The Effect of Attribute Scaling on the Performance of Support Vector Machines

Catherine Edwards and Bhavani Raskutti

Telstra Research Laboratories, Telstra Corporation,
770 Blackburn Road, Clayton, Victoria, Australia
{Catherine.A.Edwards, Bhavani.Raskutti}@team.telstra.com

Abstract. This paper presents some empirical results showing that simple attribute scaling in the data preprocessing stage can improve the performance of linear binary classifiers. In particular, a class specific scaling method that utilises information about the class distribution of the training sample can significantly improve classification accuracy. This form of scaling can boost the performance of a simple centroid classifier to similar levels of accuracy as the more complex, and computationally expensive, support vector machine and regression classifiers. Further, when SVMs are used, scaled data produces better results, for smaller amounts of training data, and with smaller regularisation constant values, than unscaled data.

1 Introduction

Data preprocessing has been recognised as critically important in data mining to improve both the speed and accuracy of the resultant model [1, 2]. In particular, and as will be shown, simple manipulations of the range of the input data by attribute scaling are computationally inexpensive, and can result in significant performance increases.

This paper presents an empirical investigation of the impact of attribute scaling on the performance of SVM classifiers. We focus on linear binary classifiers (Section 2) and measure the impact of scaling on the accuracy of the classifiers where accuracy is measured in terms of a general classifier goodness measure that is independent of the operating point of the classifier (Section 3). We consider three different scaling techniques that are linear transformations of the attribute space, and change the range and origin of the attribute space (Section 4). Using eight large datasets with differing properties, we study the effect of these scaling methods on classification accuracy when classifier parameters such as training set size are varied (Section 5). Our results show that attribute scaling can vastly improve the accuracy of even simple classifiers, and can thus provide a computationally inexpensive method for achieving high accuracy with a smaller training set size or a less complex classifier (Section 6).

2 Classifiers

Given a labelled m-sample: $\overrightarrow{xy}^m := \big((x_1, y_1), ..., (x_m, y_m)\big)$ of patterns $x_i \in X \subset \mathbb{R}^n$ and target values $y_i \in [0, 1]$, our aim is to find a "good" discriminating function $f : X \to \mathbb{R}$

G.I. Webb and Xinghuo Yu (Eds.): AI 2004, LNAI 3339, pp. 500–512, 2004.
© Springer-Verlag Berlin Heidelberg 2004

that scores the target class instances $y_i = 1$ higher than the background class instances $y_i = 0$. We focus on linear classifiers, namely two linear support vector machines, one ridge regression model and a simple centroid classifier.

2.1 Support Vector Machines (SVML1 and SVML2)

Given the training m-sample as described above, a learning algorithm used by SVMs [3, 4, 5] outputs a model $f_{\overrightarrow{xy}^m} : X \to \mathbb{R}$ defined as the minimiser of the regularised risk functional:

$$f \mapsto ||f||^2_{\mathcal{H}} + \sum_{i=1}^{m} L([1 - y_i f(x_i)]_+). \tag{1}$$

Here $\mathcal{H}$ denotes a reproducing kernel Hilbert space (RKHS) [5] of real valued functions $f : X \to \mathbb{R}$ and $||.||_{\mathcal{H}}$ the corresponding norm. $L : \mathbb{R} \to \mathbb{R}^+$ is a non-negative, convex loss function penalising for the deviation $1 - y_i f(x_i)$ of the estimator $f(x_i)$ from target y_i and $[\xi]_+ := \max(0, \xi)$.

The minimisation of (1) can be solved by quadratic programming [3] with the use of the following expansion known to hold for the minimiser (1):

$$f_{\overrightarrow{xy}^m}(x) = \sum_{i=1}^{m} \alpha_i y_i k(x_i, x)$$

$$||f_{\overrightarrow{xy}^m}||^2_{\mathcal{H}} = \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

where $k : X \times X \to \mathbb{R}$ is the kernel corresponding to the RKHS $\mathcal{H}$ [6, 7]. The coefficients α_i are unique and they are the Lagrange multipliers of the quadratic minimisation problem corresponding to the constraints $y_i f_{\overrightarrow{xy}^m}(x_i) > 0$.

