The LNM Institute of Information Technology, Jaipur

Study and Applications of Reduced Support Vector Machine

Author:
Guruvansh Choudhary

Supervisor:
Dr. M. Tanveer

Work Progress report submitted in partial fulfillment
of the requirements for the degree of

Bachelor of Technology
in
Electronics and Communication Engineering

April 2015

Declaration of Authorship

I, Guruvansh Choudhary, declare that this thesis titled, 'Study and Applications of Reduced Support Vector Machine' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"Each one of us has our own evolution of life, and each one of us goes through different tests which are unique and challenging. But certain things are common. And we do learn things from each other's experience. On a spiritual journey, we all have the same destination."

A. R. Rahman

Abstract

RSVM is an algorithm which generates a nonlinear kernel-based separating surface that requires as little as 1% of a large dataset for its explicit evaluation. After the selection of reduced dataset, the remainder of the data can be thrown away. The computational difficulties and memory usage for generating the nonlinear support vector machine classier for a large dataset can be reduced using RSVM. A small random subset from the entire dataset is selected with a fixed size to generate a reduced kernel matrix. The reduced kernel can replace the fully dense square kernel matrix used in the formulation of nonlinear support vector machine to cut the problem size and computational time and does not scarify the prediction accuracy. In this report, the standard RSVM algorithm is used for classification of massive dataset like MNIST,EEG, pendigit and BCI. The results of RSVM with a nonlinear separating surface depend on the percentage of small randomly selected portion of the dataset. The RSVM is not answerable to some essential questions like what should be the size of reduced dataset and how to deal with the situation when the randomly selected reduced dataset carries only the similar data points. In the repot some advanced methods for generating reduced dataset are discussed. Incremental Reduced Support Vector Machine (IRSVM), an improved version of RSVM begins with an extremely small reduced set and incrementally expands the reduced dataset, is an approach which can estimate the size of reduced data, is discussed. Systematic Sampling RSVM (SSRSVM) is another approach for obtaining the reduced dataset which adds a portion of misclassified points into the reduced set iteratively based on the current classifier until the validation set correctness is large enough. From the big data applications view, RSVM is equipped in the framework of MapReduce. The algorithms of RSVM-MapReduce testing and prediction are also presented.

Acknowledgements

I wish to express my sincere thanks to Dr. M. Tanveer, Assistant Professor, The LNM IIT, Jaipur, for his valuable suggestions and guidance during the course of this project. It was with his help that I was able to perform the survey and simulation works effectively for my project. His constant support, motivation and channelized guidance were instrumental in completing the project.I also wish to thank Dr. Yuh-Jye Lee, National Taiwan University of Science and Technology, for sharing the RSVM toolbox. It was with his counsel that I have completed my simulation work.

Contents

List of Figures

List of Tables

Dedicated to the philanthropists...

Chapter 1

Introduction

Machine learning is a scientific discipline that explores the construction and study of algorithms that can learn from data. Such algorithms operate by building a model from example inputs and using that to make predictions or decisions rather than following strictly static program instructions. This is often a very attractive alternative to manually constructing them, and in the last decade the use of machine learning has spread rapidly throughout computer science and beyond. Machine learning is used in Web search, spam filters, recommender systems, ad placement, credit scoring, fraud detection, stock trading, drug design, and many other applications. There is existing need of algorithms to advance the science of Machine Learning. There are various approaches to implement machine learning which includes Decision tree learning, Artificial neural networks, Support vector machine , Reinforcement learning etc. In machine learning, support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Nowadays, SVM plays a very dominant role in data classification using a kernel-based linear or nonlinear classifier. With the evolution of technology, datasets grow in size in part because they are increasingly being gathered by cheap and numerous information-sensing mobile devices, aerial (remote sensing), software logs, cameras, microphones, radio-frequency identification (RFID) readers, and wireless sensor networks.

As the data size is increasing leaps and bounds there arises a need for efficient and accurate algorithm with low time complexity for the classification of data.The two major problems that need to be dealt with large data classification by a nonlinear kernel are:

1. **Computation Time**: The total time needed for solving the mathematical programming problem.

2. **Memory Usage**: The dependence of the nonlinear separating surface on the entire dataset which creates unwieldy storage problems that prevents the use of nonlinear kernels for anything but a small dataset.

The reformulation of SVM to RSVM [1] has been done to overcome the aforementioned defficulties for classification of large datasets. Suppose for a hundred points dataset, one is confronted by a fully dense quadratic program with 101 variables and 100 constraints resulting in constraint matrix with over a million entries. In contrast, the RSVM approach reduces the problem to one which has a 11 variables and a 100 constraints which is readily solved by a smoothing technique [6] as an unconstrained 11-dimensional minimization problem. This generates a nonlinear separating surface which depends on a ten data points only, instead of the conventional nonlinear kernel surface which would depend on the entire hundred points. In [1], Eigenvalue decomposition of a randomly selected subset of the training set has been deployed. But in the RSVM the entire kernel matrix is generated within an iterative linear equation solution procedure.

In [2], the reduced set is selected randomly from the entire dataset with a user pre-specified reduced set size . It is typically much smaller than the size of entire dataset. But for an algorithm to perform well in robust cases, it must be able to determine the size of reduced dataset dynamically rather than continuing with any pre-specified value.

In this report, some advanced SVM methods are discussed that work on the principle of RSVM which follow their own methods for obtaining reduced dataset rather than choosing it randomly. Firstly, Incremental Reduced Support Vector Machine (IRSVM) that automatically and incrementally selects representative data points to form the reduced set is discussed. Secondly, Systematic Sampling RSVM (SSRSVM) is a systematic sampling mechanism to select a reduced set. Under the compatible classification performance on the test set, IRSVM and SSRSVM can generate a smaller reduced set than the one via random selection scheme. In the literature, it is shown that the IRSVM and SSRSVM can provide good discriminant function estimations in supervised learning tasks. Also it has been observed that IRSVM and SSRSVM are much faster than conventional SVM under the same level of the test set correctness.[3]

The contents of the report are outlined as follows: In Section 2 the reduced SVM approach has been outlined which includes the linear and non linear kernel classification, RSVM formulation and simulation results on MINST,EEG, BCI- competition 2 (a and b) and pendigit. Section 3 gives Reduced Dataset Selection Procedure which includes the details of various modified RSVM techniques. In Section 4, RSVM under the MapReduce framework is described, so that it has the ability to solve massive data nonlinear kernel problems. As the reduced dataset has the potential and is the only solution to

save computation time and memory usage in Bid Data era. Section 5 concludes the report.

