.01333
1 – Duration	-0.94589	-0.00709	-0.00679	-0.0057	-0.0648	0.099996	0.058722	-0.60349	-0.72603
18 - Credit Amount	-0.33725	-0.00225	-0.00253	-0.00404	-0.04935	-0.65761	-0.77481	-0.4234	-0.8176
19 - Savings Account Balance	0.280419	0.001333	0.001333	0.001333	0.001333	0.001333	0.001333	0.001333	0.001333
20 - Employment Since	0.418413	0.000513	0.001267	0.001798	0.036009	0.381561	0.439863	0.722547	0.338969
21 - Instalment rate	-0.89546	-0.00726	-0.00835	-0.00989	-0.11209	-0.57348	-0.43845	-0.81373	-0.80658
30 - Residence Since	1.316182	0.014112	0.01465	0.017694	0.15321	1.165015	1.881122	1.165456	0.97931
35 - Age	0.885631	-0.00013	0.001142	0.001063	0.016148	0.243205	-0.03338	0.198731	-0.11161
42 - # of Existing Credit Cards	-0.57952	-0.00221	-0.00358	-0.00191	-0.02947	-0.7029	-0.63172	-0.22857	-0.38767
47 - Number of Dependents	1.217864	0.006797	0.006577	0.010279	0.108277	1.380006	1.209247	0.956158	1.201656
b^*	0.97802	-0.99691	-0.95366	-0.86202	-0.80088	-0.27503	0.15576	0.42523	0.68693
$\|w\|$	3.59426	0.00051	0.14567	0.47972	0.59942	2.65509	3.19144	2.95735	3.10397
$\sum_{i=1}^{\ell} C_{y_i}\xi_i$	110.5845	8.0343	8.0343	8.0379	8.5655	10.00	20.00	39.7220	50.5038
Objective Value	123.503	8.03488	9.38724	14.4660	20.5455	49.2335	72.6676	89.5899	101.208
	Strategic Impact								
# of Positives Moved	10	69	69	65	63	52	35	24	19
# of Negatives Moved	28	0	0	0	0	1	0	1	3
# of Pos. Misclassifications	0	1	1	1	1	0	0	1	0
# of Neg. Misclassifications	30	0	0	0	0	1	2	2	5
Total Reservation Cost Used	3.01754	1035	998.71	929.70	871.55	482.76	318.61	205.60	154.01
$\sum_{i=1}^{\ell} C_{y_i}\xi_i$ (with Strategic moves)	302.9685	8.0343	8.0343	8.0379	8.5655	10.00	20.00	39.7220	50.5038
$\lambda \sum_{y_i=1} q_i$	8.37569	0	1.33162	6.19800	11.62066	32.18402	42.48227	41.12195	41.06958
Objective Value (with Strategic moves)	324.2629	8.0348	9.3872	14.4660	20.5455	49.2335	72.6676	89.5899	101.208
Seconds to solve (3.4 GHz Xeon proc.)	0.063	81.661	102.694	45.643	32.408	6.328	8.672	8.953	9.642

The 2-norm results are very similar to their 1-norm counterparts. This is interesting since the 1-norm problem is much simpler to solve.

The solutions to P4 provide several significant improvements over their non-strategic counterparts.

First, strategic solutions perform better in terms of the total number of misclassifications. In both Tables 4-6 and 4-7, we observe a drastic decrease in the number of negative misclassifications for strategic solutions when compared with their non-strategic counterparts.

Strategic solutions better separate the positive agents from the negative ones, and hence their costs of misclassification, $\sum_{i=1}^{\ell} C_{y_i} \xi_i$, is lower than the non-strategic results for all values of λ in both tables. They accomplish this by forcing a large number of positive agents to modify their attributes at a significant total costs in effort to these agents. This can be observed by the increase in number of positive points moved for the strategic solutions compared to non-strategic solutions.

As discussed after Theorem 1, a principal may want to exchange some margin (i.e., $1/\|w\|$) for lowered moves and thus effort by positive agents. The downside could be a looser bound on the principal's risk functional of the induced discriminant function and hence lower confidence in the result.

Comparing the non-strategic and strategic results for $\lambda = 1$ we see a significant drop in objective value in both tables emphasizing the high payoff gained by Strategic Learning.

Furthermore, when strategic results are compared for increasing values of λ, we see an increase in the objective values. This is a result of penalizing the objective function more for each movement of positive agents as λ is increased. This forces fewer agents to move and hence causes an increase in the positive misclassification cost. However, depending on the trade-off between an increase in λ and a decrease in $\sum_{y_i=1} q_i$ term, we observe fluctuations in the $\lambda \sum_{y_i=1} q_i$ value.

Comparing the signs of the various coefficients, we see switches in a few (Credit Amount and Age) which reflect the change in effect on the classification after agents adjust.

Table 4-8 below compares the results of non-strategic and strategic solutions for the full 1,000 point German credit dataset which was not standardized. The 1-norm and 2-norm strategic results are very similar to each other.

A decrease in the number of agents moved for both 1-norm and 2-norm solutions compared to the non-strategic case is observed. Hence, the reservation cost used for strategic solutions is substantially lower than their non-strategic counterparts. This shows that strategic solutions were able to prevent most of the agent movement. This leads to an improvement in the term $\lambda \sum_{y_i=1} q_i$ and also in the strategic objective function.

It should be noted that with $\lambda = 3.99$ which is very close to $C_{+1} = 4$, strategic and non-strategic solutions are quite similar.

Table 4-8. Strategic SVM solutions (P4) for $\lambda = 3.99$ vs non-strategic solutions for 1000 instances.

	1-Norm		2-Norm	
Attribute name	**Non-Strategic**	**Strategic**	**Non-Strategic**	**Strategic**
0 - Checking Account Balance	-0.0006	-0.00023	-0.00062	-0.00027
1 – Duration	-0.02818	-0.03106	-0.02798	-0.0312
18 - Credit Amount	-8.3E-05	-7.7E-05	-8.5E-05	-7.8E-05
19 - Savings Account Balance	0.000539	0.000225	0.000538	0.000223
20 - Employment Since	0.071218	0.067657	0.070059	0.068999
21 - Instalment rate	-0.23377	-0.19916	-0.23657	-0.20034
30 - Residence Since	-0.02751	-0.04517	-0.02634	-0.04759
35 - Age	0.007454	0.009296	0.007888	0.009507
42 - # of Existing Credit Cards	-0.16523	-0.12605	-0.16987	-0.13148
47 - Number of Dependents	-0.00372	-0.09361	-0.00321	-0.08893
b^*	2.61065	3.09950	1.75885	1.87177
$\|w\|$	12.96727	13.09071	2.83717	2.78167
$\sum_{i=1}^{\ell} C_{y_i}\xi_i$	2579.97	2524.75	2579.63	2526.27
Objective Value	2592.94	2606.92	2587.69	2601.57
	Strategic Impact			
# of Positives Moved	196	111	212	112
# of Negatives Moved	86	10	85	7
# of Positive Misclassifications	115	204	116	205
# of Neg. Misclassifications	256	259	260	259
Total Reservation Cost Used	2035.05	787.95	2030.00	775.13
$\sum_{i=1}^{\ell} C_{y_i}\xi_i$ (with Strategic moves)	2465.39	2524.75	2460.56	2526.27
$\lambda\sum_{y_i=1} q_i$	295.08	69.07	295.08	67.55
Objective Value (with Strategic moves)	2763.69	2606.92	2763.69	2601.57
Seconds	0.438	784.67	1.891	73194.671

Stochastic Versions

The assumption that the principal knows the reservation values and costs (r_i and c_i) of each agent is rather limiting. One way to relax this assumption is to assume that the principal, through experience, knows that there are different types of agents and the associated distribution function over these types. Consequently, in solving the strategic problem the principal has to take into account the fact that he/she cannot count on known r_i and c_i.values.

To model this we start by assuming that $\theta = (\mathrm{r}, \mathrm{c})$ is a random vector with finite support $s \in \{1, ..., S\}$ and a discrete density function. We indicate each agent by his/her type and if an agent is of "type s" he/she has costs $c(s)$ and reservation value $r(s)$. An alternative interpretation would be to say that the random vector $\theta(\mathrm{s})$ depends on the agent type s. Agents as usual solve the following

$$\begin{aligned} &\min c(s)' d_{is} \\ &st \qquad w'\left[x_i + D(w) d_{is} \right] + b \geq 1 \\ &\qquad d_{is} \geq 0 \end{aligned}$$

Following the same logic as the deterministic case the following can be determined for each s.

$$j_s^* = \arg \min_{j, w_j \neq 0} \frac{c(s)_{\mathrm{j}} \max\left(0, 1 - \left(b + w' x_i\right)\right)}{\left| w_j \right|}$$

then for

$$z_{is}^* (w, b) = \frac{\max\left(0, 1 - \left(b + w' x_i\right)\right)}{\left| w_{j_s^*} \right|}$$

we have

$$d_{is}^* (w, b) = \begin{cases} z_{is}^* (w, b) 1_{j_s^*} & \text{if } c(s) z_{is}^* (w, b) \leq r(s) \\ 0 & \text{otherwise} \end{cases}$$

For w equal to zero, set $d_{is}^* (0, b) = 0$.

There are many different ways of formulating the principal's stochastic Strategic Learning problem. One such approach is to model the problem by taking all possible realizations of θ into account and populating the constraints of the deterministic case for

each $s = 1,...,S$. This can be interpreted as a worst case formulation of the problem. Here, the principal's problem is written as:

$$\min_{w,b} w'w + \sum_{i=1}^{\ell} C_{y_i} \max_{s} \xi_{is}$$
$$s.t. \quad y_i \left\{ w' \left[x_i + D(w) d_{is}^*(w,b) \right] + b \right\} \geq 1 - \xi_{is} \qquad i = 1,...,\ell \text{ and } s = 1,...,S$$

Another approach might be to use expected values of random variables r(s) and c(s) to arrive at an "average agent" type using the models discussed in earlier sections where all agents have the same cost and reservation values. A third approach would be to use chance-constrained formulation and replace the original constraints by corresponding chance constraints. Let $0 < \alpha_i < 1$, then the principal's problem becomes

$$\min_{w,b} w'w$$
$$s.t. \quad \Pr ob \left\{ y_i \left\{ w' \left[x_i + D(w) d_{is}^*(w,b) \right] + b \right\} \geq 1 \right\} \geq \alpha_i \qquad i = 1,...,\ell$$

Here $(1-\alpha_i)$ represents the allowable risk that $d_{is}^*(w,b)$ takes on values that wouldn't satisfy the constraints.

