Improved Support Vector Machine Generalization Using Normalized Input Space

Shawkat Ali[1] and Kate A. Smith-Miles[2]

[1] School of Information Systems, Central Queensland University, QLD 4702, Australia
[2] School of Engineering and Information Technology, Deakin University, VIC 3125, Australia
s.ali@cqu.edu.au, katesm@deakin.edu.au

Abstract. Data pre-processing always plays a key role in learning algorithm performance. In this research we consider data pre-processing by normalization for Support Vector Machines (SVMs). We examine the normalization affect across 112 classification problems with SVM using the rbf kernel. We observe a significant classification improvement due to normalization. Finally we suggest a rule based method to find when normalization is necessary for a specific classification problem. The best normalization method is also automatically selected by SVM itself.

Keywords: Normalization, Classification, Support Vector Machines.

1 Introduction

Support Vector Machines (SVMs) [1-3] are a comparatively new machine learning tool, which adopted statistical learning theory. From the beginning SVMs are solving classification, regression and novelty detection tasks with better generalization compared with traditional learning algorithms. Let us consider, a linear SVM over a training set $\{(\vec{x}_i, y_i), i = 1, \cdots, n\}, x_i \in \Re^m, y_i \in \{-1,1\}$ where the task is to estimate a weight vector $\vec{\omega}$ and a scalar bias factor b to construct an optimal hyperplane defined by $\vec{\omega} \cdot \vec{x} + b$ as follows:

$$\min_{\vec{\omega},b} \frac{1}{2}\langle \vec{\omega} \cdot \vec{\omega} \rangle, \textit{subject to } y_i(\langle \vec{\omega} \cdot \vec{x}_i \rangle + b) \geq 1 \tag{1}$$

which leads to the dual optimisation problems such as:

$$\begin{aligned} &\max_{\alpha} \quad W(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \vec{x}_i \cdot \vec{x}_j \rangle \\ &\textit{subject to } \sum_i y_i \alpha_i = 0, \quad \alpha_i \geq 0 \end{aligned} \tag{2}$$

Now the goal is to estimate the optimisation parameter properly. The generalization performance of SVMs and many other function estimation algorithms depend on appropriate parameter estimation. Data normalization has already shown significant improvement of generalization performance with SVM [4-6]. Herbrich and Graepel

A. Sattar and B.H. Kang (Eds.): AI 2006, LNAI 4304, pp. 362 – 371, 2006.
© Springer-Verlag Berlin Heidelberg 2006

[7] have shown in terms of image classification that normalization is a data pre-processing method which plays an important role in SVM classification with real world problems. In this research we examine the issue of data normalization across a wide range of classification problems and propose a simple rule based methodology to find where normalization is beneficial. Finally we select an appropriate normalization method with the help of popular learning algorithms with priority given to SVM itself for a particular classification problem.

Our methodology seeks to understand the characteristics of the data (for particular classification problems), and explain why a normalized input space might offer better generalization. First we classify a wide range of natural classification problems with the most popular learning algorithms including SVM with radial basis function (rbf) kernel. Automatic rbf kernel parameter selection is described in [8]. After that we use all these problems with SVM rbf with modified input space with different normalized methods. Then we identify the dataset characteristics matrix by statistical measures to generate rules where normalization is necessary or not. Finally we use a set of popular learning algorithms to predict the appropriate normalization method selection for a particular classification problem with SVM. All 112 classification problems are considered from the UCI Repository [9] and Knowledge Discovery Central [10] database. Over all the experiments we consider 10 Fold Cross Validation (10FCV) performances.

Our paper is organized as follows: In Section 2, we provide some theoretical frameworks regarding SVM input space normalization. Section 3 analyses the experimental results. All statistical measures to identify the dataset characteristics matrix are summarized in Section 4. A priori normalization/non normalization method selection will be explained in 5. The most suitable normalization method selection is described in section 6. Finally we conclude our research towards the end in Section 7.