The following two types of loss function yield two different SVMs. In both cases, $c > 0$ is a regularisation constant controlling the extent of penalisation:

- **SVML1** or **L1** with "hinge loss" $L(\xi) := c\xi$ is the SVM with linear penalty, and
- **SVML2** or **L2** with the squared hinge loss $L(\xi) := c\xi^2$ is the SVM with quadratic penalty.

2.2 Ridge Regression (RR)

In addition to the SVMs we also use a regularisation network or ridge regression predictor, RR [4, 8, 6, 7]. Formally, this predictor is closely related to SVML2, the only difference being that it minimises a modified risk function (1), with loss $c(1 - y_i f(x_i))^2$ rather than $c[1 - y_i f(x_i)]^2_+$.

2.3 Centroid Classifier (CC)

The centroid classifier [9] is a simple linear classifier with the solution,

$$f_{\overrightarrow{xy}^m}(x) = \frac{\sum_{i,y_i=+1} k(x_i, x)}{2\max(1, m_+)} - \frac{\sum_{i,y_i=-1} k(x_i, x)}{2\max(1, m_-)}$$

where m_+ and m_- denote the numbers of examples with labels $y_i = +1$ and $y_i = -1$, respectively. In terms of the feature space, the centroid classifier implements the projection in the direction of the weighted difference between the centroids of data from each class. Note that the centroid solution approximates the solution obtained by SVMs at very low values of the regularisation constant c [10].

3 Performance Measures

We have used AUC, the area under the receiver operating characteristic (ROC) curve (also known as AROC) as our performance measure. We see this as the natural metric of general goodness of a classifier, capable of meaningful results even if the target class is a tiny fraction of the data [11, 12].

We recall that the ROC curve is a plot of the true positive rate, $P(f(x_i) > \theta | y_i = 1)$, (known as precision), against the false positive rate, $P(f(x_i) > \theta | y_i = -1)$, as a decision threshold θ is varied. The concept of the ROC curve originates in signal detection but it is now used in many other areas, including data mining, psychophysics and medical diagnosis (cf. review [13, 14]). In the last case, AUC is viewed as a measure of the general "goodness" of a test, formalised as a predictive model f in our context, with a clear statistical meaning. According to Bamber's interpretation [15], AUC(f) is equal to the probability of correctly ordering two points, one x_i from the negative and the other x_j from the positive class, by allocating appropriate scores, i.e. $f(x_i) < f(x_j)$. An additional attraction of AUC as a figure of merit is its direct link to the well researched area of order statistics, via U-statistics and Wilcoxon-Whitney-Mann test [15, 16].

There are some ambiguities in the case of AUC estimated from a discrete set in the case of ties, i.e. when multiple instances from different classes receive the same score. Following [15] we implement in this paper the definition

$$\begin{aligned} AUC(f) = P(f(x_i) < f(x_j) | - y_i = y_j = 1) \\ +0.5P(f(x_i) = f(x_j) | - y_i = y_j = 1) \end{aligned}$$

expressing AUC in terms of conditional probabilities. Note that the trivial uniform random predictor has an AUC of 0.5, while a perfect predictor has an AUC of 1.

4 Scaling Methods

We define scaling as applying a linear transformation to a set of data that changes its range and origin. Specifically, scaling involves multiplying by a transformation factor a_i, and subtracting a translation factor b_i, both of which are scalars. This is always done relative to each feature, i.e. if the data has m rows of training examples and n columns of features, each column will be scaled individually. This gives an equation in terms of the attribute vector $\hat{x_i}$, the ith column of the matrix, $\forall i, 1 \leq i \leq n$:

$$\hat{z_i} = a_i \hat{x_i} - b_i$$

It is important to note that the scaling factors a_i and b_i for each type of scaling are determined by the training data alone (using only the attribute vector $\hat{x_i}$ in the training set) and that these same factors are subsequently applied to the testing data.

