UCGE Reports
Number 20243

Department of Geomatics Engineering

Support Vector Machines for Land Use Change Modeling

(URL: http://www.geomatics.ucalgary.ca/research/publications/GradTheses.html)

by

Chenglin Xie

April 2006

UNIVERSITY OF CALGARY

Support Vector Machines for Land Use Change Modeling

by

Chenglin Xie

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF GEOMATICS ENGINEERING

CALGARY, ALBERTA

APRIL, 2006

ABSTRACT

Land use change modeling has been a topic of great concern in sustainable development research. It is a prerequisite to understanding the complexity of land use change. The primary objective of this research is to develop a novel approach for land use change modeling using Support Vector Machines (SVMs), with the capability to effectively address land use change data, which might be a mixture of continuous and categorical variables that might not be normally distributed. A SVMs land use change modeling framework was developed to classify land use change. Two enhancements were made to standard SVMs which improved the ability of SVMs to fit the characteristics and requirements of land use change modeling. The first improvement aimed to achieve high performance for unbalanced dataset and the second aimed to improve robustness. A case study of Calgary land use change demonstrated that the improved SVMs can achieve high and reliable performances.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Dr. Bo Huang, for his support and guidance throughout my graduate studies. His continual encouragement and advice were greatly appreciated. Also I would like to thank Dr. Richard Tay, my co-supervisor, for his support and comments on my research.

Thanks also to the professors, students, and staff of the Department of Geomatics Engineering for their discussions and suggestions on my studies. Particular thanks go to Dr. Yang Gao, Dr. Darka Mioc, Qiang Wu and Qiaoping Zhang.

I would like to thank my friends Ed Claussen, Mary Jane Claussen, and the Claussen family for their sincere help on making my time fruitful and enjoyable.

Most of all, my deepest gratitude goes to my lovely wife, Li Liu, for her love, unquestioned support and encouragement. It was her wishes and inspiration that propelled me through difficult times. I owe it all to her.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES AND ILLUSTRATIONS

LIST OF SYMBOLS, ABBREVIATIONS, NOMENCLATURE

ANN	Artificial Neural Network
CA	Cellular Automata
CT	Census Tract
ERM	Empirical Risk Minimization
GA	Genetic Algorithm
GWR	Geographic Weighted Regression
IBGP	International Geosphere-Biosphere Program
IDW	Inverse Distance Weighting
IF	Influence Function
IHDP	International Human Dimensions Program
IID	Independent and Identically Distributed
KF	Kernel Function
KKT	Karush-Kuhn-Tucker
LS-SVMs	Least Squares Support Vector Machines
LUCC	Land Use/Cover Change
MAS	Multi-Agent System
MADGIC	Maps, Academic Data, Geographic Information Center
NWMP	Northwest Mounted Police
OLS	Ordinary Least Squares
OSH	Optimal Separating Hyperplane
PAC	Probably Approximately Collect
PCP	Percentage of Correct Prediction
QP	Quadratic Programming
RBF	Radial Basis Function
RSVMs	Robust Support Vector Machines
SLR	Spatial Logistics Regress
SMO	Sequential Minimal Optimization
SRM	Structural Risk Minimization

SVs	Support Vectors
SVMs	Support Vector Machines
SVR	Support Vector Regression
TIN	Triangular Irregular Networks
VC	Vapnik-Chervonenkis
$\mathbf{w}'$	The transpose of vector $\mathbf{w}$

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

In recent years, sustainable development has become a topic of great concern among various sectors of society. Sustainable development seeks to meet the needs and aspirations of the present without compromising the ability of future generations to meet their own needs (World Commission on Environment and Development, 1987). In order to maintain sustainable development, the whole ecosystem (air, water, land, energy, flora and fauna) needs to be taken care of. Amongst all of these, land is essential not only because it is the habitat of human beings and our food and raw materials originate from it, but also because any disturbance to land by way of change in land use (e.g. conversion of forestland, agricultural into built-up land) is very difficult to reverse.

Under the umbrella of sustainable development, and stimulated by the joint international Land Use/Cover Change (LUCC) project of the International Geosphere-Biosphere Program (IBGP) and the International Human Dimensions Program on Global Environmental Change (IHDP) (Turner et al., 1995), detecting, monitoring, understanding, modeling, and projections of land use change from global to regional scale have attracted many research interests. In the past two decades, substantial work has been done in regards to land use change modeling (Baker, 1989; Agarwal et al., 2002). Various land use change models (VeldKamp and Lambin, 2001; Berling-Wolff and Wu, 2004) have been developed to help ecologists, urban planners, sociologists, administrators and policy makers better understand the complexity of land use change and to evaluate the impact of land use change on the environment.

A variety of techniques such as Markov chain analysis (Lopez et al., 2001), multiple regression analysis (Theobald and Hobbs, 1998), logistic regression (Wu and Yeh, 1997), artificial neural networks (ANNs) (Pajanowski et al., 2002; Li and Yeh, 2002), cellular automata (CA) (Wu, 1998; Wu, 2002), and genetic algorithm (GA) (Balling et al., 2004), are employed in land use change research. These models have demonstrated different levels of success in their specific applications; however, their drawbacks limit their efficiency in land use modeling. Markov chain analysis uses a transition matrix to describe the change of land use but cannot reveal the causal factors and their significance (Taha, 1997). Multiple regression requires that the error term of the regression expression be normally distributed. Logistic regression allows the factors to be a mixture of continuous and categorical variables but it assumes that the occurrence probability is linearly and additively related to the causal factors on a logistic scale (Cheng and Masser, 2003). If the assumption cannot be satisfied, the performance may significantly degrade. CA is limited by its simplicity and its inability to reveal causal factors (Torrens and O'Sullivan, 2001). Multi-agent system (MAS) is a microscopic simulation method and is therefore unable to fit the requirements of large scale modeling. Moreover, it is difficult to define the perception rule for the agent interactions in MAS (Parker and et al., 2001). Artificial neural network is a powerful method used to model nonlinear relationships but it is prone to overfitting the training data and cannot be relied upon to ensure the generalization performance (Sui, 1994; Karystinos and Pados, 2000).

To overcome the above mentioned shortcomings and to better address some important issues in land use change modeling (e.g. the imbalance of changed/unchanged land parcels and the robustness of the model), this research will develop a novel approach for land use change modeling using support vector machines (SVMs) and improving standard SVMs to ensure high performance for unbalanced and noisy data.

1.2 PROBLEM STATEMENT

Previous land use studies have demonstrated different levels of success in their specific applications; however, there are still several problems that need to be addressed to effectively and efficiently model land use change. Solutions to these problems will greatly improve the accuracy and reliability of land use change modeling.

First of all, land use change is a very complicated process through which humans and natural systems interact. Causal factors could be a mixture of continuous and categorical variables and dependent variables usually do not satisfy the normal distribution. The modeling methodology should be capable of incorporating these inputs and allowing them to deviate from the normal distribution (Cheng, 2003).

Secondly, land use change is a time-varying process. Thus, changes are generally detected and modeled in a series of relatively short time intervals. During each of the time intervals, the changed land parcels only account for a small part of the total land. Therefore, the input data for land use modeling is an unbalanced dataset. The modeling methodology should be able to handle such an unbalanced dataset, and not only achieve good performances for all of the inputs but also reveal the rules embedded in the minority with high accuracy.

Finally, land use change could be stimulated by a gamut of factors, ranging from spatial parameters to socioeconomic, political or even cultural factors. No single land use model can include all these factors. Hence, not all land use changes in the observed data are caused by the combination of the observed causal factors. Even some land parcels that are marked as low probability for change considering the causal factors in the land use change model may be affected by some unconsidered forces which might result in changed land use. These unconsidered forces may overwhelm the major factors in the model in some cases, but are not worthy to be considered in this particular model since they are not significant for the whole population or are too expensive to collect and quantify. Therefore, these samples should be treated as outliers if we only want to explore the relationship between land use change and the causal factor considered in the model. Outliers are quite common for land use change data. Special efforts should be made to take care of these outliers and give a reliable performance even when certain levels of disturbances exist in the sample data (noisy data).

1.3 RESEARCH OBJECTIVES

This research aims to effectively and efficiently model land use change by using SVMs which can effectively reveal the relationship between a mixture of continuous and categorical causal factors and a categorical/continuous outcome and can guarantee high generalization performance without making any assumption on the underlying distribution of data. Specifically, the objectives are to:

a. Investigate and assess causal factors for land use change.
b. Formulate and implement SVMs for land use change modeling.
c. Explore the optimal SVMs settings for land use change modeling, including regularization parameter selection, kernel selection, vector normalization, etc.
d. Demonstrate the strength of SVMs by comparing its performance with that of a widely accepted land use change modeling approach, namely, spatial logistics regression (SLR) (Cheng and Masser, 2003).
e. Improve SVMs to achieve better performance for unbalanced datasets.
f. Improve the robustness of SVMs.

1.4 RESEARCH SIGNIFICANCE

The aim of this research is to develop a novel method for land use change modeling with the capacity to effectively address land use change data, which might be a mixture of continuous and categorical variables that are not normally distributed. Pattern classifiers that fit the characteristic of land use change data and with a number of attractive features, namely, support vector machines, are applied to model land use change in Calgary, Canada from 1985 to 2001. Some special issues regarding the implementation of SVMs (e.g. regularization parameter selection, kernel selection, vector normalization) are carefully studied. Land use change data is generally unbalanced, which leads to low performance of classifying the minority, namely, the changed land parcels. Improvement to standard SVMs is introduced to achieve uniform performance for both the changed and unchanged data. A secondary improvement is also implemented which increases robustness and allows SVMs to cope with outliers in the land use change dataset thus providing reliable performance when there is a certain level of disturbance on land use change caused by other unobserved factors.

This research will benefit urban planners and policy makers to effectively and efficiently understand the land use change process from the unbalanced and noisy historic data, make more precise projections of future land use, and thus help them generate scientific plans which will foster sustainable development.

1.5 THESIS OUTLINE

This thesis is divided into six chapters. The remainder of this thesis is organized as follows:

Chapter 2 contains a methodological discussion that is relevant to land use change modeling. The significance of land use modeling is presented first. Then, the causal factors driving land use change adopted in the literature are summarized and discussed. Finally, there follows a literature review of prevalent methodologies for land use change modeling, such as spatial statistics, artificial neural network, multi-agent system, and cellular automata. The limitations and advantages of those techniques for land use change modeling are also presented.

Chapter 3 provides a brief introduction to support vector machines. It begins with the theoretical basis of SVMs, namely, statistical learning theory. Then, maximal margin hyperplane is presented to achieve structural risk minimization (SRM) for linearly separable data. Following this, a soft margin hyperplane is introduced to deal with imperfect data. Moreover, kernel functions are used to enable SVMs to handle non-linear classification problems. Lastly, some extensions of the standard SVMs (e.g. one class SVMs, multi-class SVMs, etc) are briefly discussed.

Chapter 4 describes the implementation of SVMs for land use change modeling. The land use change in Calgary, Canada, is used as a case study. Both data preparation and model development are presented. Moreover, some practical issues of SVMs (e.g. regularization parameter selection, kernel function (KF) selection and input vector normalization) are carefully studied to find the optimal SVMs settings for land use change modeling. The performances of SVMs are evaluated and compared with that of a well studied land use change modeling approach; namely, spatial logistic regression. This comparison clearly demonstrates the superiority of SVMs, especially on the capacity and efficiency to classify the changed land parcels.

Chapter 5 deals with the improvements made to standard SVMs to better model land use change. First, the motivations for necessary improvements to standard SVMs are discussed. Then, improvements on handling unbalanced and noisy data, which are two major characteristics of land use change data, are developed. Finally, the performances of the improved SVMs are evaluated to show their efficiency.

Lastly, chapter 6 briefly summarizes the findings of the study with reference to the research objectives set out above, along with conclusions and recommendations for further research.

CHAPTER 2: LAND USE CHANGE MODELING

2.1 INTRODUCTION

The ecosystem, one of the most important systems for the survival of human beings, is continuously changing. Amongst its numerous components, land use/land cover is a key element in the study of global change (Henderson-Sellers and Pitman, 1992). Since World War II, humans have substantially altered the land use/land cover around the world, principally through agriculture, deforestation and urbanization. Land use changes produce significant economic and environmental effects with implications for a wide variety of policy issues, including maintenance of water quality, preservation of open space, and mitigation of global climate change.

The most significant land use change in recent decades is urban growth, which converts vacant or agricultural areas into built-up land. Spurred by economic development and technological revolutions (transport, communication, information, etc), urban populations persistently increase at alarming rates and the migration from rural to urban areas continues to escalate. As a result, cities all over the world continue unabated expansion in order to cater to the needs of an ever-demanding population.

Land use change from vacant or agricultural into built-up has two conflicting factors. On the one hand, cities act as engines of economic and social growth. Urban areas contribute significantly to a nation’s economy and continue to open doors for growth and development. Hence, the escalating urban growth has often been viewed as a sign of the vitality of the regional economy (Yang and Lo, 2003). On the other hand, rapid urban growth may cause environmental and ecological degradation. The expansion of urban areas results in the conversion of farmland or open space into urban land use, which encroaches onto numerous valuable agricultural, forest and natural land. Moreover, urban growth results in increasing surface temperatures in and around cities and this leads to several other local and global climate changes. Therefore, land use change, if left unchecked, would considerably hinder sustainable development for the future.

Scientific management and planning for land use change should be based on a proper understanding of the spatial and temporal processes of land use change. Understanding the complexity of land use change and evaluating its impact on the environment involves procedures of both detection and modeling.

In recent years, progress in modern remote sensing and GIS techniques have been instrumental in opening this field for study, and significant success has already been achieved in monitoring and managing rapid land use change. With the on-going development of remote sensing techniques, image processing, artificial intelligence and machine learning, a wide variety of digital change detection algorithms have been developed over the last two decades (Zhao et al., 2004; Gong and Xu, 2003; Copppin et al., 2003). These algorithms range from frequently used univariate image differencing (Serneels et al., 2001), image ratioing (Hwarth and Wickware, 1981), post-classification comparison (Hall et al., 1991), and composite analysis (Sader, 1988), to less common algorithms such as image regression (Singh, 1989), and multi-temporal spectral mixture analysis (Adams et al., 1995).

Detecting and monitoring land use change, however, is just the first step of sustainable growth management. A further step is to identify factors that drive the land use change and explore their relative importance, analyze the land use change pattern, understand the dynamic process of land use change, and simulate “what-if” decision making based on a variety of scenarios; that is, model land use change. Modeling land use change aims to support land use development planning and sustainable land management. It is a prerequisite to understanding the complexity of land use change and forecasting future trends of land use change and its ecological impacts. Only after accurately modeling land use change can decision makers generate scientific plans which cater to the needs of an ever-demanding population and still maintain ecological balance.

2.2 CAUSAL FACTORS

Modeling means exploring the causal factors and describing the relationship between the causal factors and the outcome. Hence, exploring the causal factors driving land use change is an indispensable part in land use change modeling.

Land use change is a complex process influenced by a variety of natural and human activities. Land-use change modeling aims to explore the dynamics and causal factors of land use change and to inform policies affecting such change. In general, land use change is influenced by a number of factors which may be social, economic, or spatial variables. Earlier studies (Baker, 1989; Agarwal et al., 2002) reveal that no single set of factors can explain the changes in different places, since each context is unique. More often than not, land use studies put forward different causal forces to explain land use trends in different places. Factor selection should take into account the context of the region and period to be modeled as well as the purpose of the model.

Typically, land use changes are influenced by a few recurrent parameters that cannot be overlooked. Demographic factors (population size, population growth, and population density) are widely treated as major causal factors of land use change (Verburg et al., 2001). It is obvious that a city will grow if its population increases. Consequently, new residential areas will emerge in close proximity to transportation facilities (roads, railways and bus lines) and commercial centers also develop concurrently. In the meantime, industrial buildings develop in the vicinity of those previously existing. On the whole, urban expansion will transform vacant or low rent areas into built-up land. Additionally, the agglomeration of developed areas and the availability of exploitable sites will significantly influence land use change patterns.

Accessibility is often seen as a significant driver for land use change through its effect on transportation cost and ease of settlement (Geist and Lambin, 2001). Transport technology is an essential force for land use change from vacant or agricultural land use to urban land use. The widespread use of cars provides land parcels in proximity to advanced transportation systems more possibility to change into built-up area. Moreover, proximity to towns/markets is reported to be an important factor related to land use change because of increased employment opportunities available to the population. The proximity to settlement is also reported to be an indispensable factor because of labor availability (Chomitz and Gray, 1996).

In their case study on China, Cheng and Masser (2003) focused on land use change from rural to urban. Their literature review reported factors such as investment structure, industry structure, housing commercialization, land leasing and decentralization of decision-making. However, the model of Cheng and Masser (2003) included only the industry structure, the transportation networks and the existing developed areas. Cheng and Masser (2003) also took into consideration the constraints imposed by water and other places unfit for urban development.

It was reported that spatial detail plays an important role in land use change process (White et al., 1997). A causal force analysis conducted by Yang (2000) found that highways and shopping malls generally promote urban development. Moreover, some site characteristics, such as soil quality and terrain conditions, were found to be significant factors driving landscape changes (Yang, 2000). Landis and Zhang (2000) investigated land use change near a railway station in their small-scale example. It included only four classes of information: proximity to the transportation network, proximity to the urban structure (residential, commercial, public and industrial buildings), locations available for change and locations where no change could occur because of constraints. It also demonstrated that explanatory factors do not need to be numerous, provided they are relevant.

After examining a summary set of 250 relevant citations, Agarwal et al. (2000) gave a summarization of causal factors commonly used in different land use change models. They mentioned additional factors: 1) economic factors, such as returns to land use (cost and price), job growth, cost of land use change, and rent; 2) social factors, such as affluence, human attitudes and values; 3) collective rule making factors, such as zoning and tenure; 4) and other factors such as technology level.

Table 2.1 gives a summary of the causal factors discussed above.

Table 2.1: Causal factors for land use change

Category	**Causal factor**
Demography	Population size
	Population growth
	Population density
Proximity	Distance to road (major road/street)
	Distance to town/market
	Distance to settlement
	Distance to shopping mall
	Proximity to the urban structure
Economic	Investment structure
	Industry structure
	Housing commercialization
	Returns to land use (costs and prices)
	Job growth
	Cost of land use change
	Rent
Social	Affluence
	Human attitudes and values
Collective rule making	Zoning
	Tenure
Site characteristics	Soil quality
	Slope
Constraints	Water body
	Environment sensitive area
Neighborhoods	Availability of exploitable sites
	Agglomeration of developed areas
Others	Technology level

2.3 LAND USE CHANGE MODELS

A variety of techniques such as Markov chain analysis (Lopez et al., 2001), multiple regression analysis (Theobald and Hobbs, 1998), logistic regression (Wu and Yeh, 1997), artificial neural network (Pajanowski et al., 2002; Li and Yeh, 2002), cellular automata (Wu, 1998; Wu, 2002), and genetic algorithm (Balling et al., 2004), etc, are employed in land use change research.

2.3.1 Spatial Statistics

Traditional statistical methods, e.g. Markov chain analysis, multiple regression analysis, principal component analysis, and logistic regression, have been very successful in interpreting socio-economic activities. They were also employed in land use change modeling and demonstrated great successes in their specific applications.

2.3.1.1 Markov chain analysis

Markov chain models regard land use change as a stochastic process, and different land use categories are the states of a chain. A Markov chain is defined as a stochastic process having the property that the value of the process at time t, X_t, depends only on its value at time $t-1$, X_{t-1}, and not on the sequence of values $X_{t-2}, X_{t-3}, \ldots, X_0$ that the process passed through in arriving at X_{t-1}. That is, the future is only dependent on the present. Past and future are conditionally independent. It can be expressed as:

$$P(X_t = a_j \mid X_0 = a_0, X_1 = a_1, ..., X_{t-1} = a_i) = P(X_t = a_j \mid X_{t-1} = a_i) \qquad (2.1)$$

In Markov chain analysis, land use change is treated as discrete in time ($t = 0,1,2,...$). The probability of a land use change from a land use category (state) a_i to a land use category (state) a_j in one time period, known as ***one step transition probability***, is $P(X_t = a_j \mid X_{t-1} = a_i)$.

Generally, homogeneous Markov chain with discrete time is used in Markov chain analysis, in which the transition probability is stationary. That is, the transition probability from one state to another state $P(X_t = a_j \mid X_{t-1} = a_i)$ is independent of time and dependent only upon states a_i and a_j. In this case, the transition probability can be expressed as:

$$P(X_t = a_j \mid X_{t-1} = a_i) = P_{ij} \tag{2.2}$$

The transition probability can be estimated from historic land use change data by tabulating the number of times the land use changed from state i to j, n_{ij}, and by counting the number of times that land use category a_i occurred, n_i.

$$P_{ij} = \frac{n_{ij}}{n_i} \tag{2.3}$$

Combining all the transition probabilities between all states $a_1, a_2, ..., a_m$ results in the ***transition matrix***:

$$\mathbf{P} = (P_{ij}) = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1m} \\ P_{21} & P_{22} & \cdots & P_{2m} \\ \vdots & \vdots & & \vdots \\ P_{m1} & P_{m2} & \cdots & P_{mm} \end{bmatrix} \tag{2.4}$$

It is obvious that: 1) **P** is a square matrix; 2) each element is non-negative; and 3) the sum of any row is equal to one. Matrices with these properties are called ***stochastic***.

The probability of a land use change from a land use category (state) a_i to a land use category (state) a_j after l time periods, $P_{ij}^{(l)}$, is known as ***l* *steps transition probability***. All l steps transition probabilities compose ***l* *steps transition matrix***. According to the Chapman-Kolomogorov equation:

$$\begin{aligned} P_{ij}^{(n+s)} &= P(X_{n+s} = a_j \mid X_0 = a_i) = \sum_k P(X_{n+s} = a_j, X_n = a_k \mid X_0 = a_i) \\ &= \sum_k P(X_{n+s} = a_j, X_n = a_k \mid X_0 = a_i) \\ &= \sum_k P(X_{n+s} = a_j \mid X_n = a_k, X_0 = a_i) P(X_n = a_k \mid X_0 = a_i) \\ &= \sum_k P(X_{n+s} = a_j \mid X_n = a_k) P(X_n = a_k \mid X_0 = a_i) \\ &= \sum_k P_{ik}{}^{(n)} P_{kj}{}^{(s)} \\ \mathbf{P}^{(n+s)} &= \mathbf{P}^{(n)} \mathbf{P}^{(s)} \end{aligned} \tag{2.5}$$

The l steps transition matrix can be easily calculated by the one step transition matrix:

$$\mathbf{P}^{(n)} = \mathbf{P}^n \tag{2.6}$$

Therefore, the land use state of a period l steps after the current time can be easily predicted by the current land use state and l steps transition matrix.

$$(a_1{}^{(l)}, a_2{}^{(l)}, ..., a_m{}^{(l)}) = \mathbf{P}^{(l)} (a_1{}^{(0)}, a_2{}^{(0)}, ..., a_m{}^{(0)}) \tag{2.7}$$

In Markov chain analysis, the transition probabilities are estimated as proportions of cells that have changed state from one point in time to another. It is a useful way of estimating these probabilities despite the development of procedures for estimating transition probabilities on the basis of more complex scientific consideration. However, the Markov chain model lacks explanatory power as the causal relationships underlying the transition studies are left unexplored (Baker, 1989).

2.3.1.2 Multiple regression analysis

Regression analysis is used to investigate the association of a dependent variable with one or more independent variables. In multiple regression analysis, a linear relationship is used to represent the association of the causal factors with the probability of land use change.

The coefficients of a linear equation, corresponding to multiple causal factors, are estimated to best predict the probability of land use change. Ordinary least squares (OLS) estimation is widely used to estimate the coefficients.

Multiple regression requires that: 1) all causal factors are interval, ratio or dichotomous, and the dependent variable (land use change) is continuous; 2) the errors are normally distributed; 3) errors are independent of the causal variables; and 4) for computational stability, multiple regression does not allow multicollinearity. The generalization performances of multiple regression models are unreliable when these assumptions cannot be satisfied. Unfortunately, land use change data usually violates most of the abovementioned assumptions. Accordingly, multiple regression models cannot ensure high generalization performances for projecting future land use change (Hiroshi and et al. 1998; Frayman and et al., 2002).

2.3.1.3 Logistic regression analysis

Logistic regression is widely used to model the outcomes of a categorical dependent variable while the independent variables can be a mixture of continuous and categorical variables. Hence, it is a suitable approach to estimate the coefficients of causal factors from the observation of land use change because the land use change process does not usually follow normal assumption and its determinants are usually a mixture of continuous and categorical variables.

The general form of logistic regression is given by:

$$u = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_K x_K \tag{2.8}$$

$$u = \log(\frac{P}{1-P}) = logit(P) \tag{2.9}$$

$$P = \frac{e^u}{1+e^u} \tag{2.10}$$

where P refers to the probability of occurrence of a new unit, u is a linear-in-parameters utility function, $x_1, x_2, ..., x_k$ are K causal variables and $\beta_0, \beta_2, \ldots, \beta_K$ are $K+1$ parameters to be estimated.

Logistic regression is a frequently used methodology in land use/land cover research (Serneels and Lambin, 2001; Schneider and Pontius, 2001; Verburg et al., 2004). Besides the binary form, logistic regression also can be extended to cope with multiple land use categories. Multinomial logistic regression can be used for cases involving multiple land use change analysis (Jobson, 1992; Mertens et al., 2002).

Although logistic regression allows the causal factors to be a mixture of continuous and categorical variables, which fit land use change data quite well, it assumes that the land use change probability is linearly and additively related to the causal factors on a logistic scale. This assumption is not always valid. If the assumption cannot be satisfied, the generalization performance of logistic regression may dramatically degrade.

To summarize, traditional statistical approaches (e.g. multiple regression analysis and logistic regression) can readily identify the influence of the independent variables in the modeling process and also provide some degree of confidence regarding their contribution. They have demonstrated different levels of success in their specific applications. However, they are criticized as less effective in modeling spatial and temporal data since the spatial and temporal data often violated basic assumptions such as the normal distribution, appropriate error structure of the variables, independence of variables, and model linearity (Olden and Jackson, 2001).

2.3.2 Cellular Automata

Cellular automata are an effective bottom-up simulation tool for dynamic process modeling (Webster and Wu, 2001). It is a dynamic discrete space and time system. A classic cellular automaton system consists of four primary components: ***cell***, ***state***, ***neighborhoods*** and ***transition rule***. The basic units in a cellular automaton system are a regular grid of cells, each of which can be in one of a finite number of *k* possible states. All cells update synchronously in discrete time steps according to a local identical transition rule in relation to its neighboring cells.

The idea of CA is closely associated with that of microscopic simulation in which the behavior at a local scale gives rise to an emerging global organization (Webster and Wu, 2001). Global structure in a CA system is often seen to emerge out of purely local interactions between cells. This is attractive because it matches our intuitive sense that much human spatial activity is not centrally planned or organized, but arises from the responses of various actors, residents, developers, planners, politicians and local circumstances (O'Sullivan, 2001).

