Row and Column Selection Algorithm for SVR Model Estimation on Large Scale Business Problems

by

Kübra Yaman

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Industrial Engineering

Koç University

August, 2010

Koç University
Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Kübra Yaman

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Asst. Prof. Özden Gür Ali (Advisor)

Prof. Serpil Sayın

Asst. Prof. Serdar Kozat

Date:

...to my parents
in love and gratitude...

ABSTRACT

This study introduces an algorithm, which selects important observations and variables to estimate SVR models for very large data sets. In this two-stage methodology, namely the Row and Column Selection Algorithm, ϵ-SVR models with L_1-norm regularization are used both for selecting rows and columns. The first stage penalizes support vector weights to identify few support vectors as important points to include in the training data set. These support vectors are then used in the second stage to select the variable subset to be kept in the training data by penalizing the variable weights. The accuracy of holdout test set of the RBF-SVR models trained on this set including selected rows with all variables is significantly better than the accuracy of the same model trained on the benchmark which is the randomly sampled data set of the same size with all variables and SVMTorch.

The contribution of this thesis is the development of an algorithm which facilitates estimating SVR models with very large data sets which are accurate and low complexity. By using the proposed algorithm, it is possible to select the important observations and variables and use them for estimation. The experimental results validate that the resulting training data set works effectively and reduces the number of variables dramatically while improving the generalization error of the RBF-SVR models in the presence of redundant variables. Furthermore, we investigate how the selected points differ from others by analyzing their distribution with respect to their distance from the prediction line, target values and the input variables of data set.

This analysis demonstrates that L_1-norm ϵ-SVR provides much more sparse solution than standard ϵ-SVR. Further the observations with extreme target values are more likely to be selected than average observations. Interestingly, in contrast to standard ϵ-SVR, the L_1-norm ϵ-SVR support vectors can be located both inside and outside the ϵ-tube. Moreover, low multi-collinearity between selected columns gives face validity variable selection procedure of our algorithm, namely second part of the proposed algorithm. Lastly, we identify which points are selected with respect to variables' values. The result of this analysis indicates that the row and column selection algorithm select observations based on background knowledge.

ÖZETÇE

Bu çalışmada çok büyük veri setleri için Destek Vektör Regresyon (DVR) modellinin kurulabilmesini mümkün kılmak için önemli nokta ve değişkenlerini seçen bir algoritma geliştirilmiştir. İki aşamalı bu yöntemde, yani satır ve sütun seçme algoritmasında, hem satır hem de sütun seçimininde L_1-norm düzenlemeli ϵ-DVR modelleri kurulmuştur. İlk aşama, eğitim veri setinin destek vektörlerinin ağırlıklarını cezalandırarak veri setinin önemli noktalarından en az sayıda destek vektökterini seçer ve bu seçilen noktaları yeni eğitim veri setine dahil eder. Seçilen bu destek vektörlerinden oluşturulan yeni eğitim veri seti daha sonra ikinci aşamada değişken ağırlıklarını cezalandırarak eğitim veri setinde tutulacak olan değişken alt küme seçiminde kullanılır.

Seçilen satır ve tüm değişkenleri içeren eğitim veri seti ile çalıştırılıp kurulan Radyal Tabanlı İşlev (RTI) çekirdekli DVR modellerinin test veri seti üzerindeki doğruluğu karşılaştırma yapılan yani seçilen satır sayısı kadar satırla tüm değişkenleri içeren rassal örneklem veri setinden ve SVMTorch algoritması ile oluşturulan modellerden önemli ölüçüde daha iyi olduğu gözlenmiştir.

Bu tezin katkısı oldukça büyük veri setlerini kullanarak doğru ve düşük karmaşıklık içeren DVR modellerinin kurulmasını kolaylaştıran bir algoritma geliştirmesidir. Bu çalışmada önerilen algoritma veri setlerinin önemli gözlem ve değişkenlerini seçip ve onları tahmin modelinde kullanmayı mümkün kılmıştır. Deneysel sonuçlar satır ve sütun seçme algoritmasının etkili bir şekilde çalıştığını ve gereksiz değişkenlerin varlığında değişken sayısını önemli ölüçüde azaltırken RT-DVR modellerinin genelleme hatasını iyileştirdiğini kanıtlamıştır. Bu çalışmada ayrıca seçilen noktaların diğerler noktalardan nasıl farklı anlayabilmek için seçilen noktaların tahmin çizgisine olan uzaklıklarına, hedef değere ve veri kümesinin değişkenlerine göre nasıl dağıldıkları

analiz edilmiştir.

Yapılan analizler sonucunda, L_1-normlu ϵ-DVR standart ϵ-DVR'a göre çok daha seyrek bir çözüm sunduğunu gözlenmiştir. Ayrıca L_1-normlu ϵ-DVR'de uç noktalardaki hedef değerlere sahip olan gözlemlerin seçilmesi ortalama hedef değerlere sahip olan gözlemlerden daha olasıdır. Standart ϵ-DVR'nin aksine, L_1-normlu ϵ-DVR algoritmasının destek vektörleri ϵ tüpünün içinde ve dışında olabilir. Bunlara ilaveten, seçilen sütunlar arasındaki düşük çoklu doğrusal bağıntı algoritmamızın ikinci kısmını oluşturan değişken seçimi prosedürünün doğru bir şekilde çalıştığını desteklemektedir. Son olarak, seçilen noktalarla değişken değerleri arasındaki ilişki incenlenmiş ve bu analizin sonucunda satır ve sütun seçme algoritmasının noktaları seçimini literatürdeki bazı ön bilgilere dayalı yaptığı gözlenmiştir.

ACKNOWLEDGMENTS

Writing these few lines will be the last thing I will do as a student at Koç University. However, this is some how harder than I thought. Because I have spent tough but extraordinary two years here. During this two year period not only I get my graduate degree, but also I relearn everything I know and especially how to talk, write and read.

I am deeply indebted to my advisor Özden Gür Ali. If I am now leaving this university with full of knowledge about my research, life, and business, it is her success. She encouraged me in all steps of my graduate study and her guidance, motivation, patience and enthusiasm helped me to overcome many crisis I faced during my graduate study. I has been privilege for me to work and spend two years with her.

I am grateful to members of my thesis committee Prof. Serpil Sayın and Asst. Prof. Serdar Kozat for critical reading of this thesis and for their valuable comments. In addition I want to thank TUBITAK for providing scholarship throughout my M.Sc education and Information Resource, Inc and the leading grocery chain, Migros, for providing the data for this study.

I would like to thank to my colleagues Emin, Naciye and Özge and my officemates Abdullah, Buşra, Müge and Nihal who makes our office, CAS130, enjoyable and bearable for me.

Finally, I owe special thanks to my sister Tuğba, for leading and supporting me

all along my life and my brother Emirhan for the fun he brings to our life, without my siblings life would be pointless and unbearable for me. Lastly, I am would like to thank to two most important people in my life, my parents Ziya and Nurhayat. Without their endless support, encouragement, devotion and love for me, I would never achieved my current position and in fact I would not have been writing these sentences.

Lastly, and most importantly, I am indebted to Almighty Allah and give thanks to him for making the impossible becomes possible for me, for giving me a lovely family and for guiding me to Özden Gür Ali to entrust my graduate study and future.

TABLE OF CONTENTS

List of Tables **xii**

List of Figures **xiv**

Chapter 1: Introduction **1**

Chapter 2: Literature Review **4**

2.1 Introduction 4

2.2 Support Vector Regression 4

2.2.1 Nonlinear Support Vector Regression 9

2.2.2 Loss Functions 10

2.2.3 L_1-norm Support Vector Regression 11

2.2.4 ϵ-insensitive Linear Regression 13

2.3 Sampling Methods 14

2.4 Variable Selection 16

2.5 Previous Methods for Training SVR with Large Data 19

Chapter 3: Row and Column Algorithm **22**

3.1 Introduction 22

3.2 Row Selection 22

3.3 Column Selection 24

3.4 Model Parameter Selection Procedure 25

3.4.1 Selecting Loss Function Parameter (ϵ) 26

3.4.2 Selecting Parameter of Row Selection Step 27

3.4.3 Selecting Parameter of Column Selection Step 28

3.4.4 Selecting Parameter of standard ϵ-insensitive SVR 29

Chapter 4: Experiments and Results **30**

4.1 Introduction 30

4.2 Data Sets 31

4.3 Experimental Setup 33

4.4 Parameters of Algorithm 34

4.5 Resulting Training Data 36

4.6 Accuracy and Bias 36

4.7 Characteristics of selected row and columns 39

4.7.1 Selected Rows 39

4.7.2 Selected Columns 46

4.7.3 Effects of Variables 48

Chapter 5: Conclusion **51**

5.1 Future Works 53

Bibliography **54**

Vita **59**

LIST OF TABLES

4.1 Data Set Properties 33

4.2 Parameters used in the models 34

4.3 Complexity of Row and Column Selection Algorithm 35

4.4 Number of selected rows (observations) and columns (variables) in the training set 36

4.5 The test MAE associated with the training data set selection methods and the benchmark 37

4.6 Significance level for the paired sample z-tests on the mean difference for different models 38

4.7 The test ME associated with the training data set selection methods and the benchmark 39

4.8 Distribution of SVs to training error ranges 40

4.9 Probability of being SVs in the given error class 40

4.10 Percentage of SVs in L_1-norm vs L_2-norm 42

4.11 Comparison of SVs by L_1-norm with SVs of L_2-norm 43

4.12 Independence Test 44

4.13 Comparison of the SVs of Model II and Model III 45

4.14 Effects of chunking approach on the bias and accuracy 45

4.15 Overlapping Selected Columns among the Grocery and IRI Data Sets 47

4.16 Test of Independence of Variables 48

4.17 Effects of variables on support vectors 50

LIST OF FIGURES

2.1 Loss Functions in SVM - (a) Huber loss function and (b) ϵ-insensitive loss function 11

3.1 Estimation picture of L_1-norm(left) and L_2-norm(right) 25

4.1 Probability of being selected as SV in L_1-norm 41

4.2 Probability of being selected as SV in L_2-norm 42

Chapter 1

INTRODUCTION

With the rapid growth in size and number of available databases in many areas, there is an opportunity to use machine learning and statistical models to help the managerial decision making process. For example companies use forecasting in areas such as marketing, customer relationship management (CRM), financial planning which include large amounts of data, to increase their service levels while maintaining lower cost. To deal with these huge amounts of data, data mining algorithms are becoming increasingly attractive. Support Vector Regression (SVR) [1], a data mining algorithm, has been used successfully in many applications. Particularly, the SVR with the radial basis function (RBF) kernel is very popular, as it offers the ability to learn a variety of nonlinear relationships.

Support vector machines (SVM), developed by Vapnik [2], construct nonlinear models like the neural networks. However; due to the structural risk minimization principle in SVM, SVM differ in conjunction with the ability to generalize and strive to minimize model complexity. A significant advantage of SVM is that the solution to an SVM is global and unique. Moreover, the computational complexity of SVM does not depend on the number of the variables of the data set and SVM gives a sparse solution. The reason that SVM often outperforms other data mining algorithms is that it is less prone to overfitting, which constitutes the biggest problem with many data mining algorithms. The advantages of SVM encourage researchers to use SVM in various research fields. In 1996, a new version of SVM for prediction, called Support Vector Regression (SVR), is proposed by Vapnik et al.[2]. It has been used in time varying applications successfully. These advantages and accurate results of SVR motivates us to study on SVR in this thesis. Following section gives a detailed description of SVR formulation. On the other hand, applying this promising tool to large scale business problems, with tens of thousands of observations and hundreds of potentially useful variables, is not straightforward. The memory and time requirements grow with the square of the number of data points [3] and make it time consuming, if not infeasible.