2 SVMs Input Space

One specialty of SVM is that it transforms the data by adopting kernel. During the transformation some kernels essentially normalize the data points automatically. For example, rbf kernel [3] as follows:

$$K\left(\vec{x}_i, \vec{x}_j\right) = \exp\left(-\frac{\left\|\vec{x}_i - \vec{x}_j\right\|^2}{2\sigma^2}\right) \text{ where } \sigma > 0 \tag{3}$$

We propose some additional normalization for SVM before we transform the data points in the kernel space. The normalized data is used in the SVM kernel space rather than natural data input. We provide the preference on global attributes values normalization rather than single attribute normalization. Two types of normalization [11] are examined in this research as described in the following sections.

2.1 Min-Max Normalization

The formulation of the Min-Max normalization is:

$$D'(i) = \frac{D(i) - \min(D)}{\max(D) - \min(D)} * (U - L) + L \tag{4}$$

where D' is the normalized data matrix, D is the natural data matrix and U and L are the upper and lower normalization bound.

This type of normalization method is used to normalize the data matrix into a desired bound. The most popular bound is between 0 and 1. We also change the bound values to between 0 and –1, as well as between 1 and –1.

2.2 Zero-Mean Normalization

The formulation of the Zero-Mean normalization is as follows:

$$D' = \left(\frac{D - \overline{D}}{\sigma} \right) \tag{5}$$

where $\overline{D}$ is the mean of the natural data matrix D and σ is the standard deviation of the same data matrix. In this normalization method, the mean of the normalized data points is reduced to zero. Due to this, the mean and standard deviation of the natural data matrix is required.

The normalization performance of UCI dataset 'xab' is shown in Figure 1. The natural data distribution is highly positive skewed. But the normalized 'xab' dataset is shown balanced skewed.

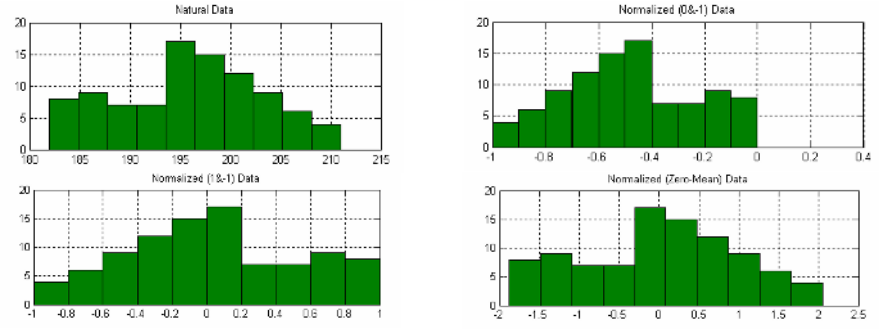
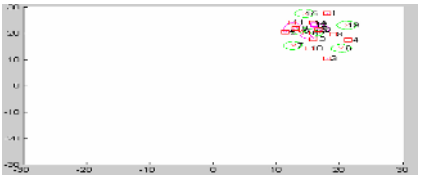


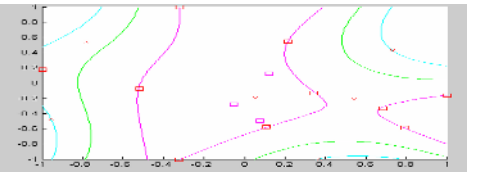
Fig. 1. Graphically represents the natural and different normalized distribution of 'xab' UCI dataset. The data distribution scale is completely different after getting normalization.

The hyperplane construction procedure of SVM with rbf kernel for UCI data set 'wpbc' is shown in Figure 2. During the classification of SVM using natural data points, several optimal decision boundaries are constructed, but only a single

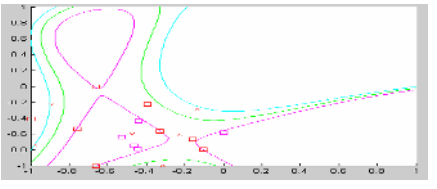
boundary is constructed for normalized data points. The misclassification error is higher for natural data points than normalized data points.