A word about the notation and background material. All vectors will be column vectors unless transposed to a row vector by a prime superscript $'$. For a vector x in the n-dimensional real space R^n, the plus function x_+ is defined as $(x_+)_i = max\{0, x_i\}$, while the step function x_* is defined as $(x_*)_i = 1$ if $x_i > 0$ else $(x_*)_i = 0, i = 1; \ldots, n$. The scalar (inner) product of two vectors x and y in the n-dimensional real space R^n will be denoted by $x^{'}y$ and the p-norm of x will be denoted by $|x|_p$. For a matrix $A \in R^{(m \times n)}, A_i$is the i^{th} row of A which is a row vector in R^n. A column vector of ones of arbitrary dimension will be denoted by e. For $A \in R^{(m \times n)}$ and $B \in R^{(n \times l)}$, the kernel $K(A, B)$ maps $R^{(m \times n)} X R^{(n \times l)}$ into $R^{(m \times l)}$. In particular, if x and y are column vectors in R^n then, $K(x^{'}, y)$ is a real number, $K(x^{'}, A^{'})$ is a row vector in R^m and $K(A, A^{'})$ is an $m * m$ matrix. The base of the natural logarithm will be denoted by ϵ.

Chapter 2

RSVM : CLASSIFICATION

2.1 Linear and Nonlinear Kernel Classification

For classification of m points in the n -dimensional real space R^n, the problem can be expressed by the $m \times n$ matrix A. A diagonal matrix D of order m with ones or minus ones along its diagonal can be considered according to membership of every point A_i in the classes +1 or -1. In [3] the standard support vector machine with a linear kernel $AA^{'}$ is given by the following quadratic program for some $v > 0$:

$$\begin{array}{lll} \min\limits_{(w,\gamma,y)\in R^{n+1+m}} & ve^{'}y + \frac{1}{2}w^{'}w & \\ \text{s.t.} & D(Aw - e\gamma) + y \geq e \\ & y \geq 0 \end{array} \tag{2.1}$$

for solving the problem.

In Figure 1, w is the normal to the bounding planes and γ represents their location relative to the origin.

$$x^{'}w - \gamma = +1 \quad \text{and} \quad x^{'}w - \gamma = -1 \tag{2.2}$$

When both the classes are linearly separable, the slack variable is null, that is, $y = 0$ and hence the first plane above bounds the class +1 points and the second plane bounds the class -1 points. The bounding planes (2.2) have a midway surface given below:

$$x^{'}w = \gamma \tag{2.3}$$

When both the classes are linearly inseparable, the slack variable is no longer null and has some positive value, that is, $y > 0$ and hence the two planes bound the two classes with a "soft margin" determined by the slack variable, that is:

$$\begin{array}{llll} x'w - \gamma + y_i \geq +1 & \text{for} & x^{'} = A_i \text{ and} & D_{ii} = +1 \\ x'w - \gamma - y_i \leq -1 & \text{for} & x^{'} = A_i \text{ and} & D_{ii} = -1 \end{array} \tag{2.4}$$

The 1-norm of the slack variable y is minimized with weight v in (2.1). The quadratic term in (2.1), maximizes the distance, called margin, between the two bounding planes of (2.2) in the n-dimensional space of $w \in R^n$ for a fixed γ. Figure 2.1 shows the separation of $A+$ and $A-$ with margin $2/ \parallel w \parallel_2$ consecutively represented by rows of A with $D_{ii} = +1$ and $D_{ii} = -1$.

In an another approach, SSVM [6], instead of the 1-norm of y as in (2.1), the square of 2-norm of the slack variable y is minimized with weight $\frac{v}{2}$. Apart from it, the distance between the planes (2.2) is estimated in the $(n+1)$-dimensional space of $(w,\gamma) \in R^{n+1}$. There is no effect on the problem as it was shown in [4]by measuring the margin in this R^{n+1} instead of R^n. By using twice the reciprocal squared of the margin instead of standard SVM formulation, modifies SVM problem as follows:

$$\begin{array}{lll} \min\limits_{(w,\gamma,y)\in R^{n+1+m}} & \frac{v}{2}y^{'}y + \frac{1}{2}(w^{'}w + \gamma^2) & \\ \text{s.t.} & D(Aw - e\gamma) + y \geq e & \\ & y \geq 0 & \end{array} \tag{2.5}$$

It has been verified by computational experiments in [5] that the reformulation of conventional support vector machine formulation (2.1) to (2.5) shows similar results to (2.1).The solution of problem (2.5), y is given by

$$y = (e - D(Aw - e\gamma))_+ \tag{2.6}$$

where $(.)_+$ replaces negative components of a vector by zeros, as described in the introduction. By replacing y in (2.5) by $(e - D(Aw - e\gamma))_+$, the SVM problem (2.5) can be converted into an equivalent SVM which is an unconstrained optimization problem as follows:

$$\min_{(w,\gamma)\in R^{n+1}} \frac{v}{2} \parallel (e - D(Aw - e\gamma))_+ \parallel_2^2 + \frac{1}{2}(w^{'}w + \gamma^2), \tag{2.7}$$

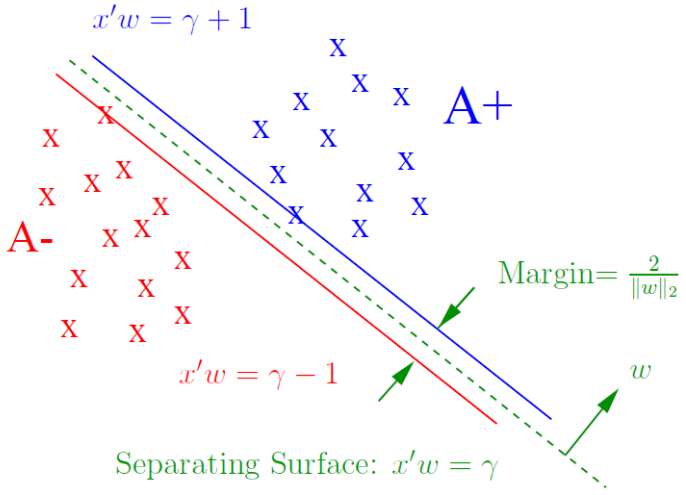

FIGURE 2.1: Data Classification by Separating Surface [8].

The equivalent SVM unconstraint optimization problem, (7), is not twice differentiable, so fast Newton Armijo method cannot be applied directly. To solve (7) using the fast Newton Armijo method which is described in Appendix 1, firstly it needs to be smoothed so that it becomes twice differentiable.