In this thesis, to address this key problem, we propose the Row and Column Selection Algorithm to select a small but informative subset from large business data sets to train SVR models. We select rows and columns; i.e. observations and variables those are likely to lead to models which generalize and predict well. The algorithm consists of two steps: 1) Row selection dividing the original data into chunks and identifying the support vectors from the non-linear SVR model with L_1 norm regularization of the support vector weights to systematically identify the most informative points 2) Column selection - using the epsilon insensitive linear regression with L_1 norm regularization of the variable weights on this reduced data set to select the informative variable subset. The resulting data set is used to train the standard ϵ-SVR model with RBF kernel. The accuracy of the algorithm is evaluated on the 7 large scale problems, 6 of which are stock keeping unit (SKU) sales volume prediction problems and the other one is a median price of house prediction problem and commonly used to test the accuracy of regression model.

In addition to accuracy evaluation, observations and variables selected by the Row and Column Selection algorithm to see what makes them different than other observations and variables to be selected. In order to see the characteristics of selected rows we explore their geometric positions according to prediction line, constructed in the first stage of the algorithm, target values and input variables. Moreover, selected variables are investigated are checked whether they are independent independence and consistent or not.

This thesis is organized as follows. In section 2, we provide relevant literature, section 3 describes the proposed algorithm. The experiments and results are described in section 4 while section 5 concludes with interpretation of results and future research opportunities.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

Solving business problems with huge amount of data require large memory and time and some algorithms, such as SVM of SVR, is not suitable since the training complexity of SVM is highly dependent on the size of data. In this thesis, in order to facilitate SVR models estimation for large data sets, we develop an algorithm which results in accurate and understandable forecasting models by selecting and using only the most informative data points and variables to construct SVR model. The relevant streams of literature for this problem are SVR, sampling methods, variable selection and specific methods for training large scale SVM and SVR models.

2.2 Support Vector Regression

Suppose that training data is given as $\{(x_1, y_1), (x_2, y_2), \ldots, (x_\ell, y_\ell)\} \subset \aleph \times \Re$, where x_i is a vector and $\aleph$ denotes the space of the input patterns and y_i is the target value for the corresponding observation i. The goal is to find the best approximation function that gives the minimum generalization error. For this purpose, at AT&T Bell Laboratories Vapnik and co-workers [4] developed Support Vector Machines(SVM). In 1996 Vapnik et al. [2] proposed a version of SVM for regression, namely Support Vector Regression(SVR), with the idea of finding the flattest function that has at most ϵ deviation from actual observations y_i for all training data. In other words, SVR locates a tube with radius ϵ around the regression function called the ϵ-tube and does not penalize the errors located inside the ϵ-tube.

In the case of linear functions, the function used in ϵ-SVR for estimation can be represented as follow:

$$f(x) = \langle \omega, x \rangle + b \tag{2.1}$$

with $\omega \in \aleph$, $b \in \Re$ where $\langle ., . \rangle$ denotes the dot product in $\aleph$. As mentioned above, the main idea behind the SVR is to find the flattest function that approximates given data with ϵ precision. In the case of (2.1), *flatness* means a small ω, which can measured by the second norm of ω, i.e. $\|\omega\|^2 = \sqrt{\omega_1 + \omega_2 + \ldots + \omega_d}^2$. This kind of SVR problems are also called as ϵ-insensitive Support Vector Regression (ϵ-SVR) [1] and can be written as convex optimization problem as follows:

$$\begin{aligned} &\min_{\omega} \frac{1}{2} \|\omega\|^2 \\ &\text{s.t.} y_i - \langle \omega, x_i \rangle - b \leq \epsilon \\ &\langle \omega, x_i \rangle + b - y_i \leq \epsilon \end{aligned} \tag{2.2}$$

.

However, in most cases finding a function that has at most ϵ deviation from the actual values for all training data may not be possible and can result with an infeasible solution. Therefore, one can introduce slack variables (ξ_i, ξ_i^*) to cope with infeasibility of this problem and arrive at the formulation stated in Vapnik [5].

$$\begin{aligned} &\min_{\omega,\xi_i,\xi_i^*} \frac{1}{2}\left\|\omega\right\|^2 + C\sum_{i=1}^{\ell}(\xi_i+\xi_i^*) \\ &\text{s.t.} y_i - \langle\omega, x_i\rangle - b \leq \epsilon + \xi_i \ i = 1, 2 \ldots \ell \\ &\langle\omega, x_i\rangle + b - y_i \leq \epsilon + \xi_i^* \ i = 1, 2 \ldots \ell \\ &\xi_i, \xi_i^* \geq 0 \end{aligned} \tag{2.3}$$

The constant C in the above formulation represents the trade-off between the flatness of the model and the amount of tolerated deviations larger than ϵ while ϵ determines the width of the ϵ-insensitive tube. That is to say that, Equation (2.3) disregards the errors if they are less than ϵ but penalizes deviations larger than ϵ.

Smola and Schölkopf [1] states the optimization problem (2.3) can be solved more easily in its dual formulation. Moreover, they show that the dual formulation provides the key for extending SV machine to nonlinear functions. Therefore they use a standard dualization method utilizing Lagrange multipliers, as described in e.g. [6]. In the formulation (2.4) L is the Lagrangian and $(\alpha_i, \alpha_i^*, \eta i, \eta_i^*)$ are Lagrange multipliers of each constraints of Equation 2.3.

$$\begin{aligned} L = \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{\ell}(\xi_i+\xi_i^*) - \sum_{i=1}^{\ell}\alpha_i\left(\epsilon+\xi_i-y_i+\omega^T x_i+b\right) \\ -\sum_{i=1}^{\ell}\alpha_i^*\left(\epsilon+\xi_i^*+y_i+\omega^T x_i+b\right) + \sum_{i=1}^{\ell}(\eta_i\xi+\eta_i^*\xi_i^*) \end{aligned} \tag{2.4}$$

By using the saddle point condition [7], when the primal objective function is minimized and the dual is maximized, the partial derivatives of Equation (2.4) with respect to $(\omega, b, \xi_i, \xi_i^*)$ have to vanish for optimality.

$$\partial_b L = \sum_{i=1}^{m} (\alpha_i^* - \alpha_i) = 0 \tag{2.5}$$

$$\partial_\omega L = \omega - \sum_{i=1}^{m} (\alpha_i^* - \alpha_i)\, x_i = 0 \tag{2.6}$$

$$\partial_{\xi_i} L = C - \alpha_i - \eta_i = 0 \tag{2.7}$$

$$\partial_{\xi_i^*} L = C - \alpha_i^* - \eta_i^* = 0 \tag{2.8}$$

where ξ and $\xi^(*)$ are primal variables and (α and α^*, η and η^* are Lagrangian varaibles. Substituting Equation (2.5) and (2.7) into L yields the following dual problem.

$$\begin{gathered}\max_{\alpha_i, \alpha_i^*} -\frac{1}{2} \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*)(\alpha_i - \alpha_i^*) \langle x_i, x_j \rangle - \epsilon \sum_{i=1}^{\ell} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{\ell} y_i (\alpha_i - \alpha_i^*) \\ \text{s.t.} \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) = 0 \\ 0 \leq \alpha_i, \alpha_i^* \leq C\end{gathered} \tag{2.9}$$

Smola and Schölkopf [1] rewrite Equation (2.6) and reach the following formulation for ω definition. This is the so-called Support Vector expansion, i.e. ω can be completely described as a linear combination of the training patterns x_i.

$$\omega = \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) x_i \tag{2.10}$$

and therefore

$$f(x) = \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b \tag{2.11}$$

According to Karush-Kuhn-Tucker (KKT) ([8], [9]) minima conditions, the product between dual variables and constraints has to vanish at the point of optimal solution [9].

$$\alpha_i \left(\epsilon + \xi_i - y_i + \langle \omega, x_i \rangle \ + b\right) = 0 \tag{2.12}$$

$$\alpha_i^* \left(\epsilon + \xi_i^* + y_i - \langle \omega, x_i \rangle \ - b\right) = 0 \tag{2.13}$$

$$\left(C - \alpha_i\right) \xi_i = 0 \tag{2.14}$$

$$\left(C - \alpha_i^*\right) \xi_i^* = 0 \tag{2.15}$$

By using the equations above, one can obtain the relationship between error term $\xi_i^{(*)}$ and lagrangian multiplier $\alpha_i^{(*)}$.

1. If $\xi_i = 0$ then $\alpha_i < C$ or if $\xi_i^* = 0$ then $\alpha_i^* < C$

2. If $\xi_i > 0$ then $\alpha_i = C$ or if $\xi_i^* > 0$ then $\alpha_i^* = C$

Based on the KKT conditions, the following properties can be learned

1. Based on the Equation (2.12) and (2.13), only for the samples $|f(x_i) - y_i| \geq \epsilon$ the coefficient $(\alpha_i - \alpha_i^*)$ will be nonzero; in other words the values of α_i and α_i^* vanish for all the data points inside the ϵ-tube. Vanished $(\alpha_i - \alpha_i^*)$ brings a sparse expansion of ω in terms of x_i and the data points whose dual variables are non-zero are called *Support Vectors*.

2. The set of dual variables of a given point (α_i, α_i^*) can never be nonzero at the same time, at most only one of them can be nonzero.

3. Samples (x_i, y_i) with corresponding $|\alpha_i - \alpha_i^*| = C$ are support vectors lying outside the ϵ-insensitive tube, while samples (x_i, y_i) with corresponding $0 < |\alpha_i - \alpha_i^*| < C$ lie on the decision boundary.

2.2.1 Nonlinear Support Vector Regression

One of the attractive features of SVR is its ability to model nonlinear relationships by mapping the given data into a high dimensional feature space F via a nonlinear mapping ϕ. An example of mapping in [1] $\phi : \Re^2 \rightarrow \Re^3$ with $\phi\left(x_1, x_2\right) = \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$. Constructing a linear model on the preprocessed features yields a quadratic function in the input space. The function of ϵ-SVR, namely Equation (2.1), takes the following form:

$$f(x) = \langle \omega, \phi\left(x\right) \rangle + b \tag{2.16}$$

where $\langle ., . \rangle$ denotes the dot product in F [1].

One can define a *kernel function*, k, such that $k\left(x_i, x_j\right) = \phi\left(x_i\right) \cdot \phi\left(x_j\right)$, and use k in the training algorithm instead of ϕ. The elegance of using the kernel function is that one can deal with high dimensional feature spaces without having to compute the map ϕ explicitly [10]. Accordingly one can rewrite the problem (2.9) in terms of the dot products in the low dimensional input space.

$$\begin{gathered}\max_{\alpha_i, \alpha_i^*} -\frac{1}{2}\sum_{i=1}^{\ell}\left(\alpha_i - \alpha_i^*\right)\left(\alpha_i - \alpha_i^*\right)k\left(x_i, x_j\right) - \epsilon\sum_{i=1}^{\ell}\left(\alpha_i + \alpha_i^*\right) + \sum_{i=1}^{\ell} y_i\left(\alpha_i - \alpha_i^*\right) \\ \text{s.t.} \sum_{i=1}^{\ell}\left(\alpha_i - \alpha_i^*\right) = 0 \\ 0 \leq \alpha_i, \alpha_i^* \leq C\end{gathered} \tag{2.17}$$

After mapping the input space to high dimensional feature space via kernel function, weight vector and regression estimate then take the form as follows [1]

$$\omega = \sum_{i=1}^{m}\left(\alpha_i - \alpha_i^*\right)\phi\left(x_i\right) \tag{2.18}$$

$$f(x) = \sum_{i=1}^{m} \left(\alpha_i - \alpha_i^*\right) k\left(x_i, x\right) + b \tag{2.19}$$

Note that unlike the linear case in nonlinear SVR, ω can not be expressed explicitly and the algorithm finds the flattest function in the *feature space* rather than input space.