Note that linear scaling of each feature independently preserves the statistical distribution of the data points for that feature, while discarding information about the range and location. Classifiers only need information about the statistical distribution of the positive and negative class data points. Discarding the range and location information has several advantages:

- The score for a particular test instance requires the calculation of a dot product with the test instance vector $\hat{x}$. If one feature has a much larger range than another, the larger feature will dominate (an effect known as feature swamping [17, 2, 18]). This will obscure any classification information contained in the feature with the smaller range. This is a problem if the feature with the larger range is not very predictive. Also, those classifiers that minimise an error function that is sensitive to scale (SVML1, SVML2 and RR) will assign more weight to those features with large ranges at the expense of those features with smaller ranges. Scaling the data reduces this problem.
- Scaling also improves the numerical conditions for those classifiers that converge to a solution (SVML1, SMVL2 and RR) [17, 18]. The algorithms have both a cap on the total number of iterations performed, and a minimum change rate, which limits the optimisation of the machine. Containing the data within a small range increases the likelihood that the solution is reached within those limitations.

This paper compares three types of scaling (Mean0Stdev1, PlusMinus1 and Class-Specific) with unscaled data.

4.1 Mean0Stdev1 Scaling

Mean0Stdev1 scaling transforms the data to have a mean of 0 and a standard deviation of 1. The transformation and translation factors are as follows:

$$a_i = \frac{1}{stdev(\hat{x}_i)} \qquad b_i = \frac{mean(\hat{x}_i)}{stdev(\hat{x}_i)}$$

4.2 PlusMinus1 Scaling

This scale transforms the range of each attribute in the training set to [-1,+1] range. Note that this may not be the range of the scaled test data since the training set may not contain the actual minimum and maximum for every attribute. This is not a significant problem, as there is no requirement that the input data be within this range, however this scale will perform better the closer the sample maximum and minimum are to the population statistics. Thus, as the training set is sampled for on a random basis, as the size of this set increases, so will the performance of this scaling method. The scaling factors a_i and b_i are computed as follows:

$$a_i = \frac{2}{\max(\hat{x}_i) - \min(\hat{x}_i)} \qquad b_i = \frac{\max(\hat{x}_i) + \min(\hat{x}_i)}{\max(\hat{x}_i) - \min(\hat{x}_i)}$$

4.3 Class-Specific Scaling

This scaling method attempts to prejudice the classifier in favour of those features that are likely to be predictive. Intuitively, if for a given feature the positive and negative

classes have small variances and significantly different means, that feature is likely to be predictive. If this is the case, we can increase its influence on the classifier as discussed above by increasing the range of its data points. This approach is essentially about making feature swamping work in favour of the classifier.

The transformation factor for each attribute is based on how predictive the model estimates that attribute will be. No translation is performed. Thus,

$$a_i = \frac{mean(\hat{x}_{i+}) - mean(\hat{x}_{i-})}{var(\hat{x}_{i+}) + var(\hat{x}_{i-})} \qquad b_i = 0$$

where $\hat{x}_{i+}$ and $\hat{x}_{i-}$ represent the attribute vector for feature i for the positive and negative class instances respectively. Note that the calculation of the a_i values for all features gives a method for determining which are the most predictive features for that data set. The least predictive features can then be discarded to reduce the amount of computational resources needed for the problem.

5 Experimental Setup

In order to understand the effect of scaling under different training conditions, we considered a number of different classifier settings. First, we explored a range of training set sizes from 50 to 6,400 observations. Next, various values for the regularisation constant were tested - the ends of range (10 and 100,000), and a mid range value (1,000).

The training sets were obtained by randomly selecting a set of 6,400 examples from the main data set of 60,000 to act as the master training set. The remaining observations then became the testing set. Training sets of the appropriate sizes were then randomly extracted from this master training set (choosing the whole set when the training set size is 6,400). The training sets were then scaled using the three methods described above. The testing set was scaled simultaneously, so that the solution produced by the training set was applicable to the testing data. Each training set was then run through the four different classifiers, once with each regularisation constant value. The results were then tested using the appropriately scaled testing set, and the AUC calculated. Note that all sets of observations extracted were stratified - i.e. the proportion of positive and negative class observations was maintained.