CA has been receiving more and more attention in land use change modeling due to its simplicity, transparency, strong capacities for dynamic spatial simulation, and innovative bottom-up approach (Clarke and Gaydos, 1998). CA has many advantages for modeling urban phenomena, including their decentralized approach, the link they provide to the complexity theory, the connection of form with function and pattern with process, the relative ease with which model results can be visualized, their flexibility, their dynamic approach, and also their affinities with geographical information systems and remotely sensed data (Torrens and O'Sullivan, 2001). Perhaps the most significant of their qualities, however, is their relative simplicity. Research has shown the great potential of CA for discovering the complexity (in particular spatial complexity) of an urban system or its subsystems.

Nevertheless, CA models focus on the simulation of spatial patterns rather than on the interpretation or understanding of the spatio-temporal processes of land use change. Generally, CA models do not explicitly deal with causal factors, such as population, policies and economic impacts on land use change. They are weak in interpreting causal factors in a complete process model. Moreover, CA models are constrained by their simplicity and their ability to represent real-world phenomena is often diluted by their abstract characteristics (Torrens and O'Sullivan, 2001). As a consequence, there are many unexplored research possibilities regarding urban growth complexity based on CA.

Fortunately, many research efforts have been made in order to improve the intricacies of cellular automata model construction, particularly in the modification and expansion of transition rules to include such notions as hierarchy, self-modification, probabilistic expressions, utility maximization, accessibility measures, exogenous links, inertia, and stochasticity (Torrens and O’Sullivan, 2001). These innovative technological advances have enabled cellular modeling to grow out of an earlier game-like simulator and to evolve into a promising tool for urban growth prediction and forecasting, as demonstrated by recent research (Batty and Xie, 1994; Couclelis, 1997; White et al., 1997; Clarke and Gaydos, 1998; Wu and Webster, 1998; Li and Yeh, 2000; Sui and Zeng, 2001). Nevertheless, further research attention needs to be shifted from technical modifications to models in several emergent key applied areas such as explorations in spatial complexity, infusing cellular automata with urban theory, new strategies for validating cellular urban models, and scenario design and simulation in relation to urban planning practices (Torrens and O’Sullivan, 2001).

2.3.3 Multi-agent System

Multi-agent systems are designed as a collection of interacting autonomous agents. All agents have their own internal data representations (***state***) and means for modifying their internal data representations (***perceptions***). Agents relate to a common environment and have abilities to modify their environment (***behavior***). Agents interact with each other via their environment. This interaction can involve communication, i.e. the passing of information from one agent and environment to another.

The basic unit of activity in an agent-based model is the agent. Usually, agents explicitly represent actors in the situation being modeled, often at the individual level. For example, an agent may represent a land manager who combines individual knowledge and values, information on soil quality and topography, and an assessment of the land management choices of neighbors to calculate a land use decision. Agents also may represent higher-level entities or social organizations such as a village assembly, local governments, or a neighboring country. Agents are capable of effective independent action, and their activity is directed towards the achievement of defined tasks or goals. They share an environment through agent communication and interaction, and they make decisions that tie behavior to the environment.

Multi-agent systems contain rules that define the relationship between agents and their environment and rules that determine sequencing of actions in the model. Agents can translate both internal and external information into internal states, decisions, or actions based on these rules.

Multi-agent systems offer a high degree of flexibility that allows researchers to account for heterogeneity and interdependencies among agents and their environment. Besides, multi-agent systems have some attractive features (White and Engelen, 2000): (1) as a tool to implement self-organizing theory such as a straightforward way of representing spatial entities or actors having relatively complex properties or behaviors; (2) an easy way to capture directly the interactive properties of many natural and human systems, as well as the complex system behavior that emerges from this interaction. The approach is useful for examining the relationship between micro-level behavior and macro outcomes.

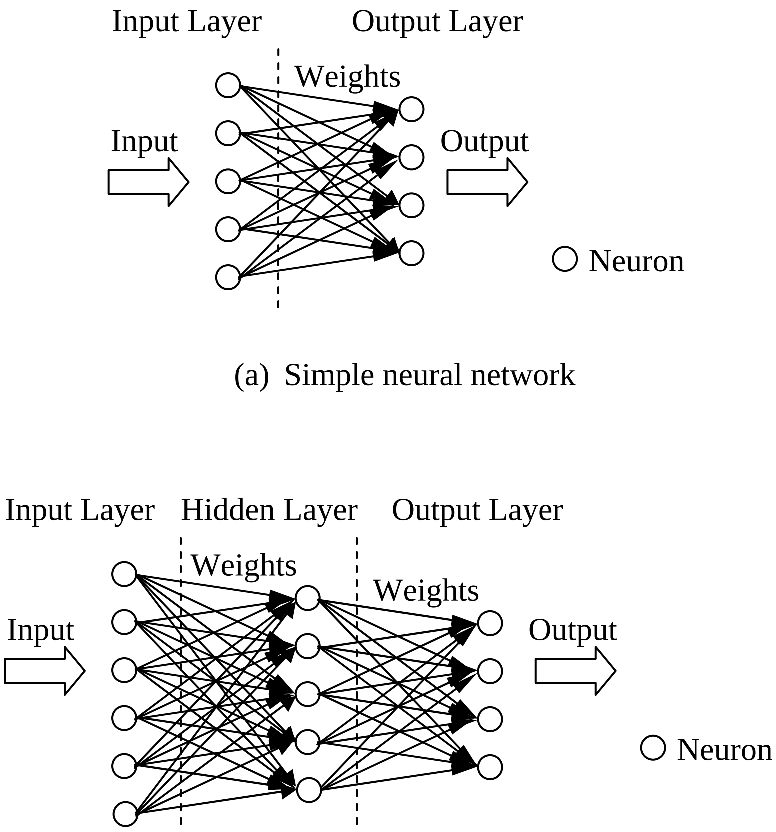
Multi-agent systems, however, are initially designed for microscopic simulation and have difficulty meeting the requirement of large scale modeling. Furthermore it is difficult to define the perception rule that determines the agent interactions. Multi-agent systems cannot guarantee a satisfactory performance if good perception rules cannot be defined.

2.3.4 Artificial Neural Network

An artificial neural network is a system composed of many simple processing and parallel operating elements, whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes. It is a powerful tool that uses a machine learning approach to quantify and model complex behavior and patterns.

Artificial neural networks were developed to model the brain’s interconnected system of neurons so that computers could be made to imitate the brain’s ability to sort patterns and learn from trial and error, thus observing relationships in data. The development of a neural network model requires the specification of a "network topology", a learning paradigm and algorithm.

Artificial neural networks consist of layers and neurons which simulate the structure of human brains and allow ANN to have learning and recall abilities like humans, especially for non-linear mapping. A neural network consists of one input layer, one output layer, and no or some hidden layers between them. The former is named a simple neural network and the latter is called a multi-layer neural network (Figure 2.1).



(a) Simple neural network

(b) Multi-layer neural network

Figure 2.1: Basic structure of artificial neural network

A neural network can be used to classify a set of inputs, $X = [x_1, x_2, ..., x_n]^T$, which consist of n causal variables. The input can be propagated through the hidden layers and the output layer in a feed-forward manner. Neurons, the basic units to process signals, are arranged in layers. The signals propagate from neurons in a sender layer (input layer or hidden layer) to neurons in a receiver layer (hidden layer or output layer) and are modified by weights associated with each neuron-neuron connection.

Each neuron in an input layer accepts a single value which corresponds to an element in X. Then each neuron generates an output value and the output value may be used as the input for all the neurons in the next layer. Weights are used to address the strengths of network interconnection between associated neurons. Each neuron in a receiver layer can receive signals from a sender layer. The receiving neuron sums the weighted inputs from all of the nodes connected to it from the previous layer as:

$$I_j = \sum_i w_{ij} S_i \tag{2.11}$$

where S_i is the signal emitted from neuron i in the sender layer, I_j is the input for neuron j in the receiver layer, and w_{ij} is the weight associated with the connection from neuron i to neuron j.

After collecting signals from the sender layer, a neuron in the receiver layer creates activation in response to the input I_j and emits an output signal. The output is computed as the function of its input, called the ***activation function***. The most widely used activation function is the sigmoid function:

$$S_j = \frac{1}{1+e^{-I_j}} \tag{2.12}$$

where I_j is the input for neuron j in the receiver layer, and S_j is the signal emitted from neuron j to next layer.

The signals move forward from neuron to neuron. Equations (2.11) and (2.12) can be used to process the signal collection and activation. The collection and activation processes continue until the final signals are obtained by the output layer.

Artificial neural networks can be used for pattern recognition or classification. Each neuron in the output layer is associated with a class. When a signal (e.g. a set of causal variables associated with a land parcel) is presented to the network, each output neuron will generate a value that indicates the similarity between the input signal and the corresponding class (e.g. certain kind of land use changes). An input signal can be classified into the class that is associated with the neuron of the highest activation level.

The determination of weights is critical to successes of applications involving artificial neural networks. Weights are determined by using a training algorithm, the most popular of which is the back propagation algorithm. This algorithm randomly selects the initial weights, and then compares the network output for a given training dataset with the expected output. The difference between the expected and network outputs across the whole training dataset is summarized using the mean squared error. Then, the weights are modified according to a generalized delta rule (Rumelhart et al., 1986), so that total error is distributed among the various neurons in the network. This process feeding forward signals and back-propagating the errors is repeated iteratively until the error stabilizes at a low level.

Unlike the more commonly used analytical methods, the ANN is independent of particular functional relationships, makes no assumptions regarding the distributional properties of the data, and requires no prior understanding of variable relationships. This independence makes the ANN a potentially powerful modeling tool for exploring nonlinear complex problems (Olden and Jackson, 2001). ANN has greater predictive and non-linear power than traditional approaches. It is an ideal method of understanding non-linear spatial patterns, on which short-term prediction may be based. Its strength lies in prediction and performing "what-if" types of experiment (Corne et al., 1999).

ANN, however, has a static nature, in which causal factors are not dynamic. Moreover, ANN’s property of a "black box" provides little explanatory insight into the relative influence of the independent variables in the prediction process. This lack of explanatory power is a major concern in spatial pattern analysis because the interpretation of statistical models is desirable for gaining knowledge of the causal factors driving spatial phenomena. Furthermore, ANN might suffer difficulties with generalization and produce models that may overfit the data (Sui, 1994; Karystinos and Pados, 2000). These major drawbacks make ANN of limited value for modeling the urban growth process.

2.4 CHAPTER SUMMARY

This chapter presented the background and significance of land use change modeling. Land use change modeling is a prerequisite to learning the complexity of land use change process and evaluating its impact on the environment. The importance of land use change modeling for sustainable development is widely accepted.

Causal factors driving land use changes reported in the literature were also summarized, which ranged from social, economic, technological factors to site specified spatial variables. It has been found that no single set of factors can explain all the changes in different places. Factor selection should consider the specific context in the area to be modeled.

A variety of techniques for land use change modeling, which included Markov chain analysis, multiple regression analysis, logistic regression, cellular automata, multi-agent system, and artificial neural network, were briefly reviewed. The advantages and limitations of these techniques were also presented. Due to the complexity of the land development process and differences in modeling objectives, there was no clearly superior approach. Each method had its strengths, weaknesses, and application domains. Therefore, the selection of methods for land use change modeling should depend on the demands of the analysis, the feasibility of the techniques and the availability or limitation of the data framework.

CHAPTER 3: SUPPORT VECTOR MACHINES

3.1 INTRODUCTION

To overcome the shortcomings of current land use change modeling methodologies and better address the research problems, a new method for land use modeling using support vector machines was developed for the purpose of this study.

Support vector machines, which were originally developed by Vapnik (1995), are a new generation of machine learning algorithms that take their inspiration from statistical learning theory (Gunn, 1998). Unlike traditional methods which minimize the empirical training error, SVMs aim at minimizing an upper bound of the generalization error through maximizing the margin between the separating hyperplane and the data. This can be regarded as an approximate implementation of the Structural Risk Minimization (SRM) principle, which endows SVMs with good generalization performances independent of underlying distributions (Joachims, 1999).

SVMs are a system for efficiently training the linear learning machines in the kernel-induced feature space (Cristianini and Shawe-Taylor, 2000). SVMs were originally designed as a linear classifier, but they are easily extended to nonlinear classifiers by mapping the space $S = \{\mathbf{x}\}$ of the input data into a high-dimensional feature space $F = \{\Phi(\mathbf{x})\}$. By choosing an adequate mapping $\Phi(\mathbf{x})$, the data points that cannot be linearly separated in the input space become linearly separable or mostly linearly separable in the high-dimensional feature space, so that one can easily apply the structure risk minimization.

Compared with the traditional way of implementing mapping functions, SVMs have incomparable advantages. We need not compute the mapped patterns $\Phi(\mathbf{x})$ explicitly, and instead we only need the dot products between mapped patterns. They are directly available from the kernel function which generates $\Phi(\mathbf{x})$ (Amari and Wu, 1999).

SVMs are an elegant and highly principled learning method for classifying nonlinear input data. SVMs are gaining increasing popularity due to a number of attractive features including (Cristianini and Shawe-Taylor, 2000):

1. SVMs are statistics-based models rather than loose analogies with natural learning systems. SVMs are theoretically related to a wide variety of study fields related to regularization theory and sparse approximation.
2. SVMs do not incorporate problem-domain knowledge. No assumption for input data distribution or error structure is needed in SVMs.
3. SVMs have a promising generalization performance. The formulation embodies the structural risk minimization (SRM) principle, as opposed to the empirical risk minimization (ERM) approach commonly employed within statistical learning methods. SVMs provide a method for controlling model complexity independently of dimensionality. It is this difference which equips SVMs with an excellent potential to generalize.
4. SVMs have the ability to model non-linear relationships in an effective and efficient way. SVMs operate in a kernel induced feature space. By using a suitable inner-product kernel, SVMs allow for constructing non-linear classifiers using only linear algorithms.
5. SVMs have the property of condensing information in the training data and providing a sparse representation by using a very small number of data points, namely, support vectors (SVs). Therefore, computations can be performed efficiently. This is especially true for huge datasets.
6. SVMs can guarantee a global and in general unique optimum. SVMs use quadratic programming to achieve maximized margin separation, which provides global minima only. The absence of local minima is a significant difference from the neural network classifiers.
7. SVMs have an extra advantage regarding automatic model selection in the sense that both the optimal number and locations of the support vectors are automatically obtained during training (Schölkopf et al., 1999).

8. SVMs are a robust tool for classification and regression in noisy, complex domains.

Owing to these prominent features, SVMs have gained growing popularity in recent years and have been successfully applied to a variety of fields such as text categorization, image recognition, hand-written digit recognition, potential disease spread prediction (Guo et al., 2005) and land cover classification (Huang et al., 2002).

3.2 STATISTICAL LEARNING THEORY

To describe the idea of SVMs, the issue of structural risk minimization principle has to be addressed first. Therefore, we will start with posing a generic statistical learning problem.

3.2.1 Statistical Learning Problem

Consider a binary classification problem: suppose we are given an empirical observations (thereafter called training set),

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_m, y_m) \in \mathbf{X} \times \mathbf{Y}, \mathbf{X} = \mathbf{R}^n, \mathbf{Y} = \{-1, +1\} \qquad (3.1)$$

where $\mathbf{X}$ is the input space of potential observations, and $\mathbf{Y}$ is the possible decision space.

Assume that the training set is drawn independently from an unknown (but fixed) probability distribution $P(\mathbf{X}, \mathbf{Y})$. This is a standard assumption in learning theory. Data generated this way is commonly referred to as IID (independent and identically distributed). The goal of classification problem is to find a classifier $y = f(\mathbf{x})$, which is a map from $\mathbf{X}$ to $\mathbf{Y}$ based on data in $\mathbf{T}$. Any future case (outside training set $\mathbf{T}$) that is also generated from $P(\mathbf{X}, \mathbf{Y})$ will be classified correctly by the map found. Of course, no classifier can classify every unseen example perfectly. Correctness of the classification is then measured by a ***loss function*** $L(\mathbf{x}, y, f(\mathbf{x}))$.

3.2.2 Loss Function

Definition 3.1 (Loss Function) Denoted by $(\mathbf{x}, y, f(\mathbf{x})) : \mathbf{x} \in \mathbf{X}, y \in \mathbf{Y}, f(\mathbf{x}) \in \mathbf{Y}$ the triplet consisting of a pattern $\mathbf{x}$, an observation y and a prediction $f(\mathbf{x})$. Then the map $L : \mathbf{X} \times \mathbf{Y} \times \mathbf{Y} \to [0, \infty)$ with the property $L(\mathbf{x}, y, y) = 0$ for all $\mathbf{x} \in \mathbf{X}$ and $y \in \mathbf{Y}$ will be called a loss function.

The well-known loss functions are squared loss used in least square algorithm and logistic loss used in logistic regression.

$$\text{Squared loss: } \; L(\mathbf{x}, y, f(\mathbf{x})) = (f(\mathbf{x}) - y)^2 \tag{3.2}$$

$$\text{Logistic loss: } \; L(\mathbf{x}, y, f(\mathbf{x})) = \ln(1 + e^{-y \cdot f(\mathbf{x})}) \tag{3.3}$$

The former uses the square of the amount of mis-prediction to determine the quality of the estimate. It satisfies the assumption that we have additive normal noise corrupting the observations. The latter uses the product $yf(\mathbf{x})$ to assess the quality of the estimate, where the sign of the prediction $\operatorname{sgn}(f(\mathbf{x}))$ denotes the class label, and the absolute value $|f(\mathbf{x})|$ describes the confidence of the prediction. No penalty occurs if $yf(\mathbf{x})$ is sufficiently large, i.e. if the patterns are classified correctly with large confidence. The logistic loss is used in order to associate a probabilistic meaning with prediction $f(\mathbf{x})$.

In a binary classification problem, another kind of loss function, namely, ***zero-one loss function*** is generally used:

$$L(\mathbf{x}, y, f(\mathbf{x})) = \frac{1}{2}\left| f(\mathbf{x}) - y \right| \tag{3.4}$$

Note that the loss is 0 if the sample $(\mathbf{x}, y)$ is classified correctly and 1 otherwise.

3.2.3 Risk Function

To sum up the total expected loss for any mapping $f:\mathbf{X}\times\mathbf{A}\rightarrow\mathbf{Y}$, where $\mathbf{A}$ is the parameter space for the mapping function, a ***risk function*** comprising the loss and the underlying probability distribution is used:

$$R(\alpha)=\int L(\mathbf{x},y,f(\mathbf{x},\alpha))dP(\mathbf{x},y)=\int\frac{1}{2}\left|f(\mathbf{x},\alpha)-y\right|dP(\mathbf{x},y) \tag{3.5}$$

where $f(\mathbf{x},\alpha)$ is a classifier from a fixed parametric family $\{f(\mathbf{x},\alpha):\alpha\in\mathbf{A}\}$.

Any choice of a particular α produces a classifier. The goal of statistical learning is to find a classifier with the minimal expected risk (or simply called risk). The difficulty of the task stems from the fact that we are trying to minimize a quantity that we cannot actually evaluate: since the underlying probability distribution $P(\mathbf{X},\mathbf{Y})$ is usually unknown, it is impossible to compute the integral (3.5) and thus to achieve the risk minimization directly.

We do not know the probability distribution $P(\mathbf{X},\mathbf{Y})$ that potential observations will be generated from. We do know, however, the training set $\mathbf{T}$ is generated from $P(\mathbf{X},\mathbf{Y})$. Thus, we can try to infer a classifier $f(\mathbf{x},\alpha)$ from the training set that is, in some sense, close to the one minimizing the risk (3.5).

3.2.4 Empirical Risk Minimization

One way to proceed is to use the empirical distribution of the training set to approximate the underlying probability distribution $P(\mathbf{X},\mathbf{Y})$ and thus to calculate an approximation for the integral in (3.5). This leads to the ***empirical risk***:

$$\mathrm{R}_{emp}(\alpha)=\frac{1}{m}\sum_{i=1}^{m}L(\mathbf{x}_i,y_i,f(\mathbf{x}_i,\alpha))=\frac{1}{2m}\sum_{i=1}^{m}\left|y_i-f(\mathbf{x}_i,\alpha)\right| \tag{3.6}$$

Most traditional methods, e.g. least square estimate, maximum likelihood estimate, and artificial neural network, aim to achieve empirical risk minimization. This makes some sense since according to the theory of uniform convergence in probability:

$$\lim_{m\to\infty} P\left\{\sup_{\alpha\in\mathbf{A}}(R(\alpha)-R_{emp}(\alpha))>\varepsilon\right\}=0,\forall\varepsilon>0 \tag{3.7}$$

The empirical risk will infinitely approximate the expected risk when the size of training set increases. However, the size of the training set is limited. We are not sure how well the empirical distribution of the training set can approximate the unknown probability distribution $P(\mathbf{X},\mathbf{Y})$. Therefore, minimizing the empirical risk does not always imply a small expected risk. For example, consider the 1D classification problem shown in Figure 3.1, with a training set of three points (marked by circles), and three test inputs (marked on the x-axis). Classification is performed by thresholding real-valued functions $g(x)$ according to $\mathrm{sgn}(g(x))$. Note that both classifiers represented using dotted line and solid line can perfectly explain the training data, but they give opposite predictions on the test inputs. That is, both classifiers can achieve empirical risk minimization but lead to quite different expected risks. Lacking any further information, the training data alone provides no means to tell which of the two functions is to be preferred.

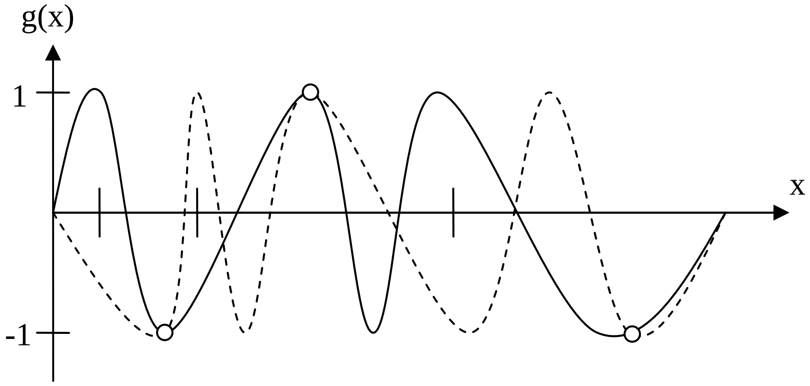


Figure 3.1: A 1D classification problem

3.2.5 Structural Risk Minimization

Although we cannot calculate the expected risk if the underlying probability is unknown, it is possible to find an upper bound for the expected risk and pose a problem for its minimization. Instead of empirical risk minimization, statistical learning theory (or Vapnik-Chervonenkis theory) aims to find a learning machine (classifier) with the minimum upper bound on the expected risk. This leads us to a method of choosing an optimal classifier for a given task. This is the essential idea of the structural risk minimization.

Prior to discussing SRM, we need to introduce the notion of function set capacity and define some means of measuring that capacity. The term ***capacity*** can be introduced as the ability of a machine (a parametric family or function set) to learn any training set without an error. Suppose we have a training set of m samples that can be assigned labels +1 or -1. Clearly, there are 2^m ways to label the training set. If, for each labeling, there is a classifier in the function set $\{f(\mathbf{x},\alpha):\alpha\in\mathbf{A}\}$ that can correctly assign those labels, we say that the training set can be shattered by the function set. Maximum cardinality of the training set that can be shattered by $\{f(\mathbf{x},\alpha):\alpha\in\mathbf{A}\}$ is called the Vapnik-Chervonenkis (VC) dimension of that function set. VC-dimension is clearly a property of the parametric family and can then be used as a measure of capacity of a particular learning machine belonging to that family (Vapnik, 1995; Cristianini and Shawe-Taylor, 2000).

The VC dimension sounds a little abstract. A simple example might be helpful in explaining it clearly. Considering a parametric family of hyperplanes in $\mathbf{R}^2$, as shown in Figure 3.2, it is capable of correctly classifying 3 samples with labels +1 or -1. There are $2^3 = 8$ ways of assigning 3 samples to two classes. For the displayed samples in $\mathbf{R}^2$, all 8 possibilities can be realized using separating hyperplanes, in other words, the function class can shatter 3 samples. This would not work if we were given 4 points, no matter how we placed the hyperplanes. Therefore, the VC dimension of the class of separating hyperplanes in $\mathbf{R}^2$ is 3. It is easy to extend this conclusion to $\mathbf{R}^n$: the VC dimension of hyperplanes in $\mathbf{R}^n$ is $VCdim(H^n) = n+1$.

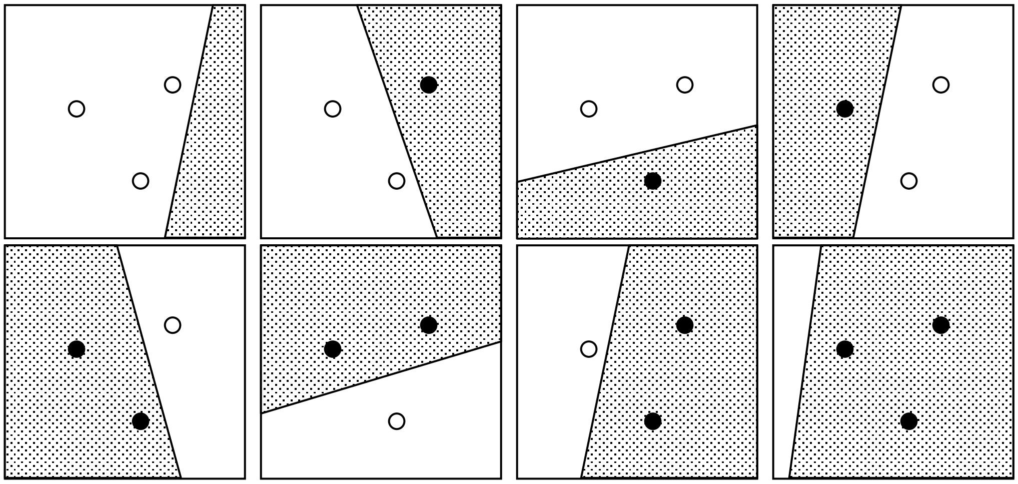


Figure 3.2: A simple VC dimension example

Vapnik and Chervonenkis applied the probably approximately collect (PAC) model to statistical inference and gave the following theorem (Vapnik, 1995) to determine the upper bound for the expected risk:

Theorem 3.1: Supposing $\{f(\mathbf{x},\alpha): \alpha \in \mathbf{A}\}$ is a parametric family for binary classification with adjustable parameters α. Then the following bound holds with a probability of at least $1-\delta$ for any underlying distribution provided $h < m$:

$$R(\alpha) \le R_{emp}(\alpha) + \sqrt{\frac{h(\ln(2m/h)+1) - \ln(\delta/4)}{m}} \tag{3.8}$$

where $R(\alpha)$ is the expected risk, $R_{emp}(\alpha)$ is the empirical risk, m is the size of the training set, and h is the VC dimension of the parametric family $\{f(\mathbf{x},\alpha): \alpha \in \mathbf{A}\}$.