2.2.2 Loss Functions

As shown in the preivous section, the lagrange multipliers (α_i, α_i^*) are often sparse in the (2.3) and related (2.10) formulations, i.e they result in non-zero values only if observations related with these lagrange multipliers are on or outside the boundary. The so called ϵ-insensitive loss function, $|\xi|_\epsilon$, described by [5].

$$|\xi|_\epsilon = \begin{cases} 0 & \text{if } |\xi| \leq \epsilon \\ \xi - \epsilon & \text{otherwise} \end{cases} \tag{2.20}$$

Note that other loss functions, such as the *huber loss* shown in Equation (2.21), can also be used in SVR. On the contrary to ϵ-insensitive, the cost function, using huber loss, has the advantage of not introducing additional bias. However, this cost function sacrifices sparsity in the lagrange multipliers (α, α^*) [11]. Furthermore, for the ϵ-insensitive loss function, SVR problem can be defined as linear programming, which is explained in the section (2.2.3), while the problem still stays quadratic for other loss functions [1].

$$|\xi|_\epsilon = \begin{cases} \frac{1}{2\sigma} & \text{if } |\xi| \leq \sigma \\ \xi - \frac{\sigma}{2} & \text{otherwise} \end{cases} \tag{2.21}$$

The loss functions are shown in Figure 2.1 Huber loss function and ϵ-insensitive loss function are shown in Figure 2.1-(a) and 2.1-(b), respectively. Note that huber loss function contributes a positive penalty to all errors other than zero. These positive penalties gives cause for non-zero ξ and ξ^*, which results in non-zero lagrangian variables (α and α^*). Hence, huber loss function sacrifices sparsity. In order to get a sparse solution Vapnik introduced ϵ-insensitive loss function, shown in Figure 2.1-(b). These benefits of ϵ-insensitive loss function motivate researchers in using ϵ-insensitive loss function in their studies.

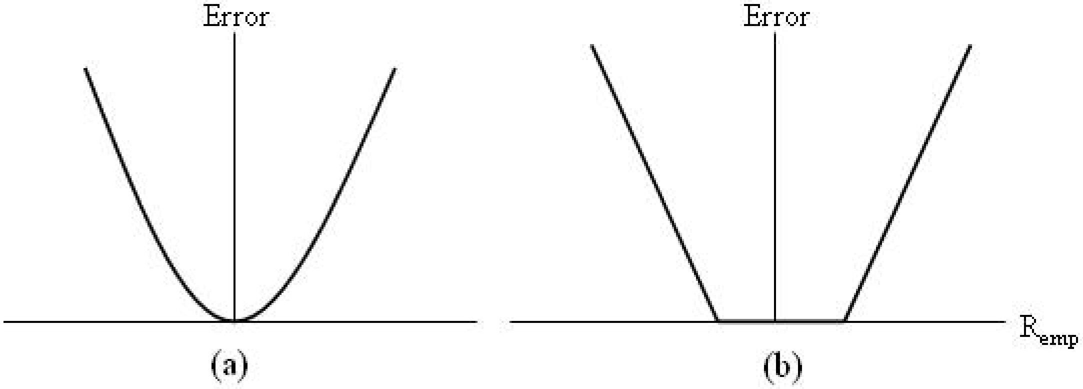

Figure 2.1: Loss Functions in SVM - (a) Huber loss function and (b) ϵ-insensitive loss function

2.2.3 L$_1$-norm Support Vector Regression

The standard SVR formulation, namely L$_2$-norm ϵ-SVR, can give good results for machine learning problems, but since the training time of SVR depends heavily on the training set size, SVR can be computationally expensive for large-scale problems (Schölkopf et. al, 1998). This quadratic programming problem can also be written as a linear program by regularizing with L$_1$-norm [12].

Smola et. al [12] gives more importance to defining ω by using the smallest subset of training patterns than choosing the flattest function as in general SVR. In other words, they control the complexity of the function in a different way by minimizing the sum of the dual variables (α, α^*) instead of ω.

$$\begin{aligned}
&\min_{\alpha,\alpha^*,\xi,\xi^*} \sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\
&\text{s.t. } y_j - \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)k(x_j, x_i) - b \leq \epsilon + \xi_i \\
&\sum_{i=1}^{l}(\alpha_i - \alpha_i^*)k(x_j, x_i) + b - y_j \leq \epsilon + \xi_i^* \\
&\alpha_i, \alpha_i^*, \xi j, \xi_j^* \geq 0
\end{aligned} \tag{2.22}$$

This L$_1$-norm formulation can also be called as sparse SVR because the optimal solution of ω can be represented by using fewer but most informative training examples than in general SVR. Training SVR with only these most important training points can bring the same accuracy level as training SVR with full training set. Note that only for the ϵ-insensitive loss function this leads to a sparse solution and the desired computational advantage [1].

Note that, there is an important difference between L$_2$-norm ϵ-SVR and L$_1$-norm ϵ-SVR in support vector definition. As mentioned in section 2.2, under L$_2$-norm ϵ-SVR the set of points not inside the tube coincides with the set of SVs. However, points lying outside the tube do not necessarily have to be support vectors in LP context. Smola et. al [12] does not explain but states that under the L$_1$-norm any point can be an SV, even if it is inside the tube. The reason behind this can be explained by the difference between the objective functions of these two formulations. L$_2$-norm ϵ-SVR tries to minimize the L$_2$-norm of the ω vector, i.e. the weights of the feature of model function, while L$_1$-norm ϵ-SVR does not minimize the features weights directly, it just tries to reduce the number of training instances that are used to define ω. Because of this property L$_1$-norm ϵ-SVR picks fewer SVs compared to

L_2-norm ϵ-SVR. In this thesis we define SV as points that have non-zero Lagrangian multipliers (α or α^*) which may or may not be inside the tube.

Besides advantages, both models also have an important drawback: they may not be applicable for large-scale problems, since the number of constraints increases as the size of the training set increases. On the other hand, as mentioned above the weight vector, namely the decision function of SVR, depends only on a small subset of training data, called *support vectors*. Therefore, removing the data points, which are irrelevant to the final decision function, does not affect the accuracy of the prediction model [13]. Wang et al. [13] select training data for SVM classification and state that if the support vectors of a data set are known, then one can obtain the same prediction function by solving a much smaller problem.

2.2.4 ϵ-insensitive Linear Regression

It is often beneficial to use a small subset of the available variables even if a large set of variables is available. The reason behind this is clear, training data sets that have a small subset of variables require much smaller memory and time complexity. Furthermore, sometimes obtaining variables can be costly, time consuming and the estimation function which uses a large number of variables can be incomprehensible. Moreover, data with large set of variables can include noise. Therefore, models with a small number of variables is generally preferable than models depending on a large number of variables.

LASSO [14], which uses L_1-norm regularization of the variable weights on the training, is often an effective technique for shrinkage and variable selection. Using L_1-norm regularizer leads to a sparse solution in the variable space, which means that the regression coefficients for most irrelevant or redundant variables are shrunk to zero.

$$\begin{aligned}\left(\hat{\alpha},\hat{\beta}\right) = \operatorname{argmin}\left\{\sum_{i=1}^{N}\left(y_i-\alpha-\sum_j \beta_j x_{ij}\right)^2\right\} \\ \text{s.t.}\sum_j \beta_j \leq t\end{aligned} \tag{2.23}$$

where $t \geq 0$ is a tuning parameter, β_i is the weights of ith input variable. And the solution for α is $\widehat{\alpha}=\overline{y}$ therefore without loss of generality that y=0 and hence α can be omitted [14]. Bi et. al. [15] used L$_1$-norm regularization to select variables in a linear model before constructing a nonlinear model. Their objective function penalizes the weights of the variables, with an L$_1$-norm regularization which results in a sparse variable set. Moreover, the epsilon insensitive loss function used in the model proposed in [15] provides additional robustness in the face of noise, resulting in fewer variables. The following formulation shows the ϵ-insensitive linear regression [1] with L$_1$-norm regularization.

$$\begin{aligned}\min_{\omega,\xi_i,\xi_i^*} \lambda\sum_{d=1}^{D}\|\omega_d\| + \frac{1}{\ell}\sum_{i=1}^{\ell}(\xi_i+\xi_i^*) \\ \text{s.t.}y_i-\langle\omega,x_i\rangle-b \leq \epsilon+\xi_i \\ \langle\omega,x_i\rangle+b-y_i \leq \epsilon+\xi_i^* \\ \xi_i,\xi_i^* \geq 0\end{aligned} \tag{2.24}$$

2.3 Sampling Methods

The typical practitioner approach for applying SVR to large data sets is to use a smaller random subsample for training. In random sampling, each instance of the data set has the same probability of being selected - with or without replacement. The main drawback of random sampling is that it may not include some of the important observations for estimation or it may give rise to a skewed sample. Stratified sampling ensures that the sample distribution for the stratification variable(s) is the same as the distribution in original data set. But it is not very practical since it requires the analyst to specify the important variables. Further when using multiple stratification variables data becomes sparse and the number of observations for in high dimensions vanishes.

Imbalanced data sets in terms of the output variable also pose a problem. For the regression problem imbalance can present itself in the form of very few observations in particular ranges of the target variable. The performance of machine learning algorithms including the SVM drops significantly when the data set is imbalanced [16], for example, the classification algorithm simply never predicts a class with very few training observations. Two popular approaches in classification to solve the imbalance problem are random oversampling of the minority class or undersampling of majority class to obtain a balanced data set. A potential problem with undersampling is deletion of highly informative data points [17], while oversampling can cause the machine learning algorithm to learn these observations by heart leading to over fitting [18]. Similar issues are present in regression problems. For example, building prediction models on imbalanced data set can result in getting the solutions that perform well only on over-represented areas of the input space. In other words, regression models constructed with imbalanced data sets can try to make accurate prediction for most existing target values while ignoring the others and regard them as outliers.

Researchers use different methodologies to identify important points for training the SVM model. Based on the fact that the prediction function only depends on the points called support vector they try to identify good candidates for SVR. Shin and Cho [19] focused on chosing the data points, which may be located near the decision boundary, based on the neighborhood properties to reduce the size of training set, which may become impractical for problems with high dimensionality. Wang et al. [13] propose two new data selection methods by using the same idea. To select data points near the decision boundary, their first method selects data points based on a statistical confidence measure, which is associated with the number of training examples that fall inside the largest possible sphere drawn centered at each training example without covering a training example of a different class, while the second method uses distance based criterion that calculates the minimal distance from a training data to another training data, which belongs to a different class. As a result of their study, they find that random sampling also performs very well when the data reduction is high, in other words, when the number of support vectors is considerably less than the data set size. These sampling methods can be assumed as related with active learning which is a subfield of artificial intelligence and actively choose the training data. The main motivation of active learning is that labeling an example requires time and wasting resources on non-informative samples is useless for learning algorithm. Therefore, the key point of active learning is chosing the data from which alogrithm learns and will perform better with less training [20].

2.4 Variable Selection

The success of machine learning algorithms on a given task can be affected by the size and relevance of the variable set. In fact, data sets with a large number of variables should result in models that fit training data better. However, in the presence of irrelevant and redundant information and noisy output functions, which is the case in business data sets, learning algorithm may perform poorly. Because of this, selection of relevant variables and reducing the dimensionality of data sets has become a challenging research topic, as many of data sets contain hundreds or thousands of variables, many of which are often redundant or irrelevant [21].