5.1 Data Sets

Experiments were performed on eight datasets, from different domains and of different levels of difficulty. The minority class was typically between 10% and 40% of the data. As some of the data sets were smaller than others, a random, stratified selection of 60,000 observations was taken out of each and used as the data set for this experiment.

Telecommunications Churn (Churn10 and Churn31): Data on mobile phone customers of a large telecommunications carrier was used to learn to distinguish between those that churned to a competitor in the following three months and those that didn't. A set of 31 continuous and ordinal variables was used for prediction, including bill and product information. To create a second task, a subset of 10 of these predictors was se-

lected via inspection of initial results, none of which were particularly predictive. This resulted in a difficult to learn task.

Handwritten Digit Recognition (Digit): Data was downloaded from the MNIST handwritten digit database. Each observation consists of a bitmap of 28×28 continuous grayscale values, representing a handwritten digit. This was converted to lower resolution (7×7 pixels) to reduce the dimensionality of the problem. The classification task was to distinguish between the digit '0' and all other digits. To make the problem more challenging, only the top 3 rows of pixels were used, and pixels near the corners which contain little information were discarded, resulting in a 17 dimensional set.

Forest Cover Type (Forest): Data was downloaded from the UCI KDD repository. 30×30 metre cells of forest are classified into one of 7 cover types based on the cell's dominant tree type. The two most populous classes were extracted, and the classification task was to distinguish between these classes. 10 continuous predictors were used.

Housing Mortgage (Housing): Data was downloaded from the U.S. Census Bureau 5% Public Use Microdata Sample (PUMS) containing individual records of the characteristics of a 5% sample of housing units for the state of Florida. Amongst all housing units which had a mortgage, the binary classification task was to distinguish between those for which the mortgage had been paid off and those for which it hadn't. There were 12 continuous or ordinal predictors.

Intrusion Detection (Intrusion): This dataset consists of a random sample of the intrusion detection data used for the 1999 KDD Cup competition. The classification task was to distinguish between normal use and intrusion. The 10 predictors used were a subset of all continuous predictors available with the data, as certain continuous predictors were omitted to make the problem more challenging.

Marital Status (Married): Data was again downloaded from the U.S. Census Bureau PUMS. From this a 1% sample of individual records from the state of California was extracted. The binary classification task was to distinguish between individuals who have been married (whether currently married or not), with individuals who have never been married. The predictors were 11 continuous variables.

Weather Season Prediction (Weather): Data, consisting of 8 continuous or ordinal predictors, was downloaded from the website of the Tropical Atmosphere Ocean project. It contains meteorological measurements from a grid of weather buoys in the Pacific Ocean. Hourly measurements for all buoys over the period from May 1999 to April 2000 were downloaded. The classification task was to distinguish readings made during the northern hemisphere Autumn months from those made in other months.

6 Results

In order to lend statistical significance to our results, performance measurements for each experimental setting were obtained using 10 different randomisations for each setting, computing the mean and standard error of the scores obtained with each set.

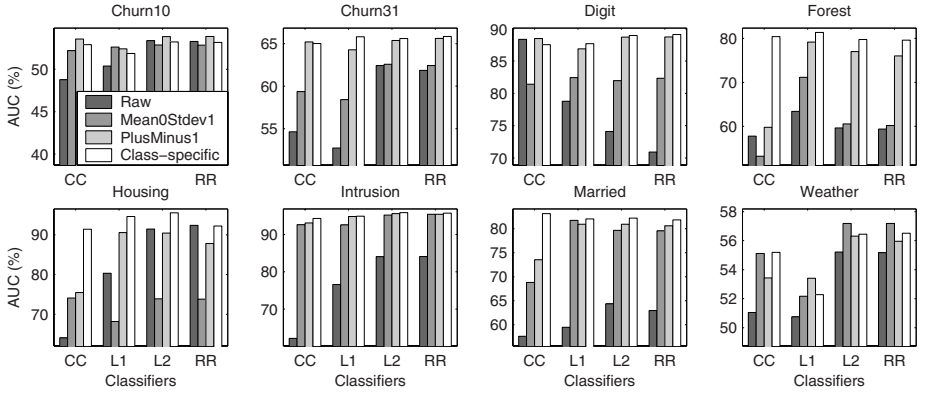

Fig. 1. Impact of scaling on different data sets. Results presented for each of the classifiers: CC (centroid), L1 (SVML1), L2 (SVML2) and RR (ridge regression) and the four scaling methods are averaged over all 10 randomisations, over all training set sizes and where appropriate, over all regularisation constants

Each of the 10 random selections were seeded to ensure that they were consistent across all other experimental settings, i.e. the test and training sets were the same across all settings for a particular randomisation, training set size and data set.