The approach we favor is a fourth approach that depends on minimizing the expected total misclassification cost by hedging against different possible agents types as shown in the following model. Let P_s be the probability that an agent is of type s. Then we solve

$$P5: \quad \min_{w,b} w'w + \sum_{i=1}^{\ell} C_{y_i} \sum_{s=1}^{S} P_s \xi_{is}$$
$$s.t. \quad y_i \left\{ w' \left[x_i + D(w) d_{is}^*(w,b) \right] + b \right\} \geq 1 - \xi_{is} \qquad i = 1,...,\ell \text{ and } s = 1,...,S$$

The counterpart of P5 (as P4 was to P3) is:

$$P6: \quad \min_{\xi_i \geq 0} w'w + \sum_{i=1}^{\ell} C_{y_i} \sum_{s=1}^{S} P_s \xi_{is} + \lambda \sum_{\substack{s=1 \\ y_i=1}}^{S} P_s q_{is}$$

s.t.

effort $(y_i = +1)$:

$$w'x_i + b + q_{is} \geq 1 - \xi_{is} \qquad i = 1,...,\ell, s = 1,...,S$$
$$M - MV_{is} \geq \xi_{is} \qquad i = 1,...,\ell, s = 1,...,S$$
$$MV_{is} \geq q_{is} \geq 0 \qquad i = 1,...,\ell, s = 1,...,S$$
$$z_i \geq q_{is} \qquad i = 1,...,\ell, s = 1,...,S$$

effort $(y_i = -1)$:

$$-w'x_i - b \geq 1 - \xi_{is} \qquad i = 1,...,\ell, s = 1,...,S$$
$$\xi_{is} \geq 2V_{is} \qquad i = 1,...,\ell, s = 1,...,S$$
$$1 - w'x_i - b + MV_{is} \geq z_i + \varepsilon \qquad i = 1,...,\ell, s = 1,...,S$$

max *adjustment*:

$$r_i \frac{\left(w^+ + w^-\right)_j}{\left(c_i\right)_j} \leq z_i \leq r_i \frac{\left(w^+ + w^-\right)_j}{\left(c_i\right)_j} + MH_{i,j} \qquad j = 1,\ldots,n, i = 1,...,\ell$$
$$\sum_j H_{i,j} = n - 1 \qquad i = 1,...,\ell$$

absolute value:

$$w = w^+ - w^-$$
$$MI_j \geq w_j^+ \geq 0 \qquad j = 1,\ldots,n$$
$$M - MI_j \geq w_j^- \geq 0 \qquad j = 1,\ldots,n$$

int*egrality*:

$$I_j, H_{i,j}, V_i \in \{0,1\}$$

Conclusion and Future Research

In this chapter we studied the effect of strategic behavior in determining linear discriminant functions. We considered two cases. In the base case, we analyzed the problem under the assumption that all agents have the same reservation costs. We showed that the optimal solution with strategic behavior is a shifted, scaled version of the solution found without strategic behavior.

For the case of equal reservation costs we find that the principal will choose a discriminant function where negative agents will have no incentive to change their true attributes but agents who are marginally positive will be forced to alter their attributes. Thus, roughly speaking, the ones who are "penalized" for engaging in strategic behavior are not the negative agents but rather the marginal positive agents. We also note that under strategic behavior the final discriminant function used by the principal will produce a bigger gap between the two classes of points.

For the general case where all agents have different reservation and cost structures,, we developed mixed integer programming models and applied our results to a credit card evaluation setting.

An issue of great importance that has not been explored yet is the application of kernel mappings under strategic behavior. It may be possible to anticipate and cancel the effects of strategic behavior by applying an appropriate kernel mapping. Of course, for an agent to anticipate a useful direction of change in some unknown feature space seems unlikely. Ramifications such as these make this approach daunting.

There are many other avenues to investigate. Usually, it might not be realistic to let each attribute be modified unboundedly without posing any constraints on how much they can actually be modified which we model in Chapter 5.

An interesting direction of research is to relax the assumptions on what the principal and the agents know. Although we have modeled a stochastic version of the Strategic Learning problem where we hedged for the uncertainty in the types of agents, we have not yet incorporated informational uncertainty in terms of the available knowledge to both parties. As an example, instead of modeling agent behavior for a

given classifier, we might assume that agents know only the signs of the coefficients of a given classifier. Also, it can be assumed that the agents can react to a classifier that is only known with some error. Hence, relaxing the assumptions on what agents and principal know about each other is an open area of research.

We assumed linear agent utilities. Relaxing these assumptions would make an interesting future research project. One interesting area for future consideration is the case of unequal reservation costs where negative agents are willing to expend any amount to be positively classified (e.g., suicide bombers trying to appear normal).

Additional areas of future research include the following. Agent collusion is not considered here. Many possibilities come to mind. For example, if agents can collude and offer side payments to other agents to make sub-optimal changes in their attributes to confuse and thwart the principal, can this be anticipated in the induction process? We studied a static situation. If the instance space changes over time (due to some exogenous factors), can we dynamically model user behavior and determine classifiers that will adapt efficiently? Another twist of this model can be to encourage real change (not just thwart superficial change). This might prove useful in public policy problems.

CHAPTER 5
USING GENETIC ALGORITHMS TO SOLVE THE STRATEGIC LEARNING PROBLEM

In this chapter, we provide a Genetic Algorithm for solving the Strategic Learning problem. We start by reducing the Strategic Learning problem to an unconstrained search over $w \in \Re^n$. Once we have accomplished this, we develop a Genetic Algorithm (GA) to perform this search.

An Unconstrained Formulation for Strategic Learning

Strategic Learning is defined as the task of a principal whose goal is to discriminate between certain type of agents which are self-interested, utility maximizing and decision making units. The following are examples of such principal and agent pairs:

- a credit card company (the principal) decides which people (agents) get credit cards.
- an anti-spam package (the principal is the package creator) tries to correctly label and then screen spam (which is agent created).
- airport security guards (the principal) try to distinguish terrorists from normal passengers (agents).

In this chapter, we focus on linear discriminant functions for binary classification which is usually performed by first determining a non-zero vector $w \in \mathbb{R}^n$ and a scalar b such that the hyperplane $w'x + b = 0$ partitions the n-dimensional Euclidian space into two half-spaces. Then, an observed vector $\mathrm{x_i}$ is assigned to the positive class if it satisfies $w'x + b \geq 0$. Otherwise it is assigned to the negative class. That is,

$(w,b):\mathbb{R}^n \to \{-1,+1\}$ where +1 denotes the positive class and -1 denotes the negative class.

Refreshing the notation in Strategic Learning setting, each agent i has a true vector of attributes x_i, a true label $y_i \in \{-1,+1\}$, reservation cost r_i and a vector of costs for modifying attributes c_i. Reservation cost can be viewed as the maximum effort that an agent is willing to exert in order to be classified as a positive agent. On the principal's side, C_{y_i} is the penalty associated with the margin shortfall (ξ_i) of an agent of true type y_i.

In Chapter 4, we proposed mixed integer programming solutions for Strategic Learning problem by focusing on a classification method known as support vector machines (Cristianini and Shawe-Taylor 2000) and we defined the general model of Strategic Learning problem as the following:

$$P3: \quad \min_{w,b} w'w + \sum_{i=1}^{\ell} C_{y_i}\xi_i + \lambda \sum_{y_i=+1} q_i(w,b)$$
$$s.t. \quad y_i\{w'x_i + q_i(w,b) + b\} \geq 1-\xi_i \qquad i = 1,\ldots,\ell$$

where

$$q_i(w,b) = \begin{cases} 0 & if \quad 1-b-w'x_i < 0 \\ 0 & if \quad 1-b-w'x_i > z_i \\ 1-b-w'x_i & otherwise \end{cases}$$

and

$$z_i \equiv r_i \max_{j \ni (c_i)_j \neq 0} \frac{|w_j|}{(c_i)_j}$$

for $\lambda \geq 0$, $C_{y_i} > 0$ and $c_i \geq 0$ with at least one j satisfying $\infty > (c_i)_j > 0$.

Now, letting $\xi_i(w,b) = \max\left(0, 1 - y_i\left[w'x_i + q_i(w,b) + b\right]\right)$ and determining b using a specialized search algorithm (see below) for a given w we get an unconstrained version of the Strategic Learning problem which is

$$f(w,b) = m(\|w\|) + \sum_{i=1}^{\ell} C_{y_i}\xi_i(w,b) + \lambda \sum_{y_i=+1} q_i(w,b)$$

where with $\|w\|$ being a norm of w and $m(\ )$ an increasing function with $m(0) = 0$. We are interested in epsilon-minimizing $f(w,b)$.

Let $b(w)$ be an epsilon optimal solution to $\varepsilon - \min_b f(b \mid w)$ where $f(b \mid w)$ denotes the function $f(w,b)$ for a fixed w. In other words, for a given w, we want to determine a value for $b(w)$ that epsilon-minimizes $f(b \mid w)$.

As noted in Chapter 4, $f(b \mid w)$ does not have any nice features such as quasiconvexity. However it is still easy to find an epsilon-optimal solution. Table 5-1 gives the possible regions of the value $C_{y_i}\xi_i(w,b) + \lambda q_i(w,b)$ positive agents and $C_{y_i}\xi_i(w,b)$ for negative agents depending on λ and z_i.

Table 5-1. Different regions of costs

Positive Cases		Negative Cases	
$\lambda = 0$	$\lambda > 0$	$z_i < 2$	$z_i \geq 2$
$(-\infty, 1 - z_i - w'x_i)$	$(-\infty, 1 - z_i - w'x_i)$	$(-\infty, -1 - w'x_i]$	$(-\infty, 1 - z_i - w'x_i)$
$[1 - z_i - w'x_i, \infty)$	$[1 - z_i - w'x_i, 1 - w'x_i)$	$[-1 - w'x_i, 1 - z_i - w'x_i)$	
	$[1 - w'x_i, \infty)$	$[1 - z_i - w'x_i, 1 - w'x_i)$	$[1 - z_i - w'x_i, 1 - w'x_i)$
		$[1 - w'x_i, \infty)$	$[1 - w'x_i, \infty)$

$f(b \mid w)$ is linear in the different regions of cost included in Table 5-1 which makes it easy to develop an algorithm to find an epsilon optimal solution to $\varepsilon - \min_b f(b \mid w)$. The key is to search over the finite starting points for the different regions in Table 5-1. For negative points at the point of discontinuity, $1 - z_i - w' x_i$, we evaluate the function at an epsilon lower point because the function jumps up at the discontinuity (See Chapter 4 for details).

There are a finite number of such starting and epsilon-neighbor points. Algorithm 1, below, examines each of these points and evaluates the function $f(b \mid w)$ at these points.

The proof of epsilon-optimality provided by Algorithm 1 follows from knowing that between two consecutive (non-epsilon) starting points the cost function is linear, meaning only the end points need be evaluated. Furthermore, we ignore points between an epsilon neighbor $b - \varepsilon$ and b (hence we achieve only epsilon optimal solutions, see Chapter 4 for details). So we march through potential values of b noting the one with the lowest total cost. This provides an epsilon optimal value of b ($b(w)$).

Below is Algorithm 1. Let $\varepsilon > 0$ be sufficiently small and fixed.

Algorithm 1

Input w.

Output an epsilon-optimal b.

Set $\alpha = \phi$

for ($i = 1, ..., \ell$) {

if ($y_i = +1$)

$\alpha \leftarrow \alpha \cup \{1 - z_i - w' x_i\}$

if $(\lambda > 0)$ $\alpha \leftarrow \alpha \cup \{1 - w' x_i\}$

else

$\alpha \leftarrow \alpha \cup \{1 - z_i - w' x_i - \varepsilon\}$

$\alpha \leftarrow \alpha \cup \{1 - w' x_i\}$

if $(z_i < 2)\, \alpha \leftarrow \alpha \cup \{-1 - w' x_i\}$

}

Set $f \leftarrow \infty$, $b \leftarrow \infty$

for each $\hat{b} \in \alpha$ {

$\hat{f} \leftarrow f\left(\hat{b} \mid w\right)$

if $\hat{f} < f$ then { $f \leftarrow \hat{f}$, $b \leftarrow \hat{b}$ }

}

Output b;

At this point we have reduced the Strategic Learning problem to an unconstrained search over $w \in \Re^n$.