The second term of (3.8) is called ***VC confidence interval***. Given a training set of finite size, we can always come up with a learning machine which achieves zero training error (provided no examples contradict each other, i.e., whenever two patterns are identical, then they must come with the same label). To correctly separate all training examples, however, this machine will necessarily require a large VC dimension h. Therefore, the VC confidence interval, which increases monotonically with h, will be large. The bound (3.8) shows that the small training error does not guarantee a small test error. To achieve good generalization performance, that is, small expected risk, both the empirical risk and VC dimension of the parametric family have to be small.

Figure 3.3 shows the relationships between VC dimension versus the empirical risk, VC confidence interval, and upper bound on the risk. Suppose we have a sequence of nested parametric families $S_1 \subset S_2 \subset ... \subset S_n \subset ...$ such that their VC dimensions satisfy $h_1 < h_2 < ... < h_n < ...$. With the increase of the VC dimension, it is possible to find a classifier in the parametric family to better fit the training set with finite size. Therefore, the empirical risk is usually a decreasing function of VC dimension h. As shown on (3.8), the VC confidence interval will monotonously increase with the increase of VC dimension h. As a result, for a given size of training set, there is an optimal value of VC dimension which can achieve a minimal upper bound on the expected risk.

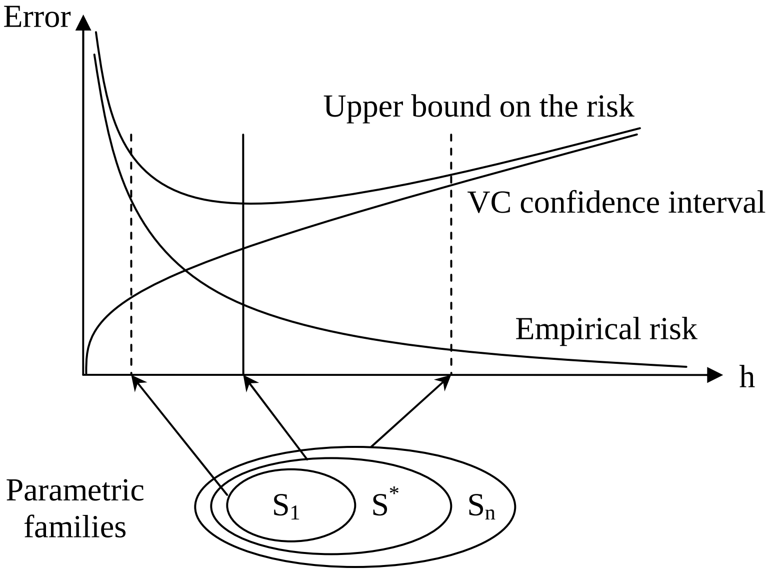


Figure 3.3: VC dimension vs. empirical risk, VC confidence, and the risk (reprinted from Vapnik, 1995)

The choice of an appropriate VC dimension, which in some techniques is controlled by the number of free parameters of the model, is crucial in order to get good generalization performance, especially when the size of the training set is small. The objective of SRM then is to find an optimal VC dimension for which the upper bound on the expected risk is minimal. That is to minimize the empirical risk and VC confidence interval simultaneously, which can be achieved through the following two-stage process:

1. For each VC dimension h_i, identify a classifier $f(\mathbf{x}, \alpha^*)$ with minimal $R_{emp}(\alpha)$
2. In all the classifiers identified, choose the classifier, for which $R_{emp}(\alpha)$ + VC confidence interval is minimal

However, finding a trade off between reducing training error and limiting model complexity is not easy because the VC dimension of a parametric family can be hard to compute and there are only a small number of parametric families for which we know how to compute the VC dimension. Moreover, even if the VC dimension of a parametric family is known, it is difficult to solve the optimization problem of minimizing the empirical risk. Hence, we usually do not follow the above steps directly but rather use some more effective and efficient strategies. Support vector machines are able to achieve the goal of minimizing the upper bound of $R(\alpha)$ by efficiently minimizing a bound on the VC dimension h and $R_{emp}(\alpha)$ at the same time.

3.3 LINEAR SVMS

Support vector machines are an approximate implementation of structural risk minimization. Based on the discussion of section 3.2, each particular choice of parametric families gives rise to a learning algorithm, consisting of performing SRM on the classifiers in parametric families of different VC dimensions. SVMs algorithms are based on parametric families of separating hyperplanes of different VC dimensions. SVMs can effectively and efficiently find the optimal VC dimension and an optimal hyperplane of that dimension simultaneously to minimize the upper bound of the expected risk.

Consider the problem of separating the training set of two separable classes of m examples:

$$\mathbf{T} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_m, y_m)\}, \mathbf{x} \in \mathbf{R}^n, y \in \{-1,1\} \tag{3.9}$$

with a hyperplane parameterized by $\mathbf{w}$ and b , $(\mathbf{w}, b) \in \mathbf{R}^n \times \mathbf{R}$,

$$\mathbf{w}' \cdot \mathbf{x} + b = 0 \tag{3.10}$$

where $\mathbf{x}_i$ is a data point in n-dimensional space, y_i is a class label, $\mathbf{w}$ is n-dimensional coefficient vector ($\mathbf{w}'$ is the transpose of $\mathbf{w}$), and b is the offset. The discriminant function of the optimal hyperplane (classifier) is:

$$f(\mathbf{x}) = \text{sgn}(\mathbf{w}' \cdot \mathbf{x} + b) \tag{3.11}$$

As shown in Figure 3.4, there exist many hyperplanes that can separate the examples perfectly. That is, many classifiers can achieve minimized empirical risk. Apparently, the generalization performances of these hyperplanes are quite different. Some hyperplanes, e.g. H_1 and H_4, have very poor generalization performance. Based on only the training set, how can we select a hyperplane which works well in general? According to structural risk minimization principle, we should select a hyperplane with a minimal VC confidence interval: that is, select a hyperplane with minimal VC dimension.

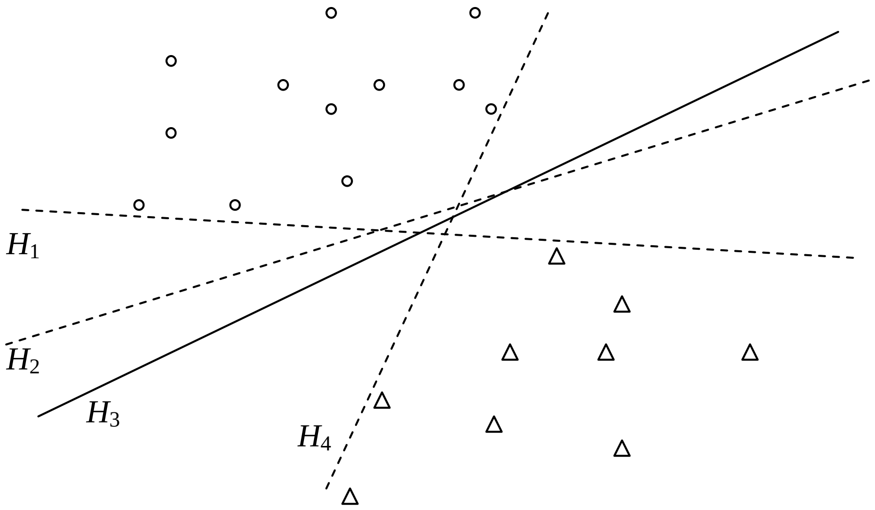


Figure 3.4: Hyperplanes perfectly separating two separable classes

Vapnik (1995) formulated another theorem to determine a separating hyperplane with the minimal VC dimension.

Theorem 3.2: Let R be the radius of the smallest ball $B_R(\mathbf{a}) = \{\mathbf{x} \in \mathbf{T} : \|\mathbf{x} - \mathbf{a}\| \le R \mid \mathbf{a} \in \mathbf{T}\}$ containing the training set $\mathbf{T} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_m, y_m)\}, \mathbf{x} \in \mathbf{R}^n, y \in \{-1,1\}$, and let

$$f_{\mathbf{w},b}(\mathbf{x}) = \operatorname{sgn}(\mathbf{w}' \cdot \mathbf{x} + b) \tag{3.12}$$

be a canonical hyperplane decision function defined on the training set $\mathbf{T}$. Then the set of separating hyperplanes $\{f_{\mathbf{w},b} : \|\mathbf{w}\| \le A\}$ has the VC dimension h bounded by

$$h \le \min(R^2 A^2, n) + 1 \tag{3.13}$$

According to theorem 3.2, in order to find a hyperplane with minimal VC dimension, we need to minimize the norm of the canonical hyperplane $\|\mathbf{w}\|$. A canonical separating hyperplane satisfied:

$$f_{\mathbf{w},b} : \min_{i=1,...,m} |\mathbf{w}' \cdot \mathbf{x}_i + b| = 1, \mathbf{x}_i \in \mathbf{T} \tag{3.14}$$

Specifically, we can find two hyperplanes parallel to the separating hyperplane and equal distances to it,

$$H_1 : y = \mathbf{w}' \cdot \mathbf{x} + b = +1 \tag{3.15}$$

$$H_2 : y = \mathbf{w}' \cdot \mathbf{x} + b = -1 \tag{3.16}$$

with the condition that there are no data points between H_1 and H_2. For any two parallel hyperplanes separating the data points, we can always scale the coefficient vector $\mathbf{w}$ and offset b so that they can be expressed as (3.15) and (3.16). As shown in Figure 3.5, the data points need to satisfy,

$$\mathbf{w}' \cdot \mathbf{x}_i + b \geq +1, \text{ for positive examples } y_i = +1, \quad i = 1, 2, \ldots, k \tag{3.17}$$

$$\mathbf{w}' \cdot \mathbf{x}_i + b \leq -1, \text{ for negative examples } y_i = -1, \quad i = 1, 2, \ldots, k \tag{3.18}$$

Conditions (3.17) and (3.18) can be combined into a single condition,

$$y_i(\mathbf{w}' \cdot \mathbf{x}_i + b) \geq 1 \tag{3.19}$$

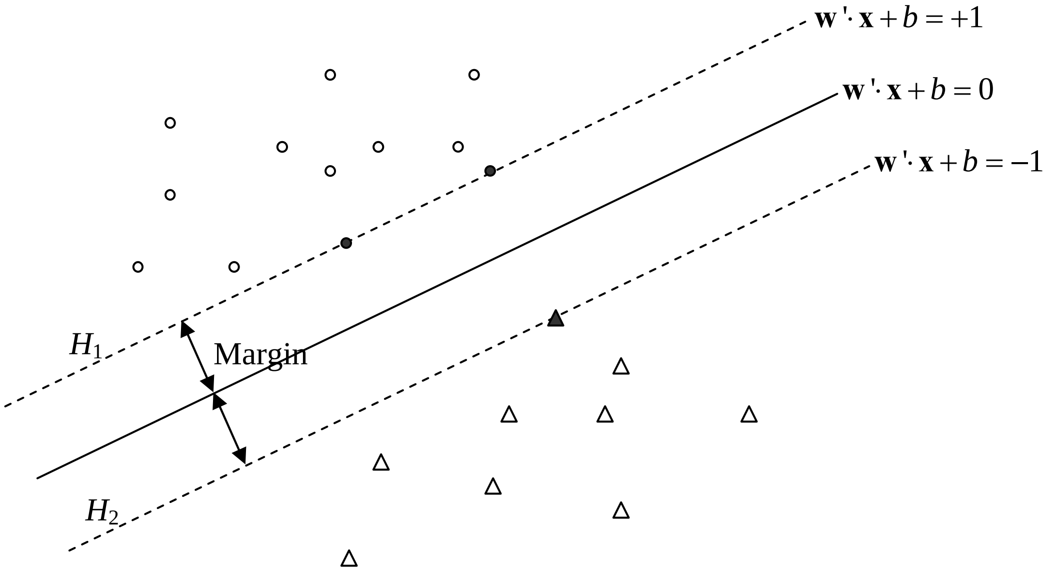

$\mathbf{w}' \cdot \mathbf{x} + b = +1$
$\mathbf{w}' \cdot \mathbf{x} + b = 0$
$\mathbf{w}' \cdot \mathbf{x} + b = -1$
H_1
Margin
H_2

Figure 3.5: Optimal separating hyperplane between two classes of separable samples

The distance of a point $\mathbf{x}_i$ from the hyperplane $\mathbf{w}'\cdot\mathbf{x}+b=0$ is,

$$d(\mathbf{w},b;\mathbf{x}_i)=\frac{|\mathbf{w}'\cdot\mathbf{x}_i+b|}{\|\mathbf{w}\|} \tag{3.20}$$

Therefore, the distance between H_1 and H_2 is given by,

$$\begin{aligned} d(H_1,H_2) &= \min_{\mathbf{x}_i:y_i=-1} d(\mathbf{w},b;\mathbf{x}_i)+\min_{\mathbf{x}_i:y_i=+1} d(\mathbf{w},b;\mathbf{x}_i) \\ &= \frac{1}{\|\mathbf{w}\|}(\min_{\mathbf{x}_i:y_i=-1}|\mathbf{w}'\cdot\mathbf{x}_i+b|+\min_{\mathbf{x}_i:y_i=+1}|\mathbf{w}'\cdot\mathbf{x}_i+b|) \\ &= \frac{2}{\|\mathbf{w}\|} \end{aligned} \tag{3.21}$$

Consequently, minimizing the norm of the canonical hyperplane $\|\mathbf{w}\|$ is equal to maximizing the margin between H_1 and H_2. That is: the purpose of implementing SRM for constructing an optimal hyperplane is to find an optimal separating hyperplane that can separate the two classes of training data with maximum margin. Hence, there will be some positive examples on H_1 and some negative examples on H_2. Only these examples determine the optimal separating hyperplanes. Other examples have no contribution to the definition of the optimal separating hyperplanes and thus can be removed from the training set. Therefore, the examples located on H_1 and H_2 are called ***support vectors***. The name ***Support Vector Machines*** originated from the name of support vector: that is, learning machines for finding support vectors.

Hence, the optimal hyperplane separating the training data of two separable classes is the hyperplane that satisfies,

$$\begin{aligned} &Minimize: F(\mathbf{w})=\frac{1}{2}\mathbf{w}'\cdot\mathbf{w} \\ &subject\ to: y_i(\mathbf{w}'\cdot\mathbf{x}_i+b)\geq 1, \qquad i=1,2,...,m \end{aligned} \tag{3.22}$$

This is a convex, quadratic programming (QP) problem with linear inequality constraints. Problems of this kind are called constrained optimization problems. It is hard to solve the inequality constraint optimization problem directly. The most common way to deal with optimization problems with inequality constraints is to introduce Lagrange multipliers to convert the problem from the primal space to dual space and then solve the dual problem (please refer to Appendix A for more information on Dual Theorem).

Introducing m nonnegative Lagrange multipliers $\alpha_1, \alpha_2, \ldots, \alpha_m \geq 0$ associated with the inequality constraints in (3.22), we have the following Lagrangean function,

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}'\cdot\mathbf{w} - \sum_{i=1}^{m}\alpha_i[y_i(\mathbf{w}'\cdot\mathbf{x}_i + b) - 1] \tag{3.23}$$

Solving the saddle point of Lagrangean function (which is unconstrained) is equivalent to solving the original constrained problem. At the saddle point of Lagrangean function, the gradient of $L(\mathbf{w}, b, \boldsymbol{\alpha})$ with respect to the primal variables $\mathbf{w}$ and b vanish,

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i = 0 \tag{3.24}$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{m}\alpha_i y_i = 0 \tag{3.25}$$

Since these are equality constraints in the dual formulation, we can substitute them into $L(\mathbf{w}, b, \boldsymbol{\alpha})$ to yield,

$$L(\boldsymbol{\alpha}) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \mathbf{x}_i'\cdot\mathbf{x}_j \tag{3.26}$$

Therefore, solving the constrained optimization problem (3.22) is converted to solving the dual optimization problem:

$$\begin{aligned} &maximize: L(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i{}' \cdot \mathbf{x}_j \\ &subject\ to: \sum_{i=1}^{m} \alpha_i y_i = 0 \\ &\qquad\qquad \alpha_i \geq 0, \qquad i = 1, 2, ..., m \end{aligned} \tag{3.27}$$

This is a quadratic optimization problem. Over the years, a number of optimization techniques have been devised to solve the quadratic optimization problem. They range from the simple gradient ascent (the steepest ascent) algorithm to more efficient algorithms, namely, the Newton method, conjugate gradient method, and primal dual interior-point method. These methods can be applied in SVMs to solve the above mentioned optimization problem. However, many of these methods require that the matrix $y_i y_j \mathbf{x}_i{}' \cdot \mathbf{x}_j$ is stored in memory. This implies that the space complexity of the algorithm is quadratic in the sample size. For large size problems, these approaches can be inefficient and sometimes impossible.

A novel algorithm called Sequential Minimal Optimization (SMO) (Platt, 1998) was designed to solve large size quadratic optimization problems. The strategy of SMO is to decompose the problem into a series of small tasks that optimize a minimal subset of just two variables at each step. An analytical solution for the two variables optimization problem is given and the original problem can be solved using iteration. For more details about SMO, please refer to Appendix B.

After obtaining an optimal solution $\boldsymbol{\alpha}^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_m^*)$ for the dual problem, the solution of an optimal coefficient vector for the primal problem can be obtained from (3.24):

$$\mathbf{w}^* = \sum_{i=1}^{m} y_i \alpha_i^* \mathbf{x}_i \tag{3.28}$$

The offset b does not appear in the QP problem and the optimal solution b^* must be solved from the primal constraints,

$$b^* = -\frac{1}{2}(\min_{\mathbf{x}_i : y_i = +1} \mathbf{w}'\cdot \mathbf{x}_i - \max_{\mathbf{x}_i : y_i = -1} \mathbf{w}'\cdot \mathbf{x}_i) \qquad (3.29)$$

Generally, we do not need to calculate $\mathbf{w}^*$ explicitly. A new example $\mathbf{x}$ can be classified using:

$$f(\mathbf{x}) = \text{sgn}(\mathbf{w}^*{}'\cdot \mathbf{x} + b^*) = \text{sgn}((\sum_{i=1}^{m} \alpha_i^* y_i \mathbf{x}_i)'\cdot \mathbf{x} + b^*) = \text{sgn}(\sum_{i=0}^{m} \alpha_i^* y_i \mathbf{x}_i{}'\cdot \mathbf{x} + b^*) \qquad (3.30)$$

Karush-Kuhn-Tucker (KKT) complementarity conditions (Taha, 1997) of optimization theory require that:

$$\alpha_i^*[y_i(\mathbf{w}^*{}'\cdot \mathbf{x}_i + b^*) - 1] = 0, \qquad i = 1, 2, ..., m \qquad (3.31)$$

Therefore, only examples $\mathbf{x}_i$ that satisfy the equalities in (3.19) can have non-zero coefficients α_i^*. Such examples lie on the two parallel hyperplanes separating two classes and thus are support vectors. Therefore, support vectors are examples whose related coefficients α_i^* are non-zero.

Since only a small part of examples are located on the two parallel hyperplanes, most examples satisfy the inequalities in (3.19), i.e., most α_i^* solved from the dual problem are null. Therefore, the coefficient vector $\mathbf{w}$ is a linear combination of a relatively small percentage of examples (support vectors). This leads to a sparse solution and it is very efficient in classifying new examples. Since only support vectors have non-zero coefficients α_i^*. The new example can be classified according to only support vectors,

$$f(\mathbf{x}) = \text{sgn}(\sum_{\text{support vector}} \alpha_i^* y_i \mathbf{x}_i{}'\cdot \mathbf{x} + b^*) \qquad (3.32)$$

3.4 SOFT MARGIN SVMS

In practice, not all training sets can be perfectly linearly separated by a hyperplane (Figure 3.6). In the case that the training set T is not linearly separable or we want to consider a general case and simply ignore whether or not the set T is linearly separable, the algorithm discussed in section 3.3 needs to be extended to solve imperfect separation problems. In that case, SVMs do not strictly require that there are no examples between separating hyperplanes H_1 and H_2. Instead, a penalty for the examples that cross the boundaries is introduced to take into account the misclassification errors.

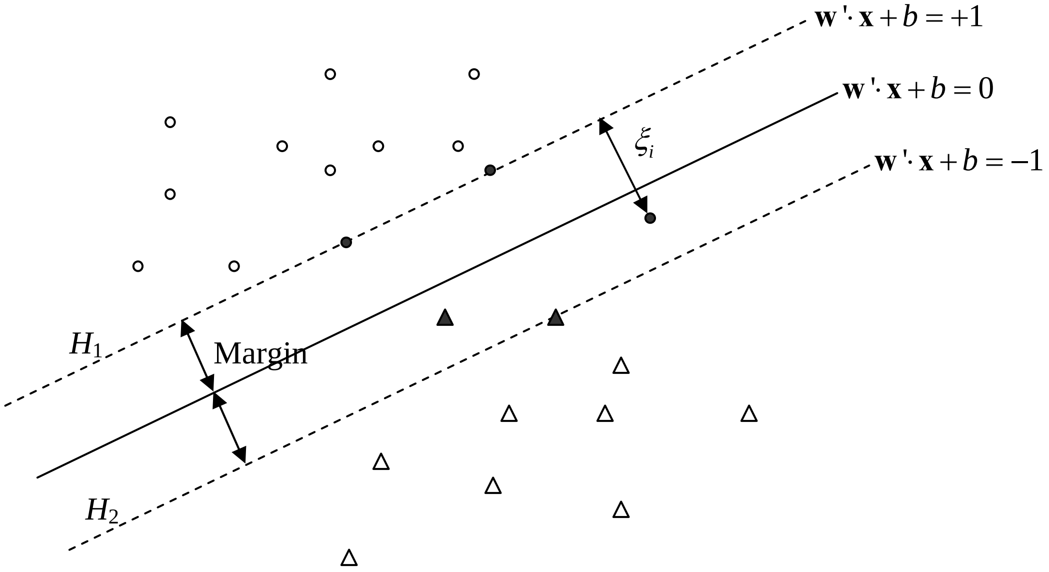

$\mathbf{w}'\cdot\mathbf{x}+b=+1$
$\mathbf{w}'\cdot\mathbf{x}+b=0$
$\mathbf{w}'\cdot\mathbf{x}+b=-1$
ξ_i
H_1
Margin
H_2

Figure 3.6: Optimal separating hyperplane between two classes of inseparable samples

This makes sense from the structural risk minimization point of view. Based on (3.8) in section 3.2, a good generalization performance can be reached when both the empirical risk and the VC confidence interval are small. In the linear SVMs discussed in section 3.3, a perfect separation is supposed. That means the empirical risk is set to be zero. Therefore, minimizing the VC dimension of the classifier by maximizing the margin can lead to a minimal VC confidence and thus a minimal upper bound for the expected risk given that (3.19) has to be met. Then, the question rises: is it possible to allow for a small number of misclassified points in order to achieve better generalization performance?

The answer is quite obvious. If the decrease in the VC confidence interval caused by a simpler classifier is larger than the increase of empirical risk caused by the misclassification error of imperfect separation, the upper bound of the expected risk will decrease and thus lead to a good generalization performance. Actually, this is a generalized optimal separating hyperplane (OSH) problem. Perfect separation problems discussed in section 3.3 are sub-optimal for certain training sets.

Soft margin SVMs aim to minimize the upper bound of the expected risk by minimizing the trade-off between margin (VC dimension, or VC confidence interval) and training error (empirical risk). To handle imperfect separation problems, non-negative slack variables ξ_i are incorporated into constraints (3.19) to consider misclassification errors:

$$y_i(\mathbf{w}' \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \qquad i = 1, 2, \ldots, m \tag{3.33}$$

$$\xi_i \geq 0, \qquad i = 1, 2, \ldots, m \tag{3.34}$$

Moreover, a penalty is added to the objective function to form a generalized expression for the upper bound of the expected risk,

$$F(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}' \cdot \mathbf{w} + C(\sum_{i=1}^{m} \xi_i)^l \tag{3.35}$$

The first term in (3.35) is related to the VC dimension of the classifier and thus corresponds to the VC confidence interval in (3.8). The second term in (3.35) is related to the misclassified points for the training set and thus corresponds to the empirical risk in (3.8). The regularization parameter C is used to control the trade-off between the empirical risk and the model complexity. A large C corresponds to stronger penalties for errors and will lead to a complex model to minimize the number of misclassified points. On the contrary, a small C corresponds to stronger penalties for complexity and will lead to a simple model to maximize the margin $\frac{2}{\|\mathbf{w}\|}$.

Usually, l is set to be 1. Hence the optimization problem becomes,

$$\begin{aligned} &minimize: F(\mathbf{w},\boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}'\cdot\mathbf{w} + C\sum_{i=1}^{m}\xi_i \\ &subject\ to: y_i(\mathbf{w}'\cdot\mathbf{x}_i + b) + \xi_i - 1 \geq 0, \quad i = 1,2,\ldots,m \\ &\qquad \xi_i \geq 0, \quad i = 1,2,\ldots,m \end{aligned} \tag{3.36}$$

Introducing Lagrange multipliers $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, we have the following Lagrangean function,

$$L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{1}{2}\mathbf{w}'\cdot\mathbf{w} + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i[y_i(\mathbf{w}'\cdot\mathbf{x}_i + b) + \xi_i - 1] - \sum_{i=1}^{m}\beta_i\xi_i \tag{3.37}$$

Similar to the linear separable case, we must now minimize $L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})$ with respect to $\mathbf{w},b,\boldsymbol{\xi}$ and simultaneously maximize $L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})$ with respect to $\boldsymbol{\alpha},\boldsymbol{\beta}$. Applying gradient vanishing conditions and simplifying yields,

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i = 0 \tag{3.38}$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{m}\alpha_i y_i = 0 \tag{3.39}$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0 \tag{3.40}$$

Substituting (3.38)-(3.40) into (3.37) yields the dual problem:

$$\begin{aligned} &maximize: L(\boldsymbol{\alpha}) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \mathbf{x}_i{}'\cdot\mathbf{x}_j \\ &subject\ to: \sum_{i=1}^{m}\alpha_i y_i = 0 \\ &\qquad 0 \leq \alpha_i \leq C, \quad i = 1,2,\ldots,m \end{aligned} \tag{3.41}$$

The only difference between perfectly separating case and the imperfectly separating case is that, the Lagrange multipliers $\alpha_i s$ are bounded above by C in an imperfectly separating case instead of unbounded in a perfectly separating case.

After the optimum Lagrange multipliers α_i have been determined, we can compute the optimum coefficient vector $\mathbf{w}^*$ and the optimal offset b^*. The solution is given by:

$$\mathbf{w}^* = \sum_{i=1}^{m} y_i \alpha_i^* \mathbf{x}_i \tag{3.42}$$

The offset b^* can be found from:

$$\alpha_i^* (y_i (\mathbf{w}^* \cdot \mathbf{x}_i + b^*) - 1) = 0 \tag{3.43}$$

for any i such that α_i^* is not zero.

Based on the new KKT conditions:

$$\alpha_i^* (y_i (\mathbf{w}^* \cdot \mathbf{x}_i + b^*) - 1 + \xi_i^*) = 0 \tag{3.44}$$

$$(C - \alpha_i^*) \xi_i^* = 0 \tag{3.45}$$

The points in the training set can be classified into three categories:

1. $\alpha_i^* = 0$: normal points (non-support vectors)
2. $\alpha_i^* > 0$: support vectors
    a. $0 < \alpha_i^* < C$: margin vectors
        - $\xi_i^* = 0$
        - The support vectors lie at a distance $\frac{1}{\|\mathbf{w}\|}$ from the OSH

b. $\alpha_i^* = C$: non-margin vectors

- $\xi_i^* > 1$: misclassified points
- $0 \le \xi_i^* \le 1$: correctly classified within margin

Figure 3.7 visually shows these three kinds of points in the training set.