In [2], a nonlinear separating surface is generated by the generalized support vector machine (GSVM) using a completely arbitrary kernel.The GSVM solves the following mathematical program for a general kernel $K(A, A^{'})$:

$$\begin{array}{lll} \min\limits_{(u,\gamma,y)\in R^{2m+1}} & ve^{'}y + f(u), & \\ \text{s.t.} & D(K(A,A^{'})Du - e\gamma) + y & \geq \quad e \\ & y & \geq \quad 0 \end{array} \tag{2.8}$$

Here $f(u)$ is some function on R^m which suppresses the parameter u. The positive number, v weights the classification error $e^{'}y$ versus the suppression of u. A solution of this mathematical program for u and γ leads to a nonlinear separating surface, given below:

$$K(x^{'}, A^{'})Du = \gamma \tag{2.9}$$

The formulation (2.1) is obtained if we let $K(x^{'}, A^{'}) = AA^{'}$, $w = A^{'}Du$ and $f(u) = 1/2u^{'}DAA^{'}Du$. We now use a different classification objective which not only suppresses the parameter u but also suppresses γ in our nonlinear formulation:

$$\begin{array}{lll} \min\limits_{(u,\gamma,y)\in R^{2m+1}} & \frac{v}{2}y^{'}y+\frac{1}{2}(u^{'}u+\gamma^2), & \\ \text{s.t.} & D(K(A,A^{'})Du-e\gamma)+y & \geq \quad e \\ & y & \geq \quad 0 \end{array} \tag{2.10}$$

At a solution of (2.10), y is given by

$$y=(e-D(K(A,A^{'})Du-e\gamma))_+ \tag{2.11}$$

where $(.)_+$ replaces negative components of a vector by zeros, as defined earlier. The y in (2.10) can be replaced by $(e-D(K(A,A^{'})Du-e\gamma))_+$ and converts the SVM problem (2.10) into an equivalent SVM which is an unconstrained optimization problem as follows:

$$\min_{(u,\gamma)\in R^{m+1}} \frac{v}{2}\| \ (e-D(K(A,A^{'})Du-e\gamma))_+ \ \|_2^2+\frac{1}{2}(u^{'}u+\gamma^2), \tag{2.12}$$

Again, as in (2.7), this problem is not twice differentiable, so there is need of its smoothening. The smoothing techniques of [7]can be used by replacing x_+ by a very accurate smooth approximation as was done in [6]for making objective function twice differentiable.

Thus we replace by The integral of the sigmoid function $\frac{1}{1+\varepsilon^{-\alpha x}}$ of neural networks, $p(x,\alpha)$ can be used to replace x_+ for some $\alpha>0$ for smoothening of the objective function. That is:

$$p(x,\alpha)=x+\frac{1}{\alpha}\log(1+\epsilon^{-\alpha x}), \alpha>0. \tag{2.13}$$

So the smooth support vector machine (SSVM) can be represented as:

$$\min_{(u,\gamma)\in R^{m+1}} \frac{v}{2}\| \ p(e-D(K(A,A^{'})Du-e\gamma),\alpha) \ \|_2^2+\frac{1}{2}(u^{'}u+\gamma^2), \tag{2.14}$$

by using p function with a smoothing parameter α, with $\alpha>0$.

As stated earlier, the objective function of (2.14) is twice differentiable. It allows the utility of a globally quadratically convergent Newton algorithm for solving the SSVM (2.14). Problem (2.14) can generate a highly nonlinear separating surface (2.9) with retaining the differentiability properties for any arbitrary kernel. Following are the two

difficulties yet to be resolved: 1. For massive datasets, m can be of the order of millions. As (2.14) is a problem in $m+1$ variables, so there the time required for solving it can very large. 2. The entire dataset is required to represent the resulting nonlinear separating surface (2.9) which creates storage difficulty for massive datasets.

To overcome the above mentioned difficulties, there is requirement of a method that needs smaller dataset which can not only reduce the time requirement but also the memory usage. The reduced support vector machine (RSVM) is one of the available methods that uses the small randomly selected dataset. The RSVM is discussed in detail in the next two subsections with algorithm.

2.2 RSVM: The Reduced Support Vector Machine

The need for RSVM arises to generate a nonlinear separating surface (2.9) for massive dataset which utilizes only a small portion of the entire dataset for its characterization. The nonlinear kernels cannot be used on massive datasets as it is the computationally impractical to solve the huge unconstrained optimization problem (2.14) which includes the kernel $K(A, A^{'})$ that typically leads to the system running out of memory. Based on the idea of chunking methods [8], use a very small subset of the dataset for generating a nonlinear surface. In RSVM, very small randomly chosen part of dataset given by $\bar{m}$ points of the original m data points with $\bar{m} << m$, called as $\bar{A}$ is used for solving unconstrained optimization problem (2.14) where $A^{'}$ can be replaced by $\bar{A}^{'}$.

The RSVM formulation can be done by reducing the full dataset $A \in R^{(m\times n)}$ with a square kernel $K(A, A^{'}) \in R^{(m\times m)}$ to the reduced dataset $\bar{A} \in R^{(m\times n)}$ with corresponding diagonal matrix and rectangular kernel $K(A, \bar{A}^{'}) \in R^{(m\times m)}$, the obtained RSVM Algorithm is shown below. This algorithm solves the RSVM quadratic program obtained from (2.10) by replacing $A^{'}$ with $\bar{A}^{'}$ as follows:

$$\begin{array}{lll} \min\limits_{(\bar{u},\gamma,y)\in R^{\bar{m}+m+1}} & \frac{v}{2}y^{'}y + \frac{1}{2}(\bar{u}^{'}\bar{u} + \gamma^2), & \\ \text{s.t.} & D(K(A, \bar{A}^{'})\bar{D}u - e\gamma) + y \geq e & \\ & y \geq 0 & \end{array} \tag{2.15}$$

2.3 RSVM Algorithm

1. Choose a random subset matrix $\bar{A} \in R^{\bar{m}\times n}$ of the original data matrix $A \in R^{m\times n}$. Typically is 1% to 10% of m.

2. Solve the following modified version of the SSVM (2.14) where $A^{'}$ only is replaced by $\bar{A}^{'}$ with corresponding $\bar{D} \in D$:

$$\min_{(\bar{u},\gamma)\in R^{\bar{m}+1}} \frac{v}{2}\| p(e - D(K(A,\bar{A}^{'})Du - e\gamma),\alpha) \|_2^2 + \frac{1}{2}(\bar{u}^{'}\bar{u} + \gamma^2), \tag{2.16}$$

which is equivalent to solving (2.10)with $A^{'}$ only replaced by $\bar{A}^{'}$

3. The separating surface is given by (2.9) with $A^{'}$ replaced by $\bar{A}^{'}$ as follows:

$$K(x^{'},\bar{A}^{'})\bar{D}\bar{u} = \gamma \tag{2.17}$$

where $(\bar{u},\gamma) \in R^{(\bar{m}+1)}$ is the unique solution of (2.16), and $x \in R^n$ is a free input space variable of a new point.