Before proceeding to the GA formulation, one result is of value. Lemma 1 provides an upper bound on $\varepsilon - \min_w f(w)$ and allows us to focus on non-zero values of w.

Lemma 1

$$f(0) = \begin{cases} 2C_{+1} \sum_{y_i = +1} y_i & \sum_i C_i y_i \le 0 \\ -2C_{-1} \sum_{y_i = -1} y_i & \sum_i C_i y_i \ge 0 \end{cases}$$

Proof:

When w=0, $z_i = 0$ and thus $q_i = 0$ for all i. We are left with choosing b. From the identities of Table 1 we get

$$b(0) = \begin{cases} -1 & \sum_i C_i y_i \le 0 \\ 1 & \sum_i C_i y_i \ge 0 \end{cases}$$

Thus

$$\begin{aligned} f(0) &= m(0) + \sum_{i=1}^{\ell} C_{y_i} \xi_i \left(0, b(0)\right) + \lambda \sum_{y_i=1} q_i \left(w, b(0)\right) \\ &= 0 + \sum_{i=1}^{\ell} C_{y_i} \xi_i \left(0, b(0)\right) + 0 \\ &= \begin{cases} 2C_{+1} \sum_{y_i=+1} y_i & \sum_i C_i y_i \le 0 \\ -2C_{-1} \sum_{y_i=-1} y_i & \sum_i C_i y_i \ge 0 \end{cases} \end{aligned}$$

□

At this point we can solve the Strategic Learning problem by searching over $w \in \Re^n / \{0\}$. However, we have found it more advantageous to search over $w \in \Re^n / \{0\}$ where we also determine a scaling factor $\gamma \in \Re$ for each w. Define the function

$$f(\gamma w, b) = m\left(\|\gamma w\|\right) + \sum_{i=1}^{\ell} C_{y_i} \xi_i (\gamma w, b) + \lambda \sum_{y_i=+1} q_i (\gamma w, b)$$

Let $b(\gamma \mid w)$ be an epsilon optimal solution to $\varepsilon - \min_b f(\gamma w, b)$ for a fixed w. Consider

$$f(\gamma \mid w) = m\left(\|\gamma w\|\right) + \sum_{i=1}^{\ell} C_{y_i} \xi_i \left(\gamma w, b(\gamma \mid w)\right) + \lambda \sum_{y_i=+1} q_i \left(\gamma w, b(\gamma \mid w)\right)$$

where

$$\xi_i(\gamma w, b(\gamma \mid w)) = \max\left(0, 1 - y_i\left[b(\gamma \mid w) + q_i\left(\gamma w, b(\gamma \mid w)\right) + \gamma w' x_i\right]\right)$$

Informally, $f(\gamma \mid w)$ is the total cost function as γ varies for a fixed w and corresponding epsilon optimal solution $b(\gamma \mid w)$. This effectively reduces the search to one over the unit sphere.

For all the problem sets we have examined we have noticed that $f(\gamma \mid w)$ is quasiconvex in γ. Whether this is true in general or under some reasonable set of assumptions remains unknown at this time. However, assuming it is true, we can perform our search as follows. Generate a normalized $w \in \Re^n / \{0\}$. Perform a univariate search over $\gamma \in \Re$ where, for each such γ we use Algorithm 1 to solve for $b(\gamma \mid w)$. This process yields a solution γw. If our assumption about the quasiconvexity of $f(\gamma \mid w)$ is not true, then we just search over $w \in \Re^n / \{0\}$.

A Genetic Algorithm Formulation for Strategic Learning

We propose a Genetic Algorithm (GA) for solving the Strategic Learning problem. Below is a sketch of the algorithm. We state the algorithm assuming we will use the γ-scaling discussed above. However, this can easily be removed from the formulation. We represent population strings as a string of bits representing an n-dimensional vector w. The real-valued coefficients are limited to those that can be represented by 32 bits. We use one bit as a sign and the rest for the magnitude. The GA produces new population members through a mixing process. Mixing consists of mutation and crossover operators. With probability, χ, two selected strings are mated. The two strings produce two children formed using an affine linear crossover operator (Davis 1989). That is,

$$w_{child\,1} = \alpha w_{parent\,1} + (1-\alpha) w_{parent\,2}$$
$$w_{child\,2} = (1-\alpha) w_{parent\,1} + \alpha w_{parent\,2}$$

for $\alpha \in (-1,1)$ randomly drawn. One of the children is randomly selected (a tournament selection could be used here instead). If the strings do not mate (with probability $1-\chi$), one is randomly selected to survive. Once all the new member strings are formed, mutation operators are applied. The GA mutation operator is a uniform mutation operator where each bit is flipped with probability μ.

We also implement a zero-coefficient mutation where w_i is changed to zero with probability μ_{zero}.

Once a string is chosen it is scaled by γ found to minimize $f(\gamma \mid w)$ using a one-dimensional search.

Parent strings are selected using rank selection (Goldberg 1989). The fitness function used in ranking is based on a lexicographic ordering involving three values. The three values, in order of importance, are $f(w)$, the margin and the number of non-zero coefficients. A string is more fit, lexicographically speaking, if, first, its $f(\gamma w)$ is lower. If the two strings have equal $f(\gamma w)$ values, we then choose the string having the larger margin. If these are also equal, we choose the string having the smaller number of non-zero values.

Table 5-2. GA sketch

Algorithm: ***(Genetic Algorithm)***

Given:

Mutation rates μ and μ_{zero}, crossover rate χ and population size $p \geq 1$.

Initialization:

Generate an initial population, population 0.
Randomly draw p strings with replacement. One with the best objective (lexicographically speaking) is designated the queen bee.

Step 1: Form a new population as follows.

(A) Repeat the following steps until the new population has p members.

(1) Randomly choose two members from the old population using the rank selection process.

(2) Form children through a mixing process consisting of crossover and mutation operations.

(B) For each string, normalize the w's represented by the string and reset the string to represent the normalized values. Compute the b value according to Algorithm 1 and an optimal γ value using a one dimensional search algorithm. Replace w with γw. Compute the string's objective value, margin and number of non-zero coefficients. If these are lexicographically better than the queen bee's make the string the new queen bee.

(C) Replace the lowest ranked member with the queen if the queen isn't already a population member.

Step 2: If stopping conditions are not met, return to Step 1.

Experimental Results

To test the efficacy of this GA approach, we performed runs on 100 point subset of German credit data set. For these runs we used the parameters $\chi = 0.2$, $\mu = 10\text{-}3$, $\mu_{zero} = 0.01$ and $p = 10$. The results of 2-norm mixed integer programming model

developed in Chapter 4 and the GA approach developed in this chapter are compared for $\lambda = 1$ in Table 5-3.

Table 5-3. 2- norm strategic SVM solutions (P4) versus GA for 100 instances

Attribute name	**P4**	**GA**
0 - Checking Account Balance	0	-0.01154
1 – Duration	0.058722	-0.23769
18 - Credit Amount	-0.77481	-0.00486
19 - Savings Account Balance	0.001333	0.001152
20 - Employment Since	0.439863	0.19678
21 - Instalment rate	-0.43845	-0.18336
30 - Residence Since	1.881121	0.130356
35 - Age	-0.03338	0.126218
42 - # of Existing Credit Cards	-0.63172	0.070311
47 - Number of Dependents	1.209247	0.140167
b^*	0.15576	-0.03833
# of Positives Moved	35	60
# of Negatives Moved	2	4
# of Positive Misclassifications	3	1
# of Neg. Misclassifications	4	12
$\sum_{i=1}^{\ell} C_{y_i} \xi_i$	20.00	61.253
$\|w\|$	1.78646	1.01176
$\lambda \sum_{y_i=1} q_i$	42.48227	51.8381
Objective Value	72.6676	116.7200

When the results of the two approaches are compared, we see that P4 outperforms GA in many ways. First, P4 forces fewer positive agents to move and the total misclassification cost $\sum_{i=1}^{\ell} C_{y_i} \xi_i$ in significantly lower compared to GA results. Also, objective value of P4 is lower than GA Thus, we conclude that GA approach is promising but needs further work.

Discussion and Future Research

In this chapter, we have reduced the Strategic Learning problem to an unconstrained search over $w \in \Re^n$ and provided a Genetic Algorithm for solving the

problem. Perhaps, the most interesting capability of this model is its power to scale up to large sample sizes in comparison to the mixed integer programming model developed in Chapter 4. Also more work needs to be completed to identify the set of parameters for which the function $f(\gamma \mid w)$ is quasiconvex in γ. Although the results presented in this chapter are promising, further analysis of the approach is required.

CHAPTER 6
STRATEGIC LEARNING WITH CONSTRAINED AGENTS

In Strategic Learning, the principal anticipates possible alteration of attributes by agents wishing to achieve a positive classification. In many cases, an agent has control over its attribute vector such that it might choose to deliberately modify one or more of its attributes in order to achieve favorable classification with respect to a classifier chosen by the principal. In that respect, the principal has no control over an agent's modifications. However, in many cases, agents are constrained on how much an attribute can be modified. For example, agents can be constrained in ways such as upper and lower bounds on the modifications or the modifications may need to belong to a certain set of moves (like in checkers or chess). In this chapter, we explore the need for anticipating attribute adjustment by constrained agents.

Introduction and Preliminaries

The goal of learning from data has a long history of investigation and the development of algorithms for that purpose still remain an important research area. Today, there exist many powerful learning algorithms such as decision trees (Quinlan 1986), neural networks (Tam and Kiang 1992), Bayesian methods (Duda and Hart 1973), and support vector machines (SVMs) (Cristianini and Shawe-Taylor 2000) among others, that are utilized in many different domains of data mining. All of these algorithms are based on the supervised learning model where the learning algorithm is supplied a training set of correctly labeled examples. An example is an attribute vector augmented with a membership identifier (the label). The task is to find an optimum classifier that

separates the set of examples according to their memberships. For example, the training set can be a collection of spam and non-spam emails where each email is represented by a binary vector of attributes each having a value of 1 if a particular word is present in the email or 0 otherwise. The aim is to discover a filter (i.e., a classifier) which correctly identifies an email as either spam or non-spam. This type of a classification task is quite common and appears in many other areas such as credit risk evaluation (Chapter 4), fraud detection (Fawcett and Provost 1997), text categorization (Dumais et al.1998 Joachims 1998) etc.

In many cases this classification task is carried out by a decision maker whose goal is to determine a function to correctly classify instances. Throughout this discussion, we will refer to this decision maker as the "principal" and these instances as "agents" which are defined as self-interested decision making units. For example, for the spam categorization problem the principal can be thought of as the spam filter and the agents as emails. Although emails are not self-interested decision making units, per se, their creators (the spammers) are.

Recent research in the area (Chapter 4, Dalvi et al.1994) consider learning where the sample space, from which the training examples are drawn, is deliberately manipulated by self-interested agents. In earlier Chapters, we defined this new paradigm of learning under such behavior as "Strategic Learning" since our formulation incorporates strategic behavior in the classical supervised learning problem. In our setting, a principal seeking an ultimate classification rule anticipates the possible strategic behavior of self-interested agents who are subject to classification. Dalvi et al. approach the same problem by defining this adversarial classification as a two period game

between two players, the classifier and the adversary. They focus on Bayesian classification method and conduct experiments on a spam problem.