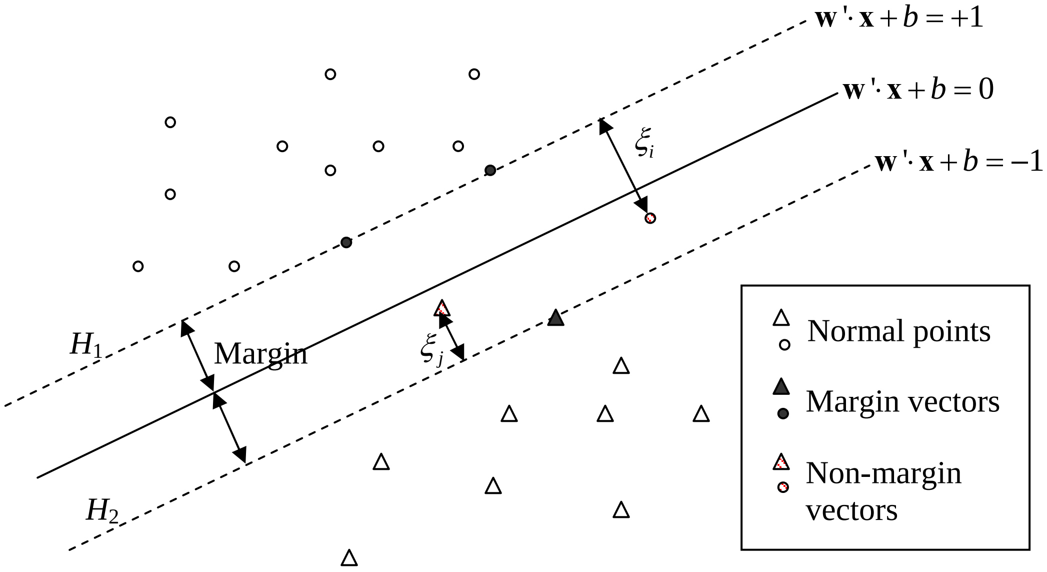

$\mathbf{w}' \cdot \mathbf{x} + b = +1$
$\mathbf{w}' \cdot \mathbf{x} + b = 0$
$\mathbf{w}' \cdot \mathbf{x} + b = -1$
ξ_i
ξ_j
H_1
Margin
H_2
Normal points
Margin vectors
Non-margin vectors

Figure 3.7: Three kinds of points in the training set

3.5 NON-LINEAR SVMS

In most practical applications, the two classes cannot be linearly separated. To extend the linear learning machine to work with non-linear cases, SVMs use a kernel method to map the non-linearly separable classes from input space to a high dimensional feature space, in which the non-linearly separable classes can be separated by a linear optimal hyperplane.

As shown in Figure 3.8, a mapping function $\Phi(\mathbf{x})$ (quadratic transform) is used to map examples in the input space $\mathbf{S} \subset \mathbf{R}^2$ into a high dimensional feature space $\mathbf{F} \subset \mathbf{R}^3$. The training set, which cannot be linearly separated in the input space, now become linearly separable in the feature space.

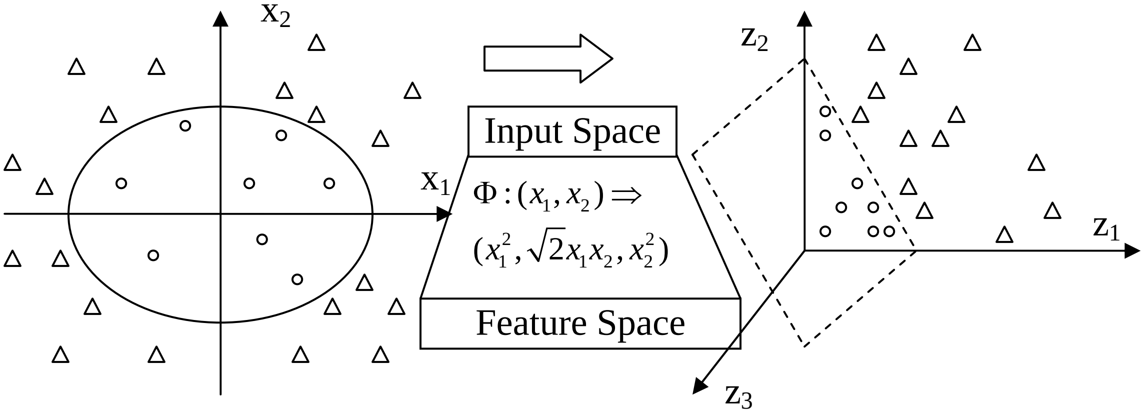


Figure 3.8: Mapping from input space to feature space (reprinted from Smola et al., 1999)

From the above example, we can find that appropriate choice of mapping function $\Phi(\mathbf{x})$, that is, appropriate construction of feature space, leads to linear separability. However, explicit use of such a mapping function would cause some efficiency problems. The dimension of the feature space is usually much higher than that of the input space, which will dramatically increase the number of parameters need to be solved. For example, polynomial transformation of degree d over N attributes in the input space leads to $\begin{pmatrix} d+N-1 \\ d \end{pmatrix} = \frac{(d+N-1)!}{d!(N-1)!}$ attributes in the feature space. Moreover, the transformation operator $\Phi(\mathbf{x})$ might be computationally expensive.

After checking the optimal separating hyperplane problem, however, it is easy to find that: an example $\mathbf{x}$ in the input space can be represented as $\Phi(\mathbf{x})$ in the feature space. Since the linear separation is performed in the feature space, the optimization problem shown in (3.41) can be rewritten as:

$$maximize: L(\boldsymbol{\alpha}) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \Phi(\mathbf{x}_i)' \cdot \Phi(\mathbf{x}_j) \qquad (3.46)$$

Since only the dot product of two vectors in the feature space appears in the optimization problem, we can define a kernel function K as,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)' \cdot \Phi(\mathbf{x}_j) \tag{3.47}$$

Hence, we do not need to know explicitly the mapping function, but can simply use a ***kernel function*** (KF) of the input space to represent the dot product in the high dimensional feature space. All the previous derivations in linear SVMs hold (substituting dot product with the kernel function), since we are still doing a linear separation, but in a different space.

The use of the kernel function greatly simplifies the mapping problem and improves the computational efficiency. While the mapping function needs to map $\mathbf{R}^n$ to $\mathbf{R}^l$ (usually $n << l$), the kernel function can map $\mathbf{R}^n \times \mathbf{R}^n$ to $\mathbf{R}$ and thus reduce the computational burden dramatically. For example, consider the following map:

$$\Phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, 1) \tag{3.48}$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)' \cdot \Phi(\mathbf{x}_j) = (\mathbf{x}_i' \cdot \mathbf{x}_j + 1)^2 \tag{3.49}$$

Using the kernel function, we can greatly simplify the computation. The kernel (3.49) is known as second order polynomial kernel.

Hence the optimization problem can be rewritten as,

$$maximize: L(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{3.50}$$

Thus the new example can be classified according to,

$$f(\mathbf{x}) = sign(\sum_{\text{support vector}} \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^*) \tag{3.51}$$

The existence of a kernel function and an appropriate feature space is problem-dependent and has to be established for each new problem. The following lists some commonly used kernels:

- Dot kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i ' \cdot \mathbf{x}_j$
- Polynomial kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i ' \cdot \mathbf{x}_j + 1)^d$
- Radial basis kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-|\mathbf{x}_i - \mathbf{x}_j|^2 / \sigma^2}$
- Sigmoid kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i ' \cdot \mathbf{x}_j + c)$

According to the definition of kernel function (3.47), it should be able to be expressed as dot product in a high dimensional space. According to Mercer's condition (Cristianini and Shawe-Taylor, 2000), any positive definite function $K(x, y)$ can be expressed as a dot product in a high dimensional space. Therefore, any kernels that meet Mercer's condition can be used to construct SVMs:

$$\iint K(\mathbf{u}, \mathbf{v}) g(\mathbf{u}) g(\mathbf{v}) d\mathbf{u} d\mathbf{v} > 0, \qquad \forall g : \int g^2(\mathbf{u}) d\mathbf{u} < \infty \tag{3.52}$$

A kernel can also be constructed by combining other kernels. Assume K_1 and K_2 are kernels, then the following expressions can generate a valid new kernel:

- $K(\mathbf{x}_i, \mathbf{x}_j) = aK_1(\mathbf{x}_i, \mathbf{x}_j)$
- $K(\mathbf{x}_i, \mathbf{x}_j) = K_1(\mathbf{x}_i, \mathbf{x}_j) + K_2(\mathbf{x}_i, \mathbf{x}_j)$
- $K(\mathbf{x}_i, \mathbf{x}_j) = K_1(\mathbf{x}_i, \mathbf{x}_j) K_2(\mathbf{x}_i, \mathbf{x}_j)$
- $K(\mathbf{x}_i, \mathbf{x}_j) = e^{K_1(\mathbf{x}_i, \mathbf{x}_j)}$

3.6 EXTENDED SVMS

SVMs have been extended to meet the requirements of different applications. Some important extensions of SVMs are:

- One-class SVMs (Schölkopf et al., 2001): One extension of SVMs to handle one-class classification problem in which only the training data of one class is available and the target class is modeled by fitting a hypersphere with minimal radius around it.
- Multi-class SVMs: SVMs are extended to deal with K-class pattern classification problem. Multi-class SVMs are usually implemented by combining several binary SVMs to solve a given multi-class problem. Popular methods are: one-versus-all method using winner-takes-all strategy (Hastie and Tibshirani, 1998), which constructs K hyperplanes between class k and the $K-1$ other classes; and one-versus-one method implemented by max-wins voting (Platt, 1999), which constructs $\frac{K(K-1)}{2}$ hyperplanes between any pairwise of two classes.
- Support Vector Regression (SVR) (Smola, 1996): One extension of SVMs to apply to regression task by the introduction of an alternative loss function—ε intensive loss.
- Reduced SVMs (Lee and Mangasarian, 2000): The reduced support vector machine (RSVM) is proposed to avoid the computational difficulties in classifying massive dataset by selecting a small random subset from the entire dataset to generate a reduced kernel (rectangular) matrix without sacrificing the prediction accuracy.
- Least Squares SVMs (Suykens and Vandewalle, 1999): Least squares support vector machines (LS-SVMs) are re-formulations to the standard SVMs by using a regularized least squares cost function with equality constraints, leading to linear Karush-Kuhn-Tucker systems. The solution can be solved efficiently by iterative methods like the conjugate gradient algorithm.

3.7 CHAPTER SUMMARY

This chapter presented a brief introduction to support vector machines: their basic idea, unique characteristics, and attractive advantages. This was followed by a detailed discussion of SVMs' mathematic foundation, namely, statistical learning theory. A statistical learning problem of binary classification was put forward, the lost function was introduced to measure the quality of prediction, and then the risk function was employed to describe the regularization performance of the classification method. The empirical risk minimization principle, which is used in most traditional methods, was discussed and its shortcomings were analyzed. A more superior principle, namely, structural risk minimization, was then introduced to circumvent these shortcomings.

As an approximate implementation of structural risk minimization principle, SVMs were discussed in detail: from a specific case, namely, linear separable optimal separating hyperplane problem, to a more general case, namely, soft margin SVMs, which allow misclassification errors. Then, kernel function was introduced to extend the linear SVMs to cope with nonlinear separation problem. Finally, some extended SVMs were listed.

CHAPTER 4: MODELING LAND USE CHANGE USING STANDARD SVMS

4.1 INTRODUCTION

Owing to its incomparable generalization performance, SVMs have steadily been gaining research attention and have been increasingly used in a wide variety of application domains ranging from pattern recognition, time series prediction, to signal processing. The great success that SVMs have achieved in these applications has attracted increased efforts by researchers to extend their applications.

However, SVMs are relatively new for the GIS community. Although some progressive researchers have attempted to introduce SVMs to GIS applications (Zhu and Blumberg, 2002; Guo et al., 2005), SVMs' power has not been fully recognized by researchers in the GIS community. For land use studies, SVMs have been used in land cover classification from remote sensing data and have shown better and more stable accuracy than traditional methods (Huang et al., 2002; Pal and Mather, 2003). However, to our knowledge, SVMs on land use change modeling is a new topic to be more fully explored.

This research aims to apply SVMs on land use change modeling. Land use change in Calgary, Canada from 1985-2001 is used as case study. This chapter will cover the model development and implementation. Some practical issues needed to be solved when applying SVMs on a specific application, namely, regularization parameter selection, kernel selection, and input vector normalization, are discussed. Performance evaluation is also addressed. The performance of SVMs is compared with that of a well studied land use change modeling approach, namely, spatial logistic regression. The comparison clearly demonstrates the superiority of SVMs, especially on the capacity and efficiency to classify the changed land parcels.

4.2 STUDY AREA

Calgary is located in southern Alberta on the eastern edge of the Rocky Mountain Foothills at the merging of the Bow and Elbow rivers (Figure 4.1). Calgary covers an area of about 720 square kilometers and thus a 25 km x 35 km rectangle should be sufficient to include it.

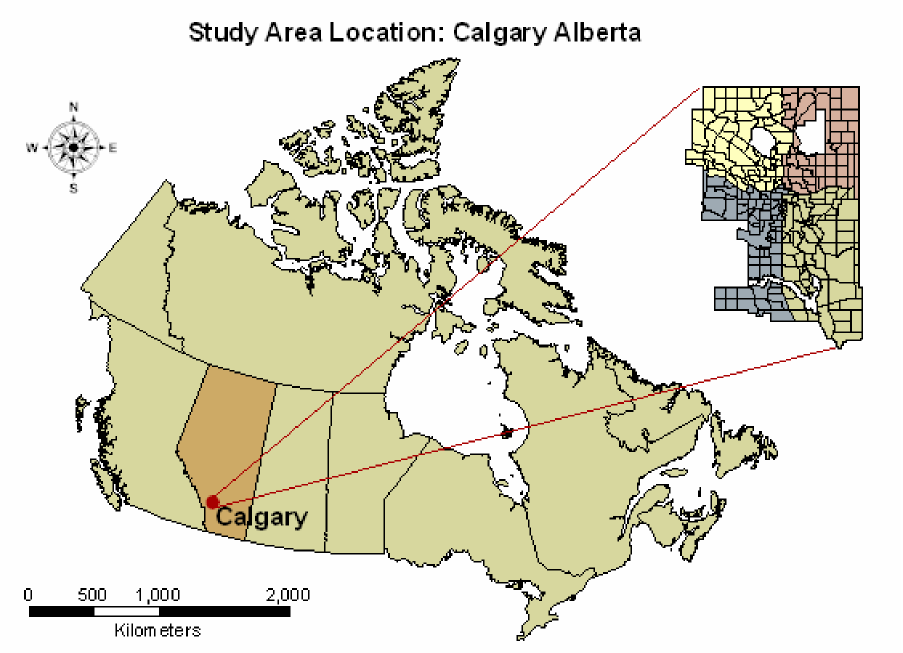


Figure 4.1: Location of Calgary City

Calgary was originally established as a frontier settlement by the Northwest Mounted Police (NWMP) in 1875. The arrival of the Canadian Pacific Railway transcontinental line in 1883 brought growth and development to Calgary. Thousands of settlers, businessmen and tourists poured into this area. The newly introduced economic grazing land leasing policy, which encouraged large ranching operations, turned Calgary into the center of Canada's cattle marketing and meatpacking industries. Agriculture was the key component of Calgary's economy until the discovery of oil and gas in 1914 in the Turner Valley area 30 miles south of Calgary. From then on, Calgary became the "oil and gas capital of Canada". After more than one hundred and thirty years of growth, the population in Calgary has grown from 1,000 to nearly 1,000,000, and Calgary has become the largest city in Alberta and the fifth largest in Canada.

In the past two decades, Calgary has experienced tremendous economic growth. This growth does not only express itself in a substantial increase in the urban population, but also in the fast expansion of the urban area. The expansion indicates a transformation of vacant and agricultural land use to construction of urban fabrics including residential, industrial and infrastructure developments.

4.3 DATA PREPARATION

Considering previous literature, the context of Calgary, and the data availability, this study included the following data in the land use change model: chronological land use data, demographic data, and transportation data (major roads and LRT lines), elevation data, community map, city amenity map, community service center map, and shopping center distribution map. Other factors, such as social and economic data, are very important for driving land use change. However, they are not considered in this study due to the difficulty in obtaining and quantifying.

4.3.1 Land Use Data

Land use data was classified from Landsat TM (thematic mapper) and ETM+ (enhanced thematic mapper plus) images using eCognition 2.1 software. Six Landsat TM and ETM+ images are available for the study area for the years of 1985, 1990, 1992, 1999, 2000, and 2001. The images were obtained from the Maps, Academic Data, Geographic Information Center (MADGIC) in the University of Calgary Library and from the internet. Detailed information for these six images is listed in Table 4.1.

Table 4.1: Detailed information of Landsat TM and ETM+ images

Acquisition Data	Format	Spatial Resolution	Projection
1985-07-26	TM	28.5m	NAD83 UTM Zone 12N
1990	TM	30m	None
1992-08-14	TM	28.5m	NAD83 UTM Zone 12N
1999-07-09	ETM+	28.5m	NAD83 UTM Zone 12N
2000-08-28	ETM+	28.5m	NAD83 UTM Zone 12N
2001-08-15	ETM+	28.5m	NAD83 UTM Zone 12N

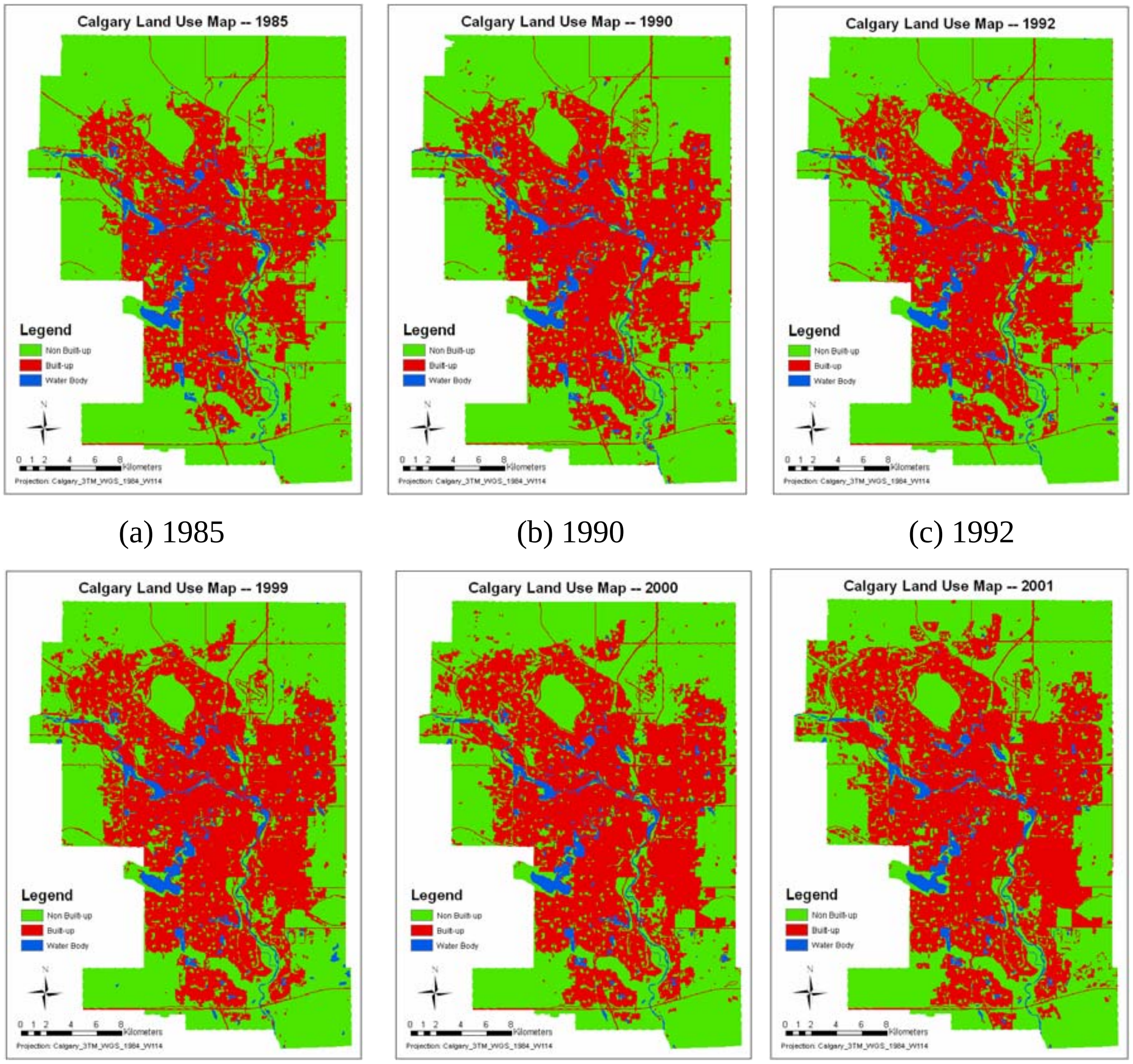
The land use classification was implemented using eCognition 2.1 software. A hybrid classification scheme (Table 4.2) was employed according to the *US Geological Survey Land Use/Land Cover Classification System* (Jensen, 1996).

Table 4.2: Land use classification scheme

Level I	Level II
Built-up	Residential and Commercial
	Industrial
	Transportation
Non Built-up	Parks
	Vacant area
Water Bodies	Water

The purpose of this study is to apply a novel method, namely, SVMs, to land use change modeling, to solve the problems that occur when implementing SVMs in a new application, and to improve the ability of SVMs to better fit the characteristics of land use change data. Therefore, a simple binary land use change modeling problem was adopted, which would serve the research objective and where the findings of binary land use change modeling can be easily extended for modeling multinomial land use change by using one of the multi-class SVMs strategies discussed in section 3.6.

Specifically, only land use change from non built-up land to built-up land is considered in this study. Water bodies are treated as constraints that are not suitable for development. The 1990 land use data was re-sampled to a raster layer with cell size 28.5m x 28.5m to be consistent with other land use data. All classified land use layers were projected to the *Calgary_3TM_WGS_1984_W114* coordinate system, which served as the coordinate system of all the data used in this study. Figure 4.2 (a) ~ (f) are land use maps of 1985, 1990, 1992, 1999, 2000, and 2001 respectively.


Calgary Land Use Map -- 1985
Calgary Land Use Map -- 1990
Calgary Land Use Map -- 1992
Calgary Land Use Map -- 1999
Calgary Land Use Map -- 2000
Calgary Land Use Map -- 2001
Legend
🟩 Non Built-up
🟥 Built-up
🟦 Water Body
N
0 1 2 4 6 8 Kilometers
Projection: Calgary_3TM_WGS_1984_W114

(a) 1985 (b) 1990 (c) 1992

(d) 1999 (e) 2000 (f) 2001

Figure 4.2: Calgary land use maps

4.3.2 Demographic Data

The demographic data was obtained from multiple sources: Statistics Canada, the City of Calgary, and MADGIC at the University of Calgary Library. Census tract maps and census profiles for 1991, 1996, 2001, and 2003 were available. In order to use the available demographic data in this research, we needed to solve spatial and temporal discrepancy problems.

The spatial problem involved in analyzing data gathered from the four different census years is the change in various attributes of each Census Tract (CT) between each year. These changes may include alterations in boundary shape and size; the population within each CT; and amalgamations or segmentations of CTs in previous years. The changes within the CTs will cause difficulty for the traditional population density calculation method, in which the population densities of land parcels in the same CT are considered to be uniform and take the value of:

$$Pop\_Dens(CT_i) = \frac{Pop(CT_i)}{Area(CT_i)} \tag{4.1}$$

Where $Pop\_Dens(CT_i)$, $Pop(CT_i)$ and $Area(CT_i)$ are the population density, total population, and area of census tract i respectively.

Since the population densities in two sides of a CT boundary vary greatly, the population density of a specific site will be artificially affected if it is located in different CTs in different years because of the changes to CTs. Spatial interpolation is suggested to address this problem and to help achieve a smooth population density distribution (Goodchild et al., 1993). This process involves interpolating over discrete points representing each of the CTs and constructing a point surface which represents the population density of each point in the given area. Several techniques are available for interpolation: Spline, Kriging, Inverse Distance Weighting (IDW) and finally, Triangular Irregular Networks (TIN). To determine which method is more appropriate for Calgary's population distribution, we tried all the methods mentioned above and compared the results of each method with the actual data in some testing area where the population density data is available. After extensive analysis, it has been determined that the best interpolation function for modeling the population density within Calgary was IDW. Therefore, IDW was used to construct population density maps for the four census years.

The temporal problem is that the census years are not consistent with the years that the land use maps were generated. The land use data we want to model is in the year 1985, 1990, 1992, 1999, 2000, and 2001 but the detailed census data we have is in the year 1991, 1996, 2001, and 2003. Therefore, temporal interpolation (Martin and Bracken, 1991) is also needed. First, an average annual population growth rate was computed between two consecutive years:

$$r_{ij} = \frac{\ln(\frac{Pop(t_j)}{Pop(t_i)})}{t_j - t_i} \tag{4.2}$$

where $Pop(t_i)$ is the total population of year t_i and r_{ij} is the annual population growth rate between year t_i and year t_j.

Then, the weighted average annual population growth rate from 1991 to 2001 was calculated using the time span of each period as the weight:

$$\overline{r} = \frac{\sum(t_j - t_i)r_{ij}}{\sum(t_j - t_i)} \tag{4.3}$$

Thirdly, the weighted average annual population growth rate $\overline{r}$ can be applied to estimate the population density of each cell from the population density of the corresponding cell in every census year and the estimated values were weighted to get the final population density for each cell according to:

$$Pop\_Dens(t) = \frac{\sum Pop\_Dens(t_i)e^{\overline{r}(t-t_i)}/(t-t_i)}{\sum 1/(t-t_i)} \tag{4.4}$$

To verify the accuracy of the temporal interpolation, the total populations of each expected year was calculated and compared with the actual populations (Table 4.3). The results showed that the interpolation scheme worked quite well.

Table 4.3: Interpolated population vs. actual population

Population	**1985**	**1990**	**1992**	**1999**	**2000**	**2001**
Interpolated	617,664	691,902	721,281	834,800	856,445	878,150
Actual	625,143	692,885	717,133	842,388	860,749	876,519
Difference	-7,479	-983	-4,148	-7,588	-4,304	-1,631

4.3.3 Other Data

Other data, including major road shape file, LRT lines and stations shape files, elevation raster layer, community boundary shape file, city amenities shape file, community service centers shape file, and shopping centers shape files, were collected from different sources: the Alberta Department of Transportation, the City of Calgary, MADGIC in the University of Calgary Library, and the internet.