The L_2-norm regularization of the ω vector in ϵ-SVR, mentioned in the Problem (2.3), ensures flatness but does not result in selection of variables, i.e. a sparse solution in terms of weights (ω). Thus, reducing the dimensionality by using a variable selection algorithm may improve learning ability of ϵ-SVR. variable selection, which is used as a preprocessing step of machine learning algorithms, has also been very effective in reducing the computational time of the learning algorithm and improving result comprehensibility [21]. Therefore, the given data should be dimensionally reduced by eliminating irrelevant and redundant data before consturcting the estimation model by solving the Problem (2.3).

Guyon and Elisseeff [21] provide an overview of the variable selection methods and classify them into three groups: filter, wrapper and embedded methods. The filter selection method was the earliest approach to variable selection. It is defined as a preprocessing step to induction that can remove irrelevant variables before training occurs. It utilizes an independent search criterion to find the appropriate variable subset before a machine learning algorithm is used [22]. The advantage of the filter model is that it does not need to re-run the algorithm for every training algorithm when choosing to run on a reduced variable data set, as a consequence, the filter approach is generally computational efficient, and practical for data sets with very high dimensionality. However, since the filter approach reduces the dimensionality before a machine learning algorithm is performed, it does not take into account the learning bias introduced by the learning algorithm. Therefore, filter selection method may not be able to select the most suitable subset for the final learning algorithm. For this reason, the wrapper model was proposed [23].

The strategy of the wrapper model is to search through the space of variable subsets by using the estimated accuracy from particular training algorithm as the measure of goodness for a variable subset. Thus, the relevance measure is directly defined from the learning algorithm. When compared to the filter methods, wrapper approaches often have better results than the filter approaches because they are tuned to the specific interaction between an induction algorithm and its training data [23]. In this way, variable selection takes into account the biases from the final learning algorithm. However, the major disadvantage of wrapper methods over filter methods is the computational time, which results from training the induction algorithm for each variable set considered.

Embedded methods differ from filter and wrapper methods in the way variable selection and learning interact. In contrast to the wrapper approach, the embedded approach embeds the selection within the basic training algorithm. Embedded methods select variables during training and are algorithm specific while, in filter and wrapper methods the learning part and the variable selection part are separated; therefore, the variable selection method can be combined with any learning machine. Least Absolute Shrinkage and Selection Operator (LASSO) method [14], which minimizes the sum of squared errors subject to a bound on the sum of absolute values of coefficients, falls in this class. In this study we use an embeded method for variable selection, which penalizes the weights of the variables with the regularization by L_1-norm and results in a sparse variable set.

Bierman and Steel [24] consider the problem of variable selection for SVMs. In their paper, they state that the classification accuracy of SVMs can be substantially improved if a smaller subset of variables is use instead of all variables. For this purpose, they proposed a two-step approach to variable selection for SVMs. During the first step best variable subsets corresponding to each possible value of the number of variables of the dataset are identified. These subsets are determined by using backward selection strategy as a search method through the different subsets of variables and variation ratio as criterion, which is a function of the kernel matrix and used to decide on an optimal subset of a given size. Then in the second stage one of the previously identified subsets is chosen as final selection by considering the number of support vectors since it is an upper bound on the expected probability of training data error of the SVM. Experimental studies of this paper proves that variable selection is very much worthwhile for SVMs by demonstrating the accuracy of the no column selection model is inferior to the model which is constructed with selected columns. The improvement in test error is especially significant in cases with many variables and small samples.

2.5 Previous Methods for Training SVR with Large Data

Support vector regression has empirically been shown to give good generalization performance on a wide variety of problems. However, the use of SVR is limited since time and memory requirement is high for training SVR with very large data. In order to make large scale problems solvable by using optimization methods Vapnik [25] suggested to break up of the problem into subproblems and then solve each of them separately. The idea behind chunking approach is based on the fact that only the support vectors play role in the SVR estimation. Therefore, knowing the support vectors, one could directly deal with very large data sets. However, the set of support vectors of a given data are not known beforehand and support vectors corresponding to non-zero lagrangian variables can only be observed only after training SVR for the given data sets.

Chunking approach divides problem into subproblems by selecting a subset of the constraints corresponding to observations of the problem. After training SVR on this first chunk of constraints is complete, all non-support vectors are discarded and the next chunk is created. This chunk contains all support vectors, patterns that violate the Kuhn-Tucker conditions, of the previous phase and additionally some new observations selected from the data. The chunking algorithm terminates when the training set only contains support vectors. Thus, for very large data sets chunking can be used efficiently if only the number of support vectors is small. But when the data set has a high noise problem, many of the slack variables ξ_i and ξ_i^* become nonzero and all the corresponding examples become SVs. In this case, chunking approach is not useful and does not bring any computational advantages; thus, decomposition algorithms [26] were proposed.

For the case of pattern recognition, Osuna et al.[26] proposed a decomposition algorithm and in 1998 [27] this decomposition algorithm is extended for SVR. The key idea of decomposition is to divide original variables into working and fixed variables. By keeping only working variables, decomposition approach optimizes a sequence of constant sized problems iteratively. The value of the objective function is improved at each iteration and the algorithm is stopped when termination criteria are met. That is to say that, the convergence is also guaranteed for the decomposition algorithm as in chunking approach. However, in contrast to chunking approach, the decomposition algorithm operates on a working set of constant size. It starts with an arbitrary subset of training patterns and solves the subproblem, while keeping the lagrangians of all other patterns constant. As long as the Kuhn-Tucker conditions are violated by at least one sample x_j from the remaining set, an arbitrary sample is chosen, both are interchanged, and the new subproblem is solved. With this method even problems with many thousands of support vectors can be handled.

By using the decomposition approach Platt [28] proposed the Sequential Minimal Optimization(SMO) algorithm. Unlike the other methods, SMO chooses the smallest optimization problem to solve, which involves two Lagrange multipliers, at every step and finds the optimal values for these multipliers. The advantage of SMO lies in the fact that one can easliy solve this simple two-variable problem without using any optimization software. However SMO is not designed for SVR. Because of this based on an idea Osuna et al. [29] and Platt [28], SVMTorch algorithm is introduced by Bengio and Collobert [3] to solve large scale problems by training SVR. In every iteration of SVMTorch a small subset of variables is selected as working set and the problem is solved on this working set. Furthermore, SVMTorch uses a shrinking phase to exclude variables that are stuck to 0 or C so that these variables will probably not change anymore. These variables can be removed from the optimization problem such that a more efficient overall optimization is obtained. However SVMTorch can not provide optimal solution if the working set size is not equal to two [3].

Chapter 3

ROW AND COLUMN ALGORITHM

3.1 Introduction

In this chapter we propose a new algorithm, Row and Column Selection Algorithm, to select important observations and variables of large business data sets in order to train SVR models. The proposed algorithm uses two steps: 1) row selection, 2) column selection. The first step, row selection, aims to select the most informative points of the large data set to reach a small but informative sub-sample to facilitate training a support vector regression model generalizing over the whole data set. Even though, once the kernel matrix is calculated, the support vector regression computation time and memory requirements will not be affected by the number of columns in the training set the accuracy of predictive model can still be affected, as a large number of redundant or irrelevant variables may result in over fitting. Therefore, the second step of the algorithm aims to select an informative set of variables for training the final SVR model. The first and the second step of the proposed algorithm can be thought as a sampling method and as a variable selection, respectively.

3.2 Row Selection

In the row selection step of the proposed algorithm, a chunking method, which divides the data set into small enough subsets that fit into memory, is used. The idea behind chunking approach that only the support vectors play a role in the SVR estimation.

In other words, if any of the other observations, which are not support vectors, are removed from the training set, the SVR solution will be exactly the same. Hence, to obtain the set of support vectors of large data set, we divide the given data sets into a number of subproblems by chunking it on the rows and train each suproblem with L$_1$-norm SVR to obtain a small set of support vectors. The support vectors of these subproblems constitute the final training data, which will be used in the second part of proposed algorithm.

To ensure that the data chunks are representative of the distribution of the main driver of the output we use stratified sampling. For example, in the case of SKU sales prediction in the presence of promotions, the major driver that is used for stratification indicates whether the SKU itself was on discount.

Then by using the L$_1$-norm ϵ-SVR with RBF kernel function, which is introduced by Smola et. al [12], support vectors of each chunk is determined and kept while other observations are discarded. The resulting support vectors from each chunk collectively form the observations of the new training data set.

$$\begin{aligned}
&\min_{\alpha,\alpha^*,\xi,\xi^*,b} \sum_{i=1}^{l}(\alpha_i+\alpha_i^*)+C_2\sum_{i=1}^{l}(\xi_i+\xi_i^*)\\
&\text{s.t. } y_j-\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)k(x_j,x_i)-b\le\epsilon+\xi_i\\
&\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)k(x_j,x_i)+b-y_j\le\epsilon+\xi_i^*\\
&\alpha_i,\alpha_i^*,\xi j,\xi_j^*\ge 0
\end{aligned} \tag{3.1}$$

Equation 3.1. *L_1-norm ϵ-SVR*

where $k\left(x_i,x_j\right)=-\gamma\left\|x_i-x_j\right\|^2$ and $\gamma=1/d$.

Note that, this ϵ-SVR formulation controls the complexity of the function by using dual variables instead of ω. Further, L_1-norm regularization is applied, which has the effect of picking a small subset of training patterns, rather than the L_2-norm which ensures flatness in terms of small coefficients [30]. Thus, L_1-norm formulation can also be called as sparse SVR because the optimal solution of ω can be represented by using fewer but most informative training examples than in standard SVR [12]. No constraint on α's trying to set them to C, therefore any point regardless of their location with respect to the ϵ-tube can become a support vector.

3.3 Column Selection

In the second step of the algorithm, we use the L_1-norm regularized ϵ-insensitive linear regression mentioned in the section (2.2.4) on the reduced data set to select an informative subset of variables. Following formulation shows he L_1-norm regularized ϵ-insensitive linear regression.

$$\begin{aligned}\min_{\omega,\xi_i,\xi_i^*,b} \lambda \sum_{d=1}^{D} \|\omega_d\| + \sum_{i=1}^{\ell} \frac{1}{\ell} (\xi_i + \xi_i^*) \\ \text{s.t.} y_i - \langle \omega, x_i \rangle - b \leq \epsilon + \xi_i \\ \langle \omega, x_i \rangle + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0\end{aligned} \tag{3.2}$$

Equation 3.2. *ϵ-insensitive Linear Regression*

Training selected points with L_1-norm regularized ϵ-insensitive linear regression results in a small but informative variable subset by eliminating redundant and irrelevant variables. The final training data set then consists of the data points identified in row selection part and the variables with nonzero weights in column selection part. Note that, L_1-norm often produce zero coefficients for variables. Figure 3.1 provides the estimation picture for L_1-norm and L_2-norm regression. The square and the circle are the constraints regions of L_1-norm and L_2-norm respectively, while the ellipses are the contours of the least square error functions (β). The picture of L_2-norm has no corners hence zero solution will rarely results while the solution of L_1-norm touches the square and lead to zero coefficient for variables [14].