We formulate the problem using rational expectations theory and thus we model an infinite game version between the principal and the agents. In Chapters 4 and 5 we developed methods to determine linear discriminant classifiers in the presence of strategic behavior by agents. In Chapter 4, we used a powerful induction method known as support vector machines (Cristiannini and Shawe-Taylor 2000) which is also the method that will be used in this chapter. In Chapter 4, we assumed agents have a linear disutility function and fixed reservation costs. Under these conditions, for separable datasets, we discovered that if the principal anticipates optimal agent behavior for a given classifier as rational expectations theory depicts then he/she can choose a classifier that can't be defeated by any actions of agents given their cost structures. We showed that the results of naïve discriminant analysis undertaken without taking anticipating the potential strategic behavior may strictly differ from the results of the analysis done with the awareness that agents may behave strategically. We concluded that the naive approach when used in the presence of strategic behavior could be ineffective and misleading.

We illustrated our results on a credit-risk evaluation problem. First, we characterized an optimal strategy for the principal on a separable data set. Second, for non-separable data sets, we provided mixed integer programming solutions and applied them to a credit-risk evaluation setting.

Strategic Learning assumes that agents have control over their attribute vectors such that they might choose to deliberately modify one or more of their attributes in order

to achieve favorable classification with respect to whichever classifier the principal chooses. In that respect, the principal has no control over an agent's actions.

However, an agent's actions are constrained by three facts. First, an agent is assumed to have a constant predefined reservation cost such that the agent can choose to modify an attribute (or collection of attributes) only if the cost of doing so does not exceed the reservation cost. Second, the agents are economically constrained by the cost of changing an attribute which may vary from attribute to attribute and agent to agent. Subsequently, an agent's actions depend highly on this cost structure. These first two facts were considered in the earlier chapters of this dissertation. In this chapter, we consider a third fact which is central to the main idea of this chapter. There might be bounds on how much an attribute can be changed by an agent (other than just those induced by the reservation costs constraint). That is, possible attribute manipulations performed by agents need to be constrained for other reasons. For example, most attributes in practical applications have upper and lower bounds. In a credit card domain, attributes such as age, number of credit cards, income, education level, etc. all have natural bounds. Often there are more involved constraints. For example, in financial domains, simple income and balance sheet relationships dictate linear interdependencies between many common attributes. Thus, usually, it might not be realistic to let each attribute be modified unboundedly which was not modeled in our earlier chapters.

In addition to natural bounds on attributes the representation chosen for the instance space might impose additional constraints. For example, in a spam domain, if one chooses a binary representation for each email (1 if a word is present, 0 if not) then the problem is automatically constrained since the agent (the spammer in this case) can

only alter an email by adding or deleting words thus changing an attribute from 0 to 1 or vice versa. So the attributes are constrained to change only by 1 unit.

Motivated by these facts, we investigate the situation where there is a need to enforce constraints on agent behavior in the context of Strategic Learning. Thus we extend the concept of Strategic Learning to constrained agents and apply our results to spam categorization problem. We focus on linear discriminant functions as our hypothesis space.

Model

Suppose $x \in X \subseteq \Re^n$ represents an agent's attributes. Each agent is a member of one of the two classes, positive (+1) or negative (-1), identified by y. The collection of ℓ examples forms the training set $S = \left(\left(x_1, y_1 \right), ..., \left(x_\ell, y_\ell \right) \right)$ where the subscript i identifies the i^{th} observation x_i (and y_i its label) sampled from X (sampling is i.i.d.). Using the training set, an induction method will determine a vector w and scalar b used to provide classifications over X as follows. An observed vector $x_i \in X$ is classified positive if it satisfies $w' x_i + b \geq 0$. Otherwise it is classified negative. That is, $\left(w, b \right) : \Re^n \to \left\{ -1, +1 \right\}$ and the goal is to find a $\left(w, b \right)$ that optimizes some measure.

We start by assuming a linearly separable sample space and use SVM to find a maximal margin classifier. This changes the decision rule to $x_i \in X$ is classified positive if it satisfies $w' x_i + b \geq 1$. Given a linearly separable sample space, a hyperplane $\left(w, b \right)$ that solves the optimization problem:

$$S_0 : \quad \min_{w,b} w'w$$
$$s.t. \quad y_i \left\{ w' x_i + b \right\} \geq 1 \qquad i = 1, ..., \ell$$

gives us a maximal margin hyperplane (Cristianini and Shawe-Taylor 2000)

Let $c'd_i$, $c > 0$, be the costs to agent i for modifying the agent's true attribute vector $x_i \in \Re^n$ to $x_i + D_i(w)d_i$ for $d_i \geq 0$ where D_i is a diagonal matrix defined by

$$D(w)_{j,j} = \begin{cases} 1 & w_j \geq 0 \\ -1 & w_j < 0 \end{cases}$$

Basically, $D(w)_{j,j}$ defines the profitable direction of change for the j^{th} attribute.

Let V be the set of feasible attribute values for agents. Given D_i, we constrain an agent's changed attributes by

$$x_i + D_i(w)d_i \in V .$$

For example, if, as in the spam case, all attribute vectors are binary valued, we have $V = \{0,1\}^n$. In other cases, it is possible to have different constraints. For example $V = \{x : l \leq x \leq u\}$ imposes simple lower and upper bounds on the attributes.

Let the reservation cost of being labeled a positive example be r_i for agent i. We say that each self-interested agent will solve the following optimization problem given (w, b), where they minimize the cost of modification subject to being classified as positive. Introducing the additional constraint, $x_i + D_i(w)d_i \in V$, into the agent's problem in Chapter 4, we get the constrained agent's problem:

$$\begin{aligned} &\min c_i' d_i \\ &st \\ &w'\left[x_i + D_i(w)d_i\right] + b \geq 1 \\ &c_i' d_i \leq r_i \\ &x_i + D_i(w)d_i \in V \end{aligned} \tag{1}$$

Note that if $w'x_i + b \geq 1$ then $d_i = 0$ is a feasible solution.

If the constrained agent's problem defined above has a feasible solution $d_i^*(w,b)$ then the agent has enough reservation cost to modify his attributes for a given (w,b). We set $d_i^*(w,b)=0$ for the case where the problem is infeasible. Taking into account this potential strategic behavior, the principal solves the following problem:

$$S_1: \quad \min_{w,b} w'w$$
$$s.t. \quad y_i\left\{w'\left[x_i + D_i(w)d_i^*(w,b)\right]+b\right\} \geq 1 \qquad i=1,...,\ell$$

In Chapter 4, for the case of unconstrained agents (agents without the additional $x_i + D_i(w)d_i \in V$ constraint), we characterized an optimal solution for the above problem for separable datasets under the assumption that r_i and c_i are constant (say r and c) across agents. We showed that if (w^*,b^*) solves S_0 then under the assumptions given, $\left(\frac{2}{2+t^*}w^*, \frac{2b^*-t^*}{2+t^*}\right)$ solves S_1 where $t^* = r\max_k \left|w_k^*\right|/c_k$. Thus a principal anticipating strategic behavior of agents will use a classifier that is a scaled and shifted version of (w^*,b^*) determined without taking into consideration strategic behavior. The shift and scaling depend on the cost structure for modifying attribute values (c), the reservation cost for being labeled as a positive case (r), and (w^*,b^*).

However, this solution is not applicable to the constrained version of interest because of several facts. First, in Chapter 4, we find that an agent optimally chooses to modify only one attribute for a given (w,b). But due to the constrained nature of (1), we find that in the case of constrained agents, each agent might need to modify multiple attributes. The decision of which to modify is highly dependent on the particular

attribute vector, x_i and the nature of the constraints imposed on the agent. Generally speaking, in constrained cases, there is no guarantee that the optimal move will be in only one direction. On the contrary, it is typically a combination of movements in different directions for the cases we have seen. Also in the unconstrained version, if the modification of a particular attribute is found to be optimal for one agent then that same particular attribute is optimal for all agents in the sample space. But in the constrained version, there is no guarantee that the optimal movement is the same across all agents. Thus, Theorem 1 as given in Chapter 4 is not applicable since the shifting and scaling may not force linear separability after the moves are realized when, especially for some marginal positive agents, the necessary modification to stay on the positive side of the scaled and shifted hyperplane is not feasible. This may allow some negative agents, with enough reservation cost, to move and blend into positive agents. Thus, we face the issue of nonseparability caused by strategic behavior. Since linear separability is not guaranteed any more, we turn our focus to the general Strategic Learning formulation developed in Chapter 4. Assuming that the principal incurs fixed costs of misclassification C_1 and C_{-1}, for positive and negative misclassifications respectively, S_1 reduces to the following:

$$S_1'\colon \quad \min_{w,b} w'w + \sum_{i=1}^{\ell} C_{y_i}\xi_i + \lambda \sum_{y_i=+1} q_i(w,b)$$
$$s.t. \quad y_i\left\{w'x_i + q_i(w,b) + b\right\} \geq 1 - \xi_i \qquad i = 1,\ldots,\ell$$

where

$$q_i(w,b)=\begin{cases} 0 & \text{if} \quad 1-b-w'x_i<0 \\ 0 & \text{if} \quad 1-b-w'x_i>z_i \\ 1-b-w'x_i & \text{otherwise} \end{cases}$$

and

$$z_i \equiv r_i \max_{j \ni (c_i)_j \neq 0} \frac{|w_j|}{(c_i)_j}$$

for $\lambda \geq 0$ and $C_{y_i} > 0$. ξ_i is the margin slack variable measuring the shortfall of a point in its margin from the hyperplane. Noticing that the objective function of S_1' has three components, $w'w$, is one over the square root of the margin, (the margin is to be maximized in statistical learning theory (Vapnik 1998)) while the second component, $\sum_{i=1}^{\ell} C_{y_i}\xi_i$, is the sum of individual misclassification costs. The third component, $\lambda \sum_{y_i=+1} q_i(w,b)$, accounts for the extra effort that needs to be exerted by positive agents as a result of strategic behavior (see Chapter 4 for details). In Chapter 5 we defined the unconstrained version of the of the Strategic Learning problem which is

$$f(w,b)=m(\|w\|)+\sum_{i=1}^{\ell} C_{y_i}\xi_i(w,b)+\lambda \sum_{y_i=+1} q_i(w,b)$$

where $\xi_i(w,b)=\max\left(0,1-y_i\left[w'\left(x_i+D(w)d_i\right)+b\right]\right)$, $\|w\|$ is a norm of w and $m(\ )$ an increasing function with $m(0)=0$. Then, we developed Algorithm 1 which finds an epsilon optimal solution to $\varepsilon - \min_b f(b \mid w)$ where $f(b \mid w)$ denotes the function $f(w,b)$ for a fixed w

In that chapter, Algorithm 1 is used by a Genetic Algorithm to develop a solution to the Strategic Learning problem for unconstrained agents. The fundamentals of Algorithm 1 are explained thoroughly in Chapters 4 and 5 but we provide a summary of the general idea of the algorithm here since in this chapter, we propose a similar algorithm for the constrained agents case..

The key point of Algorithm 1 is to search over finite inflection and discontinuity points of the objective function that are caused by strategic behavior of agents. The function is linear between two such consecutive points. In other words, an agent's strategic behavior can be explained for a fixed w (b is variable) and consists of different regions where the agent can take different strategic actions (move or don't move). Algorithm 1 is a search over these inflection and discontinuity points of these regions for each agent and the proof of optimality follows from knowing that between two such consecutive points the cost function is linear, meaning only the end points need be evaluated. Algorithm 1 examines each of these points which are potential values of b and evaluates the total cost at these points to find the one with the lowest cost. This finds a b providing an epsilon optimal value for S_1' for fixed w .