All these shape files/raster layers were compiled in ESRI ArcMap v9.1®. All layers were projected to the *Calgary_3TM_WGS_1984_W114* coordinate system. Raster layers were re-sampled using a cell size of 28.5 meters and snapped to the grid of land use data. Slope raster was generated from the elevation raster using the ArcMap spatial analyst extension. Sequential shape files of the modeled years for road networks, LRT lines/stations, city amenities, community service centers, and shopping centers were generated according to the construction years of each utility. Then the ArcMap spatial analyst extension was used to generate distance raster layers to these utilities in each modeled year according to the Euclidian distance.

4.4 MODEL DEVELOPMENT

4.4.1 Causal Factors

Like most land use change models, the model used in this study is also cell-based. Based on the causal factor discussion in section 2.2 and our previous land use change modeling research, nine causal factors were considered in this study. A summary of these causal factors are shown in Table 4.4.

Table 4.4: Summary of causal factors for the land use change model

Causal Factor	Description
Pop_Dens	Population density of the cell
Slope	Slope of the cell
Dist_LRTSta	Distance from the cell to the nearest LRT station
Dist_Road	Distance from the cell to the nearest major road
Dist_CityCen	Distance from the cell to the downtown area
Dist_Amenity	Distance from the cell to the nearest city amenity
Dist_CommServ	Distance from the cell to the nearest community service center
Dist_Shopping	Distance from the cell to the nearest shopping center
Per_Avail	Percentage of avail land in the surround area within 114m radius

Three categories of causal factors were employed: (1) site specific characteristics, (2) proximity, and (3) neighborhood characteristics. Since population is a leading force propelling global land use change, it is considered to be a chief predictor of land use change. Slope has great impact on the construction feasibility and cost. As well, slope is also an important site specific characteristic which affects land use change probability. Proximity is a prime cause of urban expansion. Proximity variables measure the minimum Euclidean distances to the nearest transportation network (road/LRT), downtown area, city amenities, community service centers, and shopping malls respectively. For neighborhood characteristics, the distance decaying mechanism of various factors is signified by the type and size of the selected neighborhood. In this study, an extended Moore neighborhood with a four cells (about 114 meters) radius was selected after considering the effect of neighboring impacts in current land use distribution.

4.4.2 SVMs Modeling Framework

A SVMs land use change modeling framework was developed using C++ programming language. The modeling framework was integrated into ArcMap as an extension so as to make use of ArcMap's powerful spatial data processing and visualization capacity.

Figure 4.3 gives an overview of the main components of the SVMs land use change modeling framework.

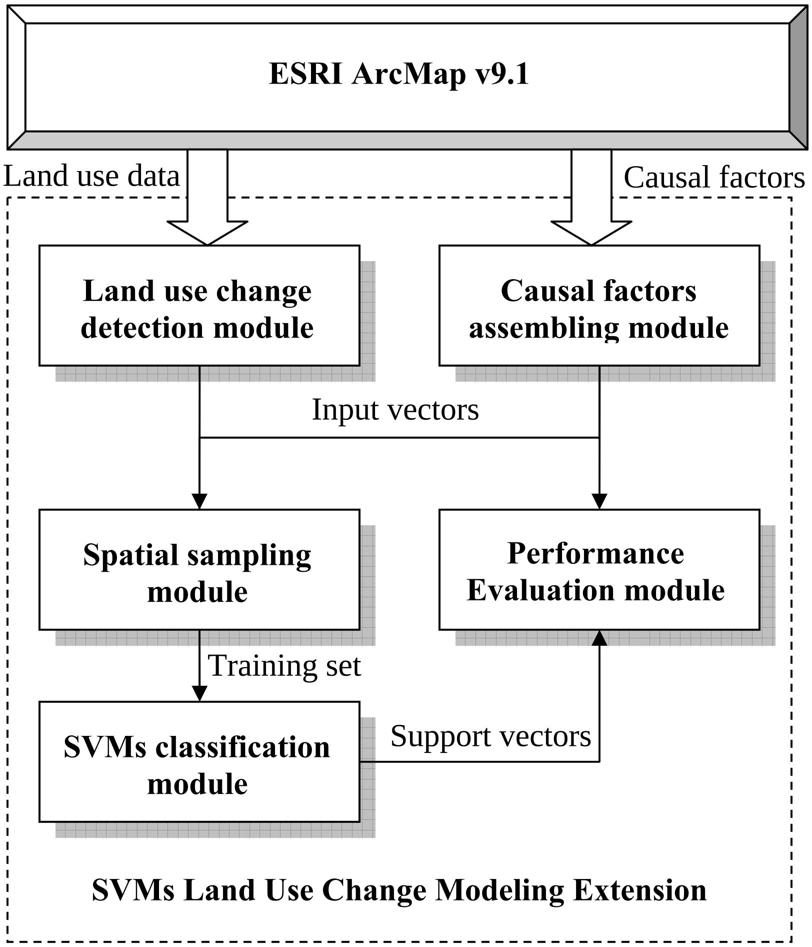

ESRI ArcMap v9.1
Land use data
Causal factors
Land use change detection module
Causal factors assembling module
Input vectors
Spatial sampling module
Performance Evaluation module
Training set
SVMs classification module
Support vectors
SVMs Land Use Change Modeling Extension

Figure 4.3: General structure of the SVMs land use change modeling framework

The land use data and causal factor data were imported into ArcMap. Then land use layers of different years were sent to the land use change detection module. A post-classification comparison method was developed to detect land use changes between 1985-1990, 1990-1992, 1992-1999, 1999-2000 and 2000-2001. Bi-temporal change maps were generated by overlaying individual classifications. The cells remained in non built-up land use and the cells that changed their land use from non built-up to built-up were collected for the binary land use changing modeling.

Different kinds of causal factor layers for different years were sent to the causal factors assembling module. The causal factors assembling module linked the related cells in different layers of the same year together and constructed an attribute vector for each cell. Then the attribute vectors were able to be combined with the land use changes detected to generate a set of labeled vectors. Each vector had a label to indicate its category: -1 refers to cells remaining in non built-up land use; +1 refers to cells changing their land use from non built-up to built-up. The vectors were candidates for model training and evaluation.

A small set of vectors was selected from the candidates as the training set for the land use change model. Nevertheless, the selection procedure cannot be performed arbitrarily. Although SVMs do not need to assume the distribution or error structure of the data, which is quite common for most statistical traditional methods, it does need to satisfy a basic condition: the data should be statistically independent and identically distributed. Land use change modeling, however, involves substantial amounts of spatial variables, and spatial land use data have the tendency to be dependent, a phenomenon known as spatial autocorrelation. This can be defined as the property of random variables to take values over distance that are more or less similar than expected for randomly associated pairs of observations, due to geographic proximity. Therefore, specific approaches should be considered to remove spatial autocorrelation. Otherwise, unreliable parameter estimation or inefficient estimates and false conclusions will result.

A spatial sampling scheme was employed in this study. On one hand, spatial sampling can expand the distance interval between sampled sites and thus mitigate the spatial autocorrelation since spatial autocorrelation should be subject to distance decay theoretically. On the other hand, spatial sampling leads to a smaller sample size that loses certain information. Nevertheless, it is a sensible approach to remove spatial autocorrelation and a reasonable design of the spatial sampling scheme will allow for a perfect balance between the two sides.

This study used the regular sampling technique of a non-overlapping moving window. The center cell was retained for each window. To examine whether observations proximate in space were spatially autocorrelated, as defined by sequential occurrences of like land use change, a check of joins was determined each time by comparing land use change types of "adjacent" cells, which are those center cells and cells to the east, west, north, and south of the center cell. A plot showing the number of joins with increasing window size is provided in Figure 4.4. A 7×7 sampling window, or sampling distances of 199.5 meters, was ultimately chosen to sample the data. At this distance, we effectively filtered out much of the spatial autocorrelation and yet had enough samples for regression analysis.

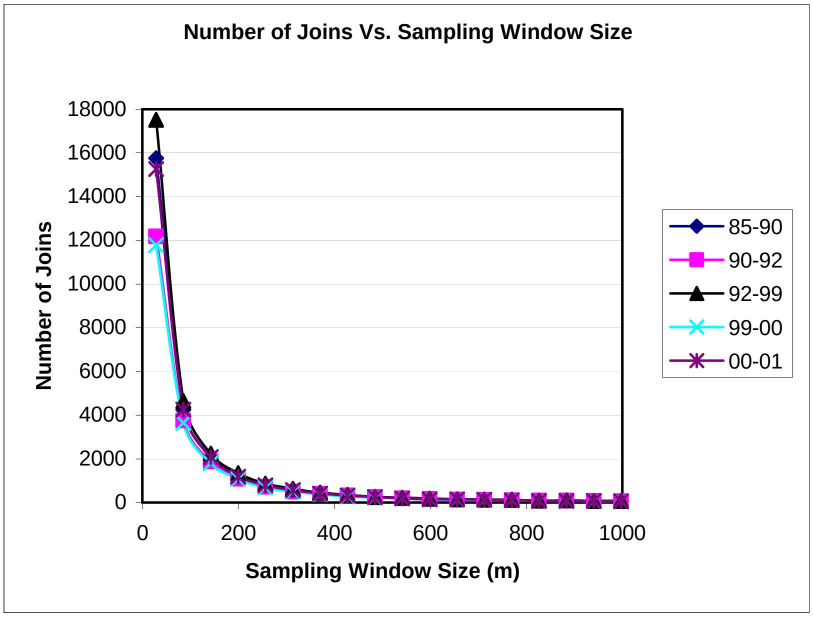


Figure 4.4: Number of joins by sampling window size

After spatial sampling, an independent and identically distributed training set was obtained. The labeled training set was sent to the SVMs classification module to find an optimal separating hyperplane in kernel incurred high dimensional feature space. Sequential minimal optimization (please check Appendix B for details) was implemented to solve the quadratic optimization problem in SVMs to calculate the optimal Lagrange multipliers for each vector in the training set. A small set of support vectors with non-zero Lagrange multipliers can be detected to represent the boundary between two classes and to predict the possible label of any unseen vector from the same distribution of the training set.

Evaluation of the model's classification accuracy based on the training set itself can only demonstrate the model's ability to describe the pattern of the training set. In order to evaluate the stability and the generalization performance of SVMs for land use change modeling, several groups of data other than the training set were used to evaluate the classification accuracy. In this study, ten sets of non-overlapped samples were randomly selected from the candidate vectors of each year. Each of these ten sets accounted for 5% of the related candidate vectors.

The percentage of correct prediction (PCP), which measures the overall concordance between a classification and the actual land use conversion, was used to assess the goodness-of-fit of the models. An efficient way to assess the goodness-of-fit of classification was used in this study, which cross tabulates predictions with observations and calculates the overall concordance, change detection capacity, and change detection efficiency. Table 4.5 shows an example of the cross evaluation table used in this study.

Table 4.5: An example of the cross evaluation table

Observed	Predicted		Total
	Non built-up	Built-up	
Non built-up	Num(NN)	Num(NB)	Num(O-N)
Built-up	Num(BN)	Num(BB)	Num(O-B)
Total	Num(P-N)	Num(P-B)	Num(Total)

In table 4.5, $Num(NN)$ is the number of cells with a label of -1 (non built-up) and was classified as non built-up. $Num(NB)$ is the number of cells with a label of -1 (non built-up) but was classified as built-up. $Num(BN)$ is the number of cells with a label of +1 (built-up) but was classified as non built-up. $Num(BB)$ is the number of cells with a label of +1 (built-up) and was classified as built-up. $Num(P-N)$ and $Num(P-B)$ are the number of cells with a label of -1 and +1 respectively. $Num(O-N)$ and $Num(O-B)$ are the number of cells that were classified as non built-up and built-up respectively. $Num(Total)$ is the size of the training set/evaluation set. Based on the cross evaluation table, three important indicators can be calculated:

$$PCP = \frac{Num(NN) + Num(BB)}{Num(Total)} \tag{4.5}$$

$$PCP1 = \frac{Num(BB)}{Num(O-B)} \tag{4.6}$$

$$PCP2 = \frac{Num(BB)}{Num(P-B)} \tag{4.7}$$

PCP indicates the overall accuracy of the classifier. $PCP1$ is measured by the percentage of changed land parcels whose land use changed can be predicted by the model. It reveals the capacity of the classifier to detect the land use change. The higher the $PCP1$, the more the changed land parcels can be correctly predicted. $PCP2$ is measured by the percentage of correctly predicted changed land parcels over the total land parcels classified as changed. It exhibits the efficiency of the classifier to detect the land use change. A higher $PCP2$ means the model can predict the changed land parcels with less incorrect predictions and thus a higher efficiency.

4.5 REGULARIZATION PARAMETER SELECTION

The performance of SVMs is very sensitive to its configuration, which includes regularization parameter *C*, kernel function type and parameter. However, configuration of SVMs is problem-dependent and has to be established for each new application. Although some efforts have been made to establish some theoretical basis for determining optimal SVMs configuration (Cherkassky and Ma, 2004; Ustun et al., 2005), no widely accepted theory is available for choosing good SVMs parameters.

Regularization parameter C is used to control the trade-off between the empirical risk and the model complexity. A large C will lead to a complex model and thus tends to overfit the training set. On the contrary, a small C will lead to a simple model and thus cannot model the underlying pattern effectively. Therefore, an optimal regularization parameter C should be identified to better trade off the model complexity and the empirical risk, and thus gain the best generalization performance.

It is suggested that a reasonable starting point and/or default value for the regularization parameter C is (Vapnik and Chapelle, 1999):

$$C_{def} = \frac{1}{\sum K(\mathbf{x}_i, \mathbf{x}_j)} \tag{4.8}$$

Cross-validation should be used to search for the optimal C on a log-scale, for example: $C = 10^{-3} C_{def}, 10^{-2} C_{def}, ..., 10^{3} C_{def}$. In this research, $C = 0.1,1,10,100,1000,10000$ were tested to explore the impact of different regularization parameters on the performance of SVMs. Figure 4.5 shows the PCPs of the standard SVMs with dot kernel and different regularization parameters in different time periods.

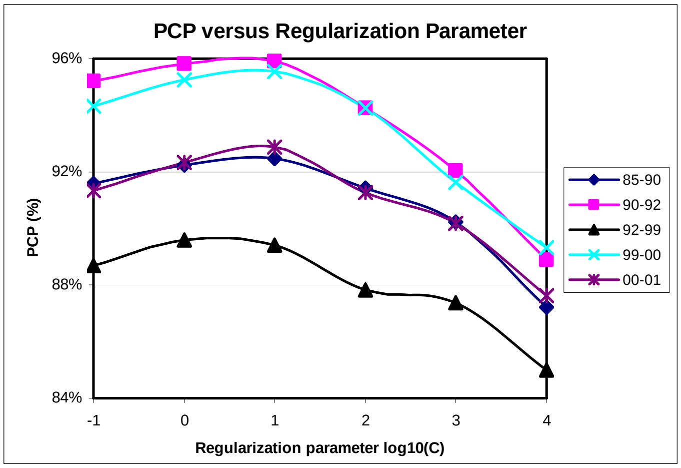


Figure 4.5: PCP versus regularization parameter

From Figure 4.5, it is clear that the standard SVMs with regularization parameter $C = 10$ have the best PCP. However, one must recognize that the optimal regularization parameter is not only dependent on the special application, but also on the selection of kernel function type and its parameter. Therefore, cross-validation over regularization parameter and kernel function (with different types and parameters) should be performed to obtain an optimal SVMs setting for the application.

4.6 KERNEL SELECTION

According to the discussion in section 3.5, kernel function maps the input space into a high dimensional feature space and converts the non-linear boundary between two classes in the input space into a linear one in the feature space. Hence, the efficiency of the kernel function to convert the non-linear boundary to a linear one will greatly affect the performance of SVMs. The proper selection of kernel function relies on the understanding of the data pattern, which requires in-depth domain knowledge. Currently, the most commonly used kernel function and parameter selection approach is a grid search approach. That is trying different kernel functions with different parameter settings and comparing their performance. This study uses a similar strategy. First, we compared the performances of SVMs for different parameter settings of two widely used kernel functions, namely, polynomial kernel and radial basis function (RBF), and found the best parameter setting for each kernel type. Then, the performances of different kernel functions, which included dot kernel, polynomial kernel, and RBF kernel, were evaluated.

The parameter of the polynomial kernel, $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i{}' \cdot \mathbf{x}_j + 1)^d$, is a polynomial of order d (degree). Figure 4.6 shows the impact of d on the SVMs performance. No obvious trend could be observed when the polynomial order d increased from 1 to higher values. When the polynomial order was increased, the performance of the SVMs oscillated slightly (PCPs varied about ±2% for each training set). Since the PCPs for the tested polynomial orders in all five data sets were fairly high, we could say that the polynomial kernel gained steady high performance for the experiment data sets and no obvious trend can be found for polynomial order selection.

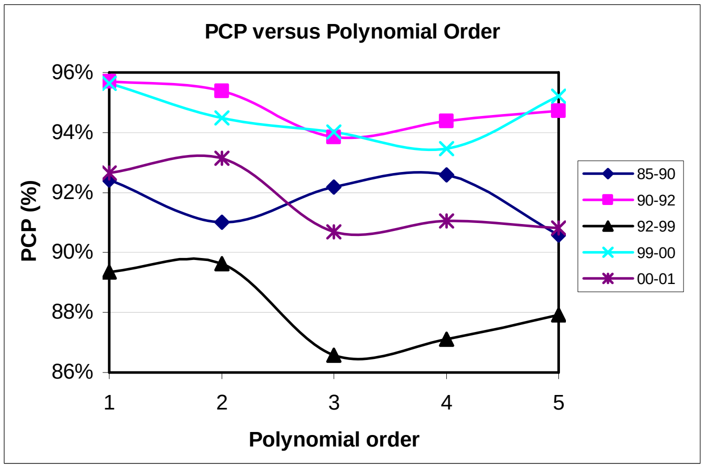


Figure 4.6: Performance of polynomial kernel as a function of polynomial order

The parameter of the RBF kernel, $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma|\mathbf{x}_i - \mathbf{x}_j|^2}$, is the gamma value γ. Figure 4.7 shows the impact of the gamma value on the SVMs performance. The performance of RBF kernel changed greatly when the gamma value varied. There were obvious trends of improved accuracy as the gamma value increased. When the gamma values increased from 0.5 to 15, the overall PCP of the SVMs model for the three data sets changed from around 72% to almost 99%.

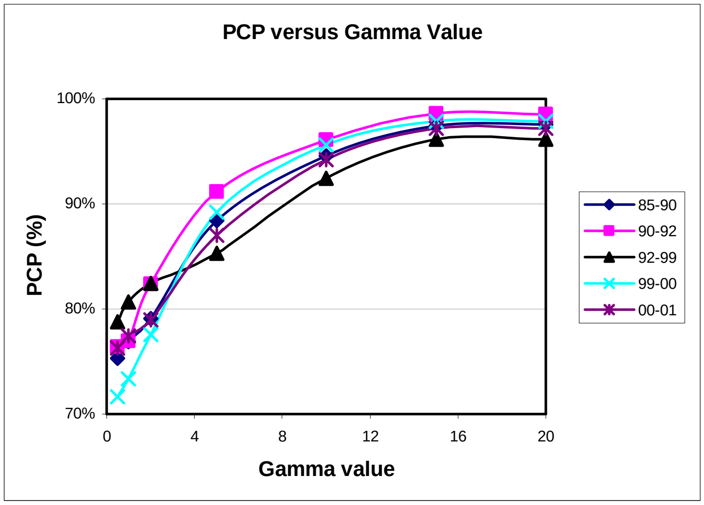


Figure 4.7: Performance of RBF kernel as a function of gamma value

Figure 4.8 shows a comparison of the performances of dot kernel, polynomial kernel, and RBF kernel with optimal parameter settings. It can be seen that the performances of each kernel function were quite different. The RBF kernel with gamma = 15 could achieve PCPs higher than 96% for all land use change training data between 1985-1990, 1990-1992, 1992-1999, 1999-2000 and 2000-2001. Thus, it can be treated as the optimal kernel setting for the land use change modeling for Calgary during period 1985-2001.

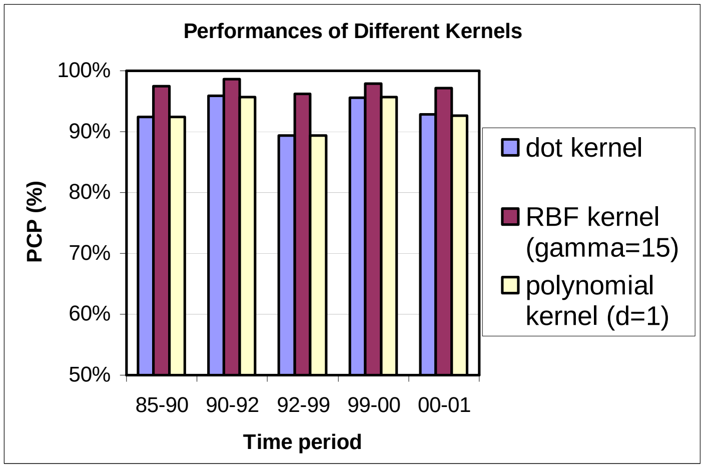


Figure 4.8: Comparison of the performances of different kernel functions

4.7 VECTOR NORMALIZATION

Normalization of the input data is a very common preprocessing technique widely used by different kinds of linear programming methods to improve the numerical stability. Previous studies (Herbrich and Graepel, 2001; Arnulf et al., 2003) have shown that normalization is a preprocessing type which plays an important role in SVMs for some applications. This study tried to figure out whether normalization could improve the performance of SVMs for land use change modeling.

The most frequently used normalization methods are: normalization to make features equally important; and normalization to bring feature vectors onto the same scale. The former regularizes the input vector by mean and variance:

$$\mathbf{x}_{norm} = \frac{\mathbf{x} - mean(\mathbf{X})}{\sqrt{\mathrm{var}(\mathbf{X})}} \tag{4.9}$$

The latter regularizes the input vector by normalizing the length of the vector according to some norm:

$$\mathbf{x}_{norm} = \frac{\mathbf{x}}{\|\mathbf{x}\|} \tag{4.10}$$

SVMs are designed to find the optimal separating hyperplane in the feature space which is obtained by a nonlinear mapping from the input space. If we do the normalization in the input space, in most cases we will lose the normalization in the feature space considering the nonlinearity of such a mapping (Vapnik, 1995).

As suggested by Herbrich and Graepel (2001), normalization in the feature space is a possible solution. Unlike in linear programming methods where normalization acts on rows or columns of the design matrix, normalization in SVMs can be applied to kernel functions so as to simultaneously rescale rows and columns to obtain a matrix with all diagonal entries set to one. Therefore, any kernel function can be revised to its normalized form:

$$\tilde{K}(\mathbf{x}_i,\mathbf{x}_j)=\frac{K(\mathbf{x}_i,\mathbf{x}_j)}{\sqrt{K(\mathbf{x}_i,\mathbf{x}_i)}\sqrt{K(\mathbf{x}_j,\mathbf{x}_j)}} \tag{4.11}$$

It is easy to see that $\tilde{K}(\mathbf{x}_i,\mathbf{x}_i)=1$. That means all vectors in the feature space lie on a unit hypersphere, showing that the length of the vector in the feature space is normalized. Clearly, normalized kernels satisfy Mercer's condition. They are still valid kernels for SVMs. In addition, the normalization of kernels is a conformal transformation of the original kernels. Thus, the angles between vectors of the feature space are invariant with respect to normalization of the kernel functions.

Figure 4.9 shows the comparison of the performances of SVMs with and without normalization for RBF kernel with gamma = 15, which was proved to be the optimal kernel for land use change modeling in Calgary. For comparison, we found that normalization cannot help to improve the performance of SVMs when applying it to model the land use change in Calgary from 1985 to 2001.

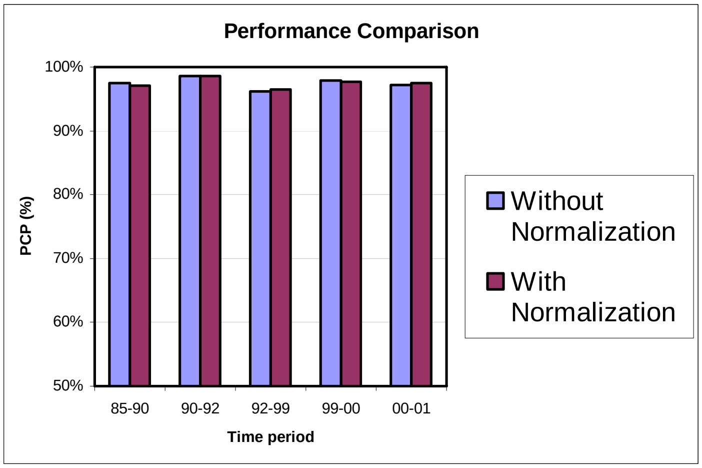


Figure 4.9: Comparison of SVMs performance with/without normalization

4.8 PERFORMANCE COMPARISON AND EVALUATION

The experiment results (Figure 4.8) clearly showed the high performance of SVMs on modeling land use change in Calgary. However, before drawing a conclusion on the suitability of a new method for a specific application, it is important to compare its performance with other widely accepted methods in the same application.

Spatial logistic regression, a statistical method widely used in land use change modeling for many years, was selected to be compared with SVMs. There were two main reasons for selecting spatial logistic regression. The first was that SLR has received extensive study and shown great success in land use change modeling (Wu and Yeh, 1997; Xie et al., 2005; Xie et al., 2006). The second reason is due to SLR's close relation to SVM. Both methods aim to solve binary classification problems and use similar loss functions. The major difference is that SLR just aims to minimize the empirical risk and SVMs aim to minimize the upper boundary of expected risk which includes the empirical risk and the model complexity.

The performances of SVMs with optimal configuration (regularization parameter $C = 10$; RBF kernel with gamma = 15) were compared with those of SLR. Figure 4.10 shows the comparison of overall prediction accuracy. The comparison clearly shows SVMs were superior to SLR regarding overall prediction accuracy.

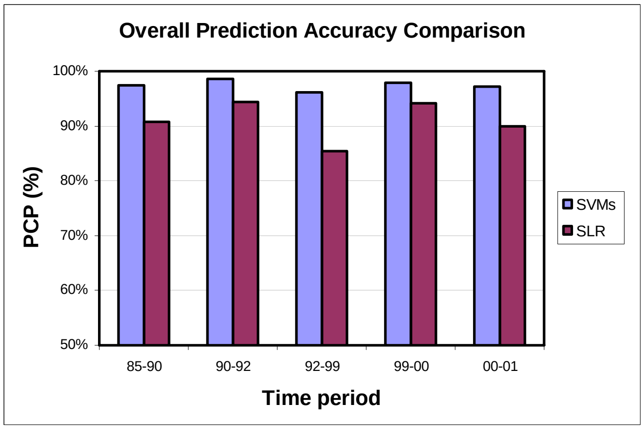


Figure 4.10: Performance comparison for SVMs and SLR (PCP)

Another important indicator, standard deviation, can show the stability of performance and was used to demonstrate the superiority of SVMs in this study. In this research, the standard deviation of PCP is calculated from the results of ten sets of non-overlapped testing data. Figure 4.11 shows that SVMs achieved a more stable performance than SLR.