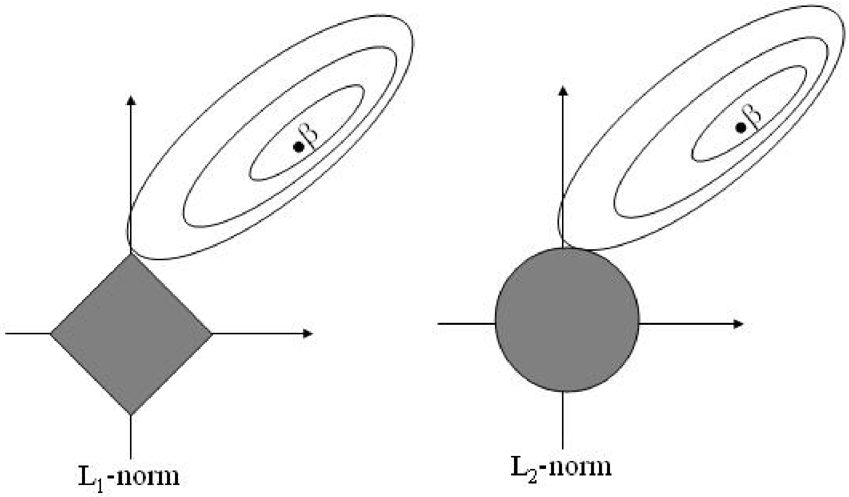

Figure 3.1: Estimation picture of L_1-norm(left) and L_2-norm(right)

3.4 Model Parameter Selection Procedure

Notice that in both steps of the training data set selection algorithm, we use the same ϵ insensitive loss function as in the (ϵ-SVR) model (2) that we want to build. However, instead of using the L_2-norm regularization of the standard SVR, we use the L_1-norm regularization to facilitate selection of rows and columns with formulations that penalize the support vector (Row Selection SVR), and variable weights (Column Selection SVR) respectively.

There are several parameters in our algorithm whose selection is an important issue for ϵ-SVR because the accuracy of SVR models can be drastically affected by the choice of model parameters. Parameter C_2, C and λ control the trade-off between the training error and the model complexity in standard ϵ-SVR model, row selection step and column selection step, respectively. Parameter ϵ controls the width of the ϵtube, used for training data. Poor choice of these parameters may result in a model with an inferior accuracy level; therefore in order to obtain good generalization, it is necessary to use a proper setting of model parameters.

3.4.1 Selecting Loss Function Parameter (ϵ)

Cherkassy and Mulier [31] and Scholkopf et al. [32] used cross-validation for parameter selection procedure, but this approach is not found efficient because of high computational time requirements. For determining ϵ, a more efficient approach is proposed by Scholkopf et al. [12] in 1998. They proposed a variant of the SVR algorithm called ν-SVR, which determines ϵ automatically by using another user defined parameter ν. They showed that, in regular SVR case, ν represents an upper bound on the fraction of errors and lower bound on the fraction of points outside the ϵ-tube, namely support vectors. Hence, using ν-SVR may be useful if prior knowledge about what percent of data points will be support vectors is available. From the statistical literature it is known that noise i.e. variation in the target values which is unpredictable from the input data limits the accuracy of the learning algorithm. Therefore, preventing the model from fitting the noise it is essential to cope with the noise by identifying the optimal value of ϵ for the data set. By considering this idea, Kwok [33] and Smola et al [34] proposed that asymptotically optimal ϵ values have a linear relationship with the noise in the data. However, the main drawback of their approach is that it does not reflect the sample size and that higher sample size should be associated with smaller ϵ. From the previous studies, it is known that ϵ should be proportional to the input noise level ([33], [34]). Cherkassky and Ma [30] claimed that the value of ϵ depends on the standard deviation of the noise and the size of training data, as follows

$$\epsilon = 3\sigma\sqrt{\frac{ln(n)}{n}} \quad \text{where} \quad \sigma^2 = \frac{n}{n-d}\frac{1}{n}\sum_{i}^{n} = 1\,(y_i - \hat{y}_i) \tag{3.3}$$

In the equation above $\hat{y}_i$ is the estimate based on the *low bias model*, which is operationalized as nearest neighbor or one variable regression, d is the degrees of freedom and n is the size of training data. By following this idea, we set the ϵ value as follow

$$\epsilon = \frac{1}{10}\sigma \tag{3.4}$$

3.4.2 Selecting Parameter of Row Selection Step

Tradeoff parameter of row selection part of our algorithm determines a tradeoff between the ϵ insensitive error and the weights of selected points. The most proper value for C_2 parameter is established starting with 0, increasing by 0.5 at each step, and comparing mean absolute error of validation set with the value from previous step. If the reduction percentage in error is less than 0.001, the line search is terminated and the C_2 value is set as one that gives less than 0.1% reduction in error. The following algorithm show steps of Row Selection Step:

1. Take the first chunk of the data set and set C_2=0, i=1 and MAE$_0 = \infty$

2. Solve the problem by using the "'Row Selection SVR formulation"' to construct the prediction model

3. Test the model on the validation set and calculate the MAE and compute im-

provement by using the following formulation.

$$improvement = \frac{MAE_i - MAE_{i-1}}{MAE_{i-1}}$$

4. If improvement $\geq$ -0.001 then stop, else increase C_2 by 0.5 and go to step 3.

3.4.3 Selecting Parameter of Column Selection Step

In the column selection part of our algorithm λ is used for determining the tradeoff between the L$_1$-norm of variable weights and the sum of ϵ insensitive absolute errors. We use the line search method to set λ. We start with a λ value that is high enough to give rise to 0 variables and decrease it gradually. Decreasing the value of λ starts to penalize variable weights less; therefore the number of selected variables is increased with respect to the previous step. Larger variable sets generally bring better accuracy level for training set but cause over fitting and reduce accuracy level for the test set.

In order to get the most variable subset we generate a criterion that depends on the random variables and can be thought as a similar approach as Bi et al [15]. Before starting line search for the value of λ, we augment the training set with random variables from different distributions that are independently generated of the response variable and other variables. Specifically, we add $[0.1\times d]$, where d represents the size of input space, random noise variables to determine stopping point of the grid search. The value of λ is decreased by $e^{-0.25}$ at each step and a new variable subset is constructed until a random variable is chosen. If one of these random variables is selected then this means that the model starts to select irrelevant variables and the tradeoff parameter (λ) is too small to control the selected variables. In other words, decreasing the value of tradeoff parameter (λ) beyond the value that selects the random variable starts to select irrelevant variables.

3.4.4 Selecting Parameter of standard ϵ-insensitive SVR

For determining tradeoff parameter used in the standard SVM formulation in classification, Mattera and Haykin [35] set the C value to the range of output values of the training data. However, such a selection of C does not take into account possible effect of outliers in the training data therefore this approach may not work well for the data with outliers. By considering these studies Cherkassky and Ma [30] proposed the following equation for parameter C:

$$max\left(\left|\bar{y}-3\sigma_y\right|,\left|\bar{y}+3\sigma_y\right|\right) \tag{3.5}$$

where $\bar{y}$ and σ_y are the mean and the standard deviation of the y values, namely target values of the training data. In this thesis we use Equation (3.5) to determine C value of Equation (3.6), which is the formulation of non-linear support vector regression. [5].

$$\begin{aligned} &\min_{\omega,\xi_i,\xi_i^*} \frac{1}{2}\left\|\omega\right\|^2 + C\sum_{i=1}^{\ell}\left(\xi_i+\xi_i^*\right) \\ &\text{s.t.} y_i - \langle\omega,\phi_i\rangle - b \leq \epsilon + \xi_i \; i = 1,2\ldots\ell \\ &\langle\omega,\phi_i\rangle + b - y_i \leq \epsilon + \xi_i^* \; i = 1,2\ldots\ell \\ &\xi_i,\xi_i^* \geq 0 \end{aligned} \tag{3.6}$$

Chapter 4

EXPERIMENTS AND RESULTS

4.1 Introduction

In this chapter we evaluate the Row and Column selection algorithm performance on three different problems which have large data sets with thousands of observations and hundreds of variables. We evaluate the impact of each step of algorithm on the accuracy and bias of the RBF-SVR model. The bias of proposed algorithm is calculated as Mean Error (ME) while the accuracy is calculated as Mean Absolute Error (MAE). Both the accuracy and bias are compared with an alternative training sample selection method, random sampling, and the SVM-Torch [3] method, which is introduced to train large scale problems.

We analyzed rows and columns selected by the algorithm to see what makes them different than other rows or columns to be selected. In order to see the characteristics of selected rows we explore their geometric positions according to prediction line, constructed in the first stage of the algorithm, target values and input variables.

From section 2.2.3 we know that there is a difference between L_1-norm and L_2-norm in SV definition; therefore, to see this difference we also compare the selected points with the SVs of L_2-norm ϵ-SVR and make independency test to check whether the probability of being selected in L_1-norm is independent from the probability of being SVs by L_2-norm. Lastly, selected columns are checked for independence and consistency.

In this study we use three different softwares. The statistical software SAS 9.1.3

is used for the analysis and preparation of data sets and as well as the random variables and subsamples generation. Row selection problem is coded and solved under MATLAB R2008a software while BMRM [36] is used for the column selection stage of the Row and Column Selection algorithm. Construction of final RBF-SVR model is trained by using LibSVM via MATLAB R2008a software. Because of the heuristic methods, BMRM and LibSVM, used in the steps of algorithm, Row and Column Selection algorithm is also a heuristic method.

4.2 Data Sets

We use 7 different data sets from three domains with thousands of observations and hundreds of variables to evaluate the impact of the proposed training data selection algorithm on the accuracy of the RBF-SVR model. The first two data sets come from different countries' grocery stores and are used for SKU sales prediction for a product category and the last data set is dealing with house price prediction. The grocery data sets have three maingroups, which can be further used for consistency check within each grocery data set.

The first data comes from the leading grocery store chain in Turkey and includes daily sales, price and promotion information of SKUs in the black tea category in 5 different stores from September 6, 2006 to September 20, 2008. Following the work by Gür Ali et al [37] 116 variables are created that describe the current prices and promotions for the SKUs in the particular store; historical sales, promotion and price statistics for the best selling namely focal SKU, and others in the store, along with SKU and store characteristics, and seasonality variables. We use the first 18 months of the data as the training data, and remaining 7 months as the test data with exception of the randomly sampled 1000 observations that were used as validation set to set parameters. Separate models are constructed for the 3 maingroups in the black tea category (loose tea, teabags, and bags for teapots respectively) to evaluate the algorithm performance.

The second set of data is provided by the Information Resource, Inc. (IRI) and deals with a large scale SKU sales volume prediction problem. The data set covers 6 years of weekly data and includes weekly sales, price and promotion information for each SKU (i.e., SKU denoted by unique universal product code (UPC)). We focus on the coffee category in 38 stores in the Chicago area. We use the second through the fourth years of the data as the training data set, and the fifth as the test data after setting aside 1000 observations for validation set. Moreover, we divide coffee data into 3 maingroups (ground caffeinated, ground decaf and coffee bean) and construct models for each of the 3 maingroups independently to evaluate the algorithm performance.

The last data set that we used to test our proposed algorithm accuracy is Census House Data set, which is constructed from the 1990 US Census and for predicting the median house price in a small survey region, is obtained from website of Data for Evaluating Learning in Valid Experiments (Delve). Census-house data set has 137 variables which mainly represent information about the people who live in that region. We set aside 1000 observations as validation set and randomly divided the remaining data to training and test data set to give roughly a 2:1 ratio. The importance of this public data set is that it is a standard regression test problem and used in some other studies such as [38],[39],[40].

Table 4.1 provides the number of observations and the average of target value for each data set. As it can be seen from the table, our data sets have thousands of rows and hundreds of columns. As data preparation, we standardized the variable of each data set by subtracting the mean and dividing by the standard deviation of the variable. Each data set was standardized with its specific mean and standard deviation values. The target of grocery data set is set to a multiple of its historical

Table 4.1: Data Set Properties

	data set	# of rows	# columns	Average of target value
Train	Grocery-1	71729	116	1.006
	Grocery-2	36733	116	0.9954
	Grocery-3	31346	116	1.022
	IRI-1	36000	131	1.0015
	IRI-2	15777	131	0.9896
	IRI-3	28071	131	0.9845
	Census-House	14000	137	-0.0054
Test	Grocery-1	33520	116	0.9867
	Grocery-2	17034	116	0.9446
	Grocery-3	12525	116	0.9701
	IRI-1	47155	131	1.0246
	IRI-2	9685	131	0.9921
	IRI-3	19415	131	1.0023
	Census-House	7784	137	0.0051

average sales to remove the effects best selling SKUs and the target of Census-House data is also standardized.