In this chapter, we contribute by bringing in additional constraints to an agent's problem which affects the way the movement of each agent is determined with respect to a given hyperplane. In that respect, the general idea of Algorithm 1 is applicable to our case but with the emphasis that the calculation of inflection and discontinuity points is dependent on the additional constraints imposed on the agent.

In the next section, we first look at the agent behavior under additional constraints specific to spam categorization problem and develop Algorithm 2 that finds an optimal

solution to constrained agent's problem. Later, we develop Algorithm 3 that alters Algorithm 1 to take into account the specific form of agent constraints that arise in a spam categorization application. The Genetic Algorithm developed in Chapter 5 is not affected by the issues that arise due to the introduction of additional agent constraints and thus generalizes to our case of constrained agents.

Application to Spam Filtering

In this section we apply our results to the spam categorization problem and develop algorithms specific to that problem. We choose to present emails in binary format where $x_i[j]=1$ when the jth word is present in the ith email or $x_i[j]=0$ otherwise.

The emails are labeled such that spam emails corresponding to negative cases are assigned a label -1 whereas the non-spam emails corresponding to positive cases are assigned the label +1. In our scenario, spammers might modify their emails by adding or deleting words (a third possibility is replacing words which corresponds to deletion of one word and addition of another at the same time). Given this setting, the spammer's problem becomes:

$$\begin{aligned}
&\min c_i' d_i \\
&st \\
&w'\left[x_i + D_i(w)d_i\right] + b \geq 1 \\
&c_i' d_i \leq r_i \\
&x_i + D_i(w)d_i \in \{0,1\}^n
\end{aligned}$$

By setting

$$D_i(w)_{j,j} = \begin{cases} 1 & w_j > 0 \text{ and } x_{i,j} = 0 \\ -1 & w_j < 0 \text{ and } x_{i,j} = 1 \\ 0 & \textit{otherwise} \end{cases}$$

we get a simpler version:

$$\min c_i' d_i$$
$$st$$
$$w'\left[x_i + D_i(w)d_i\right] + b \geq 1$$
$$c_i' d_i \leq r_i$$
$$d_i \in \{0,1\}$$

In this case there are only two possible modifications that a spammer can make on a particular attribute $x_i[j]$. If the true attribute value is 1 then the change can be either -1 or 0 or if the attribute value is 0 then the change can be either +1 or 0. These facts are both captured by the design of $D_i(w)$ along with the constraint $d_i \in \{0,1\}$ guaranteeing that $D_i(w)d_i$ gives a legitimate modification.

As an example, consider the following spam email

Spam $(y_i = -1)$: $x_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$, and $w = \begin{bmatrix} -0.8 \\ 1.2 \\ 0 \\ -0.5 \end{bmatrix}$.

Let $\overline{x_i} = x_i + D_i(w)d_i$. Clearly,

$$D_i(w)_{j,j} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}$$

and one possible modification that the spammer can make is

$$d_i = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

changing x_i to

$$\overline{x}_i = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

We assume that r_i is an integer value and also c_i, the unit cost of changing an attribute, is identical for each attribute and is equal to 1. It is hard to argue to the contrary since changing any one attribute is no harder than changing any other. With these assumptions, the agent problem reduces to a form of a binary knapsack problem and we develop Algorithm 2 which is a greedy algorithm for finding an optimal solution to agent's problem for a given (w, b).

The agent's problem resembles the binary knapsack problem with a similar goal of packing the knapsack with objects in order to maximize the total value of the packed objects without exceeding the knapsack's capacity. However, in our case, the goal is to minimize $c_i' d_i$ while we satisfy the constraint $w'\left[x_i + D_i(w)d_i\right] + b \geq 1$ which can be thought of as the knapsack capacity with the exception that we allow for the last object added to exceed the knapsack capacity. This corresponds to a surplus in the constraint $w'\left[x_i + D_i(w)d_i\right] + b \geq 1$. Essentially, in constrained agent's problem, each $|w_i|$ can be thought of as an object and our goal is to satisfy the constraint $w'\left[x_i + D_i(w)d_i\right] + b \geq 1$ without violating the reservation cost constraint ($c_i' d_i \leq r_i$) which makes it more involved than a regular knapsack problem. More specifically, if the reservation cost constraint is violated before $w'\left[x_i + D_i(w)d_i\right] + b \geq 1$ is met, then $d_i^* = 0$

Although greedy algorithms fail to always provide an optimal solution to binary knapsack problems, in our case since we assume $c_i = 1$ and allow for the last object

added to exceed the capacity, a greedy algorithm proves optimal. A key point is that since $c_i = 1$ (i.e., the cost of modification is equal for all attributes), the agent is indifferent to choose a modification among feasible ones since they all cost the same. Hence, the proof of optimality follows from the fact that the agent will choose a modification that will give a maximum movement since the goal is to satisfy $w'\left[x_i + D_i(w)d_i\right] + b \geq 1$ while minimizing $c_i' d_i$.

Now, define L to be the list of indexes such that $\left|w_{L_i}\right| \geq \left|w_{L_{i+1}}\right|$ for $\mathrm{L}_i, \mathrm{L}_{i+1} \in L$. In other words, L is the list of attribute indexes sorted according to the magnitude of the corresponding w_i (in descending order). For example, for

$$w = \begin{bmatrix} -0.8 \\ 1.2 \\ 0 \\ -0.5 \end{bmatrix},$$

L would be $L = \{1, 0, 3, 2\}$.

Essentially, Algorithm 2 takes (w, b) and x_i (an arbitrary hyperplane and an attribute vector) as inputs and then outputs $d_i^*(w, b)$ which is the minimum cost modification for an agent in order to be classified as positive, if possible depending on the reservation cost. By sequentially visiting each attribute index drawn from the list L, the algorithm checks if a modification is possible on the current attribute given $D_i(w)_{j,j}$. If true, the necessary modification is made. The algorithm stops when the reservation cost constraint will be violated or a positive classification is achieved or when the indexes contained in L are all visited. The last step is to check if the movement is big enough for

the agent to be able to get classified as a positive case otherwise the agent will prefer not to move (setting $d_i^*(w,b)=0$).

Algorithm 2

Input: (w,b) and x_i

Set $d_i=0$, $sum=w'x_i+b$, $r=0$ and $j=1$

while (($sum<1$) and ($j\le|L|$) and ($r\le r_i$)){

if $D(w)_{L_j,L_j}\neq 0$ {

$sum \leftarrow sum + D(w)_{L_j,L_j} * \mathrm{w}_{\mathrm{L}_j}$, $d_{i,L_j}=1$, $r{+}{+}$

}

$j{+}{+}$

}

if $(sum<1)$ $d_i=0$

Output: $d_i^*(w,b)$

Algorithm 2 helps to explain why Theorem 1 does not apply in our case. If the above algorithm finds a non-zero solution, $d_i^*(w,b)$ corresponding to the modification of a set of attributes (or words) then the spammer has enough reservation cost to modify an email. However, for Theorem 1 to hold, the same set of attributes has to be optimal for every email. This is not guaranteed in our case for several reasons. First, for example, if the deletion of ith word is found to be a part of an optimal modification for a particular email, then this implies that ith word exists in that email. However, there is no guarantee that it exists in the rest of the emails too. In fact, the ith word may not even exist in any of

the remaining emails which makes the deletion of that word infeasible. Thus, the optimal modification for each email may be different which contradicts the basic idea of Theroem 1.

Given the agent's problem, the principal's problem can be written as

$$\min_{w,b} w'w + \sum_{i=1}^{\ell} C_{y_i}\xi_i$$
$$s.t. \quad y_i\left\{w'\left[x_i + D_i(w)d_i\right] + b\right\} \geq 1 - \xi_i \qquad i = 1,...,\ell$$
$$0 \leq x_i + d_i \leq 1$$
$$c_i'd_i \leq r_i$$
$$d_i \in \{0,1\}$$

Recall that the objective here is to minimize $f(w,b) = m(\|w\|) + \sum_{i=1}^{\ell} C_{y_i}\xi_i(w,b) + \lambda \sum_{y_i=+1} q_i(w,b)$. However, in the spam categorization problem, it is important to realize that positive agents are non-strategic since it is reasonable to think that the author of a non-spam email is unlikely to change the content of his/her email fearing that it might be detected by the spam filter as spam. The legitimate users of internet (users other than spammers, hackers etc.) normally do not engage in such strategic behavior so the term $\lambda \sum_{y_i=+1} q_i(w,b)$ is zero for the spam categorization problem since no effort will be exerted by positive agents.

Using Algorithm 1 ideas, we develop Algorithm 3 to produce an optimal shift in determination of an optimal b for $\varepsilon - \min_b f(b \mid w)$ where

$$f(b \mid w) = m(\|w\|) + \sum_{i=1}^{\ell} C_{y_i}\xi_i(b \mid w).$$

Figure 6-1 illustrates how the cost $\xi_i(w,b) = \max\left(0, 1 - y_i\left[w'\left(x_i + D(w)d_i\right) + b\right]\right)$ vary with b for the following case

Spam $(y_i = -1)$: $x_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$,

for $w = \begin{bmatrix} -0.8 \\ 1.2 \\ 0 \\ -0.5 \end{bmatrix}$, $c = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ and $C_{-1} = 1$.

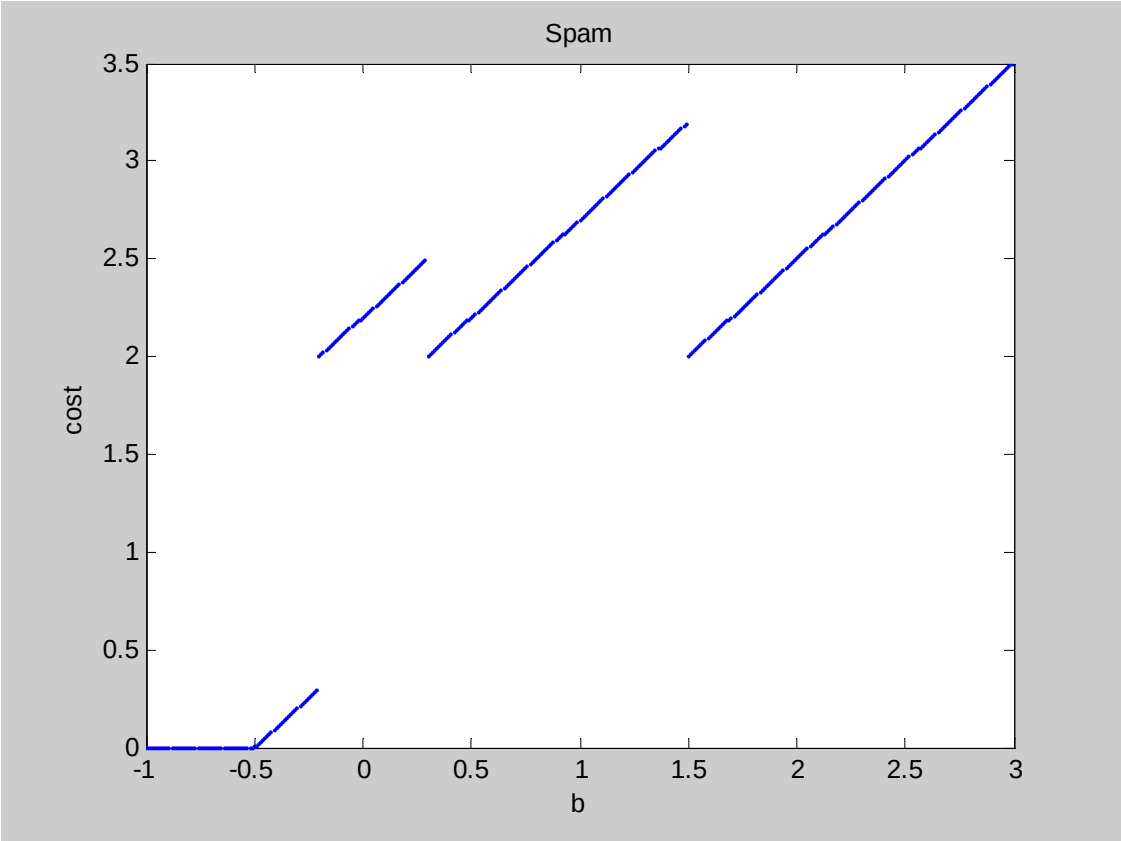


Figure 6-1. Spam email with $r_i = 2$.