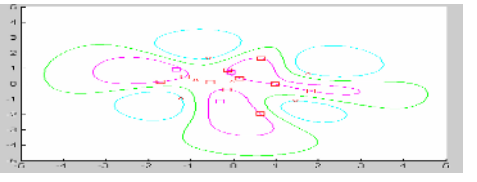
(a) SVM with rbf kernel: natural data points learning.



(b) SVM with rbf kernel: Min-Max normalized (1&-1) data points learning.



(c) SVM with rbf kernel: Min-Max normalized (0&-1) data points learning.



(d) SVM with rbf kernel: Zero-Mean normalized data points learning.

Fig. 2. The hyperplane construction procedure for natural and normalized data points of UCI 'wpbc' data set. SVM considers the rbf kernel with width 1.

We now compare the scaling method with normalization performance. This method also transforms the data points within a certain range.

2.3 Log Scaling

The formulation of the log scaling [12] method of the data matrix is as follows:

$$D'(i) = \log(D(i)) \tag{6}$$

This type of scaling transforms the data points within a logarithmic scale as shown in Figure 3.

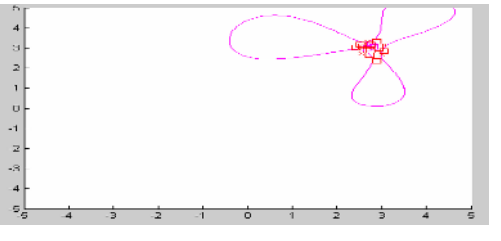


Fig. 3. The hyperplane construction procedure for log scaled data points of UCI 'wpbc' data set. SVM considers the rbf kernel width 1.

It is observed that log scaled data points are very difficult to classify by SVM. The classification error is higher than normalized data classification performance.

3 Experimental Results

3.1 Classification and Computational Performance

We tested the performance of the different normalization methods with SVM. All the methods average performance based on accuracy and computational time is shown in Figure 4 and 5. We observed normalization performance is generally much better than natural data sets classification performance for SVM. The log scaling performance is not carried out the similar significance with normalization. In terms of computational cost, SVM needed more time to classify the normalized data points than natural data. The log scaling method need less time but the accuracy performance is the worst comparing with others methods.

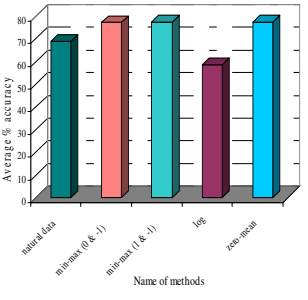


Fig. 4. Average accuracy performance of different normalization and scaling methods compared to natural data points classification performance

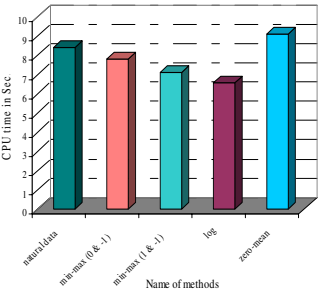


Fig. 5. Average computational performance of different normalization and scaling method with comparing natural data points classification performance

We observed among the different methods of normalization 84% of the 112 data sets performed better with some kind of normalization. The rest of the data sets showed better classification performance without normalizing or scaling. We found some visual reason why normalization is not always effective. First, those data sets with a combination of negative continuous and discrete attributes values, second a combination of categorical and continuous and finally those holding categorical attributes values that all are not aspect data normalization. We observed after getting normalization these data points become very close in the feature space, where optimal hyperplane construction procedure is complex.

In the following section we describe the methodology we use to assist in the appropriate selection of the best method for a given dataset. First each dataset is described by a set of measurable meta characteristics; we then combine this information with the performance results; and finally use a rule-based induction method to provide rules describing when each method for SVM is likely to perform well.

4 Datasets Characteristics Measurement

Each dataset can be described by simple, distance and distribution based statistical measures [13,14]. These three sets of measures characterized the datasets in different

ways. First, the simple classical statistical measures identify the data characteristics based on variable to variable comparisons. Then, the distance based measures identify the data characteristics based on sample to sample comparisons. Finally, the density based measures consider the single data point from a matrix to identify the datasets characteristics. We average most of statistical measures over all the variables and take these as global measures of the dataset characteristics.