6.1 Classifier and Data Set Interactions

Figure 1 shows the effect of scaling on classifier performance for each data set. The x-axis shows bars grouped into four clusters corresponding to the four classifiers used: CC (centroid), L1 (SVML1), L2 (SVML2) and RR (ridge regression). Each cluster contains four scores, one for each different scaling method: unscaled, Mean0Stdev1, PlusMinus1 and Class-Specific, as shown in the legend. The performance is measured using the mean of the AUC over all randomisations (y-axis). Scores are averaged over all training set sizes, and all regularisation constant values (where appropriate).

Error bars have not been shown in this graph, as the standard error is consistently low. The mean standard error is 0.28% AUC, and the maximum is 1.28% AUC for the forest data set, for SVML1, using the Mean0Stdev1 scaling method. Further, the standard error only exceeds 1% AUC in 3 out of 128 cases.

As seen from Figure 1, there is no single classifier that is the best for all data sets. However, scaled data results in better performance than unscaled data in 30 of the 32 classifier/data set combinations. In particular, the Class-Specific scaling method tends to produce consistent improvements in performance compared to unscaled data, for all classifiers, for six datasets: churn31, forest, housing, intrusion, married and weather. In general, it tends to be the best scaling choice with best or equal best performance in 24 out of the 32 experimental situations shown. PlusMinus1 scaling also tends to

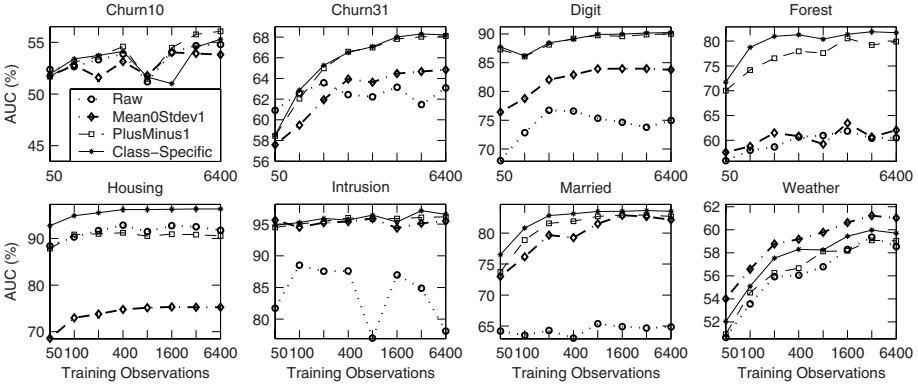

Fig. 2. Impact of scaling and training set sizes on classifier performance for different data sets. Results presented are the mean AUCs, where the means are computed over all randomisations for all classifiers and all regularisation constant values

considerably improve classifier performance and produces the best or equal best score in 10 of the 32 classifier/data set combinations. Overall one of these scales is the best choice in 29 out of 32 situations, and generally there is not a great difference between the performance of these two scaling methods.

The centroid classifier performs better when the Class-Specific scale is used in six data sets, and improves its performance to the point where it is comparable with that of the SVMs and RR. This is noteworthy as the centroid classifier requires no training time, and is thus extremely efficient computationally. Indeed, as a general tendency, the scores for a particular data set are relatively similar over the four classifiers when the data is scaled using the Class-Specific scale. This type of scaling thus seems to reduce the impact of classifier choice significantly.

Mean0Stdev1 scaling varies in effect considerably. It tends to produce improved scores relative to raw data, but not to the same extent as the other two scaling methods. In some cases (e.g. the housing data set) it impairs classifier performance. However for certain data set/classifier combinations (e.g. the weather data set) it actually outperforms the other scaling methods.