Essentially, Figure 6-1 consists of linear regions corresponding to different modifications ($d_i^*(w,b)$) of the spam email assuming the spammer uses the greedy strategy (Algorithm 2). Each region is characterized by the corresponding $d_i^*(w,b)$ where the discontinuities occur as spammer needs to make one more modification to $d_i^*(w,b)$ in order to achieve or maintain the positive classification gained by strategic behavior.

As an example, in Figure 1 there are three discontinuity points. The first discontinuity point at $b = 1.5$ occurs when agent makes his first available modification to achieve positive classification (i.e., he sets

$$d_i^* = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

to

$$d_i^* = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}).$$

The second discontinuity point at $b = 0.3$ occurs when spammer makes his second available modification by changing the current

$$d_i^* = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

to

$$d_i^* = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

and thus maintains the positive classification.

The last discontinuity point at $b = -0.2$ is when the agent can not make any more modifications to maintain positive classification so chooses not to make any modifications by setting

$$d_i^* = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

As pointed out earlier, in the spam categorization problem, positive agents are non-strategic so we safely assume that $d_i^*(w,b) = 0$ for non-spam cases. Thus, this leads us to only focus on negative (spam) cases. However, we still analyze non-spam cases to determine the points of inflections of the cost function $\xi_i(w,b)$.

Figure 6-2 illustrates how $\xi_i(w,b)$ changes with b for the following non-spam email with no strategic behavior.

Non-spam $(y_i = 1)$: $x_i = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$, for $w = \begin{bmatrix} -0.8 \\ 1.2 \\ 0 \\ -0.5 \end{bmatrix}$, $c = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ and $C_1 = 1$.

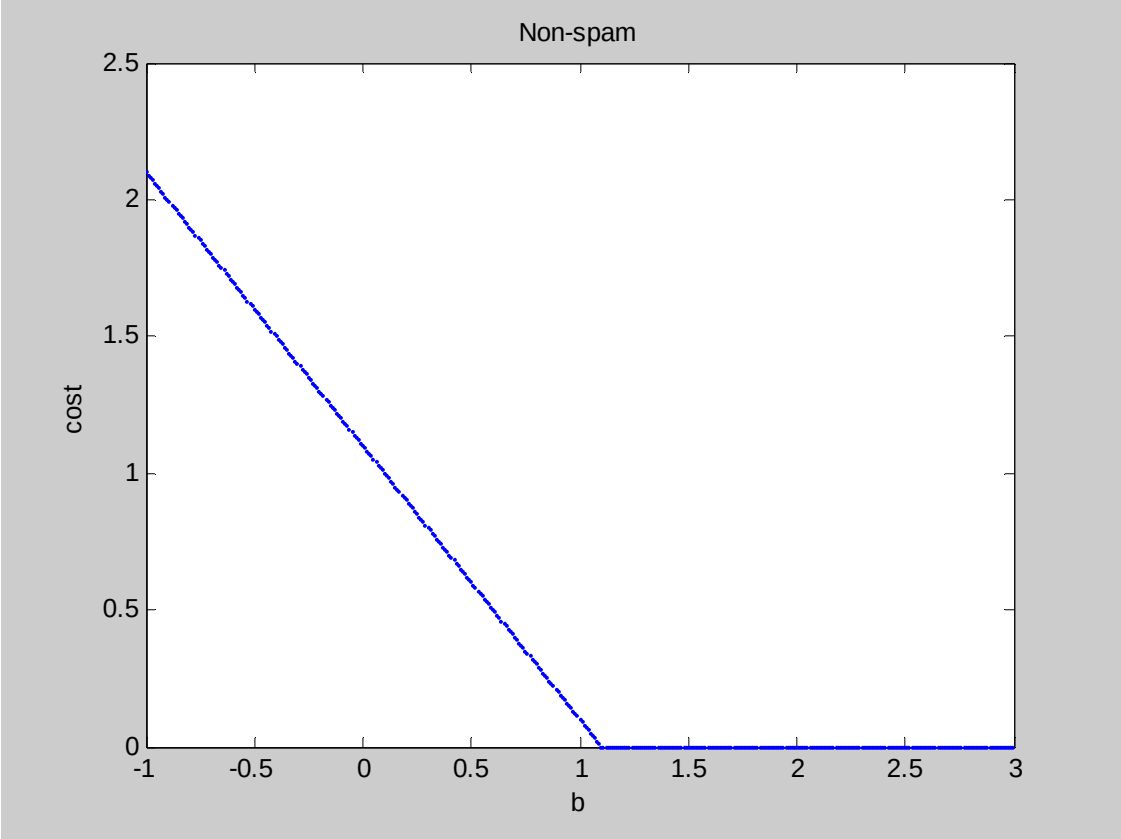


Figure 6-2. Non-Spam email without strategic behavior.

In the following, we develop Algorithm 3 which is a search over the discontinuity and inflection points of each agent. As was mentioned before, Algorithm 3 is a modified form of Algorithm 1 with the exception that Algorithm 3 is developed for constrained agents for spam categorization problem. Details of Algorithm 1 can be found in Chapters 4 and 5. The key idea is to search over the finite discontinuity and inflection points of the different regions of agent behavior. For non-strategic positive agents, the only inflection point is at $1-w'x_i$. For negative agents, at the last point of discontinuity (when all available modifications are done), we evaluate the function $f(b \mid w)$ at an epsilon lower point because the function is right-continuous at all the discontinuity points and our goal is to minimize the function. There are a finite number of such points and Algorithm 3

examines each of these points and evaluates the total cost function $f(b \mid w)$ at these points.

Let $\varepsilon > 0$ be sufficiently small and fixed.

Algorithm 3

Input: (w)

Set $\alpha = \phi$

for ($i = 1, \ldots, \ell$) {

if ($y_i = +1$)

$\alpha \leftarrow \alpha \cup \{1 - w' x_i\}$

else{

Set $z_i = 0$, $r = 0$, $j = 1$

$\alpha \leftarrow \alpha \cup \{1 - w' x_i\}$

while ($j \le |L|$ and $r \le r_i$){

if $D_i(w)_{L_j, L_j} \neq 0$ {

$z_i \leftarrow z_i + D_i(w)_{L_j, L_j} * \mathrm{w}_{\mathrm{L}_j}$, $\alpha \leftarrow \alpha \cup \{1 - z_i - w' x_i\}$,

$r{+}{+}$

}//if

$j{+}{+}$

}//while

$\alpha \leftarrow \alpha \cup \{1 - z_i - w' x_i - \varepsilon\}$

if ($z_i < 2$) $\alpha \leftarrow \alpha \cup \{-1 - w' x_i\}$

}//else

}//for

Set $f \leftarrow \infty$, $b \leftarrow \infty$

for each $\hat{b} \in \alpha$ {

$\hat{f} \leftarrow f\left(\hat{b} \mid w\right)$

if $\hat{f} < f$ then { $f \leftarrow \hat{f}$, $b \leftarrow \hat{b}$ }

}//for

Output: b

Conclusion

In this chapter we investigated the situation where there is a need to enforce constraints on agent behavior in the context of Strategic Learning and we extend the concept of Strategic Learning to constrained agents. We have selected spam categorization problem as a particular example and developed algorithms specific to that problem.

This chapter presents one crucial extension of Strategic Learning and stands as an example of how one can modify the general Strategic Learning model to accommodate for particularities that arise due to the nature of the specific application. In that respect, it is an attempt to deviate from the ideal setting to help analyze more realistic situations like the case of constrained agents.

The problem of constrained agents obviously is not limited to spam categorization and can be observed in many other problems which involve agents with limited strategic behavior such as fraud detection.

CHAPTER 7
CONCLUSION

In this study we investigated the Strategic Learning problem which is defined as the task of a principal whose goal is to discriminate between certain type of agents which are self-interested, utility maximizing and decision making units. We concentrated on linear discriminant functions for binary classification and focused on support vector machines.

In both Chapters 2 and 3, we provide an overview of Strategic Learning with the exception that we intend to reach different type of readers. In Chapter 4, we give a comprehensive and intricate study of Strategic Learning and provide the details of the model. In Chapter 5, we develop a Genetic Algorithm for Strategic Learning to solve larger versions of the problem. In Chapter 6, we extend the Strategic Learning model to handle more complex agent behaviors.

This study is organized as a collection of articles, each of which corresponds to one chapter of the entire study. Each chapter is complete within itself and includes a conclusion and future work section related with the aspects of the study covered in that specific chapter. Due to this self-contained style of preparation, we ask the reader to refer to those sections for a detailed description of possible future work areas.

APPENDIX
PROOF OF THEOREM 1

Theorem 1 states that when the training set is separable, a solution to the non-strategic SVM problem (P1) can be perturbed to a solution to the strategic SVM problem (P2). We start our proof by first showing a simple result.

Lemma 1

If $\left(\bar{w},\bar{b}\right)$ solves P2 then

- (a) $d_i^*\left(\bar{w},\bar{b}\right)=0$ for all $y_i=-1$
- (b) $d_i^*\left(\bar{w},\bar{b}\right)=r\max_k\left|\bar{w}_k^*\right|/c_k$ for at least one i where $y_i=1$

Proof:

- (a) Since $\left(\bar{w},\bar{b}\right)$ solves P2 no truly negative agent can achieve a positive classification so there is no incentive for such agents to move.
- (b) At least one positive support vector will have been moved its maximal amount or else the svm margin can be increased, which would be a contradiction. The maximum move is $r\max_k\left|\bar{w}_k\right|/c_k$.

□

Theorem 1

$\left(w^*,b^*\right)$ solves P1 if and only if $\left(\frac{2}{2+t^*}w^*,\frac{2b^*-t^*}{2+t^*}\right)$ solves P2 where t^* is given by $t^*=r\max_k\left|w_k^*\right|/c_k$

Proof:

Since S has at least two elements having opposite labels, P1 must have $w^* \neq 0$. Since c and r are positive, $t^* > 0$. Let $k = \arg\max_k \left|w_k^*\right| / c_k$. We start by first showing feasibility of $\left(\frac{2}{2+t^*} w^*, \frac{2b^* - t^*}{2+t^*}\right)$ to P2. We break this part of the proof into two parts, one handling positively labeled agents and the other negatively labeled ones.