4.1 Simple Statistical Measures

Descriptive statistics can be used to summarise any large dataset into a few numbers that contain most of the relevant characteristics of that dataset. The following table lists the statistical measures used in this work as provided by the Matlab Statistics Toolbox [12] and some other different sources [15, 16] as follows: Geometric mean, Harmonic mean, Trim mean, Standard deviation, Interquartile Range, Max. and Min. eigenvalue, Skewness, Kurtosis, Correlation Coefficient and Prctile.

4.2 Distance Based Measures

Distance based measures calculate the dissimilarity between samples. We measure the euclidean, city block and mahalanobis distance between each pair of observations for each dataset as follows: Euclidean distance, City Block distance and Mahalanobis distance.

4.3 Distribution Based Measures

The probability distribution of a random variable describes how the probabilities are distributed over the various values that the random variable can take on. We measure the probability density function (pdf) and cumulative distribution function (cdf) for all datasets by considering different types of distributions as follows: Chi-square pdf, Normal pdf, Binomial pdf, Exponential pdf, Gamma pdf, Lognormal pdf, Rayleigh pdf, Chi-square cdf, Normal cdf, Discrete uniform cdf, F pdf, Poisson pdf, Student's t pdf, and Noncentral T cdf (nctcdf).

These measures are all calculated for each of the datasets to produce a dataset characteristics matrix. Finally by combining this matrix with the performance results we can derive rules to suggest when certain methods are appropriate.

5 Rule Generations

The trial-and-error approach could be a very common procedure to select the normalization or non normalization method for SVM classification. It is a computationally complex task to find an appropriate method by following this procedure. If we are interested in applying a specific method to a particular problem we have to consider which method is more suitable for which problem. The suitability test can be done from rules developed with the help of the data characteristics properties.

Rule based learning algorithms, especially decision trees (also called classification trees or hierarchical classifiers), are a divide-and-conquer approach or a top-down

induction method, that have been studied with interest in the machine learning community. Quinlan [17] introduced C4.5 and then C5.0 algorithms to solve classification problems. C5.0 works in three main steps. First, the root node at the top node of the tree considers all samples and passes them through to the second node called 'branch node'. The branch node generates rules for a group of samples based on an entropy measure. In this stage C5.0 constructs a very big tree by considering all attribute values and finalizes the decision rule by pruning. It uses a heuristic approach for pruning based on statistical significance of splits. After fixing the best rule, the branch nodes send the final class value in the last node called the 'leaf node' [17,18]. C5.0 has two parameters: the first one is called the pruning confidence factor (c) and the second one represents the minimum number of branches at each split (m). The pruning factor has an effect on error estimation and hence the severity of pruning the decision tree. The smaller value of c produces more pruning of the generated tree and a higher value results in less pruning. The minimum branches m indicates the degree to which the initial tree can fit the data. Every branch point in the tree should contain at least two branches (so a minimum number of $m = 2$). For detail formulations see [17].

Now that the characteristics of each dataset can be quantitatively measured, we can combine this information with the empirical evaluation of normalization/natural classification performance and construct the dataset characteristics matrix. Thus, the result of the jth method on the ith dataset is calculated as:

$$R_{ij} = 1 - \frac{e_{ij} - \max(e_i)}{\min(e_i) - \max(e_i)} \tag{7}$$

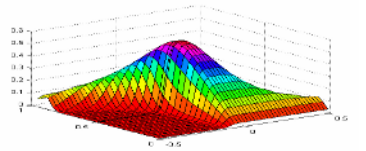
where e_{ij} is the percentage of correct classification for the jth method on dataset i, and e_i is a vector of accuracy for dataset i. The class values in the matrix are assigned based on the performance. If the normalization method is showed better performance than natural data set classification, then class value is 1, otherwise 2. Based on the 112 classification problems we can then train a rule-based classifier (C5.0) to learn the relationship between dataset characteristics and normalization/natural method performance. We split the matrix 90% to construct the model tree. The process is then repeated using a 10 fold cross validation approach so that 10 trees are constructed. From these 10 trees, the best rules are found for normalization/non normalization method selection based on the best test set results. The generalization of these rules is then tested by applying each of the randomly extracted test sets and calculating the average accuracy of the rules as discussed below in Table 1.