Churn10 is the only data set for which scaling has a limited effect. This set performs poorly, with scores only a little better than what a random classification would produce. As described previously, the churn10 data set contains the 10 least predictive attributes from the churn31 data set. Thus it is not surprising that the scores are very low.

6.2 Training Set Size Interactions

Figure 2 shows the effect of training set size increase (x-axis) on AUC (y-axis) for eight different data sets, with the four different types of scaling: unscaled (dotted line), Mean0Stdev1 (dot-dash line), PlusMinus1 (dashed line) and Class-Specific (unbroken

Table 1. Impact of training set size on the standard error of the AUC averaged across all other experimental conditions

Training Set Size	50	100	200	400	800	1600	3200	6400
Standard Error ($\times 10^{-2}$)	5.9	4.2	3.6	2.8	2.7	2.1	1.9	1.5

line). The data shown has been averaged over all randomisations, all classifiers, and all regularisation constant values. Again, the standard error is very low and as such error bars are not shown. However, as shown in Table 1, the mean standard error over all other experimental conditions (as calculated above) drops significantly as the size of the training sample increases.

The trend across all data sets and all scales is for AUC to increase rapidly over the first small increases of training set size (50 to 200 samples), then plateau as the training set size increases further. Scaling seems to have a positive impact on this behaviour, reducing the amount of training data required before the curve flattens out. In particular, the Class-Specific scale tends to decrease the training set size at which the curve plateaus. As such, it is possible to get very good results with small sample sizes.

Raw data is often erratic, depending on the data set (e.g. churn10, churn31, digit, forest and intrusion). However, in seven of the eight data sets, scaling smoothes the curve out significantly. The greatest smoothing effect is seen with the Class-Specific and PlusMinus1 scales. Mean0Stdev1 scaling tends to produce some smoothing effects, but this is often not as pronounced as the other scaling methods (e.g. churn31, forest).

Note that increasing the training set size improves the classifier performance in two key ways. Firstly, more information is available to the classifier, presenting it with a sample that is more representative of the population. Secondly, as the training set size increases, more information becomes available to calculate the scaling statistics. As such, they become progressively closer to the the population statistics. This improves the quality of the scaling method, and thus the classifier performance.

6.3 Effect of Regularisation Constant Values

Figure 3 shows the effect of changing the regularisation constant (c) (x-axis) on AUC (y-axis) for different scaling methods, data sets and classifiers. Each group of twelve graphs corresponds to a data set. The three columns correspond to the different regularisation machines: L1 (SVML1), L2 (SVML2) and RR (ridge regression). The four rows show the four different scaling types: unscaled (Raw), Mean0Stdev1 (M0S1), PlusMinus1 (PM1) and Class-Specific (C-S). Data has been averaged over all training set sizes, and error bars have not been shown as the error is very small.

As we would expect, the trend across all variables is that scores improve as c increases. However, there is significant interaction between the classifier used and the impact of changing c on the performance. For SVML1, scores are often impaired by increasing the regularisation constant, particularly when the data has been preprocessed using the Mean0Stdev1 and PlusMinus1 scaling methods. This is in contrast to SVML2 and RR, which consistently show either improvement or no change as c is increased, across all scaling methods.