Case $y_i = 1$:

We start by noting that

$$w^{*\prime} x_i + b^* \geq 1$$

for all $y_i = 1$ and is exactly satisfied for positive support vectors. Let i index a positive support vector. Now

$$z_i\left(\frac{2}{2+t^*} w^*, \frac{2b^* - t^*}{2+t^*}\right) = \frac{\max\left(0, 1 - \left(\frac{2b^* - t^*}{2+t^*} + \frac{2}{2+t^*} w^{*\prime} x_i\right)\right)}{\left|\frac{2}{2+t^*} w_k^*\right|}$$

and

$$1 - \left(\frac{2b^* - t^*}{2+t^*} + \frac{2}{2+t^*} w^{*\prime} x_i\right) = 1 + \frac{t^*}{2+t^*} - \frac{2}{2+t^*}\left(w^{*\prime} x_i + b^*\right) = \frac{2t^*}{2+t^*}$$

Furthermore we need $c_k z_i\left(\frac{2}{2+t^*} w^*, \frac{2b^* - t^*}{2+t^*}\right) \leq r$. Now

$$c_k z_i\left(\frac{2}{2+t^*} w^*, \frac{2b^* - t^*}{2+t^*}\right) = \frac{c_k}{\left|w_k^*\right|} \frac{2t^*}{2+t^*} = r \frac{2}{2 + r\left|w_k^*\right| / c_k} \leq r$$

Thus, $d_i^*\left(\frac{2}{2+t^*}w^*,\frac{2b^*-t^*}{2+t^*}\right)=z_i\left(\frac{2}{2+t^*}w^*,\frac{2b^*-t^*}{2+t^*}\right)1_k$ is a feasible move for agent i with $\left(\frac{2}{2+t^*}w^*,\frac{2b^*-t^*}{2+t^*}\right)$. Now consider

$$
\begin{aligned}
& y_i\left\{\frac{2}{2+t^*}w^*\left[x_i+D(\bar{w})d_i^*\left(\bar{w},\bar{b}\right)\right]+\frac{2b^*-t^*}{2+t^*}\right\} \\
&=\frac{2}{2+t^*}w^{*\prime}x_i+\frac{2t^*}{2+t^*}+\frac{2b^*-t^*}{2+t^*} \\
&=\frac{2}{2+t^*}\left(w^{*\prime}x_i+t^*+b^*-t^*/2\right) \\
&=\frac{2}{2+t^*}\left(1+t^*/2\right)=1
\end{aligned}
$$

So after the move we see that positive support vector agents become positive support vectors of the new LDF $\left(\frac{2}{2+t^*}w^*,\frac{2b^*-t^*}{2+t^*}\right)$. Any other positive agent will either not have to adjust attributes (because it was far enough from the original LDF hyperplane) or will have to adjust to a value no greater than $\frac{2t^*}{2+t^*}$.

Case $y_i=-1$:

We start by noting that

$$-w^{*\prime}x_i-b^*\geq 1$$

for all $y_i=-1$ and is exactly satisfied for negative support vectors. Let i index a negative support vector. There are two cases. In the first, the margin of the P1 solution may be larger than the maximal move a negative agent is willing to make so the agent would gain nothing by moving. This means that $d_i^*\left(w^*,b^*\right)=0$. The second case has $d_i^*\left(w^*,b^*\right)\neq 0$. As in the positive case, we focus just on the negative support vectors

noting that any other negative agent will either not have to adjust attributes (because it was too far from the original LDF hyperplane to make a difference) or will have to adjust to a value no greater than at least as much as the support vectors.

Case $d_i^*\left(w^*, b^*\right) = 0$:

Assume $d_i^*\left(w^*, b^*\right) = 0$. Then we know that $1 - \left(b^* + w^{*\prime} x_i\right) = 2$ and then that

$$\frac{c_k}{\left|w_k^*\right|} 2 > r$$

(since this agent cannot move). Now consider

$$1 - \left(\frac{2b^* - t^*}{2 + t^*} + \frac{2}{2 + t^*} w^{*\prime} x_i \right) = 1 - \frac{2b^* - t^*}{2 + t^*} - \frac{2}{2 + t^*} w^{*\prime} x_i$$

$$= 1 + \frac{t^*}{2 + t^*} - \frac{2}{2 + t^*}\left(w^{*\prime} x_i + b^*\right) = 2$$

Then $d_i^*\left(\frac{2}{2 + t^*} w^*, \frac{2b^* - t^*}{2 + t^*} \right) = 0$ since no move is possible also. Thus we have

$$y_i \left\{ w' \left[x_i + D\left(\bar{w}\right) d_i^*\left(\bar{w}, \bar{b}\right) \right] + \frac{2b^* - t^*}{2 + t^*} \right\}$$

$$= -\frac{2}{2 + t^*} w^{*\prime} x_i - \frac{2b^* - t^*}{2 + t^*}$$

$$= \frac{t^*}{2 + t^*} - \frac{2}{2 + t^*}\left(w^{*\prime} x_i + b^*\right) = 1$$

This shows that a negative support vector under P1 remains one under the new LDF $\left(\frac{2}{2 + t^*} w^*, \frac{2b^* - t^*}{2 + t^*} \right)$.

Case $d_i^*\left(w^*, b^*\right) \neq 0$:

Assume $d_i^*\left(w^*, b^*\right) \neq 0$. Then we know

$$1-\left(\frac{2b^*-t^*}{2+t^*}+\frac{2}{2+t^*}w^{*\prime}x_i\right)=1+\frac{t^*}{2+t^*}-\frac{2}{2+t^*}\left(w^{*\prime}x_i+b^*\right)=2$$

and that

$$d_i^*\left(\frac{2}{2+t^*}w^*,\frac{2b^*-t^*}{2+t^*}\right)=\frac{2}{\left|w_k^*\right|}1_k$$

so

$$\begin{aligned}
&y_i\left\{\frac{2}{2+t^*}w^{*\prime}\left[x_i+D\left(\bar{w}\right)d_i^*\left(\bar{w},\bar{b}\right)\right]+\frac{2b^*-t^*}{2+t^*}\right\}\\
&=-\left(\frac{2}{2+t^*}w^{*\prime}x_i+2+\frac{2b^*-t^*}{2+t^*}\right)\\
&=-\left(-\frac{t^*}{2+t^*}+2+\frac{2}{2+t^*}\left(w^{*\prime}x_i+b^*\right)\right)\\
&=-1
\end{aligned}$$

which shows, under $\left(\frac{2}{2+t^*}w^*,\frac{2b^*-t^*}{2+t^*}\right)$ a negative support vector wouldn't move.

Then we have

$$\begin{aligned}
&y_i\left\{\frac{2}{2+t^*}w^{*\prime}\left[x_i+D\left(\bar{w}\right)d_i^*\left(\bar{w},\bar{b}\right)\right]+\frac{2b^*-t^*}{2+t^*}\right\}\\
&=-\left(\frac{2}{2+t^*}w^{*\prime}x_i+\frac{2b^*-t^*}{2+t^*}\right)\\
&=-\left(-\frac{t^*}{2+t^*}+\frac{2}{2+t^*}\left(w^{*\prime}x_i+b^*\right)\right)\\
&=1
\end{aligned}$$

This shows that a negative support vector under P1 remains one under the new

LDF $\left(\frac{2}{2+t^*}w^*,\frac{2b^*-t^*}{2+t^*}\right)$.

We now start with a solution, $\left(\bar{w},\bar{b}\right)$, to P2. As above, we break this part of the proof into two parts, one handling positively labeled agents and the other negatively labeled ones. We start with $\left(\bar{w},\bar{b}\right)$ and consider $\left(\frac{2+t^*}{2}\bar{w},\frac{\left(2+t^*\right)\bar{b}+t^*}{2}\right)$ as a feasible solution of P1.

Case $y_i = 1$:

We start by noting that

$$\bar{w}'\left(x_i + D\left(\bar{w}\right)d_i^*\left(\bar{w},\bar{b}\right)\right)+\bar{b} \geq 1$$

for all $y_i = 1$ and is exactly satisfied for positive support vectors. By Lemma 1b we have there is an i such that

$$\begin{aligned}
&\frac{2+t^*}{2}\bar{w}'x_i + \frac{\left(2+t^*\right)\bar{b}+t^*}{2} \\
&= \frac{2+t^*}{2}\left(\bar{w}'x_i+\bar{b}\right)+\frac{t^*}{2} \\
&= \frac{2+t^*}{2}\left(1-r\max_k\left|\bar{w}_k\right|/c_k\right)+\frac{t^*}{2} \\
&= 1+t^*-r\max_k\left|\frac{2+t^*}{2}\bar{w}_k\right|/c_k
\end{aligned}$$

Thus we see that this maximally shifted support vector can be a support vector of an unshifted problem provided

$$t^* = r\max_k\left|\frac{2+t^*}{2}\bar{w}_k\right|/c_k$$

Now consider any other positive agent, call him agent j. Then

$$\frac{2+t^*}{2}\bar{w}'x_j+\frac{\left(2+t^*\right)\bar{b}+t^*}{2}$$
$$=\frac{2+t^*}{2}\left(\bar{w}'x_j+\bar{b}\right)+\frac{t^*}{2}$$
$$\geq\frac{2+t^*}{2}\left(1-D\left(\bar{w}\right)d_j^*\left(\bar{w},\bar{b}\right)\right)+\frac{t^*}{2}$$
$$\geq\frac{2+t^*}{2}\left(1-D\left(\bar{w}\right)d_i^*\left(\bar{w},\bar{b}\right)\right)+\frac{t^*}{2}$$
$$=\frac{2+t^*}{2}\left(1-r\max_k\left|\bar{w}_k\right|/c_k\right)+\frac{t^*}{2}$$
$$=1+t^*-t^*=1$$

showing feasibility to P1.

Case $y_i=-1$:

We start by noting that

$$-\bar{w}'\left(x_i+D\left(\bar{w}\right)d_i^*\left(\bar{w},\bar{b}\right)\right)-\bar{b}\geq 1$$

for all $y_i=-1$ and is exactly satisfied for negative support vectors. By Lemma 1a we know $d_i^*\left(\bar{w},\bar{b}\right)=0$ for all $y_i=-1$. Thus

$$-\bar{w}'x_i-\bar{b}\geq 1$$

and is equality for negative support vectors. Consider the following for a negative support vector.

$$-\frac{2+t^*}{2}\bar{w}'x_i-\frac{\left(2+t^*\right)\bar{b}+t^*}{2}$$
$$=-\frac{2+t^*}{2}\left(\bar{w}'x_i+\bar{b}\right)-\frac{t^*}{2}$$
$$=\frac{2+t^*}{2}-\frac{t^*}{2}=1$$

Similarly, for other negative agents we get

$$-\frac{2+t^*}{2}\bar{w}'x_i-\frac{\left(2+t^*\right)\bar{b}+t^*}{2}\geq 1$$

showing feasibility to P1.

Above we showed that an optimal solution, $\left(w^*,b^*\right)$, to P1 provides a feasible solution, $\left(\frac{2}{2+t^*}w^*,\frac{2b^*-t^*}{2+t^*}\right)$, to P2 and in the second part we showed that an optimal solution, $\left(\bar{w},\bar{b}\right)$, to P2 provides a feasible solution, $\left(\frac{2+t^*}{2}\bar{w},\frac{\left(2+t^*\right)\bar{b}+t^*}{2}\right)$, to P1. The following inequalities must hold.

$$w^{*\prime}w^*\leq\left(\frac{2+t^*}{2}\right)^2\bar{w}'\bar{w}$$

$$\bar{w}'\bar{w}\leq\left(\frac{2}{2+t^*}\right)^2 w^{*\prime}w^*$$

so

$$\left(\frac{2}{2+t^*}\right)^2 w^{*\prime}w^*\leq\bar{w}'\bar{w}\leq\left(\frac{2}{2+t^*}\right)^2 w^{*\prime}w^*$$

and then

$$\left(\frac{2}{2+t^*}\right)^2 w^{*\prime}w^*=\bar{w}'\bar{w}$$

completing the proof.