4.3 Experimental Setup

For each data set and maingroup, we train the ϵ-SVR model with RBF kernel on three different training data sets as follows:

1. Random sampling of rows with full variable set, which constitutes the benchmark (RR Model)

2. Selected rows based on step 1 of the algorithm, with full variable set (SR Model)

3. Selected rows and columns as a result of applying steps 1 and 2 of the Row and Column Selection Algorithm (SRSC Model)

For the benchmark (RR Model) we randomly sample as many rows from the original data as the number of selected rows using the Row and Column Selection Algorithm to ensure a fair comparison in terms of the training sample size.

We also train our data sets with SVMTorch [3], and compare the accuracy of Row and Column Selection Algorithm with the results of SVMTorch[3]. Moreover, we use mean error (ME) and mean absolute error (MAE) to make comparison between the accuracy of proposed algorithm and benchmark methods.

$$\text{ME} = \frac{\sum_{i=1}^{n}(\text{actual-predicted})}{n} \tag{4.1}$$

$$\text{MAE} = \frac{\sum_{i=1}^{n}|\text{actual-predicted}|}{n} \tag{4.2}$$

4.4 Parameters of Algorithm

Table 4.2 shows the value of parameters used in the Row and Column Selection Algorithm for SR and SRSC Models and the value of parameter C of RR Model.

Table 4.2: Parameters used in the models

	ϵ	C_2	λ	C	C for RR Model
Grocery-1	0.1234	1.25	0.0143	7.6544	5.0606
Grocery-2	0.1044	1	0.0183	6.6613	5.0776
Grocery-3	0.1067	1	0.0087	6.8686	5.115
IRI1	0.0761	1.5	0.0235	3.4175	3.521
IRI2	0.0672	2	0.0639	3.1167	3.4061
IRI3	0.0692	2	0.0639	3.0871	3.1964
Census-House	0.017	1.5	0.0087	3.084	3.105

Following the algorithm procedure in section 3.4 within the IRI and grocery data sets the parameters are similar across maingroups except for λ. Note that, for RR Model we use the same ϵ value as in SR and SRSC Models but the value of C used in the RBF-SVR model differs since RR Model has different observation than SR and SRSC Models.

A big driver of difference is the frequency of the observations while the IRI data provide weekly observations, in the grocery data they are daily. Looking at the value of ϵ, we observe that grocery data is more noisy than IRI data. Interestingly, when the errors are assumed to be independent, the daily versus weekly data would imply a factor of $\sqrt{7}$ for the noise. The factor is slightly less than $\sqrt{7}$. We can not make comparison with Census-House data since its target value is different than grocery and IRI data. Table4.3 shows the number of iterations of selection of parameter C and λ and the number of chunks in each data set.

Table 4.3: Complexity of Row and Column Selection Algorithm

	#of chunks	# of iteration for setting up parameter C	# of iteration for setting up parameter λ
Grocery-1	36	5	13
Grocery-2	19	4	12
Grocery-3	16	4	15
IRI-1	18	6	11
IRI-2	8	7	7
IRI-3	14	7	7
Census-House	7	6	15

4.5 Resulting Training Data

Table 4.4 shows the number of selected rows and columns by the methods I through III, described in the previous section, for each data set. Row and Column Selection Algorithm provides big reduction both for rows and columns. The algorithm retains between 3% and 26% of the observations and 5% and 17% of the columns. Such a big reduction in columns makes the prediction model clearer and more understandable. Reduction in rows makes training the standard SVR easy to implement and saves time and memory space since just the small amount of data is required.

Table 4.4: Number of selected rows (observations) and columns (variables) in the training set

	Row Selection			Column Selection		
	Training Data Size	# of Selected Rows	% of selected Rows	# of Variables	# of Selected Columns	% of selected Columns
Grocery-1	71729	13371	19%	116	16	14%
Grocery-2	36733	5605	15%	116	17	15%
Grocery-3	32346	7044	22%	116	19	16%
IRI 1	36000	4046	11%	131	22	17%
IRI 2	15777	1309	8%	131	10	8%
IRI 3	28071	877	3%	131	14	11%
Census-house	16000	4118	26%	137	7	5%

4.6 Accuracy and Bias

Table 4.5 reports the test set MAE values associated with the training data set selection method. This experiment demonstrates that selecting data points by using Row Selection SVR produces more accurate predictions than the training data set generated by randomly selected data points RR Model. This is evidenced by the reduction in the MAEs by up to 48%. Moreover, comparison to SVMTorch [3], which uses all data points to construct a prediction model, shows SR Model produces much more accurate predictions in four data sets out of seven and is better than RR Model according to average prediction accuracy level. The results of the other 3 data sets are still comparable with SVMTorch [3], but slightly worse. Therefore, we can conclude that selecting data points from a large data set works well and generally produce more accurate predictions than using the whole training data set or randomly selected data points.

Table 4.5: The test MAE associated with the training data set selection methods and the benchmark

	RR Model	SR Model	SRSC Model	SVMTorch
Grocery-1	1.2964	0.6724	0.5552	0.6111
Grocery-2	0.5778	0.3327	0.29	0.5458
Grocery-3	0.3765	0.2362	0.1909	0.5477
IRI 1	0.6049	0.5377	0.6795	0.482
IRI 2	0.1115	0.1038	0.1178	0.4961
IRI 3	0.201	0.1983	0.2254	0.4613
Census-house	0.1002	0.1016	0.0168	0.0759
Average	0.467	0.312	0.297	0.46

Table 4.6 shows the significance values of the paired sample Z-test for the differences between the test MAEs associated with the SR and SRSC models and the benchmark models, which are RR Model and SVMTorch for each maingroup. Accordingly, the proposed SR model has significantly better holdout accuracy performance than the benchmark methods RR Model and SVMTorch in all maingroups of Grocery and IRI data sets. However for Census House, while SRSC Model outperforms SVM-Torch, pairwise z-test between SR Model and the RR Model indicates that difference is not significantly different from 0 since the p-value is greater than significance level,

which is equal 0.05.

Table 4.6: Significance level for the paired sample z-tests on the mean difference for different models

	Random vs SR	SR vs SRSC	SRSC vs SVMTorch
Grocery 1	0	0	0
Grocery 2	0	0	0
Grocery 3	0	0	0
IRI 1	0	1	0
IRI 2	0	1	0
IRI 3	0.0188	1	0
Census-House	0.9101**	0	0

The second experiment, which is done by using the model SR Model and SRSC Model, illustrates the effects of Column Selection SVR, namely variable selection. According to the test MAE, eliminating redundant columns and construct the model based on the selected columns results in more accurate models for Grocery and Census-House data sets. This experimental result demonstrates that the existence of redundant variables can decrease the predictive power. On the other hand, for the IRI data set models with weekly basis observations and long training data time frame eliminating variables does not improve the accuracy of prediction model. The SRSC model dramatically reduces the number of variables used in the model between 3% and 17% of the original size. This is an important operational gain for the MIS operations considering the costs of maintaining the data. However, contrary to Grocery and Census-House data sets findings, omitting variables in IRI data set reduces the predictive power of the training set. Therefore, in the presence of redundant variables Column Selection SVR works effectively and reduces the number of variables dramatically while improving the generalization error.

Table 4.7 reports the test ME associated with the training data set selection

Table 4.7: The test ME associated with the training data set selection methods and the benchmark

	RR Model	SR Model	SRSC Model	SVMTorch
Grocery-1	0.0355	-0.1421	-0.0724	0.2626
Grocery-2	0.02	-0.1128	-0.1031	0.1306
Grocery-3	-0.0403	-0.0259	0.0172	0.1156
IRI 1	-0.0139	-0.0234	0.234	0.1071
IRI 2	-0.0015	0.0128	0.0254	0.0792
IRI 3	-0.0008	0.002	0.0913	-0.0065
Census-house	0.0007	0	-0.0023	0.0018
Average	0	-0.042	0.027	0.099

method. The random rows method RR Model provides better or similar unbiasedness compared to all other methods, which is not surprising as the L_1-norm regularization is known to introduce bias. However, note that for Grocery and Census-House data sets MEs of SRSC Model and for IRI data sets MEs of SR Model are adequate, as all MEs are smaller than ϵ, which was deemed an admissible error. Therefore, by considering the experiments related with MAE, both SR Model and SRSC Model are assumed to be more accurate than RR Model. In addition to these, Row and Column Selection Algorithm outperforms SVMTorch when compared to average value of ME among 7 data sets.

4.7 Characteristics of selected row and columns

4.7.1 Selected Rows

Next, we investigate how the rows selected by the algorithm differ from others according to (1) distance of the training observations from the predicted value, (2) target values, (3) SVs in L_2-norm case and (4) the effects of variables. First of all, in order to see geometric position of selected points, we calculate the distance between actual values of them and predicted values which comes from the first part of our algorithm. We classified the training data errors into 5 groups according to the value of the ϵ.

Table 4.8: Distribution of SVs to training error ranges

	no error	within ϵ	between ϵ and 2ϵ	between 2ϵ and 3ϵ	beyond 3ϵ	total
Grocery-1	65%	9%	5%	3%	19%	100%
Grocery-2	60%	11%	6%	3%	20%	100%
Grocery-3	64%	10%	5%	4%	17%	100%
IRI 1	53%	6%	5%	5%	29%	100%
IRI 2	49%	6%	6%	6%	33%	100%
IRI 3	38%	8%	7%	6%	41%	100%
Census-house	59%	2%	2%	2%	34%	100%

Table 4.8 indicates that a large portion (38% to 65%) of the selected points (support vectors) are on the prediction line. This results contradicts with the support vector definition of standard ϵ-SVR, using L_2 regularization on ω, where the support vectors are located on the decision boundary and outside the ϵ-tube.

Table 4.9: Probability of being SVs in the given error class

	0	errors between 0 and ϵ	errors between ϵ and 2ϵ	errors between 2ϵ and 3ϵ	errors beyond 3ϵ	Overall
Grocery-1	0.27	0.11	0.10	0.08	0.13	0.19
Grocery-2	0.24	0.11	0.08	0.06	0.11	0.15
Grocery-3	0.34	0.15	0.12	0.11	0.15	0.22
IRI 1	0.29	0.09	0.08	0.09	0.08	0.11
IRI 2	0.28	0.01	0.01	0.11	0.08	0.08
IRI 3	0.20	0.09	0.08	0.08	0.09	0.03
Census-house	0.29	0.29	0.29	0.32	0.29	0.26

Table 4.9 shows the percentage of points that are selected as SVs within each error range. This analysis indicates that across all data sets observations on the line have the highest probability of being selected (support vectors). Interestingly, for the Census data set the density of support vectors does not seem to depend on the size of the residual. On the other hand, in the other two data domains the density of support vectors (i.e. the percentage of points that are support vectors) is highest on the line.

Figure 4.1 and 4.2 show the probability of being selected as SV according to target values for L_1-norm and L_2-norm ϵ-SVR, respectively. These figures indicate that both in L_1-norm and L_2-norm, the probability of being selected as SV tends to increase or remains nearly same as the target value deviates from average value. However, in the case of L_1-norm the target values has no effect or little effect on the selected points for IRI data sets, because of this the probability remains nearly same for each range of target values.