4. A new input point $x \in R^n$ is classified into class +1 or -1 depending on whether the step function:

$$(K(x^{'},\bar{A}^{'})\bar{D}\bar{u} - \gamma)_*, \tag{2.18}$$

is +1 or zero, respectively.

In the next section, the RSVM algorithm is applied on MNIST, EEG, BCI-competition 2 and EEG datasets for classification.

2.4 Simulation Results

In this section, the results and observations for classification through RSVM on four publicly available massive datasets are presented. All the datasets, MNIST, pendigit, EEG, and BCI- competition 2 are publicly available on internet for research in the field of data mining and have been taken from UCI and their respective websites. To verify and analyse the performance efficiency of RSVM, all the datasets were simulated at first without reducing the datasets i.e. original datasets were employed for simulations. In case of EEG, the dataset was not separated i.e. both training and testing were in same file. Thus dataset was divided into two parts, training and testing, in the ratio of 80% and 20% respectively before starting the simulations.

Table 2.1: Classification Results on MNIST dataset

Used Dataset(Percentage)	Error Rate (Percentage)	Accuracy (Percentage)	Time Taken (Minutes)
1	7.0450	92.9550	0.3276
2	6.5433	93.4567	0.8472
3	6.2900	93.7100	1.5929
4	6.1500	93.8500	9.9660
5	6.0600	93.9400	24.0773
Complete dataset	System Runs Out of Memory		

Table 2.2: Classification Results on EEG dataset

Used Dataset(Percentage)	Error Rate (Percentage)	Accuracy (Percentage)	Time Taken (Seconds)
1	15.1916	84.8084	0.5423
2	9.3919	90.6081	1.5495
3	7.1936	92.8064	2.6724
4	6.3865	93.6135	5.5956
5	5.7223	94.2777	9.8643
6	5.5093	94.4907	14.0990
7	5.3258	94.6742	21.1456
8	5.2069	94.7931	25.8904
9	5.0951	94.9049	38.0031
10	5.0203	94.9797	42.5544
Complete dataset	N.A.	N.A.	N.A.

For Mnist, the big size of the training dataset i.e. 60000×784 forced the system run out of the memory and stop responding to any input, eventually all the running programs ceased responding. Indeed, it was due to high memory usage for which RSVM is the prime solution as mentioned in the previous sections. Therefore RSVM emerged out to be the primary tool for data classification in this case. The training dataset was brought down randomly and then was used for further processing. The reduction ratio was varied from 1% to 5% with a step size of 1%. Table 2.1 presents the observed results, which includes error rate, accuracy, and time taken (including both training and testing).

The size of the training dataset for EEG also caused the system choking to any input, evoking no response from the running programs and hence the need of RSVM was identified for classification. The training dataset was reduced randomly for further processing. This time the reduction ratio was varied from 1% to 10% with a step size of 1%. Table 2.2 presents the observed results, which includes error rate, accuracy, and time taken (including both training and testing).

In fig. 2.2 and 2.3, Accuracy versus Reduction ratio and Time taken versus Reduction ratio are plotted, respectively. As fig. 2.2 depicts, The accuracy of the model almost

saturates above a certain amount,5% of dataset. Thus, the complete dataset was not required at all for classification and hence helped saving a lot of time.

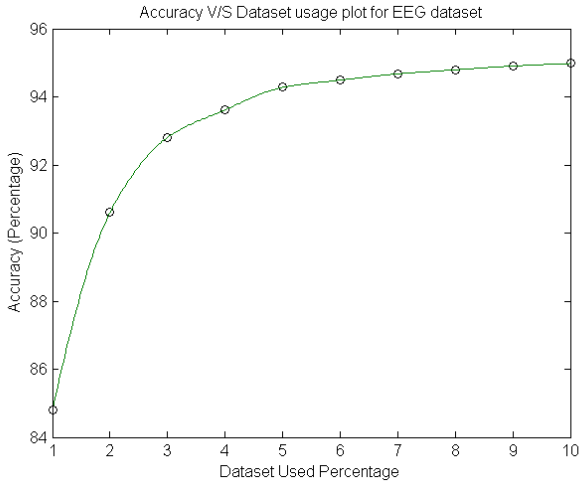

FIGURE 2.2: Accuracy Plot for EEG Dataset.

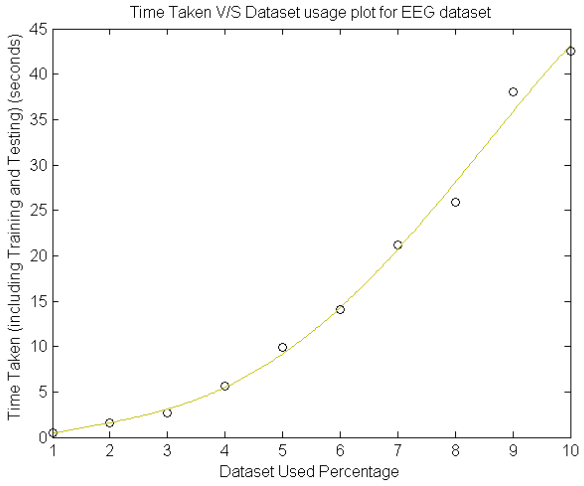

FIGURE 2.3: Time Taken plot for EEG Dataset.

Lately, the pendigit dataset was simulated, analogous to EEG. Initially it was utilized for classification with complete dataset. The overall time taken by the process was 433.79 seconds (including both training and testing) with accuracy of 98.86 %. Classification was done successfully but on the cost of large amount of processing time. To minimize the processing time, RSVM was used by breaking down the same dataset at first and then further processing was initiated. Again, the reduction ratio was varied from 1% to 10% with a step size of 1%. Table 2.3 presents the observed results, which includes error rate, accuracy, and time taken (including both training and testing).

TABLE 2.3: Classification Results on PENDIGIT dataset

Used Dataset(Percentage)	Error Rate (Percentage)	Accuracy (Percentage)	Time Taken (Seconds)
1	4.4969	95.5031	0.7156
2	3.3711	96.6289	1.5513
3	3.2333	96.7667	2.1459
4	3.0583	96.9417	2.9480
5	3.0126	96.9874	3.5970
6	2.9668	97.0332	4.4981
7	2.9851	97.0149	5.3806
8	2.9748	97.0252	6.8450
9	2.9383	97.0617	8.0763
10	2.8868	97.1132	9.2553
Complete dataset	2.1441	97.8559	1170.3

In fig. 2.4 and 2.5, Accuracy versus Reduction ratio and Time taken versus Reduction ratio are plotted, respectively. As fig. 2.5 depicts, Difference between the accuracy of the model with reduced dataset and complete dataset is less than 1.6% above a certain amount of dataset, 5% and time taken has been reduced by a factor of 230.