□

LIST OF REFERENCES

Abdel-Khalik, A. R., K. M. El-Sheshai. 1980. Information choice and utilization in an experiment on default prediction. *Journal of Accounting Research* **18(2)** 325-342.

Agrawal, R., T. Imielinski, A. Swami. 1993. Mining association rules between sets of items in large databases. *Proceedings of the ACM SIGMOD Conference on Management of Data,* Washington, D.C., 207-216.

Arnt, A., S. Zilberstein. 2005. Learning policies for sequential time and cost sensitive classification. *Proceedings of the KDD-05 Workshop on Utility-Based Data Mining*, Chicago, IL, 39-46.

Bajgier, S. M., A.V. Hill. 1982. An experimental comparison of statistical and linear programming approaches to the discriminant problem. *Decision Sciences* **13** 604-618.

Bennett, K. P., E. J. Bredensteiner. 1997. A parametric optimization method for machine learning. *INFORMS Journal on Computing* **9** 311-318.

Bhattacharyya, S., K.K. Tharakunnel. 2005. Reinforcement learning in learder-follower multiagent systems: Framework and an algorithm, Information and Decision Sciences, University of Illinois, Chicago.

Bishop, C. 1995. *Neural networks for pattern recognition*. Oxford University Press, New York, NY.

Ciraco, M., M. Rogalewski, G. Weiss. 2005. Improving classifier utility by altering the misclassification cost ratio. *Proceedings of the KDD-05 Workshop on Utility-Based Data Mining*, Chicago, IL, 46-53.

Cristianini, N., J. Shawe-Taylor. 2000. *An introduction to support vector machines and other kernel-based methods*. Cambridge University Press, Cambridge, UK.

Crone, S., S. Lessmann, R. Stahlblock. 2005. Utility based data mining for time series analysis: Cost sensitive learning for neural network predictors. *Proceedings of the KDD-05 Workshop on Utility-Based Data Mining*, Chicago, IL, 59-69.

Dalvi, N., P. Domingos, Mausam, S. Sanghai, D. Verma. 2004. Adversarial classification. *Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD),* Seattle, 99-108.

Davis, L. 1989. Adapting operator probabilities in genetic algorithms. *Proceedings of the Third International Conference on Genetic Algorithms*. Schaefer, J. D., ed. Morgan Kaufman, Los Altos, California, 61-69.

Duda, R. O., P. E. Hart. 1973. *Pattern classification and scene analysis.* Wiley, New York, NY.

Dumais, S., J. Platt, D. Heckerman, M. Sahami. 1998. Inductive learning algorithms and representations for text categorization. *Proceedings of the 7th International Conference on Information and Knowledge Management,* ACM Press, Bethesda, Maryland, 148-155.

Ehtamo, H.,M. Kitti, P. R. Hämäläinen. 2002. Recent studies on incentive design problems in game theory and management science. *Optimal Control and Differential Games. Essays in Honor of Steffen Jørgensen*. 121-134.

Eisenbeis, R. 1987. Discussion, Supplement to Srinivasan, V. and Kim, Y. H. 1987. Credit granting: A comparative analysis of classification procedures. *J. Fin*. **42(3)** 665-680.

Elkan, C. 2001. The foundations of cost-sensitive learning. *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI'01),* Seattle, 973-978.

Fan, M., J. Stallaert, A. B. Whinston. 2003. Decentralized mechanism design for supply chain organizations using auction market. *Information System Research* **14(1)** 1-22.

Fawcett, T. 2003. "In vivo" Spam filtering: A challenge problem for KDD. *SIGKDD Explorations*. **5(2)** 140-148.

Fawcett. T., F. Provost. 1997. Adaptive fraud detection. *Data Mining and Knowledge Discovery*. **1(3)** 291-316.

Fisher, R. A. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics.* **7** 179-188.

Freed, N., F. Glover. 1981a. A linear programming approach to the discriminant problem. *Decision Sciences* **12** 68-74.

Freed, N., F. Glover. 1981b. Simple but powerful goal programming models for discriminant problems. *European Journal of Operational Research* **7** 44-60.

Freed, N., F. Glover. 1982. Linear programming and statistical discrimination -- the LP side. *Decision Sciences* **13** 172-175.

Freed, N., F. Glover. 1986a. Evaluating alternative linear programming models to solve the two-group discriminant problem. *Decision Sciences* **17** 151-162.

Freed, N., F. Glover. 1986b. Resolving certain difficulties and improving the classification power of LP discriminant analysis formulations. *Decision Sciences* **17** 589-595.

Fung, G., O. L Mangasarian. 2002. A feature selection Newton method for support vector machine classification. *Data Mining Institute Technical Report 02–03*, University of Wisconsin, Madison.

Genton, M.G. 2001. Classes of kernels for machine-learning: A statistics perspective. *Journal of Machine Learning Research* **2** 299-312.

Glover, F. 1990. Improved linear and integer programming models for discriminant analysis. *Decision Sciences* **21** 771-785.

Goldberg, D. E. 1989. Genetic *Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley, Reading, MA.

Hand, D. J. 1981. *Discrimination and Classification*. John Wiley & Sons, New York, NY.

Holte, R., C. Drummond. 2005. Cost-sensitive classifier evaluation. *Proceedings of the KDD-05 Workshop on Utility-Based Data Mining*, Chicago, IL, 3.

ILOG, ILOG CPLEX. 2005. *Reference manual and user manual*. V9.5, ILOG, Gentilly, France.

Joachims, T. 1998. Text categorization with support vector machines. *Proceedings of European Conference on Machine Learning (ECML)*, 137-142.

Kaelbling, L. P., M.L Littman, A. W. Moore. 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* **4** 1039-1069.

Kapoor, A., R. Greiner. 2005. Reinforcement learning for active model selection. *Proceedings of the KDD-05 Workshop on Utility-Based Data Mining*, Chicago, IL, 17-24.

Keyhani, A. 2003. Leader-follower framework for control of energy services. *IEEE Transactions on Power Systems* **18(2)** 837-841.

Koehler, G. J. 1989a. Characterization of unacceptable solutions in LP discriminant analysis. *Decision Science* **20** 239-257.

Koehler, G. J. 1989b. Unacceptable solutions and the hybrid discriminant model. *Decision Science* **20** 844-848.

Koehler, G. J. 1991. Linear discriminant functions determined by genetic search. *Journal on Computing* **3** 345-357.

Koehler, G. J., S.S. Erenguc. 1990a. Minimizing misclassifications in linear discriminant analysis. *Decision Science* **21(1)** 63-85.

Koehler, G. J., S.S. Erenguc. 1990b. Survey of mathematical programming models and experimental results for linear discriminant analysis. *Managerial and Decision Economics* **11** 215-225.

Laffont, JJ., D. Martimort. 2002. *The Theory of Incentives: The Principal-Agent Model*. Princeton University Press, Princeton, NJ.

Littman, M. L. 1994. Markov games as a framework for multi-agent reinforcement learning. *Proceedings of the Eleventh International Conference on Machine Learning*, New Brunswick, NJ, 157-163.

Mangasarian, O. 1965. Linear and nonlinear separation of patterns by linear programming. *Operations Research* **13** 444-452.

Mangasarian, O. 1994. Misclassification minimization. *Journal of Global Optimization* **5** 309-323.

Markowski, E. P., C. A. Markowski. 1985. Some difficulties and improvements in applying linear programming formulations to the discriminant problem. *Decision Sciences* **16** 237-247.

McCarthy, K., B. Zabar, G. Weiss. 2005. Does cost-sensitive learning beat sampling for classifying rare classes?. *Proceedings of the KDD-05 Workshop on Utility-Based Data Mining*, Chicago, IL, 69-78.

Melville, P., M. Saar-Tsechansky, F. Provost, R. Mooney. 2005. Economical active feature-value acquisition through expected utility estimation. *Proceedings of the KDD-05 Workshop on Utility-Based Data Mining*, Chicago, IL, 10-17.

Messier, W.F., Jr., J.V. Hansen. 1988. Inducing rules for expert system development: An example using default bankruptcy data. *Management Science.* **34(12)** 1403-1415.

Morrison, C., P. Cohen. 2005. Noisy information value in utility-based decision making. *Proceedings of the KDD-05 Workshop on Utility-Based Data Mining*, Chicago, IL, 34-39.

Muth, J. A. 1961. Rational expectations and the theory of price movements. *Econometrica.* **29 (6)** 315-35.

Provost, F. J. 2005. Toward Economic machine learning and utility based data mining. *Proceedings of the ACM SIGKDD Workshop on Utility-based Data Mining*, Chicago, IL, 1.

Rosenblatt, F. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*. **65(6)** 386-408.

Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning* **1** 81-106.

Quinlan, J. R. 1996. Decision trees and instance-based classifiers. *In CRC Handbook of Computer Science and Engineering.* A. B. Tucker, Ed., CRC Press, Boca Raton, FL.

Samuel, A. L. 1959. Some studies in machine learning using the game of checkers. *IBM Journal on Research and Development* **49** 210-229.

Schölkopf, B., A. J. Smola. 2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, MA.

Stam, A., E. A. Joachimsthaler.1989. Solving the classification problem in discriminant analysis via linear and nonlinear programming methods. *Decision Sciences* **20** 285–93.

Sutton , R. S.,A. G. Barto. 1998. *Reinforcement learning: An introduction.* MIT Press, Cambridge, MA.

Tam, K.Y., M.Y. Kiang. 1992. Managerial applications of neural networks: The case of bank failure predictions. *Management Science.* **38(7)**.

Vapnik, V. 1998. *Statistical learning theory*. John Wiley & Sons, New York, NY.

Vapnik, V. 1999. An Overview of Statistical Learning Theory. *IEEE Transactions on Neural Networks* **10** 988-999.

Vapnik,V. and Chervonenkis, A. 1981. The necessary and sufficient conditions for uniform convergence of means to their expectations. *Theory of Probability and its Applications* **26** 532-553.

Von Stackelberg, H. 1952. *The theory of market economy.* Oxford Univ. Press, London, UK.

Weiss, G., F. Provost. 2003. Learning when training data are costly: the effect of class distribution on tree induction. *Journal of Artificial Intelligence Research* **19** 315-354.

Zadrozny, B. 2005. One-benefit learning: Cost-sensitive learning with restricted cost information. *Proceedings of the KDD-05 Workshop on Utility-Based Data Mining*, Chicago, IL, 53-59.

BIOGRAPHICAL SKETCH

Fidan Boylu has received a B.S. degree in electrical and electronic engineering from Middle East Technical University in Turkey. After getting her MBA degree in 2002, she started working as a research and teaching assistant at the University of Florida. Her research involves classification of observations when the associated data will be deliberately manipulated by self interested agents. This novel idea expands results from statistical learning theory and the well-known principal agent problem. Her research shows that results from the statistical learning theory have to be modified to account for this strategic behavior. Applications of this idea include credit rating, spam filtering, and college admissions to name a few. She has presented her results at three international conferences, INFORMS 2005, DSI 2005 and HICCS 2006. Her work was nominated for a best paper award in HICCS 2006. Fidan is set to graduate and receive her PhD degree from University of Florida in the summer of 2006 and start an academic career in the Operations and Information Management Department at the University of Connecticut.