5.1 Rules for Normalization/Non Normalization Selection

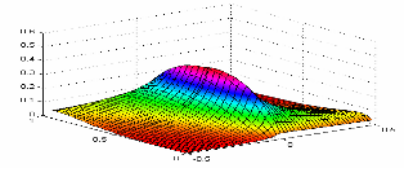
The rules for normalization method selection are generated with c = 85% and m = 4 as follows:

Rule 1: IF (nctcdf > 0) OR (median > 68.9) THEN do normalization.
Rule 2: IF (nctcdf < 0) OR (median < 68.9) THEN don't do normalization.

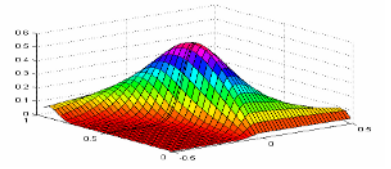
Include principle rules are data driven was to determine if data will benefit from normalization, or is it already normalized enough. For example, we can explain the nature of the rules by following noncentral T cumulative distribution function (nctcdf) value p as shown in Figure 6.



(a) Noncentral T cumulative distribution, when $p = 0.15$.



(b) Noncentral T cumulative distribution, when $p < 0$.



(c) Noncentral T cumulative distribution, when $p > 0$.

Fig. 6. Noncentral T cumulative distribution with different p values that explains the nature of data distribution

Noncentral T cumulative distribution [20-21] is highly positively skewed with respect to normal distribution as shown in Figure 6(a). We observed from the experiment, when p value of the noncentral T cumulative distribution function is less than 0, then the distribution nature of the data is closely normal. In that case, data normalization is not required for SVM classification. On the other hand, when p value is greater than 0 that means data is closely distributed as like noncentral T cumulative distribution. These types of data are needed to normalization for SVM classification.The 10 fold cross validation performance of the rule generation process is summarized in Table 1.

Table 1. Confusion matrix based on 10FCV results for the normalization/nonnormalization method selection rule (Accuracy = 89.40%)

Data Condition Satisfied		Normalization Method Best	
		Y	N
	Y	88	6
	N	6	12

We summarized the best rules from 10FCV performance. Average 10FCV performance is more than 89%. However, which method is best for individual datasets has been shown to be quite data dependent. These rules might be useful to determine where normalization is most appropriate for which problem.

Now, we can decide either normalization is necessary or not. Then we need to find which normalization method is suitable for a particular problem. We analyzed the performance of each normalization and scaling method with comparing natural datasets classification performance.

We found the log scaling method is not a good way to transformed SVM input space. Due to this we consider Min-Max both method and zero-mean method to normalize the SVM input space before start mining a problem. We used a set of learning algorithms including neural network (NN), decision tree (C4.5), Naive Bayes (NB) (for details see [19]) and SVM to predict the appropriate normalization method selection for a particular classification problem. We repeat the meta learning technique with using same above data characteristics as described in section 4 and individual performance results of different normalization methods for this prediction. The class membership attribute has designed by {+1 & -1}. If any learning algorithm predict +1 that means the current method is appropriate for the present dataset. The 10FCV learning algorithms performance is shown in Table 2.

Table 2. Normalization method selection performances with different learning algorithms

Test set classification performance (% Accuracy)				
	MLP	NB	C4.5	SVM
Max-Min method [0&-1]	75	66.67	66.67	83.33
Max-Min method [1&-1]	75	50	91.67	100
Zero-Mean method	91.67	50	83.33	100

SVM has shown best performance for a specific normalization method selection. So, SVM can help itself when this algorithm needs proper normalization method.