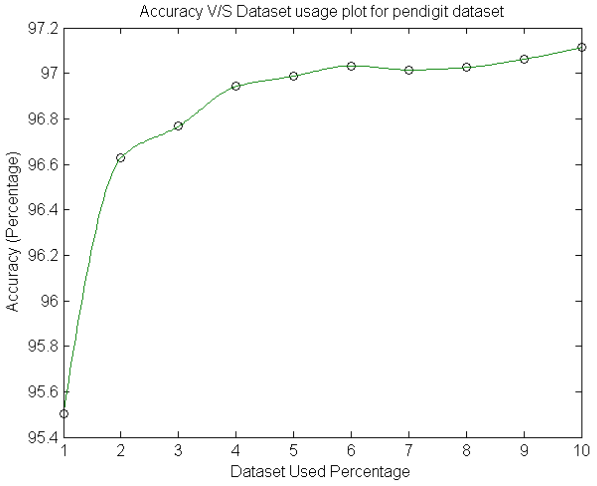

FIGURE 2.4: Accuracy Plot for PENDIGIT Dataset.

Lastly, RSVM was simulated for classification of BCI- competition 2 datasets. From table 2.4, in case of BCI-a, a deviation of 6.34% in accuracy was observed with reduction in total time by a factor of 3.32 by using 10% dataset . In case of BCI-b, with 10% dataset, a deviation of 0.39% in accuracy was observed with reduction in total time by a factor of 4.

It can easily be noticed for BCI datasets, RSVM algorithm did not work as well as it did for EEG, Pendigit and MNIST. As the size of BCI-a dataset was 268×5376 and that of BCI-b dataset was 200×8064. Apparently, both are massive datasets. But the number

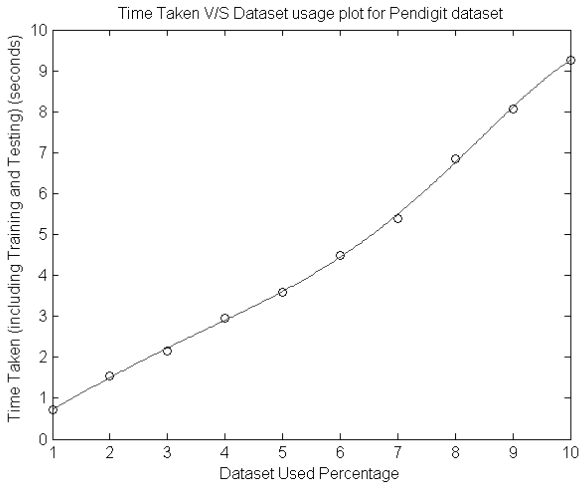

FIGURE 2.5: Time Taken plot for PENDIGIT Dataset.

TABLE 2.4: Classification Results on BCI dataset

Dataset	BCI a		BCI b	
Used Dataset(%)	Accuracy (%)	Time Taken *(sec)*	Accuracy(%)	Time Taken *(sec)*
1	71.79	0.1010	51.09	0.0589
2	78.87	0.1115	50.06	0.0589
3	80.16	0.1162	51.35	0.0679
4	80.73	0.1299	50.67	0.0693
5	80.86	0.1333	50.57	0.0707
6	81.22	0.1298	50.94	0.0709
7	81.28	0.1310	51.17	0.0724
8	81.50	0.1390	51.22	0.0730
9	81.36	0.1479	51.47	0.0759
10	81.37	0.1639	51.83	0.1081
Complete dataset	87.71	0.5928	52.22	0.4302

of attributes in both cases are very less, 268 and 200 whereas the number of features are very much, 5376 and 8064. Consequently, RSVM can be used on massive datasets provided the number of attributes are in abundance, and is independent of number of features.

Chapter 3

Reduced Dataset Selection Procedure

In the previous sections, the usage of RSVM has been observed, how it helps in saving memory usage and computation time with very minute degradation in accuracy. The reduced datasets used for simulations were chosen randomly and with variable reduction ratio, as per RSVM algorithm. But it is not as much reliable to select the dataset randomly, as in case if in one case out of hundreds, randomly selected dataset has all attributes *correlated*or *similar*, that is, very small amount of information has been recapitulate from original dataset. To overcome such issues there is requirement of advanced approaches to obtain reduced dataset from given dataset. In this section, some advanced approached will be discussed with algorithms for obtaining the reduced dataset. These approaches are answerable to the following questions that RSVM can not do:

1. What should be the best way for choosing the reduced dataset?
2. How to determine the size of reduced dataset automatically or dynamically?
3. How to reduce the chances of choosing the reduced dataset that are *highly correlated*, that is a possible outcome of choosing dataset *randomly*?

3.1 Incremental Reduced Support Vector Machine

In this section, an advanced reduced SVM algorithm that is Incremental Reduced Support Vector Machine (IRSVM) algorithm[10] will be discussed. The mainline of IRSVM is that it automatically and incrementally selects informative data points from the

entire training dataset to generate the rectangular kernel matrix for further processing. The nonlinear separating surface (17) is a linear combination of kernel functions $\{1, K(., \bar{A}_1^{'}, \bar{A}_2^{'} \ldots, , \bar{A}_{\bar{m}}^{'})\}$. This separating surface is of the form:

$$\sum_{i=1}^{\bar{m}} K(x^{'}, \bar{A}_i^{'}) \bar{D}_{ii} \bar{u}_i = \gamma \tag{3.1}$$

As (3.1) is a linear combination of a catalogue function set [11] that consists of kernel functions $\{1, K(., \bar{A}_1^{'}, \bar{A}_2^{'} \ldots, , \bar{A}_{\bar{m}}^{'})\}$ generated by the reduced set $\bar{A}$. It can be easily inferred, if the kernel functions in the catalogue function set are very "highly correlated", the hypothesis space spanned by this catalogue function set will be very limited.To overcome this problem of using limited information and use "uncorrelated" points IRSVM is an appropriate available solution. As IRSVM concatenates sequentially the kernel function into the catalogue function set, only when the function is "unsimilar" to the current set and carrying sufficient extra information over the current set. The algorithm of IRSVM starts with a very small reduced set $\bar{A}$, typically a size of 2 and concatenate a new data point A_i into the reduced set only when the extra information carried in the vector $K(A, A_i^{'})$ with respect to the column space of $K(A, \bar{A}_i^{'})$ is significant. The decision for adding the new data point in present data set can be done by solving a least square problem. In this way the similar function sets are rejected from further usage and hence the size of the dataset gets reduced, and very efficiently. For convenience, $K(A, \bar{A}_i^{'}) \in R^{m*\bar{m}}$ is represented by $\bar{K}$. The least squares problem that needs to be solved is