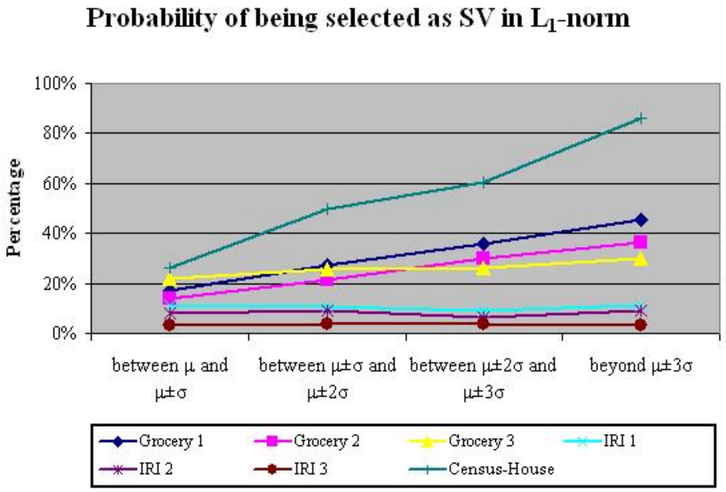

Figure 4.1: Probability of being selected as SV in L_1-norm

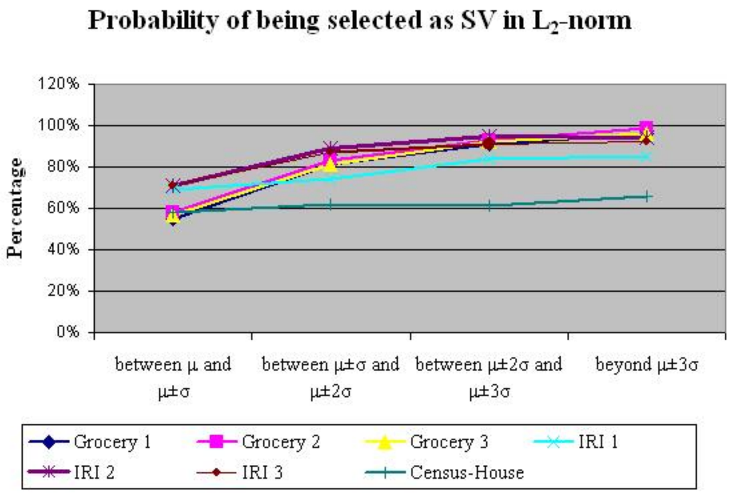

Figure 4.2: Probability of being selected as SV in L$_2$-norm

Up to now, we analyze the selected points by considering their geometric positions. Now we want to compare the selected points with support vectors of each data set, which are obtained by training each chunk of data set with L$_2$-norm ϵ-insensitive SVR, namely standard ϵ-SVR.

Table 4.10: Percentage of SVs in L$_1$-norm vs L$_2$-norm

	L$_1$-norm	L$_2$-norm
Grocery-1	19%	58%
Grocery-2	15%	62%
Grocery-3	22%	61%
IRI-1	11%	72%
IRI-2	8%	76%
IRI-3	3%	76%
Census-House	29%	58%

As mentioned in section 2.2.3, the reason behind using L$_1$-norm regularization of the support vector weights is to select a small but a representative sample, which coincides with the SVs of L_2-norm SVR. In order to compare the L_1-norm SVs with the L_2-norm support vectors of the given data, we train each chunk of grocery data sets with L_2-norm SVR and construct the table 4.10. Table 4.10 shows that L_1-norm selects fewer SVs than as expected. It is not surprising because from section 2.2.3, we know that L_1-norm ϵ-SVR picks fewer SVs compared to L_2-norm ϵ-SVR. We also test SVs for independence. The hypothesis is that the probability of being selected in L_1-norm is independent from the probability of being SVs by L_2-norm. The multiplication rule says that if two events were independent, then the probability of both occurring was the product of the probabilities of each occurring. By considering this rule, we test the SVs for independence by using the significance level as 0.05.

Table 4.11: Comparison of SVs by L_1-norm with SVs of L_2-norm

	L_1 SV and L_2 SV	L_1 SV not L_2 SV	L_2 SV not L_1 SV	not L_1 SV not L_2 SV
Grocery 1	7113	6258	34518	23840
Grocery 2	3404	2201	19230	11898
Grocery 3	4271	2773	14708	9594
IRI 1	2873	1173	23169	8785
IRI 2	993	316	11044	3647
IRI 3	677	200	20792	7325
Census House	2207	1911	5927	3955

In order to test SVs for independence, we compare the selected points coming from the first step of the algorithm with the SVs of the L_2-norm of the given data and construct the table 4.11, which is further used for independence test. Table 4.12 indicates that in this analysis in three out of seven data sets the probability of being selected in L_1-norm is independent of the probability of being SVs by L_2-norm but for other four data sets the hypothesis is not true. However, the data sets, which show dependency, does not show overlapping pattern i.e. for one data set expected value is lower than actual while for the rest not. Therefore, we can not conclude that L_1-norm and L_2-norm are dependent and the data sets overlap in dependency.

Table 4.12: Independence Test

Grocery-1	P-value = 0
Grocery-2	$0.1 \leq$ P-value ≤ 0.2
Grocery-3	P-value = 1
IRI-1	$0.025 \leq$ P-value ≤ 0.05
IRI-2	$0.5 \leq$ P-value ≤ 0.6
IRI-3	$0.025 \leq$ P-value ≤ 0.05
Census-House	P-value = 0

Finally, we investigate whether the chunking approach of row selection step converges to the same solution as training the algorithm with whole data, without chunking and the prediction accuracy of the algorithm is affected by the chunking approach. In order to make this analysis we compare three different models resulting from the same data consisting 2000 observations. These three models differs from each other in the data used to construct the final model. Model I includes all data points, Model II consists of only SVs of the data obtained by training the data with L_1-norm ϵ-insensitive without chunking and Model III includes the SVs of the data obtained by applying the first step of Row and Column Selection algorithm after dividing it into 10 chunks of equal size. When the selected points coming from training each chunk with L_1-norm ϵ-insensitive SVR (Model III) is compared with the SVs of the whole data (2000 points) obtained by L_1-norm ϵ-insensitive SVR (Model II), it is observed that dividing the data into small chunks and solving them separately does not yield the same solution as training the whole data by using same learning algorithm.

Table 4.13 summarizes the comparison of SVs of Model II and Model III. By analyzing this table, we can observe that Model III results in larger set SVs than Model

Table 4.13: Comparison of the SVs of Model II and Model III

	SV of Model II and SV of Model III	SV of Model II - not SV of Model III	SV of Model III - not SV of Model II	not SV of Model II not SV of Model III
Grocery 1	213	165	574	1048
Grocery 2	130	300	455	1115
Grocery 3	80	206	546	1168
IRI 1	55	191	376	1378
IRI 2	42	142	338	1478
IRI 3	8	53	244	1695
Census House	199	392	473	936

II and it does not include all the SVs of Model II. Therefore, we can conclude that the resulting data set coming from the first step of algorithm does not converge to the SV set of the whole data. Furthermore in order to see the effects of chunking approach on the prediction accuracy, we compare accuracy of test set for three different RBF-SVR models stated above.

Table 4.14: Effects of chunking approach on the bias and accuracy

	ME			MAE		
	Model I	Model II	Model III	Model I	Model II	Model III
Grocery-1	-0.3762	-0.4063	-0.3878	0.7454	0.7108	0.7376
Grocery-2	-0.0166	0.2306	-0.0421	0.5995	0.571	0.5847
Grocery-3	0.175	0.185	0.2486	0.5808	0.5721	0.5706
IRI-1	-0.0376	-0.1309	-0.2281	0.7235	0.7623	0.7737
IRI-2	-0.0642	-0.2871	-0.1932	0.6338	0.6574	0.6427
IRI-3	0.0937	0.1578	-0.2704	0.5869	0.6402	0.6498
Census House	0.0101	-0.012	-0.0004	0.1379	0.1278	0.1342

Table 4.14 shows the prediction accuracy of the Model I through Model III. Look-

ing at the MAE value of Model I and Model III, we observe that dividing the problem into small chunks and solving them separately to select the important observations of each chunk results in more accurate models for Grocery and Census-House data sets. However, for IRI data sets, which are based on weekly basis and have a long training data time frame, chunking approach does not improve the accuracy of prediction model. But it dramatically reduces the number of observations (12.6% to 21.55%) with only 0.2ϵ to 1.9ϵ increase in MAE for IRI data sets. On the other hand, we can conclude that selecting important observations of a data set by training whole algorithm without diving it into small chunks (Model II) results gives the best accuracy among three models for Grocery and Census House data sets while Model I, which includes the whole data, gives the best accuracy level for IRI data sets. In addition to these, Model I gives best ME value among three models but this is not surprising since the bias generally increases in absolute value as the number of observations decreases.

4.7.2 Selected Columns

Finally, we investigate the columns that are selected by the Row and Column Selection Algorithm. While the specific columns selected for each maingroup differ, 4 and 7 of the variables are present in all maingroup models for Grocery and IRI data sets, respectively. Variables common in all grocery data sets include an indicator variable showing whether the item was sold yesterday, its current discount level, whether the sale of product is fast, and its share with the maingroup in the same store. On the other hand, overlapping selected variables among IRI data sets include the historical sales of the product for non discount days and four weeks ago, number of days passed since last discount, whether the product has a discount or not for the given day, number of the SKUs in the maingroup and size of the product. Historical discount probabilities for the SKU and maingroup, discount amounts for the SKU yesterday, and number of the SKUs in the maingroup are also used in two out of the three models. Some prediction models also use store as a variable to predict the sales.

Table 4.15: Overlapping Selected Columns among the Grocery and IRI Data Sets

	# of variables	common with all other maingroups	common with another maingroup
Grocery 1	16	4	7
Grocery 2	17	4	4
Grocery 3	19	4	5
IRI 1	22	7	8
IRI 2	10	7	3
IRI 3	14	7	5

Moreover, in order to see the independency between selected columns we construct regression models for each independent variable and use the Variance Inflation Factor (VIF), which is a measure of the degree of multi-collinearity of the ith independent variable with the other independent variables in a regression model [41], to check whether there is a dependency between selected columns or not.

$$VIF_i = \frac{1}{1 - R_i^2} \tag{4.3}$$

VIF provides a reasonable and intuitive indication of the effects of multi-collinearity on the variance of the ith regression coefficient. The VIF value 1 means that there is no correlation among the ith predictor and the remaining predictor variables and the VIF exceeding 4 is a sign of dependence between ith predictor and the remaining predictor [41].

In this analysis, we use original data sets and resulting data sets, which include only selected rows and columns of the original data sets. Comparison of the results of

Table 4.16: Test of Independence of Variables

	Original rows and columns			Selected rows and columns		
	Max VIF	Mean VIF	% columns with VIF $\geq$ 4	Max VIF	Mean VIF	% columns with VIF $\geq$ 4
Grocery 1	455.469	22.917	56%	2.2504	1.4208	0%
Grocery 2	546.561	24.941	52%	2.0331	1.4613	0%
Grocery 3	239.025	19.761	57%	4.1508	1.9955	5%
IRI 1	70.9232	8.6092	56%	2.5322	1.6956	0%
IRI 2	217.4418	11.3313	53%	1.4278	1.1956	0%
IRI 3	548.0794	21.7451	56%	7.5757	2.8738	36%
Census House	1378.9733	92.31	79%	5.8417	3.5376	14%

these two data sets, verifies that the column selection part of our algorithm eliminates or reduces multi-collinearity. Strong evidence for this statement can be found in table 4.16, if we compare the percentages of variables with VIF value grater than 4 for original data sets to the corresponding entries for resulting data sets. Furthermore, analyzing max VIF values of selected rows and columns indicates that there is no dependency between any selected columns for four out of seven data sets. For Grocery 1, Grocery 2, IRI 1 and IRI 2 the variables with highest VIF values are smaller than 4, which is the sign of dependence. However, for Grocery 3, IRI 3 and Census House the variables with highest VIF values tells us that some variables can be correlated with at least one of the other predictors in the model.