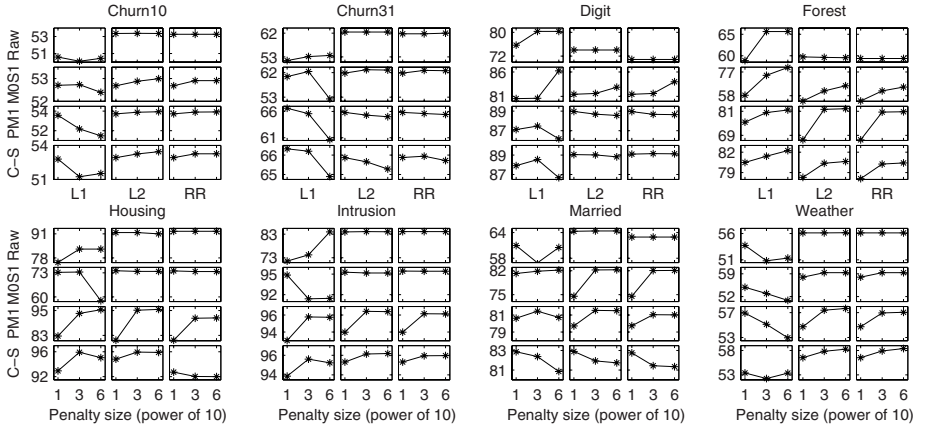

Fig. 3. Effect of regularisation constant (c) for different scaling methods, data sets and classifiers. Results presented are AUCs averaged over all training set sizes and correspond to three machines: L1 (SVML1), L2 (SVML2) and RR (ridge regression) and four scaling methods: unscaled (Raw), Mean0Stdev1 (M0S1), PlusMinus1 (PM1) and Class-Specific (C-S)

There is no clear trend in the interaction between different scaling methods and regularisation constant values. Ideally, the results suggest tuning for the regularisation constant value that gives the best results for a particular data set, classifier and scaling method. However, this requires an investment of time and resources, which is not always possible. We see that there is no evidence that an increase in AUC is likely to result from an increase in c, particularly if the data is scaled. As such, and given that an increase in regularisation constant value correlates to a significant increase in computational resources required, low values of c are recommended in the general case.

7 Discussion and Recommendations

Table 2 shows the overall performance of each form of scaling. The mean AUC and standard error over all experimental conditions is shown in rows 2 and 3 respectively. The last row shows the percentage of times a scaling method provides the best AUC. Notably, all methods of scaling significantly improve the mean score and decrease the mean standard error. Further, scaled data produces the best results in 93.78% of cases. It is clear that simple attribute scaling greatly improves the performance of linear binary classifiers. In particular, the Class-Specific scale tends to be the best choice of scale. It produces the best results in half the experimental conditions, and substantially reduces the effect of experimental vaiable (such a training set size), and thus the standard error.

Table 2. Performance of the three scaling methods, averaged across all experimental conditions

Scaling Method	Unscaled	Mean0Stdev1	PlusMinus1	Class-Specific
Mean AUC %	65.05	69.49	74.30	76.63
Standard Error ($\times 10^{-2}$)	4.1	2.2	1.6	1.4
Best Method (%)	6.22	18.14	25.65	49.99

7.1 Statistical Significance

To test for the statistical significance of these results, paired one-tailed Student's t-tests were used to compare the classification performance of data with and without scaling. All observations were divided into four sets (S_R, S_M, S_P and S_C) based on the form of scaling that the data was preprocessed with (raw, Mean0Stdev1, PlusMinus1 and Class-Specific respectively). The observations in any two of these sets were paired based on the value of the settings of the other experimental variables (data set, classifier, training set size, regularisation constant value and randomisation).

Three null hypotheses were tested for, $H_0^M : \mu_R \geq \mu_M$, $H_O^P : \mu_R \geq \mu_P$ and $H_0^C : \mu_R \geq \mu_C$, where μ_R, μ_M, μ_P and μ_C are the means of S_R, S_M, S_P and S_C respectively. At the significance level $\alpha = 0.001$, all three hypotheses were rejected in favour of the alternative hypotheses, $H_1^M : \mu_R < \mu_M$, $H_1^P : \mu_R < \mu_P$ and $H_1^C : \mu_R < \mu_C$. Thus we can say that, at the 0.1% confidence level, all forms of scaling improve the classification performance.

Further, the extent of the percentage improvement from raw data given by scaling was tested, again using a one tailed Student's t-test with $\alpha = 0.001$. The null hypotheses tested were $H_0^M : 8 \geq \frac{100(\mu_M - \mu_R)}{\mu_R}$, $H_O^P : 15 \geq \frac{100(\mu_P - \mu_R)}{\mu_R}$ and $H_0^C : 18 \geq \frac{100(\mu_C - \mu_R)}{\mu_R}$. Again, these hypotheses were rejected, and thus we can be confident that on average, and at the 0.1% significance level, the three scaling methods, Mean0Stdev1, PlusMinus1 and Class-Specific, improve the classification performance at baseline by 8%, 15% and 18% respectively.