$$\min_{\beta \in R^{\bar{m}}} \parallel \bar{K}\beta - K(A, A_i^{'}) \parallel_2^2 \tag{3.2}$$

where $\beta \in R^{\bar{m}}$ is a free vector variable and $\bar{K}\beta \in R^m$ is a linear combination of the functions $K(A, \bar{(A)}_i^{'})$, $i = 1, \ldots, \bar{m}$ that represents the column space of $K(A, \bar{(A)}^{'})$. From [12],it is evident that calculating the optimal solution β^* of problem (3.2) is equivalent to solve equation:

$$\bar{K}^{'} \bar{K} \beta = \bar{K}^{'} K(A, \bar{A}_i^{'}) \tag{3.3}$$

As in IRSVM, all the data points are "unsimilar", it can be deduced that the columns of the rectangular kernel matrix generated by the initial reduced set are linear independent. So the IRSVM algorithm (the sequential version) keeps the independence property throughout the whole process, and hence the least squares problem (3.2) has a unique solution β^* given as,

$$\beta^* = (\bar{K}^{'}\bar{K})^{-1}\bar{K}^{'}K(A, \bar{A}_i^{'}) \tag{3.4}$$

The distance r from $K(A, A_i^{'})$ to the column space of $\bar{K}$can be calculated as

$$r = \| \bar{K}\beta^* - K(A, A_i^i) \|_2 \tag{3.5}$$

The excess information carried in $K(A, A_i^{'})$ over $K(A, \bar{A}_i^{'})$ is given by r^2 [13]. As the size of the reduced set is very small, hence it does not lead to any computational difficulty in solving the least squares problem, though it is to be solved multiple times in the entire process. There are two methods for sequentially increasing the dataset, IRSVM- Sequential Versions and IRSVM- Batch Version. The algorithm of each version is discussed below.

3.1.1 IRSVM Algorithm-Sequential Version

Let $\delta > 0$ be a given threshold.

1. Choose a very small random subset matrix $\bar{A}_0 \in R^{\bar{m}*n}$ from the training data matrix $A \in A \in R^{m*n}$, say $\bar{m} = 2$, as an initial reduced set, and generate the reduced kernel matrix$K(A, \bar{A}_0^{'})$. Let $\bar{A}_{new} = \bar{A}_0$.

2. Select $A_j \in A \setminus \bar{A}_0$ and compute the distance r from the kernel vector$K(A, \bar{A}_j^{'})$ to the column space of$K(A, \bar{A}_{new}^{'}$ by using (3.5).

3. If$r > \delta$ then $\bar{A}_{new}^{'} = \bar{A}_{new}^{'} \cup A_j$.

4. Repeat Step 2) until several successive failures happened in 3), then the resulting $K(A, \bar{A}_{new}^{'})$ is our final reduced kernel.

5. Apply the Newton-Armijo Algorithm [section 5] to solve the objective function (16):

$$\min_{(\bar{u},\gamma)\in R^{\bar{m}+1}} \phi_{\alpha,v}(\bar{u}\gamma) \tag{3.6}$$

   where the reduced kernel $K(A, \bar{A}^{'})$ in (16) is that obtained in Step 4).

6. The separating surface is given as follows:

$$K(x^{'}, \bar{A}')\bar{D}\bar{u} = \gamma \tag{3.7}$$

   where $(\bar{u}, \gamma) \in R^{\bar{m}+1}$ is the unique solution to (3.6).

7. A new point $x \in R^n$ is classified into class $+1$ or -1 depending on whether the step function:

$$(K(x^{'}, \bar{A}')\bar{D}\bar{u} - \gamma)_* \tag{3.8}$$

is $+1$ or zero, respectively.

In Algorithm 3.1, the equation (3.3) has to be solved multiple times. The time complexity of (3.3) is $O(\bar{m}^3)$, where $\bar{m}$ is the current size of the reduced set. The main cost of solving this equation depends on $\bar{K}^{'}\bar{K}$, but not on $K(A, A_j^{'})$. [13].

In order to speed up Algorithm 3.1the batch of data points can be used to generate the new reduced set. The algorithm of IRSVM based batch method is as follows:

3.1.1.1 IRSVM Algorithm -Batch Version

Let $\delta > 0$ be a given threshold and l be a given batch size.$\bar{B}_i \in R^{l*n}$ denotes a batch of data and r denotes a distance vector.

1. Choose a very small random subset matrix $\bar{A}_0 \in R^{\bar{m}*n}$ from the training data matrix $A \in R^{m*n}$, say $\bar{m} = 2$, as an initial reduced set, and generate the reduced kernel matrix $K(A, \bar{A}_0^{'})$. Let $\bar{A}_{new} = \bar{A}_0$.

2. For A_j to $A_{j+1} \in A \backslash \bar{A}_0$, form a batch $\bar{B}_i$.

3. For each $\bar{B}_j$ compute the distance vector r, which consists of individual distances from each of the columns of $K(A, \bar{B}_j^{'})$ to the column space of $K(A, \bar{A}_{new}^{'})$ by using (3.5).

4. For each $A_j \in \bar{B}_i$, $\bar{A}_{new} = \bar{A}_{new} \cup A_j$ if the corresponding distance value exceeds the threshold δ.

5. Repeat Step 3) until several successive failures in adding new points. Then $K(A, \bar{A}_{new}^{'})$ is our resulting reduced kernel.

6. Apply the Newton-Armijo Algorithm [section 5] to solve the objective function (16):

$$\min_{(\bar{u},\gamma) \in R^{\bar{m}+1}} \phi_{\alpha,v}(\bar{u}, \gamma) \tag{3.9}$$

where the reduced kernel $K(A, \bar{A}^{'})$ in (16) is that obtained in Step 4).

7. The separating surface is given as follows:

$$K(x^{'}, \bar{A}')\bar{D}\bar{u} = \gamma \tag{3.10}$$

where $(\bar{u}, \gamma) \in R^{\bar{m}+1}$ is the unique solution to (3.6) and $x \in R^n$ is a free input space variable of a new point.

8. A new point $x \in R^n$ is classified into class $+1$ or -1 depending on whether the step function:

$$(K(x^{'}, \bar{A}')\bar{D}\bar{u} - \gamma)_* \tag{3.11}$$

is $+1$ or zero, respectively.

This modified algorithm significantly reduces the frequency of solving (3.3). From this algorithm, it can be noticed that "similar" points in the same batch might be added into the reduced set, eventually resulting in increment in size of reduced set. But for the reasons of speed, this algorithm can be considerable.