4.7.3 Effects of Variables

In this section, we identify the effects of variables on the selected points only for grocery data set. Before starting statistical data analysis of the variables of the gorcery data set, we categorize continuous variables based on the standard deviations away from the mean and generate a new target showing whether the point is SV or not. Once a data set with categorical variables is obtained, two-way contingency tables can be constructed between each categorical predictor and the new response variable.

In all three maingroups of grocery data set, according to chi-square goodness-of-fit test, slow moving SKUs are more likely to become support vectors. Also, as the share of a SKU or brand increases with respect to all stores the probability of being selected starts to decrease. In other word when the SKU becomes a fast moving product, then it is less likely to be selected as a support vector. This observation supports the first claim which is about the size of SKU. Conversely, presence and amount of discounts for the SKU and competitors at different time periods such as today, yesterday and a week ago has a positive effect on probability of becoming support vectors. Moreover, the days with zero and the observations of some specific brand and store sales play an important role in determining the rows more likely to be support vectors. Besides the points mentioned above, we see that the algorithm selects observations that are located on the extreme values rather than selecting the observations located on the average values. This will facilitate measurement of the impact of the high and low share SKUs or brand in the subgroup and as well as the number of sales yesterday of the SKU and the average sales for the last four weeks. Lastly, discounts for the best selling SKUs have also effect on row selection. As the variation of discount amount of specific competitor SKU form mean increases, the probability of the given observation being support vector starts to decrease. That is to say that, the probability of being support vector is highest for the observations with the average value of discount amount of specific competitor SKUs when compared to higher or lower value of this variable. As a result of this, we see that the algorithm selects observations that will facilitate measurement of the impact of discounts on sales in the same time period, in the future time periods, as well as impact of discounts and promotions of competitors on focal SKU sales, one day and four week sales history of the SKU, and the specific brand and store. Interestingly these are the main components of promotion impact that are studied in the marketing literature.

Table 4.17: Effects of variables on support vectors

Variable Name	seen in # of data sets	Effect Type
Size of SKU	3	negative effect
SKU or brand share in all stores	2	negative effect
non zero sales yesterday	3	negative effect
absolute or relative discount	3	positive effect
competitor weighted discount	3	positive effect
specific brand or store	3	positive effect
SKU or brand share in subgroup	3	u-shaped effect
sales history of SKU	3	u-shaped effect
specific competitor SKU discount	2	n-shaped effect
# of days SKU has been on discount in the store	3	w-shaped effect

Table 4.17 summarizes the effects of variables on being support vector. According to results stated in table 4.17, we can conclude that the effects of variables have consistent effects on each maingroup of grocery data set.

Chapter 5

CONCLUSION

In this thesis, we use support vector regression to construct a forecasting model for grocery chain to predict the daily sales of a particular category. Our objective is to develop a methodology to estimate SVR models for very large data sets, which have low complexity and give accurate predictions. To accomplish this objective, we developed a two stage methodology, namely Row and Column Selection algorithm. In the first stage, the algorithm divides the given data into small chunks and solves each of them independently by using linear programming models and obtain the important points of data set. Then, in the second stage of proposed algorithm, a small variable subset is selected by training ϵ-insensitive linear regression with L_1-norm regularization of the variable weights on these selected points. The data complexity reduction approach of proposed algorithm makes it possible to train standard ϵ-SVR for large data set.

The *SR Model*, which includes selected points coming from the first stage of the algorithm with all columns, is more accurate and as unbiased as random sampling. The promising result of *SR Model* comes from utilising the fact that only the SVs play role in the SVR estimation. Moreover, the result of SVMTorch are not as accurate as *SR Model*.

From background knowledge, it is known that accuracy training of the algorithm can deteriorate in the presence of redundant or irrelevant variables. Eliminating the redundant or irrelevant variables and selecting the best variable subset by implement-ing second stage of proposed algorithm decreases the data complexity dramatically while improving the generalization error. Therefore, Row and Column Selection algorithm provides respectable accuracy while dramatically reducing the number of variables, which is an important operational gain for the MIS considering the costs of maintaining the data. As in *SR Model*, data set with seleted rows and columns (SRSC Model) also outperforms the SVMTorch.

The analysis of selected points demonstrates that unlike the standard SVR model, where the support vectors are just located on the decision boundary and outside the ϵ-tube [1], for the non-linear SVR model with L_1-norm regularization of the support vector weights, SVs can be located both inside and outside the ϵ-tube with large portion on the line. Another analysis related with the geometric position of selected points according to the target value of observations shows that in the case of L_1-norm, observations with extreme target values of Grocery and Census-house data sets are more likely to be selected while the target values has no effect or little effect on the selected points for IRI data sets. The result of this analysis is also consistent with L_2-norm. As in L_1-norm case, the probability of being SV in L_2-norm increases as the target value deviates from mean. Furthermore, the difference between SVs of L_1-norm and L_2-norm are compared. According to result of this experiment, we can conclude that L_1-norm SVR selects fewer points compared to L_2-norm ϵ-SVR and probability of being selected in L_1-norm is not independent from the probability of being SVs by L_2-norm.

Moreover, variables which are selected by the Row and Column Selection Algorithm is also analyzed to check whether there is a multi-collinearity between the selected variables or not. The result of this analysis indicates that there is no or low multi-collinearity between selected variables. In other words, our algorithm can accomplish dimensionality reduction via eliminating redundant and irrelevant variable and selecting only the most informative subset of variables. Selected variables are also consistent within Grocery and IRI data sets. This consistency enables the retailer to understand the main drivers of sales and retain less data.

Lastly, effects of variables on selected points is analyzed to identify those that carry the most information about support vectors. The result of this analysis indicates that the row and column selection algorithm select observations that are located on the extreme values rather than selecting the observations located on the average values. The effects of variables on selected points are consistent in all Grocery and IRI data sets. As a result of this analysis we see that for Grocery data sets the algorithm selects observations that will facilitate measurement of the impact of the product discounts on the sales in different time periods as well as impact of discounts and promotions of competitors on focal SKU sales, sales history of the SKU, and the specific brand and store. Interestingly these are the main components of promotion impact that are studied in the marketing literature.

5.1 Future Works

This research can be extended into at least two directions. The first involves using Row Selection SVR as active learning to select the points as the new observations added to data set and determine a rule for when to retrain the model. A second possible research direction is to repeat the analysis in this paper for some other public research data sets to determine under which conditions this training data selection algorithm provides more favorable results.

BIBLIOGRAPHY

[1] A.J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[2] H. Drucker, C.J.C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. *Advances in neural information processing systems*, pages 155–161, 1997.

[3] R. Collobert and S. Bengio. SVMTorch: Support vector machines for large-scale regression problems. *The Journal of Machine Learning Research*, 1:160, 2001.

[4] V. Vapnik. The nature of statistical learning, 1995.

[5] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.

[6] R. Fletcher. *Practical methods of optimization.* Wiley, 1987.

[7] O.L. Mangasarian. *Nonlinear programming.* McGraw-Hill, 1969.

[8] W. Karush. Minima of functions of several variables with inequalities as side constraints. *Master's thesis, Department of Mathematics, University of Chicago*, 1939.

[9] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proc.* 2^{nd} *Berkeley Symposium on Mathematical Statistics and Probabilistics*, pages 481–492, Berkeley, 1951. University of California Press.

[10] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[11] P.J. Huber and E.M. Ronchetti. *Robust statistics*. John Wiley & Sons Inc, 2009.

[12] Alex J. Smola, Bernard Schölkopf, and Gunnar Rätsch. Linear programs for automatic accuracy control in regression. pages 575–580, London, 1999.

[13] J. Wang, P. Neskovic, and L.N. Cooper. Training data selection for support vector machines. *Lecture Notes in Computer Science*, 3610:554, 2005.

[14] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[15] J. Bi, K. Bennett, M. Embrechts, C. Breneman, and M. Song. Dimensionality reduction via sparse support vector machines. *The Journal of Machine Learning Research*, 3:1229–1243, 2003.

[16] Z. Lu, X. Wu, and J. Bongard. Active learning with adaptive heterogeneous ensembles. In *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM 2009)*, pages 6–9.

[17] H. He and E.A. Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263, 2009.

[18] D. Mease, A.J. Wyner, and A. Buja. Boosted classification trees and class probability/quantile estimation. *The Journal of Machine Learning Research*, 8:439, 2007.

[19] H. Shin and S. Cho. Invariance of neighborhood relation under input space to feature space mapping. *Pattern recognition letters*, 26(6):707–718, 2005.

[20] B. Settles. Active Learning Literature Survey. *Science*, 10(3):237–304, 1995.

[21] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[22] G.H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the 11th International Conference on Machine Learning*, pages 121–129, 1994.

[23] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

[24] S. Bierman and S. Steel. Variable selection for support vector machines. *Communications in Statistics-Simulation and Computation*, 38(8):1640–1658, 2009.

[25] V.N. Vapnik. *Estimation of dependences based on empirical data*. Springer-Verlag New York.

[26] Edgar Osuna, Robert Freund, and Federico Girosi. An improved training algorithm for support vector machines. pages 276–285. IEEE, 1997.

[27] E. Osuna and F. Girosi. Reducing the run-time complexity of support vector machines. *Advances in Kernel Methods: Support Vector Learning, MIT press, Cambridge, MA*, pages 271–284, 1999.

[28] John C. Platt. Sequential minimal optimization: Fast training of support vector machines using sequential minimal optimization. pages 185–208, 1999.

[29] E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In *Neural networks for signal processing VIIProceedings of the 1997 IEEE workshop*, pages 276–285. Citeseer, 1997.

[30] V. Cherkassky and Y. Ma. Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, 17(1):113–126, 2004.

[31] V.S. Cherkassky and F. Mulier. *Learning from data: concepts, theory, and methods*. Wiley-Interscience, 1998.

[32] B. Scholkopf and A. Smola. Advances in kernel methods: Support vector learning. 1999.

[33] J.T. Kwok. Linear Dependency between e and the Input Noise in e-Support Vector Regression. In *Artificial neural networks–ICANN 2001: International Conference, Vienna, Austria, August 21-25, 2001: proceedings*, page 405. Springer Verlag, 2001.

[34] A. Smola, N. Murata, B. Scholkopf, and KR Muller. Asymptotically optimal choice of ε-loss for support vector machines. In *Proceedings of the International Conference on Artificial Neural Networks, Perspectives in Neural Computing*, pages 105–110, 1998.

[35] Davide Mattera and Simon Haykin. Support vector machines for dynamic reconstruction of a chaotic system. pages 211–241, 1999.

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

[37] Özden Gür Ali, Serpil Sayın, Tom Van Woensel, and Jan Fransoo. SKU demand forecasting in the presence of promotions. *Expert Systems with Applications*, 36(10):12340–12348, 2009.

[38] D.R. Musicant and A. Feinberg. Active set support vector regression. *IEEE Transactions on Neural Networks*, 15(2):268–275, 2004.

[39] I.W.H. Tsang, J.T.Y. Kwok, and JA Zurada. Generalized core vector machines. *IEEE Transactions on Neural Networks*, 17(5):1126–1140, 2006.

[40] M. Tohmé and R. Lengellé. F-SVR: a new learning algorithm for support vector regression. In *ICASSP, Proc. IEEE international conference on acoustics speech and signal processing*, 2008.

[41] J. Neter, W. Wasserman, M.H. Kutner, et al. *Applied linear statistical models.* Irwin Burr Ridge, Illinois, 1990.

VITA

Kübra Yaman was born in Istanbul, Turkey, on December 8, 1985. She graduated from Yeşilköy Anatolian High School in 2003. She received her B.S degree in Industrial Engineering from Bahçeşehir University, Istanbul, in 2008. Same year, she joined the M.S program in Industrial Engineering at Koç University as a research and teaching assistant.