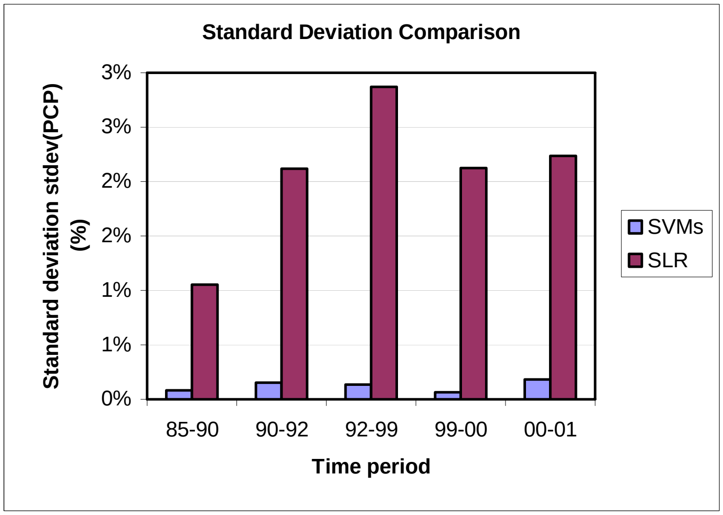


Figure 4.11: Performance comparison for SVMs and SLR (standard deviation)

Besides overall PCP, land use change modeling also attaches importance to the model's capacity and efficiency to predict the land use change. The former is measured by $PCP1$. The higher the $PCP1$, the more the changed land parcels can be correctly predicted. The latter is measured by $PCP2$. A higher $PCP2$ means the model can predict the changed land parcels with less incorrect predictions and thus has higher efficiency. Figure 4.12 and Figure 4.13 show that SVMs are better than SLR in both change prediction capacity and efficiency. To summarize, SVMs is superior to SLR when they are used to model the land use change in Calgary from 1985 to 2001.

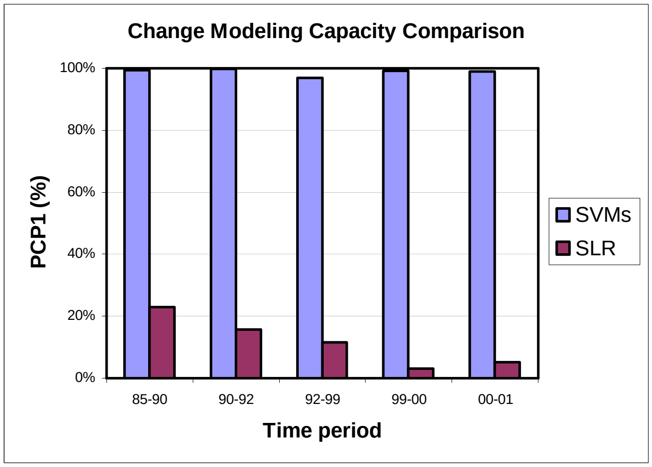


Figure 4.12: Performance comparison for SVMs and SLR (PCP1)

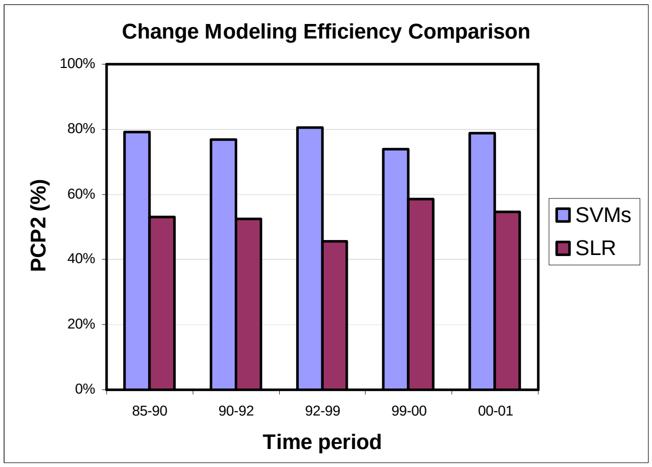


Figure 4.13: Performance comparison for SVMs and SLR (PCP2)

4.9 CHAPTER SUMMARY

In this chapter, SVMs were used to model the land use change in Calgary from 1985 to 2001. The study area was briefly introduced. This was followed by a description of both data collection and processing procedures. Then, the model development was discussed. The discussion included: the selection of causal factors which determined the main components of the land use change model; and the development of SVMs modeling framework, which was used to generate an optimal model. The performance evaluation system was also presented in this chapter.

After the development of the modeling framework, cross-validation tests were performed to find the optimal setting for the SVMs algorithm, which included selection of regularization parameter, selection of kernel function and its parameter, and the validity of normalization. Then the performances of SVMs with optimal configuration were compared with those of SLR from different points of view: overall prediction accuracy, standard deviation, change prediction capacity, and change prediction efficiency. The comparison showed that SVMs were superior to SLRs in all aspects.

CHAPTER 5: IMPROVEMENTS OF STANDARD SVMS

5.1 INTRODUCTION

In chapter 4, a SVMs modeling framework was developed to model the land use change in Calgary from 1985 to 2001. Implementation issues, e.g. data processing, framework development, and optimal algorithm configuration were discussed and solved. A SVMs modeling framework with RBF kernel (gamma = 15) was found to achieve a high performance for all land use change training data.

In this chapter, we attempt to tailor SVMs to better fit the characteristics and requirements of land use change modeling by improving the standard SVMs, which includes improvement for unbalanced datasets and improvement for robustness. These improvements can effectively address two important issues in land use change modeling, namely, the unbalance of change/unchanged land parcels and the robustness of the model, which were frequently overlooked in previous studies.

5.2 MOTIVATION

The performance comparison in section 4.8 not only revealed that SVMs were superior to SLR but also gave us some other information. One obvious finding was that: unlike SVMs that give similar performances on PCP , $PCP1$ and $PCP2$, SLR gave quite different performances. The PCP of SLR was comparable to that of SVMs and was relatively high. However, the $PCP1$ of SLR was rather low. After careful checking of the training datasets, we credited the unbalanced performance of SLR to the imbalance of positive/negative data in the training set.

Land use change is a long-drawn-out process. Generally, land use changes gradually as a whole, except in some special situations (e.g. war, plague, etc.) However, in order to reduce time-varying impacts and to accurately grasp the land use change pattern, most land use change studies adopt a relatively short time period. During a short time period to be modeled, only a very small amount of land parcels experience land use changes. It leads to a rare positive data (changed land parcels) situation in the binary land use change classification. Table 5.1 shows the details of candidate land parcels in each modeling time period. The unbalance of positive/negative data is quite apparent.

Table 5.1: Details of candidate land parcels

Time Period	1985-1990	1990-1992	1992-1999	1999-2000	2000-2001
Changed cells	50,208	32,080	70,181	25,899	45,343
Unchanged cells	492,187	477,795	434,488	423,666	401,523
Total cells	542,395	509,875	504,669	449,565	446,866
Positive rate	9.26%	6.29%	13.91%	5.76%	10.15%

Since the amount of unchanged cells (negative data) is much larger than that of changed cells (positive data), the objective function of the optimization problem will initially be dominated by the negative data. The optimization process will sacrifice the performance of positive data (change modeling capacity) in order to minimize the overall loss. This can explain why the PCP1 is much lower than PCP for SLR. Apparently, such cases are not only applied to SLR, but also impact SVMs. Since rare positive data is one inherent nature of land use change modeling, the impacts of unbalanced data on the performance of SVMs and the way to eliminate such impacts require careful study.

Another inherent nature of land use change modeling is its complexity. Land use change is caused by a gamut of factors, ranging from spatial parameters to socioeconomic, political or even cultural factors. No single land use model can include all these factors. Hence, not all land use changes in the observed data are the result of the combination of the observed causal factors. Some land parcels that are marked as low probability for change considering the causal factors in the land use change model, might be affected by previously unconsidered forces. This in turn will impact their land use. These unconsidered forces, although might overwhelm the major factors of the model in some cases, is not worthy to be considered in the model since they are not significant for the whole population or are too expensive to collect and quantify corresponding variables. Therefore, these samples should be treated as outliers. Outliers are quite common to land use change data. Special effort should be made to take care of these outliers and to give a reliable performance even when certain levels of disturbance exist within the sample data.

5.3 IMPROVEMENT FOR UNBALANCE DATASET

In section 5.2, we discussed the unbalanced performance of SLR due to the imbalance of positive/negative data in the training set. Although the SVMs with optimal configuration demonstrated quite uniform performances on PCP, PCP1, and PCP2, it is worthy to investigate the impacts of unbalanced data on the performance of SVMs.

The literature study showed that, although SVMs are known to perform well regarding misclassification error, they also have been recognized to provide skewed decision boundaries for unbalanced classification losses (Grandvalet et al., 2005). Recall the objective function of SVMs optimization problem:

$$F(\mathbf{w},\boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}'\cdot\mathbf{w} + C\sum_{i=1}^{m}\xi_i \qquad (5.1)$$

The upper bound of the expected risk of the model consists of two parts: the first term is the VC confidence interval determined by the complexity of the model; and the second term is the empirical risk caused by the misclassification error. When the training set can be linearly separated or approximately be linearly separated in the feature space, the second term equals or is close to zero. The model complexity dominates the optimization problem. The unbalanced dataset has no impact or just a slight impact on the performance. This explains why the SVMs with RBF kernel (gamma = 15) can achieve uniform performances for classifying the unbalanced land use change data in Calgary from 1985 to 2001.

However, when the training data is non-separable or the kernel used is not effective in mapping the input space to a separable feature space, there exists a mixed range with positive points (changed land parcels) and negative points (unchanged land parcels). Thus misclassification is significant and might account for an major part in the objective function. Similar to the case of SLR, if the amount of negative data is much larger than that of positive data, the objective function of the optimization problem will be dominated by negative data. The optimization process will sacrifice the performance of positive data in order to minimize the overall loss.

Huang et al. (2002) suggested that replicating the samples of the smaller class such that the two classes have approximately the same size can avoid the performance degradation for an unbalanced dataset. However, the replication of the samples will result in an increase of the size of the training set and thus increase the computational burden. Therefore, an elegant approach is preferred to handle this case. In this research, an optimal separating hyperplane using different losses for positive and negative examples (Morik et al., 1999) was adopted to address unbalanced training data. Figure 5.1 visually explains the main idea of this approach.

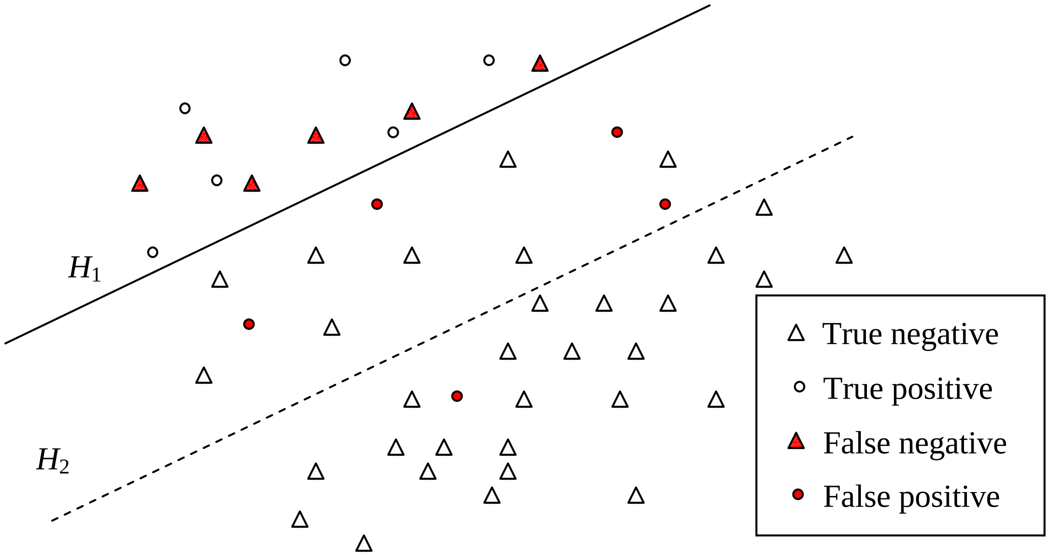


Figure 5.1: Optimal separating hyperplane for unbalanced training set

In Figure 5.1, hyperplane H_1 is the optimal separating hyperplane obtained from the standard SVMs. Using such a classifier, the number of misclassified positive data (false positive) and that of misclassified negative data (false negative) are comparable. However, since the data is unbalanced, the misclassified positive data accounts for a large percentage of the positive data and thus leads to a low performance for classifying positive data. By changing the objective function and giving higher weight for the losses caused by positive data, we can make the same amount of positive data play a more important role than the negative data and thus push the optimal separating hyperplane move from H_1 to a new position H_2, which is closer to the negative side. Therefore, more samples are classified to be positive. From Figure 5.1, we find that the false negatives increase and the false positives decrease. Although the total number of misclassifications increase, the increase in the false negative only accounts for a very small part of the large negative data and thus causes the accuracy of classifying negative data to decrease slightly. Contrarily, for the positive data, the decrease of the false positive data will greatly improve the accuracy of classifying positive data. Therefore, a higher weight for the losses caused by positive data will lead to a slight decrease of the overall classification accuracy but will greatly improve the change modeling capacity. This is worthwhile for land use change modeling since revealing the pattern of change is very important in land use study.

Considering different weights for positive loss and negative loss, the optimization problem of SVMs is changed to:

$$\begin{aligned} &\mathit{minimize}: F(\mathbf{w},\boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}'\cdot\mathbf{w} + C_+\sum_{i=1}^{m_1}\xi_i^+ + C_-\sum_{i=1}^{m_2}\xi_i^- \\ &\mathit{subject\ to}: \mathbf{w}'\cdot\mathbf{x}_i + b \geq 1-\xi_i^+, \quad i=1,2,\ldots,m_1 \\ &\qquad\qquad \mathbf{w}'\cdot\mathbf{x}_i + b \leq -1+\xi_i^-, \quad i=1,2,\ldots,m_2 \\ &\qquad\qquad \xi_i^+ \geq 0, \quad i=1,2,\ldots,m_1 \\ &\qquad\qquad \xi_i^- \geq 0, \quad i=1,2,\ldots,m_2 \end{aligned} \tag{5.2}$$

And its dual form is:

$$
\begin{aligned}
&maximize: L(\boldsymbol{\alpha}) = \sum_{i=1}^{m_1}\alpha_i - \frac{1}{2}\sum_{i=1}^{m_1}\sum_{j=1}^{m_1}\alpha_i^+\alpha_j^+ y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^{m_2}\alpha_i - \frac{1}{2}\sum_{i=1}^{m_2}\sum_{j=1}^{m_2}\alpha_i^-\alpha_j^- y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\
&subject\ to: \sum_{i=1}^{m_1}\alpha_i^+ y_i + \sum_{i=1}^{m_2}\alpha_i^- y_i = 0 \\
&\qquad\qquad 0 \le \alpha_i^+ \le C_+, \qquad i = 1, 2, ..., m_1 \\
&\qquad\qquad 0 \le \alpha_i^- \le C_-, \qquad i = 1, 2, ..., m_2
\end{aligned}
\tag{5.3}
$$

The new problem can be solved using a technique similar to the standard SVMs. Another issue for the improved SVMs concerning unbalanced data is the selection of the trade-off between the loss of positive data and negative data. Based on the Bayes' decision theory, Lin et al. (2002) suggested that the ratio of the coefficients C_+ and C_- should be equal to the ratio of both the false positive and negative losses. An approximate setting is to let the ratio of the coefficients C_+ and C_- equal that of the ratio of the number of negative and positive samples.

Figures 5.2—5.4 show the performance comparisons of the standard SVMs and the unbalanced SVMs, both of which used dot kernel. From these figures, it is easy to tell that the overall PCPs of the unbalanced SVMs are slightly lower than those of the standard SVMs. However, the change modeling capacities (PCP1) of the unbalanced SVMs are greatly improved. The change modeling efficiencies (PCP2) might increase or decrease slightly, depending on the specific dataset. For the RBF kernel (gamma = 15), since the kernel function can effectively map the input space to a feature space in which the training set can be linearly separated with few misclassifications. The dominant component in the objective function is the model complexity. Both the losses caused by the false positive and negative classification are insignificant. Hence, the performance of the standard SVMs and the unbalanced SVMs are very close. Both of them can achieve uniform performance on overall classification accuracy, change modeling capacity, and efficiency.

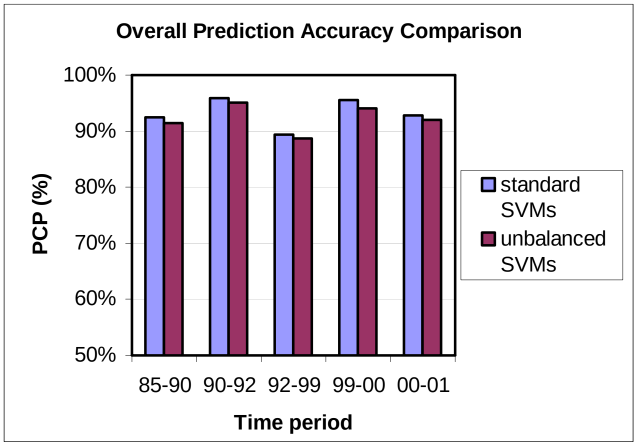


Figure 5.2: Performance comparison for standard SVMs and unbalanced SVMs (PCP)

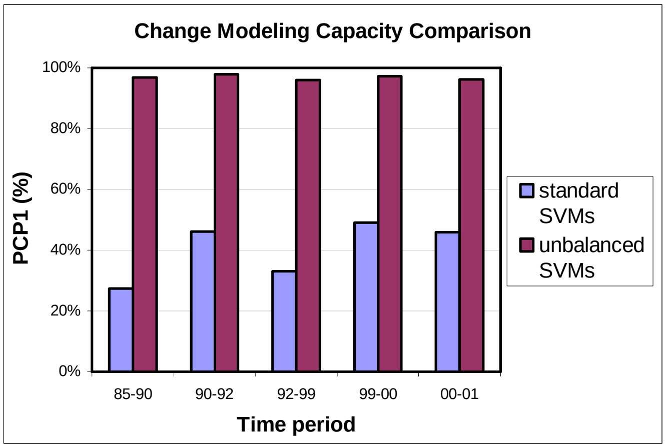


Figure 5.3: Performance comparison for standard SVMs and unbalanced SVMs (PCP1)

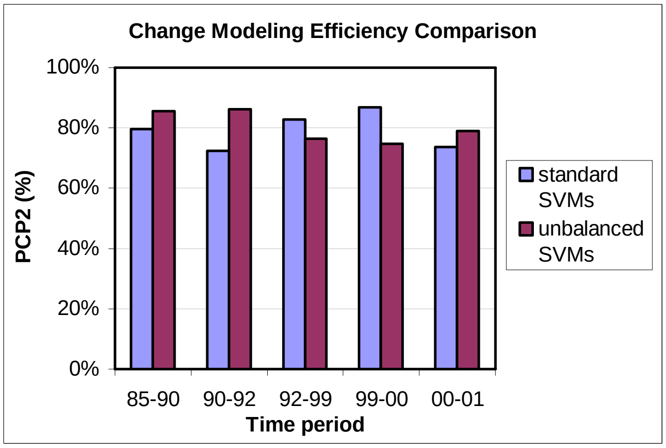


Figure 5.4: Performance comparison for standard SVMs and unbalanced SVMs (PCP2)

5.4 IMPROVEMENT FOR ROBUSTNESS

As discussed in section 5.2, land use change data is apt to be contaminated with noise. Hence a good method for land use change modeling should be robust. That is, small deviations in the data do not cause dramatic performance degeneration.

SVMs are well-known in regards to robustness (Malossini et al., 2004). Strictly speaking, the robustness of an algorithm should be estimated by its Influence Function (IF) (Huber, 1981). However, the non-differentiable characteristic of SVMs loss function makes it hard to evaluate its robustness using normal approach. It would be meaningful to analyze its loss function against the misclassification error and some behavior to get an indication of the SVMs' robustness. From SVMs' objective function (3.36), it is easy to know the loss corresponding to the misclassification error:

$$L(y_i(\mathbf{w}'\cdot\mathbf{x}_i+b))=\max\{0,1-y_i(\mathbf{w}'\cdot\mathbf{x}_i+b)\} \tag{5.4}$$

It can be shown from (5.4) that SVMs loss is linear. Therefore, it is less sensitive to extreme data points compared with squared loss which is widely used in various statistical methods including the famous least squares method. As well, the solution of SVMs is only dependent on the support vectors and the influences of support vectors are bounded. Therefore, SVMs are insensitive to small noise and thus are robust in noisy, complex domains.

In order to test the robustness of SVMs, a controlled experiment was conducted under different noise settings. We randomly performed the flipping of the original training set for percentages of 1%, 2%, 5% and 10% to simulate different noise levels. Figure 5.5 shows the performance of the standard SVMs under different noise levels. Figure 5.6 shows the performance of SLR under different noise levels, which was used as a reference to show the robustness of the standard SVMs.

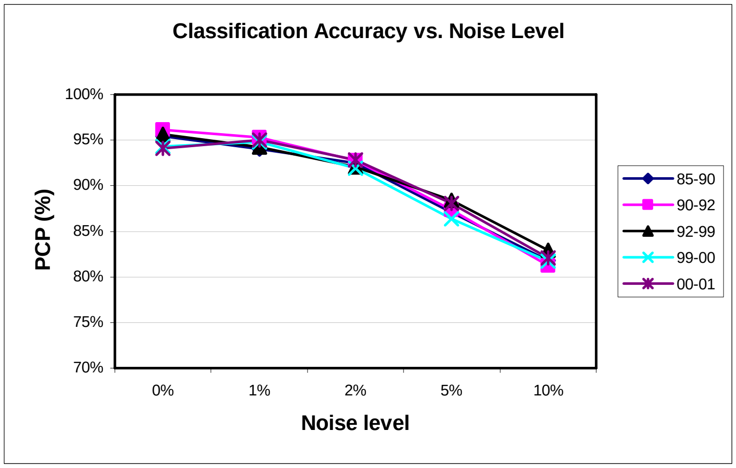


Figure 5.5: Performance of the standard SVMs under different noise levels

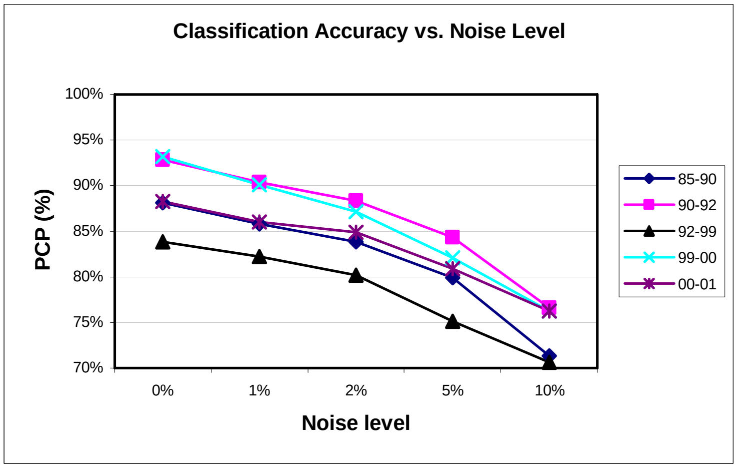


Figure 5.6: Performance of SLR under different noise levels

Figure 5.5 and Figure 5.6 show that the degradation of performance is less for the standard SVMs than that of SLR. Moreover, previous studies (Zhang and Yang, 2003) indicated that the loss function of SLR is close to linear and thus, SLR is not very sensitive to outliers. Therefore, the robustness of SVMs was confirmed.

However, we also noticed that the degradation of performance for SVMs increased when the noise level was greater than 2%. This might be due to the way we manipulated the data. The flipping of negative data would overwhelm the amount of remaining positive data when the data was unbalanced. The noise’s impact on the positive and negative data was quite different. The smaller class suffered more from a certain level of noise. It was very hard to identify the pattern correctly for the smaller class, which led to the increasing degradation of performance. In order to solve this problem, a specific method needs to be introduced to detect the outliers and remove them from the support vectors, which were used to determine the boundary between positive and negative data. In this study, the Robust Support Vector Machines (RSVMs) earlier developed by Song et al. (2002) was introduced to tackle the outlier problems and improve the robustness of SVMs for unbalanced data.

The basic idea of RSVMs was to use an adaptive separation margin and to minimize only the margin of the weights $\mathbf{w}$ instead of minimizing the sum of the margin and misclassification error in the standard SVM training. A new slack variable $\lambda D^2(\mathbf{x}_i, \mathbf{x}^*_{y_i})$ was introduced in place of ξ_i in the standard SVM training. Then the optimization problem became:

$$\begin{aligned} &\textit{minimize}: F(\mathbf{w}, \xi) = \frac{1}{2}\mathbf{w}' \cdot \mathbf{w} \\ &\textit{subject to}: y_i(\mathbf{w}' \cdot \mathbf{x}_i + b) \geq 1 - \lambda D^2(\mathbf{x}_i, \mathbf{x}^*_{y_i}), \qquad i = 1, 2, \ldots, m \end{aligned} \tag{5.5}$$

where $\lambda \geq 0$ was a user-defined regularization parameter measuring the influence of averaged distance to the class center, and $D^2(\mathbf{x}_i, \mathbf{x}^*_{y_i})$ was the normalized distance from each sample $\mathbf{x}_i$ to the center of the respective class $(\mathbf{x}^*_{y_i}, y_i \in \{-1, +1\})$ in the feature space, which was calculated by:

$$\begin{aligned} D^2(\mathbf{x}_i, \mathbf{x}^*_{y_i}) &= |\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}^*_{y_i})|^2 / D^2_{\max} \\ &= \left[\Phi(\mathbf{x}_i)' \cdot \Phi(\mathbf{x}_i) - 2\Phi(\mathbf{x}_i)' \cdot \Phi(\mathbf{x}^*_{y_i}) + \Phi(\mathbf{x}^*_{y_i})' \cdot \Phi(\mathbf{x}^*_{y_i})\right] / D^2_{\max} \\ &= \left[K(\mathbf{x}_i, \mathbf{x}_i) - 2K(\mathbf{x}_i, \mathbf{x}^*_{y_i}) + K(\mathbf{x}^*_{y_i}, \mathbf{x}^*_{y_i})\right] / D^2_{\max} \end{aligned} \tag{5.6}$$

where $\Phi(\mathbf{x}_i)$ was the transformation function mapping the input space to the feature space, $K(\mathbf{x}_i,\mathbf{x}_j)=\Phi(\mathbf{x}_i)'\cdot\Phi(\mathbf{x}_j)$ was the kernel function, $D^2_{\max}$ was the maximum distance between the center $\mathbf{x}^*_{y_i}$ and training data points of the respective class in the kernel space.