3.2 Systematic Reduced Set Selection

In this section, Systematic Reduced Set Selection (SSRSVM) [14] , another approach to generate the reduced set is discussed with algorithm. The inspiration for developing SSRSVM is arises with the key idea of SVM, the SVM classifier can be represented by support vectors with the misclassified points as a part of support vectors. Instead of randomly generating the reduced set, it begins with an extremely small reduced set and concatenate the misclassified points into the reduced set iteratively using existing classifier. In Fig. 3.1, there are two different types of misclassified points, the algorithm uses those points to update the reduced set. The algorithm only needs to augment the columns which are generated by the new points in the reduced set.

3.2.0.2 Systematic Sampling RSVM Algorithm

1. Randomly select an extremely small portion data points, such as $\bar{m} = 5\}$, from the entire training data matrix $A \in R^{m*n}$ as an initial reduced set which is represented by $\bar{A}_0 \in R^{\bar{m}*n}$.

2. Generate the reduced kernel matrix $K_\sigma(A, \bar{A}_0^{'})$ and perform RSVM algorithm to generate a tentative separating surface represented by $f(x) = 0$

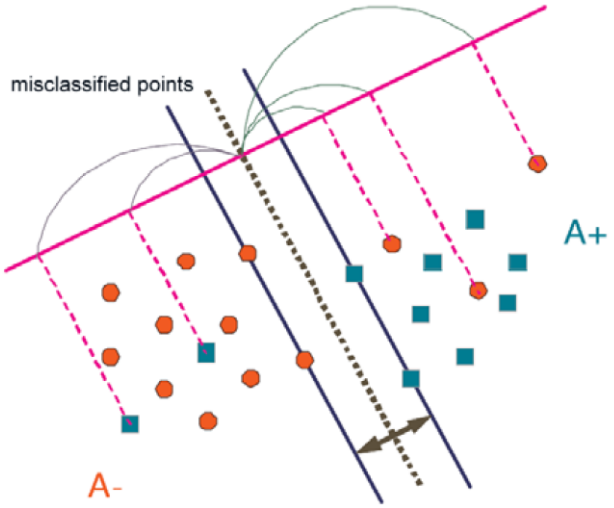

FIGURE 3.1: Illustration of the idea of systematic sampling RSVM algorithm [15] .

3. Use the separating surface to classify the point which is in the training set but not in the current reduced set. Let $\bar{I}_+$ be the index set of misclassified points of positive example. That is, $\bar{I}_+ = \{i|f(A_i \leq 0, A_i \in A_+\}$. Similarly, $\bar{I}_- = \{i|f(A_i < 0, A_i \in A_-\}$.

4. Sort the set $\bar{I}_+$ by the absolute value of $f(A_{\bar{I}_+})$ and the set $\bar{I}_+$ by $f(A_{\bar{I}_+})$ respectively. We named the resulting sorted sets $\bar{S}_+)$ and $\bar{S}_-)$

5. Partition $\bar{S}_+)$ and $\bar{S}_-)$ into several subsets respectively such that each subset has nearly equal number of elements just like Fig. 1. That is, let $\phi \neq \bar{sp}_i \subset \bar{S}_+, \forall 1 \leq I \leq kwherekisthenumberofsubsets._+) = \bar{sp}_1 \cup \bar{sp}_2 \cup \ldots.\bar{sp}_k \cup$ and $\bar{sp}_i \cap \bar{sp}_j = \phi, \forall ij, 1 \leq i, j \leq k$. Similarly, $\bar{S}_-) = \bar{sn}_1 \cup \bar{sn}_2 \cup \ldots.\bar{sn}_k \cup$ and $\bar{sn}_i \cap \bar{sn}_j = \phi, \forall ij, 1 \leq i, j \leq k$. Then, choose one point from each subset and add these points into $\bar{A}_0$ to generate a new reduced set in place of $\bar{A}_0$.

6. Repeat steps 2-5 until the validation set correctness has arrived at the threshold which is user pre-specified.

7. Output the final classifier, $f(x) = 0$ Fig. 1. Illustrate the idea of systematic sampling RSVM algorithm.

As mentioned above, the misclassied points can be treated as parts of support vectors and added into the reduced set. Thus for taking the information of the misclassied points uniformly, systematic sampling approach selects the same number of misclassified points in different distance from them to separating surface.The algorithm uses incremental model selection scheme that prevents addition of so many misclassied points in the same distance level.

Chapter 4

Big Data Applications

Sensor networks, Social networks, Internet documents, etc. are the prominent source to collect large amount of data. The analysis of this massive data is a very critical and cumbersome task. The MapReduce framework for nonlinear model is one of the available solutions to efficiently solve a nonlinear support vector machine with massive dataset. MapReduce is a programming model for processing and generating large data sets with a parallel, distributed algorithm on a cluster. MapReduce is useful in a wide range of applications, including distributed pattern-based searching, distributed sorting, web link-graph reversal, Singular Value Decomposition. MapReduce program mainly composed of two subprograms, Map() and Reduce(). Map procedure performs filtering and sorting and Reduce() procedure performs summary operation, aggregates subresults. By sub dividing the big problem into smaller segments, it is possible to save memory requirements and reduce computational complexity. The entire data is split into n stratified disjoint subsets at first which signifies the Map-step. RSVM model is associated with a reduced subset which is much smaller than the corresponding data subset. After processing these n RSVM models independently, these models will be combined and concluded in the Reduce-step. In the Reduce-step, firstly all the instances from these n reduced subsets are collected, and then encoding of each of these instances is done as an n-dimensional vector by its predicted values made by these n RSVM models. In fig. 4.1, there is a Big Table which has a much more compact representation for the reduced subsets, where the data size is equal to the size of all the reduced subsets and the n-dimensional features are described by those n RSVM models. Once the compressed data is extracted as described above, final MRRSVM can be concluded by applying linear SVM to compressed data. The detailed training procedure is presented in Algorithm 4.1. MRRSVM prediction phase also consists of Map and Reduce steps. In Map-step, each testing data instance is first encoded as an n-dimensional vector by using the RSVM model prediction values learned from MRRSVM training phase. The testing

data prediction is then made by the MRRSVM model (i.e., the Reduce-step), presented in Algorithm 4.2.

4.1 MapReduce RSVM training

Require: The big dataset A.
Ensure: RSVM models $\{model_i\}_{(i=1)}^{n}$, the reduced set of each subset $\{\tilde{A}\}$, the linear SSVM model (final model) with new features generated by $\{\text{model}_i\}_{(i=1)}^{n}$.

Split A into n subsets and their associated reduced sets: $\{A_i\}_{(i=1)}^{n}$ and $\{\tilde{A}_i\}_{(i=1)}^{n}$.