7.2 Recommendations

Based on these results, we can put forward several recommendations, as follows. Scaling should generally be used to improve the performance of linear binary classifiers. Ideally, an initial test run should be performed to determine the optimum scaling method, however if this is not possible, the Class-Specific scaling method should be used.

If only limited resources are available and it is acceptable to achieve classifications that are suboptimal, the centroid classifier with Class-Specific scaling should be used.

If there is only limited training data available (less than 400 samples), the Class-Specific scaling method will typically give the best results.

If the data is scaled, it is generally unnecessary to use large values for the regularisation constant. This will again reduce the computational resources needed for the training task. However, note that if resources permit, small improvements in score can be gained by tuning the value of c to the particular classifier, data set and scaling method.

If the feature space is large, the value of the transformation factor for the Class-Specific scale can be used as a way to choose which features can be discarded. This can further reduce the resources needed for the training.

8 Conclusions and Future Work

We have shown that simple attribute scaling can significantly improve the performance of linear binary classifiers. Furthermore, a particular scaling method, introduced here as the Class-Specific scale, is the best choice of scales to use across all experimental conditions.

Given the extent of the improvements shown here, it would be useful to investigate the effect of attribute scaling on the performance of SVMs with different kernels. Other scaling methods, particularly those that utilise information about the class distribution of each attribute, would also be worth studying further.

Acknowledgements

The permission of the Managing Director, Telstra Research Laboratories (TRL) to publish this paper is gratefully acknowledged. The technical advice and feedback of Herman Ferra, TRL, is gratefully acknowledged.

References

[1] Berry, M., Linoff, G.: Data Mining Techniques: For Marketing, Sales and Customer Support. Wiley, New York (1997)
[2] Pyle, D.: Data Preparation for Data Mining. Morgan Kaufmann Publishers, Inc., California (1999)
[3] Corte, C., Vapnik, V.: Support-vector networks. Machine Learning **20** (1995) 273 – 297
[4] Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press, Cambridge (2000)
[5] Vapnik, V.: Statistical learning theory. Wiley, New York (1998)
[6] Kimeldorf, G., Whaba, G.: A correspondence between Bayesian estimation of stochastic processes and smoothing by splines. Ann. Math. Statist. **41** (1970) 495–502
[7] Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond. MIT Press (2001)
[8] Girosi, F., Jones, M., Poggio, T.: Regularization theory and neural networks architectures. Neural Computation **7** (1995) 219–269
[9] Rocchio, J.J.: Relevance feedback in information retrieval. In Salton, G., ed.: The SMART Retrieval System: Experiments in Automatic Document Processing, Englewood Cliffs, N J, Prentice-Hall Inc. (1971) 313–323
[10] Kowalczyk, A., Raskutti, B.: Exploring Fringe Settings of SVMs for Classification. In: Proceedings of the Seventh European Conference on Principle and Practice of Knowledge Discovery in Databases (PKDD03). (2003)
[11] Bradley, A.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition **30(7)** (1997) 1145–1159
[12] Weiss, G., Provost, F.: The effect of class distribution on classifier learning. Technical report, Rutgers University (2001)

[13] Centor, R.: Signal detectability: The use of ROC curves and their analysis. Med. Decis. Making **11** (1991) 102 – 106
[14] Fawcett, T.: ROC Graphs: Notes and practical considerations for data mining researchers. In: HP Labs Tech Report HPL-2003-4. (2003)
[15] Bamber, D.: The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. J. Math. Psych. **12** (1975) 387 – 415
[16] Hand, D., Till, R.: A simple generalisation of the area under the ROC curve for multiple class classification problems. Machine Learning **45** (2001) 171 – 186
[17] Hsu, C., Chang, C., Lin, C.: A practical guide to support vector classification. http://www.csie.ntu.tw/ cjlin/papers/guide/guide.pdf (2003)
[18] Sarle, W.: Neural network FAQ. ftp://ftp.sas.com/pub/neural/FAQ2.html (1997)