Using Lagrange multipliers $\boldsymbol{\alpha}$, the dual problem of (5.6) can be obtained:

$$\begin{aligned}&maximize: L(\boldsymbol{\alpha})=\sum_{i=1}^{m}\alpha_i(1-\lambda D^2(\mathbf{x}_i,\mathbf{x}^*_{y_i}))-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j)\\&subject\ to: \sum_{i=1}^{m}\alpha_i y_i=0\\&\qquad\qquad \alpha_i\ge 0,\qquad i=1,2,...,m\end{aligned}\tag{5.7}$$

Comparing with the dual problem in the standard SVM, the only difference lies in the additional part $-\lambda D^2(\mathbf{x}_i,\mathbf{x}^*_{y_i})$ in the maximization function. Therefore, the optimization problem (5.7) can be solved using the program of the standard SVMs with small changes.

The mechanism of RSVMs is to detect the outliers and to remove them from the support vectors. The support vectors are those particular samples with $\alpha_i>0$. Based on the KKT complementarity condition, these samples satisfy the equalities in the constraint of (5.5):

$$y_i(\mathbf{w}'\cdot\mathbf{x}_i+b)=1-\lambda D^2(\mathbf{x}_i,\mathbf{x}^*_{y_i})\tag{5.8}$$

For each sample, the separation margin can be thought of as an adaptive value $1-\lambda D^2(\mathbf{x}_i,\mathbf{x}^*_{y_i})$. Suppose a sample is an outlier that is located on the wrong side and far away from the separable hyperplane. The distance between this sample and the center of the class is longer than that of the other normal sample in the same class. The augmented term $\lambda D^2(\mathbf{x}_i,\mathbf{x}^*_{y_i})$ is relatively large. Therefore, the inequality in (5.5) is satisfied and the coefficients associated with the sample should be toward zero. Therefore, this outlier may not become a support vector.

Figure 5.7 shows the performance of the RSVMs under different noise levels. Compared to Figure 5.5, the performance degradation with the noise level is much slower. Therefore, the RSVMs' improvement in the robustness is quite obvious.

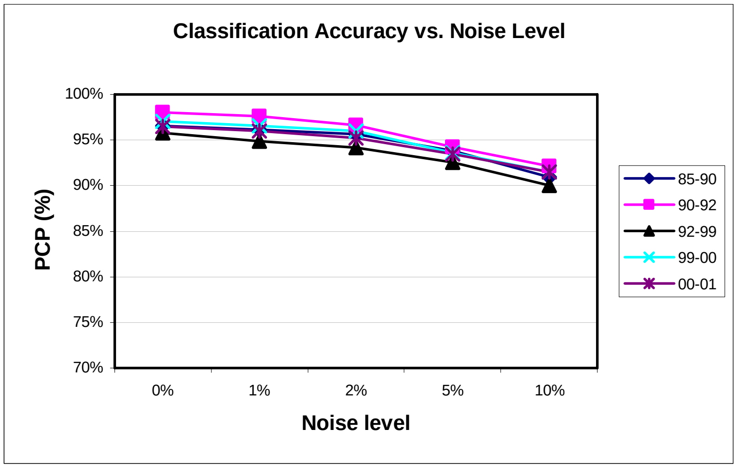


Figure 5.7: Performance of the RSVMs under different noise levels

5.5 CHAPTER SUMMARY

In this chapter, two improvements towards the standard SVMs were discussed. The motivation for making necessary improvements to standard SVMs was provided. This was followed by a detailed discussion of the two improvements: improvement to deal with an unbalanced dataset and improvement for robustness.

For improvement in dealing with unbalanced datasets, the impacts of unbalanced datasets were analyzed. Then, methods to eliminate these impacts were provided. The implementation of an efficient approach was discussed in detail. After that, experimental results showing the significant performance improvement validated the unbalanced SVMs on tackling unbalanced datasets.

In regards to the robustness improvement, theoretic analysis and experimental tests on the robustness of the standard SVMs were performed. Then the limitation of its robustness on the land use change dataset was given after analyzing the experiment results. An efficient approach was then introduced to remove the outliers and thus improve robustness. Finally, this was followed by experimental results which validated the improvement.

CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS

6.1 SUMMARY AND CONCLUSIONS

In an effort to better address some important issues in land use study, this research aimed to develop a novel method for land use change modeling using support vector machines. Modeling land use change is a prerequisite to understanding the complexity of land use change, forecasting future trends of land use change, and evaluating the ecological impacts of land use change. This research will benefit urban planners and policy makers in their efforts to effectively and efficiently understand the land use change process from the unbalanced and noisy historic data, to make more precise projections for future land use, and thus ensure the generation of scientific plans which will foster sustainable development.

A review of previous land use change modeling studies was done in this research. The causal factors driving land use change reported in literature was discussed and summarized. Then a variety of techniques widely used in land use change modeling were introduced. Advantages and limitations of a variety of techniques were also given. It was found that some important issues which have great impact on the accuracy and reliability of land use change modeling, still need to be further addressed in order to effectively and efficiently model land use change. Two of these problems were: how to achieve uniform high performance for the unbalanced land use change dataset and how to retain high performance when different levels of noise appear in the training data.

This research presented a detailed discussion on a novel method, namely, SVMs, which have the potential to effectively address the research problem. SVMs are an elegant classification algorithm based on statistical learning theory. Minimizing the upper bound of structural risk instead of minimizing just the empirical risk endows SVMs with a good generalization performance without assuming the underlying distribution of data to be classified. Using kernel tricks, SVMs are able to handle nonlinear situations in an easy and computationally efficient way. Moreover, the optimization problem in SVMs can be transformed to a quadratic programming problem, which can be solved using a highly efficient algorithm that can ensure global minima. Furthermore, SVMs can condense information in the training data and use just a very small number of data points to determine the model. All these attractive features make SVMs a promising approach for land use change modeling.

To investigate the performance of SVMs on land use change modeling, a SVMs land use change modeling framework was implemented and applied to a case study of modeling land use change in Calgary from 1985 to 2001. Data regarding Calgary land use change modeling was collected and processed. Raster layers including land use data and different causal factors were compiled using ESRI ArcMap. A SVMs land use change modeling framework, which consisted of a land use change detection module, a causal factor assembling module, a spatial sampling module, a SVMs classification module and a performance evaluation module, were developed and integrated in ArcMap to perform land use change modeling. Three implementation issues for SVMs, namely, regularization parameter selection, kernel function selection, and vector normalization, were carefully addressed. The performance of SVMs was compared with that of a well-studied land use modeling approach, namely, spatial logistic regression. The comparison showed that SVMs were superior to SLR.

Two improvements of standard SVMs were developed to tailor SVMs to better fit the characteristics and requirements of land use change modeling. The first improvement aimed to improve the accuracy of classifying smaller class when the training set was unbalanced. By changing the objective function and giving different weights for positive and negative data, the improvement proved to be effective in providing uniform high performance for unbalanced land use change data. The other improvement aimed to improve the robustness of SVMs especially in the case of unbalanced data. A robust SVMs algorithm that detected outliers and removed them from the support vectors was introduced and tested. The result showed that the robust SVMs could efficiently improve robustness.

The research has led to the following findings:

1. There is a need for a novel land use modeling approach to better address some important issues specific to land use data, namely, a mixture of continuous and categorical causal factors, non-normal distribution of causal factors, imbalance of the training dataset, and the existence of outliers in the training dataset. Moreover, a good land use change modeling approach should promise a high generalization performance regardless of the underlying distribution of data.
2. SVMs can be applied to land use change modeling applications and have demonstrated high overall concordance, stable performance, high land use change modeling capacity and high land use change modeling efficiency when the optimal model configuration was employed. For land use change modeling in Calgary, the optimal SVMs settings are: regularization parameter $C = 10$ and RBF kernel with gamma = 15.
3. A land use change modeling framework developed with C++ and closely related to ArcMap cannot only achieve high computational efficiency but also make use of ArcMap's powerful spatial data processing and visualization capacity. It is a promising framework for spatial analysis.
4. Standard SVMs may suffer degradation on land use change modeling capacity when the optimal kernel is not adopted and the training data is unbalanced. By assigning different weights to the positive and negative classes, SVMs can be improved to effectively handle unbalanced datasets. This improvement enabled SVMs to provide uniform high performances for separating both classes.
5. Standard SVMs demonstrate certain level of robustness when the noise level is small but the robustness will degrade when the noise level in unbalanced data exceeds a certain threshold. By introducing a new slack variable describing the distance from a sample to its class center, RSVMs can effectively detect the outliers, remove them from support vectors, and thus achieve a robust performance.

6. The improved SVMs can greatly improve the accuracy and reliability of land use change modeling especially when the underlying data distribution is unknown and the dataset is significantly unbalanced.

We also recognize that the above mentioned conclusions are drawn from the results of a case study using Calgary’s land use change data from 1985 to 2001. Limited by the data availability, we didn’t consider some factors that might be important for land use change, e.g. social factors, economic factors, etc. The land use model we built in this study might be biased due to the incompleteness of the causal factors. Therefore, experiments with more complete causal factors and case studies in other cities/regions are needed in order to draw a more general conclusion.

6.2 RECOMMENDATIONS FOR FUTURE WORK

While SVMs have been applied for land use modeling in this study, there are still a number of unanswered questions regarding the application of SVMs for land use change modeling:

Although the standard SVMs are designed to solve binary classification problems, they can also be extended to multi-class SVMs which are capable of classifying multiple class data. Future work on extending SVMs to modeling multinomial land use change is recommended.

Future work should also include introducing new elements in SVMs modeling framework to model the temporal complexity of land use change. Alternative solutions may include exploring an effective way to combine several bi-temporal SVMs models along the time axis to form a smoothed model or employing a time-varying model. Such a model with a temporal complexity modeling capacity could achieve higher accuracy when projecting future land use.

A challenging task deserving future study is the incorporation of spatial and temporal correlations in the SVMs model. Building a complex autogressive structure to model the autocorrelation could provide more relevant information in the model and thus prove superior to using spatial sampling to remove the autocorrelation.

BIBLIOGRAPHY

Adams, J. B., Sabol, D. E., Kapos, V., Almeida, F. R., Robers, D. A., Smith, M. O., and Gillespie, A. R., 1995. Classification of multi-spectral images based on fractions of end members: applications to land-cover change in the Brazilian Amazon. *Remote Sensing of Environment*, 12, 137-154.

Agarwal, C., Green, G. L., Grove, J. M., Evans, T., and Schweik, C., 2000. A review and assessment of land use change models: dynamics of space, time, and human choice. In *4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4): Problems, Prospects and Research Needs*, Banff, Canada, September 2-8.

Amari, S. and Wu, S., 1999. Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, 12, 783-789.

Arnulf, B., Graf., A., Smola, A. J., and Borer, S., 2003. Classification in a normalized feature space using support vector machines. *IEEE Transactions on Neural Networks*, 14(3), 597-605.

Baker, W. L., 1989. A review of models in landscape change. *Landscape Ecology*, 2(2), 111-133.

Balling, R., Powell, B., and Saito, M., 2004. Generating future land-use and transportation plans for high-growth cities using a genetic algorithm. *Computer Aided Civil and Infrastructure Engineering*, 19, 213-222.

Batty, M., and Xie, Y., 1994. From cells to cities. *Environment and Planning B: Planning and Design*, 21, 31-48.

Berling-Wolff, S, and Wu, J., 2004. Modeling urban landscape dynamics: a review. *Ecological Research*, 19, 119-129.

Bruntland, G. (eds.), 1987. *Our common future: the world commission on environment and development*, Oxford University Press, Oxford.

Cheng, J., 2003. Modeling spatial and temporal urban growth. Doctoral Dissertation, Faculty of Geographical Sciences, Utrecht University, Netherlands.

Cheng, J., and Masser, I., 2003. Urban growth modeling: a case study of Wuhan City, PR China. *Landscape and Urban Planning*, 62, 199-217.

Cherkassky, V., and Ma, Y., 2004. Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, 17, 113-126.

Chomitz, K. M., and Gray, D. A., 1996. Roads, land use, and deforestation: a spatial model applied to Belize. *The World Bank Economic Review*, 10(3), 487–512.

Clarke, K. C., and Gaydos, L. J., 1998. Loose-coupling a CA model and GIS: long-term urban growth prediction for San Francisco and Washington/Baltimore. *International Journal of Geographical Information Science*, 12(7), 699-714.

Coppin, P., Jonckheere, I., Nackaerts, K., and Muys, B., 2003. Digital change detection methods in ecosystem monitoring: a review. *International Journal of Remote Sensing*, 24, 1-33.

Corne, S., Murray, T., Openshaw, S., See, L., and Turton, I., 1999. Using computational intelligence techniques to model subglacial water systems. *Journal of Geographical System*, 1, 37-60.

Couclelis, H., 1997. From cellular automata to urban models: new principles for model development and implementation. *Environment and Planning B: Planning and Design*, 24, 165-174.

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

Frayman, Y., Rolfe, B. F., Webb, G. I., 2002. Solving regression problems using competitive ensemble models. In McKay, B, Slaney, J. (Eds.), *Advances in Artificial Intelligence*, Springer-Verlag, Berlin Heidelberg.

Geist, H. J., and Lambin, E. F., 2001. *What drives tropical deforestation*, LUCC International Project Office.

Gong, P., and Xu, B., 2003. Remote sensing of forests over time: change types, methods, and opportunities. In Woulder, M., Franklin, S. E. (eds.), *Remote Sensing of Forest Environments: Concepts and case studies*, Kluwer Press, Amsterdam.

Goodchild, M. F., Anselin, L. and Deichmann, U., 1993. A framework for the areal interpolation of socioeconomic data. *Environment and Planning A*, 25, 383-397.

Grandvalet, Y., Mariéthoz, J., and Bengio, S., 2005. A probabilistic interpretation of SVMs with an application to unbalanced classification. IDIAP Research Report 05-26.

Gunn, S. R, 1998. Support vector machines for classification and regression. ISIS Technical Report, Image Speech & Intelligent Systems Group, University of Southampton.

Guo, Q., Kelly, M., and Graham, C. H., 2005, Support vector machines for predicting distribution of sudden oak death in California. *Ecological Modeling*, 182, 75-90.

Hall, F. G., Botkin, D. B., Strebel, D. E., Woods, K. D., and Goetz, S. J., 1991. Large-scale patterns of forest succession as determined by remote sensing. *Ecology*, 72, 628-640.

Hastie, T., and Tibshirani, R., 1998. Classification by pairwise coupling. In Jordan, M.I., Kearns, M.J., Solla, A.S. (eds.), *Advances in Neural Information Processing Systems*, Vol. 10, MIT Press, Cambridge.

Henderson-Sellers, A., and Pitman, A.J., 1992. Land-surface schemes for future climate models specification, aggregation and heterogeneity. *Journal of Geophysical Research*, 97, 2678-2696.

Herbrich, R., and Graepel, T., 2001. A PAC-Bayesian margin bound for linear classifiers: why SVMs work. *Advances in Neural Information Processing Systems*, 13.

Hiroshi, S., Jun, R, and Mitsuru, N., 1998. Improving the generalization performance of the MCE/GPD learning. In ICSLP'98, Australia, Dec 1998.

Howarth, P. J., and Wickware, G. M., 1981. Procedures for change detection using Landsat. *International Journal of Remote Sensing*, 2, 227-291.

Huang, C., Davis, L. S., and Townsheng, J. R. G., 2002. An assessment of support vector machines for land cover classification. *International Journal of Remote Sensing*, 23(4), 725-749.

Huber, P. J., 1981. *Robust Statistics*, John Wiley and Sons, New York.

Jensen, J., 1996. *Introductory Digital Image Processing*, Prentice Hall, New Jersey.

Jobson, J. D., 1992. *Applied multivariate data analysis*, Springer, New York.

Joachims, T., 1999. Estimating the Generalization Performance of an SVM Efficiently. In *Proceedings of ICML-00, 17th International Conference on Machine Learning*, Morgan Kaufmann Publishers, San Francisco.

Karystinos, G. N., and Pados, D. A., 2000. On overfitting, generalization, and randomly expanded training sets. *IEEE Transactions on Neural Networks*, 11(5), 1050-1057.

Landis, J. H., and Zhang, M., 2000. Using GIS to improve urban activity and forecasting models: three examples. Chapter five in: Fotheringham, A. S., and M. Wegener (eds.), *Spatial Models and GIS – New Potential and New Models*, Taylor & Francis, London.

Lee, Y. J., and Mangasarian, O. L., 2000. RSVM: reduced support vector machines. Technical Report 00-07, Data Mining Institute, Computer Sciences Department, University of Wisconsin-Madison, Wisconsin.

Li, X., and Yeh, A. G., 2000. Modeling sustainable urban development by the integration of constrained cellular automata and GIS. *International Journal of Geographical Information Science,* 14(2), 131-153.

Li, X., and Yeh, A. G., 2002. Neural-network-based cellular automata for simulating multiple land use changes using GIS. *International Journal of Geographical Information Science*, 16(4), 323-343.

Lin, Y., Lee, Y., and Wahba, G., 2002. Support vector machines for classification in non-standard situations. *Machine Learning*, 46, 191-202.

Lopez, E., Bocco, G., Mendoza, M., and Duhau, E., 2001. Predicting land cover and land use change in the urban fringe: a case in Morelia City, Mexico. *Landscape and Urban Planning*, 55, 271-285.

Malossini, A., Blanzieri, E., and Ng, R. T., 2004. Assessment of SVM reliability for microarrays data Analysis. In *Proceedings of the Second European Workshop on Data Mining and Text Mining in Bioinformatics*, Pisa, Italy, September 2004.

Martin, D. and Bracken I., 1991. Techniques for modeling population-related raster databases. *Environment and Planning A,* 23, 1069-1075.

Mertens, B., Poccard-Chapuis, R., Piketty, M. G., Lacquies, A. E. and Venturieri, A., 2002. Crossing spatial analyses and livestock economics to understand deforestation processes in the Brazilian Amazon: The case of São Félix do Xingú in South Pará. *Agricultural Economics*, 27(3), 269–294.

Morik, K., Brockhausen, P., and Joachims, T., 1999. Combining statistical learning with a knowledge based approach - a case study in intensive care monitoring. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML-99),* Bled, Slovenia, June, 1999.

Olden, J. D., and Jackson, D. A., 2001. Fish–habitat relationships in lakes: gaining predictive and explanatory insight by using artificial neural networks. *Transactions of the American Fisheries Society*, 130, 878-897.

O'Sullivan, D., 2001. Graph-cellular automata: a generalized discrete urban and regional model. *Environment and Planning B: Planning and Design*, 28, 687-705.

Pajanowski, B. C., Brown, D. G., Shellito, B. A., and Manik, G. A., 2002. Using neural networks and GIS to forecast land use changes: a land transformation model. *Computers, Environment and Urban Systems*, 26, 553-575.

Pal, M., and Mather, P. M., 2003. Support vector classifiers for land cover classification. In *Proceedings of Map India 2003 Conference*, New Delhi, India, January, 2003.

Parker, D. C., Berger, T., and Manson, S. M., 2001. Agent-based models of land use and land cover change. Report No. 6, LUCC Report Series .

Platt, J., 1998. Sequential minimal optimization: a fast algorithm for training support vector machines. Microsoft Research Technical Report MSR-TR-98-14, Microsoft.

Platt, J., 1999. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In Smola, A.J., Bartlett, P., Schölkopf, B., Schuurmans, D. (eds.), *Advances in Large Margin Classifiers*, 61–74, MIT Press, Cambridge.

Rumelhart, D., Hinton, G., and Williams, R., 1986. Learning internal representations by error propagation. In Rumelhart, D. E., and McClelland, J. L. (eds.), *Parallel distributed processing: Explorations in the microstructures of cognition*, MIT Press, Cambridge.

SADER, S. A., 1988. Remote sensing investigations of forest biomass and change detection in tropical regions. In *Satellite imageries for forest inventory and monitoring: experiences, methods, perspectives*, 31-42, Research Notes No.21, Department of Forest Mensuration and Management, University of Helsinki, Helsinki, Finland.

Schneider, L.C., and Pontius, Jr. R.G., 2001. Modeling land use change in the Ipswich watershed, Massachusetts, USA. *Agriculture, Ecosystems and Environment*, 85, 83–94.

Schölkopf, B., Burges, C. J. C., and Smola, A. J., 1999. *Advances in Kernel Methods*, MIT Press, Cambridge.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A., and Williamson, R. C., 2001. Estimating the support of a high-dimensional distribution. *Neural Computer*, 13(7), 1443–1471.

Serneels, S., and Lambin, E. F., 2001. Proximate causes of land use change in Narok District, Kenya: a spatial statistical model. *Agriculture, Ecosystems and Environment*, 85, 65–81.

Serneels, S., Said, M., and Lambin, E. F., 2001. Land-cover changes around a major East Africa wildlife reserve: the Mara ecosystem. *International Journal of Remote Sensing*, 22, 3397-3420.

Singh, A., 1989. Digital change detection techniques using remotely-sensed data. *International Journal of Remote Sensing*, 10, 989-1003.

Smola, A. J., 1996. Regression estimation with support vector learning machines. Master’s thesis, Technique University of Munich.

Smola, A. J., Bartlett, P., Schölkopf, B., and Schuurmans, C., 1999. *Advances in Large Margin Classifiers*, MIT Press, Cambridge.

Song, Q., Hu, W., and Xie, W., 2002. Robust support vector machine for bullet hole image classification. *IEEE Transaction on Systems, Man and Cybernetics Part C,* 32(4), 440-448.

Sui, D.Z., 1994. Recent applications of neural networks for spatial data handling. *Canadian Journal of Remote Sensing*, 20, 368-380.

Sui, D. Z., and Zeng, H., 2001. Modeling the dynamics of landscape structure in Asia's emerging desakota regions: a case study in Shenzhen. *Landscape and Urban Planning*, 53, 37-52.

Suykens, J. A. K., and Vandewalle, J., 1999. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3), 293–300.

Taha, H. A., 1997. *Operations research: an introduction*, Prentice Hall, New Jersey.

Theobald, D. M., and Hobbs, N. T., 1998. Forecasting rural land-use change: a comparison of regression and spatial-based models. *Geographical and Environmental Modeling*, 2, 65-82.

Torrens, P. M., and O'Sullivan, D., 2001. Cellular automata and urban simulation: where do we go from here? *Environment and Planning B: Planning and Design*, 28, 163-168.

Turner, B. L. II, Skole, D., Sanderson, S., Fischer, G., Fresco, L., and Leemans, R., 1995. *Land-Use and Land-Cover Change; Science/Research Plan*. IGBP Report No.35, HDP Report No.7. IGBP and HDP, Stockholm and Geneva.

Ustun, B., Melssen, W. J., Oudenhuijzen, M., and Buydens, L. M. C., 2005. Determination of optimal support vector regression parameters by genetic algorithms and simplex optimization. *Analytical Chinica Acta*, 544, 292-305.

Vapnik, V., 1995. *The nature of statistical learning*, Springer-Verlag, New York.

Vapnik, V., and Chapelle, O., 1999. Bounds on error expectation for SVM. In Smola, A.J., Bartlett, P., Schölkopf, B., and Schuurmans, C. (eds.), *Advances in Large Margin Classifiers*, MIT Press, Cambridge.

VeldKamp, A., and Lambin, E. F., 2001. Predicting land use change. *Agriculture, Ecosystems and Environment*, 85, 1-6.

Verburg, P. H., Koning, G. H. J., Kok, K., Veldkamp, A., and Priess, J., 2001. The CLUE modelling framework: an integrated model for the analysis of land use change. Chapter 2 in Singh, R. B., Jefferson, F., and Himiyama, Y. (eds.), *Land Use and Cover Change*, Science Publishers, Inc., Enfield.

Verburg, P. H., Schot, P., Dijst, M., and Veldkamp, A., 2004. Land use change modeling: current practice and research priorities. *Geojournal*, 61(4), 309–324.

Webster, C., and Wu, F., 2001. Coarse, spatial pricing and self-organizing cities. *Urban Studies*, 38(11), 2037–2054.

White, R., and Engelen, G., 2000. High-resolution integrated modeling of the spatial dynamics of urban and regional systems. *Computers, Environment and Urban Systems*, 24, 383-400.

White, R., Engelen, G., and Uijee, I., 1997. The use of constrained cellular automata for high-resolution modeling of urban land use dynamics. *Environment and Planning B: Planning and Design*, 24, 323–343.

Wu, F., 1998. SimLand: a prototype to simulate land conversion through the integrated GIS and CA with AHP-derived transition rules. *International Journal of Geographical Information Science,* 12(1), 63-82.

Wu, F., 2002. Calibration of stochastic cellular automata: the application to rural-urban land conversions. *International Journal of Geographical Information Science*, 16, 795-818.

Wu, F., and Webster, C. T., 1998. Simulation of land development through the integration of cellular automata and multi-criteria evaluation. *Environment and Planning B: Planning and Design*, 25.

Wu, F., and Yeh, A. G., 1997. Changing spatial distribution and determinants of land development in Chinese cities in the transition from a centrally planned economy to a socialist market economy: a case study of Guangzhou. *Urban Studies*, 34, 1851-1879.

Xie, C., Huang, B., Claramunt, C., and Chandramouli, M., 2005. Spatial logistic regression and GIS to model rural-urban land conversion. In *Second International Colloquium on the Behavioral Foundations of Integrated Land-use and Transportation Models: Frameworks, Models and Applications,* Toronto, Canada, July 2005.

Xie, C., Huang, B., and Tay, R., 2006. Evaluation of Newton-Raphson method versus genetic algorithm for logistic analysis with application to rural-urban land conversion modeling. In *Proceedings of the 85th Annual Meeting of the Transportation Research Board*, Washington, USA, January 2006.

Yang, X., 2000. Integrating image analysis and dynamic spatial modeling with GIS in a rapidly suburbanizing environment. PhD Dissertation, University of Georgia, Athens, Georgia.

Yang, X., and Lo, C. P., 2003. Modeling urban growth and landscape changes in the Atlanta metropolitan area. *International Journal of Geographic Information Science*, 17, 463-488.

Zhang, J., and Yang, Y., 2003. Robustness of regularized linear classification methods in text categorization. In *Proceedings of SIGIR'03,* Toronto, Canada.

Zhao, G. X., Lin, G., and Warner, T., 2004, Using thematic mapper data for change detection and sustainable use of cultivated land: a case study in the Yellow River Delta, China. *International Journal of Remote Sensing*, 25, 2509-2522.

Zhu, G., and Blumberg, D. G., 2002. Classification using ASTER data and SVM algorithms: the case study of Beer Sheva, Israel. *Remote Sensing of Environment*, 80, 233-240.

APPENDIX A: DUALITY THEORY

A Linear Program (LP) is an important branch of applied mathematics. It is a linear optimization problem which aims to optimize (maximize or minimize) a linear objective function subject to linear constraints (inequalities or equalities). The standard inequality form of a maximum LP is the following:

$$\begin{aligned} &\textit{maximize} \quad \mathbf{c}^T\mathbf{x} \\ &\textit{subject to} \quad \mathbf{A}\mathbf{x} \le \mathbf{b} \\ &\qquad\qquad\quad \mathbf{x} \ge 0 \end{aligned} \tag{A.1}$$

where $\mathbf{x}$ is a n -dimension variable $(x_1,\ldots,x_n)^T$, $\mathbf{c}$ is a n -dimension coefficient determines the objective function. The $m \times n$ matrix $\mathbf{A}$ and the n -dimension vector $\mathbf{b}$ determine the m linear constraints, which are here only inequalities. Furthermore, all variables $\mathbf{x}_i$ are assumed to be nonnegative.