Learn the RSVM $\{model_i\}$ for each subset A_i with its reduced set $\tilde{A}_i$

Generate a new representation $B \in R^{(\sum \tilde{l}_i)*n}$ (i.e., the Big Table) for the reduced subsets and the j^{th} row of B is represented as follows:
$x_j^{new} = [model_1(x_j), model_2(x_j), \ldots, model_n(x_j)]^T \in R^n$ for $x_j \in \{\tilde{A}_i\}_{(i=1)}^{n}$.

Learn the linear SSVM model with B: final-model ⟵linear-SVM-train(B)

Return $\{model_i\}_{(i=1)}^{n}$, $\{\tilde{A}_i\}_{(i=1)}^{n}$ and final-model.

4.2 MapReduce RSVM prediction

Require: A testing instance x_t , $\{model_i\}_{(i=1)}^{n}$, $\{\tilde{A}_i\}_{(i=1)}^{n}$ and final-model.
Ensure: Predicted label

1. Generate the new representation for the testing instance x_t:
   $x_j^{new} = [model_1(x_j), model_2(x_j), \ldots, model_n(x_j)]^T \in R^n$
2. Predicted label ⟵sign$(final - model(\mathrm{x}_t^{new}))$

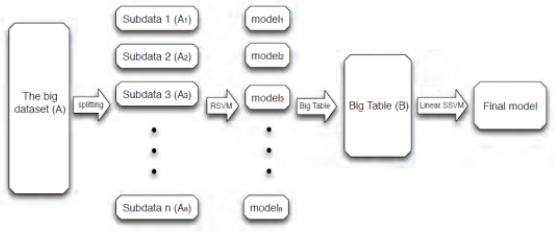

FIGURE 4.1: MapReduce RSVM Framework .

Chapter 5

Conclusion

In today's era of technology and development of data mining, there is a requirement of efficient, robust and low complexity algorithms with high accuracy. SVM is one of the best tools available which provides good results with low computation time. There are various modified SVM formulations available, out of which SSVM is a formulation that is a smooth unconstrained optimization reformulation of the traditional quadratic program associated with SVM. The fast Newton-Armijo method is required for solving the SSVM which needs the optimization fuction to be twice differentiable. In SSVM the size and time problem. To overcome these problems, RSVM is an emerging solution as it uses very small subset of data typically 10Even with enormous applications of RSVM as were demonstrated in simulation section, RSVM is not answerable to certain questions like how much percentage of data is enough for characterization, what if the randomly selected dataset have very minute information. IRSVM and SSRSVM are two modified RSVM approaches that were discussed as both the approaches help to overcome the issues associated with RSVM. The IRSVM approach provides a mechanism to determine the size of the reduced set automatically and the generated reduced set has more representative than one by purely random election. SSRSVM selects the informative data points to form the reduced set *iteratively*. SSRSVM also starts with an extremely small initial reduced set and adds a portion of missclassified points into reduced set iteratively based on the current classifier until the validation set correctness is large enough.In the last section, as an application of RSVM, RSVM is embed in the MapReduce framework because it is expected that the MapReduce framework can be an important technique in Big Data aeon.

Appendix A

Newton–Armijo Algorithm

Newton–Armijo Algorithm for solving the optimization problems (2.16), (3.6), and (3.9).
Start with any $(w^0, \gamma^0) \in R^{(n+1)}$. Having (w^i, γ^i), stop if the gradient of the objective function of (2.16, 3.6, and 3.9) is zero, that is $\nabla\Phi_a(w^i, \gamma^i) = 0$. Else compute $(w^{(i+1)}, \gamma^{(i+1)})$ as follows:

1. **Newton direction**: Determine direction $d^i \in R^{(n+1)}$ by setting equal to zero the linearization of $\nabla\Phi_a(w, \gamma)$ around (w^i, γ^i) which gives $n+1$ linear equations in $n+1$ variables:

$$\nabla^2\Phi_a(w^i, \gamma^i)d^i = -\nabla\Phi_a(w^i, \gamma^i)' \tag{A.1}$$

2. **Armijo Step size**: Choose a step size $\lambda_i \in R$ such that

$$\begin{gathered}(w^{i+1}, \gamma^{i+1}) = (w^i, \gamma^i) + \lambda_i d^i \\ \text{where} \quad \lambda_i = \max\{1, \tfrac{1}{2}, \tfrac{1}{4}, ...\} \quad \text{such that}\end{gathered} \tag{A.2}$$

$$\begin{gathered}\phi_a(w^i, \gamma^i) - \phi_a(w^{i+1}, \gamma^{i+1}) \geq -\delta\lambda_i\nabla\Phi_a(w^i, \gamma^i)d' \\ \text{where} \quad \delta \in (0, \tfrac{1}{2})\end{gathered} \tag{A.3}$$

Bibliography

[1] C. K. I. Williams and M. Seeger. Using the nystrom method to speed up kernel machines. *Advances in Neural Information Processing Systems*, 2000. URL http://www.kernel-machines.org.

[2] P. Bartlett B. Scholkopf ___, A. Smola and eds. D. Schuurmans. Generalized support vector machines. *MIT Press,Cambridge*, 2000.

[3] V. Cherkassky and F. Mulier. Learning from data - concepts, theory and methods. *John Wiley Sons, New York*, 1998.

[4] ___. Successive overrelaxation for support vector machines. *IEEE Transactions on Neural Networks*, 1999.

[5] ___. Lagrangian support vector machines. *Machine Learning Research*, 2000.

[6] Yuh-Jye Lee and O. L. Mangasarian. Ssvm: A smooth support vector machine. *Computational Optimization and Applications*, 1999. URL ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/99-03.ps.

[7] Chunhui Chen and O. L. Mangasarian. Smoothing methods for convex inequalities and linear complementarity problems. *Mathematical Programming*, 1995.

[8] P. S. Bradley and O. L. Mangasarian. Massive data discrimination via linear support vector machines. *Optimization Methods and Software*, 2000.

[9] Y.-J. Lee and O. L. Mangasarian. Rsvm: Reduced support vector machines. technical report. *Proceedings of the First SIAM International Conference on Data Mining, Chicago*, 2001.

[10] Increased SVM Paper. Review paper. *Review paper*, Review paper.

[11] Review paper. Review paper. *Review paper*, Review paper.

[12] O. L. Mangasarian. Nonlinear programming. *SIAM, Philadelphia*, 1994.

[13] G. H. Golub and C. F. Van Loan. Matrix computations. *The John Hopkins University Press, Baltimore*, 1996.

[14] Variant Method Paper. Review paper. *Review paper*, Review paper.

[15] Y. Y. Ou C. Y. Chen Y. J. Oyang, S. C. Hwang and Z. W. Chen. An novel learning algorithm for data classification with radial basis function networks. *Proceedings of the 9th International Conference on Neural Information Processing*, 2001.