6 Discussions

We investigate the normalization affect across a large scale classification problem with one of the popular classification algorithm SVM. The normalization of SVM input space can significantly influence the higher accuracy performance of the classification procedure. We find out the reason why some problems are not required normalization method. Some normalization method required less computational time to classify the normalized data sets rather than natural data sets. We proposed a priori rule based method when normalization is necessary for SVM with a specific problem. SVM is also used by itself to find out the most suitable normalization methods for a specific problem. The limitation of this research is we have not considered different types of data, for instance gene expression data. This methodology could be examined with bioinformatics problem. This research could also be re-examining with other popular SVM kernels.

References

1. Vapnik, V.: The Nature of The Statistical Learning Theory, Springer-Verlag, New York. (1995).
2. Vapnik, V.N.: Statistical Learning Theory, John Wiley & Sons, Inc. (1998).
3. Vapnik, V.N.: An Overview of Statistical Learning Theory, IEEE Transaction on Neural Networks, 10(5) (1999) 988-999.

4. Graf, A. and Borer, S.: Normalization in Support Vector Machines, in Proc. DAGM Pattern Recognition. Berlin, Germany: Springer-Verlag. (2001).
5. Pontil M. and Verri, A.: Support Vector Machines for 3-D Object Recognition, IEEE Trans.Pattern Anal. Machine Intell. 20 (1998) 637–646.
6. Graf, A.B.A., Smola, A.J. and Borer, S.: Classification in a Normalized Feature Space Using Support Vector Machines, IEEE Transactions on Neural Networks, 14(3) (2003) 597-605.
7. Herbrich R. and Graepel, T.: A PAC-bayesian margin bound for linear classifiers: Why SVM's work. Advances in Neural Information Processing Systems. 13 (2001).
8. Ali, S. and Smith, K.A.: Kernel Width Selection for SVM Classification-A Meta-Learning Approach, International Journal of Data Warehousing and Mining, Idea Publishers, USA. (2005) 78-97.
9. Blake, C. and Merz, C.J.: UCI Repository of Machine Learning Databases. http://www.ics.uci.edu /~mlearn/MLRepository.html, Irvine, CA: University of California. (2002).
10. Lim, T.-S.: Knowledge Discovery Central, Datasets, http://www.KDCentral.com/. (2002).
11. Kennedy, R.L., Lee, Y., Roy, B.V., Reed, C.D. and Lippman, R.P.: Solving Data Mining Problems Through Pattern Recognition, Prentice-Hall, NJ. (1997).
12. Statistics toolbox user's guide, Version 3, The MathWorks, Inc. USA. (2001).
13. Smith, K.A., Woo, F., Ciesielski, V. and Ibrahim, R.: Modelling The Relationship Between Problem Characteristics and Data Mining Algorithm Performance Using Neural Networks, C. Dagli et al. (eds.), Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Complex Systems, ASME Press, 11 (2001) 357-362.
14. Smith, K.A., Woo, F., Ciesielski, V. and Ibrahim, R.: Matching Data Mining Algorithm Suitability to Data Characteristics Using a Self-Organising Map, in A. Abraham and M. Koppen (eds.), Hybrid Information Systems, Physica-Verlag, Heidelberg, (2002) 169-180.
15. Mandenhall, W. and Sincich, T.: Statistics for Engineering and The Sciences, 4th eds. Prentice Hall. (1995).
16. Tamhane, A.C. and Dunlop, D.D.: Statistics and Data Analysis, Prentice Hall. (2000).
17. Quinlan, R.: C4.5: Programs for Machine Learning, Morgan Kaufman Publishers, San Mateo, CA. (1993).
18. Duin, R.P.W.: A note on comparing classifier, Pattern Recognition Letters, 1 (1996) 529-536.
19. Witten, I.H. and Frank, E.: Data Mining: practical machine learning tool and technique with Java implementation, Morgan Kaufmann, San Francisco, (2000).
20. Evans, M., Hastings, N. and Peacock, B.: Statistical Distributions, Second Edition, John Wiley and Sons. (1993).
21. Johnson, N. and Kotz, S.: Distributions in Statistics: Continuous Univariate Distributions-2, John Wiley and Sons. (1970).