Every LP problem that aims to maximize the objective function gives rise to a related problem, called ***dual problem***, which aims to minimize an objective function, and vice versa. The dual of an LP is motivated by finding an upper bound to the objective function of the given LP (which is called the ***primal problem***). In general, the dual LP for the primal LP (A.1) is obtained as follows:

- Multiply each primal inequality by some nonnegative number y_i .
- Add each of the n columns and require that the resulting coefficient of x_j for $j = 1,...,n$, $\sum_{i=1}^{m} y_j a_{ji}$, be at least as large as the coefficient c_j in the objective function. Since $x_j \ge 0$, this will set an upper bound for the objective function.
- Minimize the resulting right hand side $y_1 b_1 + \cdots + y_m b_m$.

So the dual problem of the primal problem (A.1) is:

$$\begin{aligned} &\textit{minimize} \quad \mathbf{y}^T\mathbf{b} \\ &\textit{subject to} \quad \mathbf{y}^T\mathbf{A} \geq \mathbf{c}^T \\ &\qquad\qquad\quad \mathbf{y} \geq 0 \end{aligned} \tag{A.2}$$

Theorem A.1 (Weak LP Duality) If $\mathbf{x}$ is a feasible solution of the primal LP (A.1) and $\mathbf{y}$ is a feasible solution of the dual LP (A.2), then their objective functions satisfy:

$$\mathbf{c}^T\mathbf{x} \leq \mathbf{y}^T\mathbf{b}$$

If $\mathbf{c}^T\mathbf{x} = \mathbf{y}^T\mathbf{b}$ (equality holds), these two solutions are optimal for both LPs.

***Proof*.**

$$\left.\begin{aligned} \mathbf{Ax} \leq \mathbf{b}, \mathbf{y} \geq 0 \Rightarrow \mathbf{y}^T\mathbf{Ax} \leq \mathbf{y}^T\mathbf{b} \\ \mathbf{y}^T\mathbf{A} \geq \mathbf{c}^T, \mathbf{x} \geq 0 \Rightarrow \mathbf{y}^T\mathbf{Ax} \geq \mathbf{c}^T\mathbf{x} \end{aligned}\right\} \Rightarrow \mathbf{c}^T\mathbf{x} \leq \mathbf{y}^T\mathbf{Ax} \leq \mathbf{y}^T\mathbf{b}$$

Theorem A.2 (Strong LP Duality) If a standard LP is bounded feasible, its dual LP is also bounded feasible. They have optimal solutions with equal value of objective functions.

The above theorem is the central theorem of duality theory. Its proof is not simplistic and is provided in this thesis.

Duality is of great theoretical importance. Some LP problems, which are difficult to solve directly, may be solved much more easily by converting them to their dual form. For computational and other reasons, LP is often considered in equality form:

$$\begin{aligned} &\textit{maximize} \quad \mathbf{c}^T\mathbf{x} \\ &\textit{subject to} \quad \mathbf{Ax} = \mathbf{b} \\ &\qquad\qquad\quad \mathbf{x} \geq 0 \end{aligned} \tag{A.3}$$

LP in equality form is a more generalized case. Any LP in inequality form (A.1) can be converted to equality form by introducing a slack variable z_i for each constraint:

$$\begin{aligned} \max imize \quad & \mathbf{c}^T\mathbf{x} \\ subject\ to \quad & \mathbf{Ax}+\mathbf{z}=\mathbf{b} \\ & \mathbf{x} \quad \geq 0 \\ & \quad \mathbf{z} \geq 0 \end{aligned} \tag{A.4}$$

This amounts to extending the constraint matrix $\mathbf{A}$ to the right by an identity matrix and by adding coefficients 0 in the objective function for the slack variables.

A dual problem should be considered for the LP in a more general equality form.

From the steps of constructing the dual LP, it is obvious that, since the equality constraint is preserved even when multiplied with a negative number, the corresponding dual variable is unrestricted in sign for any primal equality constraint. An LP in general form has inequalities and equalities as constraints, as well as nonnegative and unrestricted variables. In the dual LP, the inequalities correspond to nonnegative variables and the equalities correspond to unrestricted variables and vice versa. The full definition of an LP in general form is as follows. Let M and N be finite sets (whose elements denote rows and columns, respectively), $I \subseteq M$, $J \subseteq N$, $\mathbf{A} \in \mathbf{R}^{M\times N}$, $\mathbf{b} \in \mathbf{R}^{M}$, $\mathbf{c} \in \mathbf{R}^{N}$.

Here the indices in I denote primal inequalities and corresponding nonnegative dual variables, whereas those in $M-I$ denote primal equality constraints and corresponding unconstrained dual variables. The sets J and $N-J$ play the same role with “primal” and “dual” interchanged. The feasible sets for primal and dual LP can be defined as follows:

$$\begin{aligned} P = \{ \mathbf{x} \in \mathbf{R}^N \mid & \sum_{j\in N} \mathbf{a}_{ij}\mathbf{x}_j \leq \mathbf{b}_{i,} \quad i \in I, \\ & \sum_{j\in N} \mathbf{a}_{ij}\mathbf{x}_j = \mathbf{b}_{i,} \quad i \in M-I, \\ & \mathbf{x}_j \geq 0, \quad j \in J\}. \end{aligned} \tag{A.5}$$

Any $\mathbf{x}$ belonging to P is called primal feasible. The primal LP is the problem:

$$\begin{aligned} &\textit{maximize} \quad \mathbf{c}^T\mathbf{x} \\ &\textit{subject to} \quad \mathbf{x} \in P \end{aligned} \tag{A.6}$$

Then, the feasible set of the corresponding dual LP can be expressed as:

$$\begin{aligned} D = \{\mathbf{y} \in \mathbf{R}^M \mid \sum_{i\in M} \mathbf{y}_i \mathbf{a}_{ij} \geq \mathbf{c}_{j,} \quad & j \in J, \\ \sum_{i\in M} \mathbf{y}_i \mathbf{a}_{ij} = \mathbf{c}_{j,} \quad & j \in N - J, \\ \mathbf{y}_i \geq 0, \quad & i \in I\}. \end{aligned} \tag{A.7}$$

and the corresponding dual LP is:

$$\begin{aligned} &\textit{minimize} \quad \mathbf{y}^T\mathbf{b} \\ &\textit{subject to} \quad \mathbf{y} \in D \end{aligned} \tag{A.8}$$

Then, the duality theorem of linear programming states: a) for any primal and dual feasible solutions, the corresponding objective functions are mutual bounds; and b) if the primal and the dual LP both have feasible solutions, then they have optimal solutions with the same value of their objective functions.

Theorem A.3 (General LP duality) Consider the primal-dual pair of LPs (A.6), (A.8). Then

(a) (Weak duality) $\mathbf{c}^T\mathbf{x} \leq \mathbf{y}^T\mathbf{b}$ for all $\mathbf{x} \in P$ and $\mathbf{y} \in D$.

(b) (Strong duality) If $P \neq \emptyset$ and $D \neq \emptyset$, then $\mathbf{c}^T\mathbf{x} = \mathbf{y}^T\mathbf{b}$ for some $\mathbf{x} \in P$ and $\mathbf{y} \in D$.

APPENDIX B: SEQUENTIAL MINIMAL OPTIMIZATION

Sequential Minimal Optimization (SMO) is a simple algorithm used to efficiently solve the QP problem in SVMs. It decomposes the overall QP problem into sub-problems, which involve only two Lagrange multipliers. At every step, SMO chooses two Lagrange multipliers to jointly optimize, finds the optimal values for these multipliers, and updates the SVMs to reflect the new optimal values. Since the solving of two Lagrange multipliers is done analytically, the numerical QP optimization and any extra matrix storage are avoided. Therefore, the computational and storage efficiencies of SMO are very good. The following sections discuss the main components in SMO: an analytic method of solving the two Lagrange multipliers optimization problem, updating after a successful optimization, and a heuristic for choosing two multipliers to optimize.

B.1 SOLVING OF TWO LAGRANGE MULTIPLIERS

The SMO algorithm searches through the feasible region of the dual problem and maximizes the objective function:

$$\begin{aligned} &L(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i ' \cdot \mathbf{x}_j \\ &0 \le \alpha_i \le C, \qquad i = 1, \ldots, m \end{aligned} \tag{B.1}$$

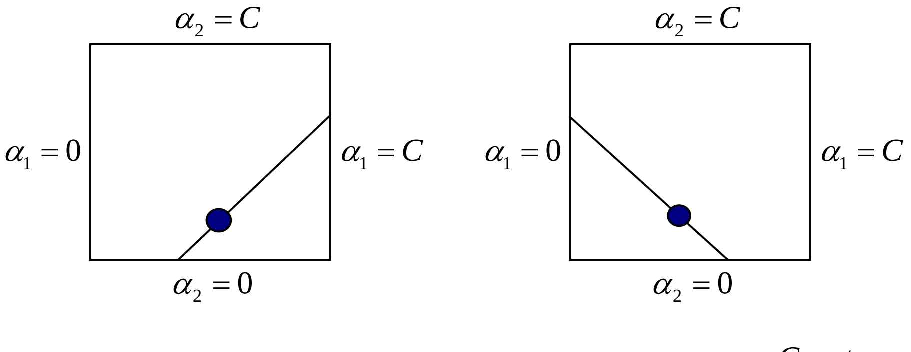
SMO decomposes the problem into a set of smallest possible optimization problems and works by optimizing only two Lagrange multipliers at a time with the other Lagrange multipliers fixed. In order to solve the two Lagrange multipliers optimization problem, SMO first computes the constraints on the multipliers and then solves the constrained minimum.

Initially, we can set $\alpha_i = 0, i = 1, \ldots, m$, which is a feasible solution. Without loss of generality, suppose we are optimizing α_1 and α_2 from an old set of a feasible solution:

$\alpha_1^{old}, \alpha_2^{old}, \alpha_3, \ldots, \alpha_m$. Considering the bound constraints on the multipliers: $\sum_{i=1}^{m} y_i\alpha_i = 0$, we have:

$$y_1\alpha_1 + y_2\alpha_2 = y_1\alpha_1^{old} + y_2\alpha_2^{old} = -\sum_{i=3}^{m} y_i\alpha_i = Const \tag{B.2}$$

This confines the optimization to lie on a diagonal line segment, as shown in Figure B.1:



$y_1 \neq y_2 \Rightarrow \alpha_1 - \alpha_2 = Const$ $\qquad$ $y_1 = y_2 \Rightarrow \alpha_1 + \alpha_2 = Const$

Figure B.1: Two Lagrange multipliers optimization problem

Bearing in mind that $y \in \{-1,+1\}$, let $s = y_1y_2$ and multiply (B.2) by y_1, and we have

$$\alpha_1 = \gamma - s\alpha_2 \tag{B.3}$$

where $\gamma = \alpha_1 + s\alpha_2 = \alpha_1^{old} + s\alpha_2^{old}$ is a constant.

Fixing the other $\alpha_i^{'}s$, the objective function now can be written as:

$$\begin{aligned} L(\boldsymbol{\alpha}) = \alpha_1 + \alpha_2 + Const_1 - \frac{1}{2}(y_1y_1\mathbf{x}_1^T \cdot \mathbf{x}_1\alpha_1^2 + y_2y_2\mathbf{x}_2^T \cdot \mathbf{x}_2\alpha_2^2 \\ +2y_1y_2\mathbf{x}_1^T \cdot \mathbf{x}_2\alpha_1\alpha_2 + 2\left(\sum_{i=3}^{m}\alpha_i y_i \mathbf{x}_i^T\right)\cdot(y_1\mathbf{x}_1\alpha_1 + y_2\mathbf{x}_2\alpha_2) + Const_2) \end{aligned} \tag{B.4}$$

To be convenient, denote $K_{11} = \mathbf{x}_1^T \cdot \mathbf{x}_1$, $K_{22} = \mathbf{x}_2^T \cdot \mathbf{x}_2$, $K_{12} = \mathbf{x}_1^T \cdot \mathbf{x}_2$, and

$$\begin{aligned} v_j &= \sum_{i=3}^{m} \alpha_i y_i \mathbf{x}_i^T \cdot \mathbf{x}_j \\ &= \mathbf{x}_j^T \cdot \mathbf{w}^{old} - \alpha_1^{old} y_1 \mathbf{x}_1^T \cdot \mathbf{x}_j - \alpha_2^{old} y_2 \mathbf{x}_2^T \cdot \mathbf{x}_j \\ &= (\mathbf{x}_j^T \cdot \mathbf{w}^{old} + b^{old}) - b^{old} - \alpha_1^{old} y_1 \mathbf{x}_1^T \cdot \mathbf{x}_j - \alpha_2^{old} y_2 \mathbf{x}_2^T \cdot \mathbf{x}_j \\ &= u_j^{old} - b^{old} - \alpha_1^{old} y_1 \mathbf{x}_1^T \cdot \mathbf{x}_j - \alpha_2^{old} y_2 \mathbf{x}_2^T \cdot \mathbf{x}_j \end{aligned} \tag{B.5}$$

where $u_j^{old} = \mathbf{x}_j^T \mathbf{w}^{old} + b^{old}$ is the output of $\mathbf{x}_j$ under old parameters. Therefore, the objective function can be expressed as:

$$\begin{aligned} L(\alpha) &= \alpha_1 + \alpha_2 - \frac{1}{2}\left(K_{11}\alpha_1^2 + K_{22}\alpha_2^2 + 2sK_{12}\alpha_1\alpha_2 + 2y_1v_1\alpha_1 + 2y_2v_2\alpha_2\right) + Const \\ &= \gamma - s\alpha_2 + \alpha_2 - \frac{1}{2}(K_{11}(\gamma - s\alpha_2)^2 + K_{22}\alpha_2^2 + 2sK_{12}(\gamma - s\alpha_2)\alpha_2 \\ &\quad + 2y_1v_1(\gamma - s\alpha_2) + 2y_2v_2\alpha_2) + Const \\ &= (1-s)\alpha_2 - \frac{1}{2}K_{11}(\gamma - s\alpha_2)^2 - \frac{1}{2}K_{22}\alpha_2^2 - sK_{12}(\gamma - s\alpha_2)\alpha_2 - y_1v_1(\gamma - s\alpha_2) \\ &\quad - y_2v_2\alpha_2 + Const \\ &= (1-s)\alpha_2 - \frac{1}{2}K_{11}\gamma^2 + sK_{11}\gamma\alpha_2 - \frac{1}{2}K_{11}s^2\alpha_2^2 - \frac{1}{2}K_{22}\alpha_2^2 - sK_{12}\gamma\alpha_2 + s^2K_{12}\alpha_2^2 \\ &\quad - y_1v_1\gamma + sy_1v_1\alpha_2 - y_2v_2\alpha_2 + Const \end{aligned} \tag{B.6}$$

Since $s^2 = (y_1y_2)^2 = y_1^2y_2^2 = 1$ and $sy_1 = y_1^2y_2 = y_2$, we have:

$$\begin{aligned} L(\alpha) &= (1-s)\alpha_2 + sK_{11}\gamma\alpha_2 - \frac{1}{2}K_{11}\alpha_2^2 - \frac{1}{2}K_{22}\alpha_2^2 - sK_{12}\gamma\alpha_2 + K_{12}\alpha_2^2 + y_2v_1\alpha_2 \\ &\quad - y_2v_2\alpha_2 + Const \\ &= (-\frac{1}{2}K_{11} - \frac{1}{2}K_{22} + K_{12})\alpha_2^2 + (1 - s + sK_{11}\gamma - sK_{12}\gamma + y_2v_1 - y_2v_2)\alpha_2 + Const \\ &= \frac{1}{2}(2K_{12} - K_{11} - K_{22})\alpha_2^2 + (1 - s + sK_{11}\gamma - sK_{12}\gamma + y_2v_1 - y_2v_2)\alpha_2 + Const \end{aligned} \tag{B.7}$$

Let $\eta = 2K_{12} - K_{11} - K_{22}$ is the coefficient of the second order term. The coefficient of the first order term is:

$$\begin{aligned}
\rho &= 1 - s + sK_{11}\gamma - sK_{12}\gamma + y_2 v_1 - y_2 v_2 \\
&= 1 - s + sK_{11}(\alpha_1^{old} + s\alpha_2^{old}) - sK_{12}(\alpha_1^{old} + s\alpha_2^{old}) + y_2(u_1^{old} - b^{old} - \alpha_1^{old} y_1 K_{11} - \alpha_2^{old} y_2 K_{12}) \\
&\qquad - y_2(u_2^{old} - b^{old} - \alpha_1^{old} y_1 K_{12} - \alpha_2^{old} y_2 K_{22}) \\
&= 1 - s + sK_{11}\alpha_1^{old} + K_{11}\alpha_2^{old} - sK_{12}\alpha_1^{old} - K_{12}\alpha_2^{old} + y_2 u_1^{old} - y_2 b^{old} - sK_{11}\alpha_1^{old} - K_{12}\alpha_2^{old} \\
&\qquad - y_2 u_2^{old} + y_2 b^{old} + sK_{12}\alpha_1^{old} + K_{22}\alpha_2^{old} \\
&= 1 - s + (sK_{11} - sK_{12} - sK_{11} + sK_{12})\alpha_1^{old} + (K_{11} - 2K_{12} + K_{22})\alpha_2^{old} + y_2(u_1^{old} - u_2^{old}) \\
&= y_2^2 - y_1 y_2 + (K_{11} - 2K_{12} + K_{22})\alpha_2^{old} + y_2(u_1^{old} - u_2^{old}) \\
&= y_2(y_2 - y_1 + u_1^{old} - u_2^{old}) - \eta\alpha_2^{old} \\
&= y_2((u_1^{old} - y_1) - (u_2^{old} - y_2)) - \eta\alpha_2^{old} \\
&= y_2(E_1^{old} - E_2^{old}) - \eta\alpha_2^{old}
\end{aligned} \tag{B.8}$$

where $E_i^{old} = u_i^{old} - y_i$ is the prediction error on $\mathbf{x}_j$ under old parameters.

Hence, the objective function can be simply expressed as:

$$L(\alpha) = \frac{1}{2}\eta\alpha_2^2 + (y_2(E_1^{old} - E_2^{old}) - \eta\alpha_2^{old})\alpha_2 + Const \tag{B.9}$$

The first and second derivatives of the objective function are:

$$\frac{dL(\alpha)}{d\alpha_2} = \eta\alpha_2 + (y_2(E_1^{old} - E_2^{old}) - \eta\alpha_2^{old}) \tag{B.10}$$

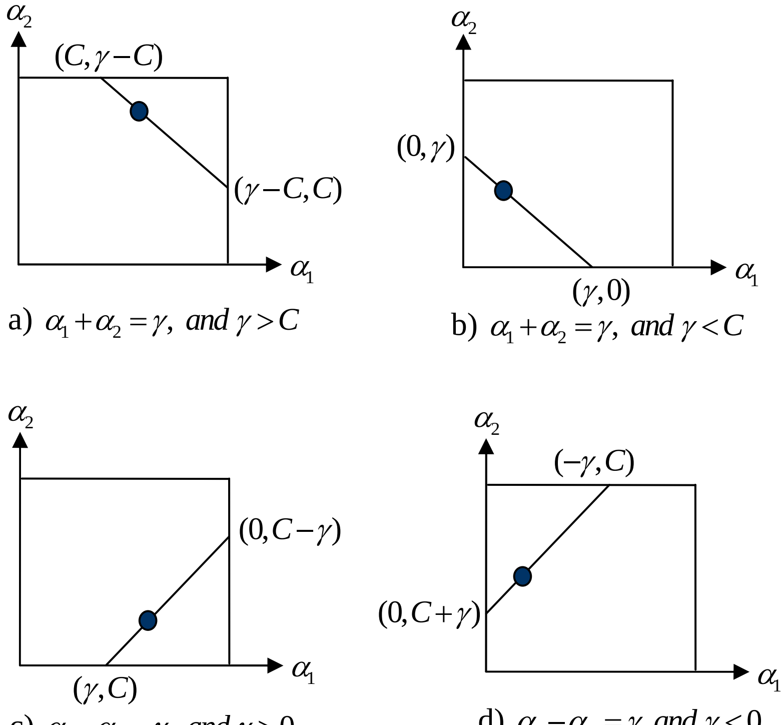
$$\frac{d^2L(\alpha)}{d\alpha_2^2} = \eta \tag{B.11}$$

Since $\eta = 2K_{12} - K_{11} - K_{22} = -(\mathbf{x}_2 - \mathbf{x}_1)^T \cdot (\mathbf{x}_2 - \mathbf{x}_1) = -\|\mathbf{x}_2 - \mathbf{x}_1\|^2 \leq 0$, the second derivative is always negative or null. To maximize the objective function, let the first derivative to be null, and we have:

$$\alpha_2^{new} = -\frac{y_2(E_1^{old} - E_2^{old}) - \eta\alpha_2^{old}}{\eta} = \alpha_2^{old} + \frac{y_2(E_2^{old} - E_1^{old})}{\eta} \qquad \text{(B.12)}$$

If $\eta < 0$, the above equation gives us the unconstrained maximum point α_2^{new}. The constrained maximum is found by clipping the unconstrained maximum to the feasible range $0 \le \alpha_2^{new} \le C$, which is determined as follows:

- If $s = 1$, then $\alpha_1 + \alpha_2 = \gamma$.
    - If $\gamma > C$, then the range of α_2 is $[C, \gamma - C]$ (Figure B.2.a).
    - If $\gamma < C$, then the range of α_2 is $[0, \gamma]$ (Figure B.2.b).
- If $s = -1$, then $\alpha_1 - \alpha_2 = \gamma$.
    - If $\gamma > 0$, then the range of α_2 is $[0, C - \gamma]$ (Figure B.2.c).
    - If $\gamma < 0$, then the range of α_2 is $[-\gamma, C]$ (Figure B.2.d).



a) $\alpha_1 + \alpha_2 = \gamma,\ and\ \gamma > C$

b) $\alpha_1 + \alpha_2 = \gamma,\ and\ \gamma < C$

c) $\alpha_1 - \alpha_2 = \gamma,\ and\ \gamma > 0$

d) $\alpha_1 - \alpha_2 = \gamma, and\ \gamma < 0$

Figure B.2: Constrained maximum point under different conditions

Since the second derivative is always negative or null, the $L(\alpha)$ curve is convex. Let the minimum feasible value of α_2 be L, maximum be H. We have:

$$\alpha_2^{new,clipped} = \begin{cases} H, & if\ H < \alpha_2^{new} \\ \alpha_2^{new}, & if\ L \le \alpha_2^{new} \le H \\ L, & if\ \alpha_2^{new} < L \end{cases} \qquad \text{(B.13)}$$

To summarize, given α_1, α_2 and the corresponding y_1, y_2, K_{11}, K_{12}, K_{22}, $E_2^{old} - E_1^{old}$, we can optimize the two $\alpha_i^{'}s$ by the following procedure:

1. $\eta = 2K_{12} - K_{11} - K_{22}$
2. If $\eta < 0$,

   $$\Delta\alpha_2 = \frac{y_2(E_2^{old} - E_1^{old})}{\eta} \qquad \text{(B.14)}$$

   and clip the solution within the feasible region. Then

   $$\Delta\alpha_1 = -s\Delta\alpha_2. \qquad \text{(B.15)}$$
3. If $\eta = 0$, we need to evaluate the objective function at the two endpoints of the feasible region, and set α_2^{new} to be the one with larger objective function value.

B.2 UPDATING AFTER A SUCCESSFUL OPTIMIZATION

When α_1, α_2 are changed by $\Delta\alpha_1$, $\Delta\alpha_2$, we can update the prediction error $E(\mathbf{x}, y)$, the coefficient vector $\mathbf{w}$ (for linear kernel), and the offset b. The prediction error $E(\mathbf{x}, y)$ is:

$$E(\mathbf{x}, y) = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i^T \cdot \mathbf{x} + b - y \qquad \text{(B.17)}$$

The change in E is

$$\Delta E(\mathbf{x}, y) = \Delta\alpha_1 y_1 \mathbf{x}_1^T \cdot \mathbf{x} + \Delta\alpha_2 y_2 \mathbf{x}_2^T \cdot \mathbf{x} + \Delta b \tag{B.18}$$

Obviously, when the α_i^{new} is not at the bounds, the output of $\mathbf{x}_i$ is forced to be y_i. Hence, the change in the offset b can be computed by forcing $E_1^{new} = 0$ if $0 < \alpha_1^{new} < C$ (or $E_2^{new} = 0$ if $0 < \alpha_2^{new} < C$):

$$\begin{aligned} 0 &= E(\mathbf{x}, y)^{new} \\ &= E(\mathbf{x}, y)^{old} + \Delta E(\mathbf{x}, y) \\ &= E(\mathbf{x}, y)^{old} + \Delta\alpha_1 y_1 \mathbf{x}_1^T \cdot \mathbf{x} + \Delta\alpha_2 y_2 \mathbf{x}_2^T \cdot \mathbf{x} + \Delta b \end{aligned} \tag{B.19}$$

Therefore, we have

$$\Delta b = -E(\mathbf{x}, y)^{old} - \Delta\alpha_1 y_1 \mathbf{x}_1^T \cdot \mathbf{x} - \Delta\alpha_2 y_2 \mathbf{x}_2^T \cdot \mathbf{x} \tag{B.20}$$

If $\alpha_1 = 0$, we can only say $y_1 E_1^{new} \geq 0$; similarly, if $\alpha_1 = C$, we have $y_1 E_2^{new} \leq 0$. If both α_1 and α_2 take values 0 or C, the SMO algorithm computes two values of the new b for α_1 and α_2 using (B.20), and takes the average.

For the coefficient vector of linear kernels,

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{B.21}$$

$$\Delta\mathbf{w} = \Delta\alpha_1 y_1 \mathbf{x}_1 + \Delta\alpha_2 y_2 \mathbf{x}_2 \tag{B.22}$$

B.3 CHOOSING TWO LAGRANGE MULTIPLIERS TO OPTIMIZE

There are two separate heuristics for choosing two Lagrange multipliers for optimization: one for the first Lagrange multiplier and the other for the second Lagrange multiplier.

The choice of the first Lagrange multiplier provides the outer loop of the SMO algorithm. It first iterates over the entire training set, determining whether each example violates the KKT conditions:

$$\begin{aligned} &\alpha_i = 0 \Leftrightarrow y_i u_i \geq 1 \\ &0 < \alpha_i < C \Leftrightarrow y_i u_i = 1 \\ &\alpha_i = C \Leftrightarrow y_i u_i \leq 1 \end{aligned} \tag{B.23}$$

If an example violates the KKT conditions, it is then eligible for optimization.

Once the first Lagrange multiplier α_1 is chosen, the inner loop looks for a non-boundary example ($0 < \alpha_2 < C$) that maximizes $| E_2 - E_1 |$. Under unusual circumstances, SMO cannot make positive progress using the second heuristic described above. If SMO does not make positive progress, it starts a sequential scan through the non-boundary examples, starting at a random position, and searches for a second example that can make positive progress. If none of the non-bound examples make positive progress, SMO starts a sequential scan through all the examples, also starting at a random position, until a second example is found that can make positive progress. In extremely degenerate circumstances, none of the examples will make positive progress. In this case, the first example is skipped and SMO continues with another potential first example.