Extreme Re-balancing for SVMs: a case study

Bhavani Raskutti BHAVANI.RASKUTTI@TEAM.TELSTRA.COM
Adam Kowalczyk ADAM.KOWALCZYK@TEAM.TELSTRA.COM
Telstra Corporation, 770 Blackburn Road, Clayton, Victoria, Australia

Abstract

This paper reports work in progress on balancing training data which has the two classes of interest in heavily unbalanced proportions. We focus on the case of supervised learning with support vector machines. We consider the impact of both sampling and weighting imbalance compensation techniques and then extend the balancing to situations when one of the classes is ignored completely and learning is accomplished using examples from a single class.

Our investigation shows that for some real world problems, such as the gene knock-out experiments for understanding Aryl Hydrocarbon Receptor signalling pathway that provided the data for the second task of the KDD 2002 Cup, minority one-class SVMs significantly outperform models learnt using examples from both classes. We investigate this anomalous behaviour through an extensive analysis of this data as well as text benchmarks such as Reuters Newswire data.

1. Introduction

A standard recipe for two class discrimination is to take examples from both classes, then generate a model for discriminating them. This approach is so entrenched in machine learning that practitioners often will not consider data unless it contains examples of both classes. Moreover, many machine learning algorithms, such as decision trees, naive Bayes or multilayer perceptron, do not function unless the training data includes examples from two classes. However, there are many applications where obtaining examples of a second class is difficult and the data has heavily unbalanced representatives of the two classes of interest (unbalanced priors). A supervised learning algorithm applied to such a problem has to implement some form of balancing. The point we want to make in this paper is that sometimes it is beneficial to design re-balancing even more radically than warranted by unequal priors. The extreme case here is learning two class discrimination with one class data only. We show in this paper that in some real life learning problems, such as the Aryl Hydrocarbon Receptor data used for the second task of the KDD 2002 Cup (Craven, 2002), Support Vector Machines (SVM) do benefit from such an extreme approach (Kowalczyk & Raskutti, 2002).

The paper is organised as follows. Section 2 places our research in the context of existing work. Section 3 introduces the basic support vector machines in the particular form used for this research. We then discuss in Section 4 the two forms of imbalance compensation techniques investigated in this paper. The experimental setup including data collections and performance measures is described in Section 5. In Section 6 we present results on experiments for both sample balancing and weight balancing techniques, and discuss the implications of our experimental findings in Section 7.

2. Related Research

The problem of discrimination of unbalanced classes is encountered in a large number of real life applications of machine learning, e.g., detection of oil spills in satellite radar images (Kubat et al., 1997), information retrieval and filtering (Lewis & Catlett, 1994) and biological domains (Craven, 2002; Kowalczyk & Raskutti, 2002). Many solutions have been proposed to address the imbalance problem including sampling and weighting examples, cf. (Japkowicz & Stephen, 2002) for a thorough survey. However, these methods typically focus on cases when the imbalance ratio of minority to majority class is around 10:90. In this paper, we focus on extreme imbalance, where the minority class consists of around 1-3% of the data, then extend the sampling to situations when one of the classes is ignored completely and learning is accomplished using examples from a single class.

The possibility of single class learning with support vector machines (SVM) has been noticed previously. In particular, (Schölkopf et al., 1999) have suggested a method of adapting the SVM methodology to *one-class* learning by treating the origin as the only mem-

Workshop on Learning from Imbalanced Datasets II, ICML, Washington DC, 2003.

ber of the second class. This methodology has been used for image retrieval (Chen et al., 2001) and for document classification (Maneivitz & Yousef, 2002). In both cases, modelling is performed using examples from the positive class only, and the one-class models perform reasonably, although much worse than the *two-class* models learnt using examples from both classes. In contrast, in this paper, we show that for certain problems such as the gene knock-out experiments for understanding Aryl Hydrocarbon Receptor (AHR) signalling pathway, minority one-class SVMs significantly outperform models learnt using examples from both classes. We investigate this peculiar behaviour through a thorough analysis of the AHR data and text benchmarks such as Reuters Newswire data.

3. Support Vector Machines

In this section we recall basic concepts of *Support Vector Machines (SVM)* in a form suitable for this paper. Given a labelled training sample

$$\vec{\mathbf{x}y}^m := \big((x_1, y_1), ..., (x_m, y_m)\big) \in (X \times \{\pm 1\})^m \qquad (1)$$

drawn from a high dimensional feature space $X \subset \mathbb{R}^n$, with $n \gg m$. The case of prime interest is when the target class, labelled "+1", is much smaller than the background class (labelled "−1"), e.g. when the prior of the target class is of the order $\approx 1\%$ of the data. Our aim is to find a direction of "good" discrimination, such that the target class instances have scores $w \cdot x_i$ higher than the scores for the background class. The solution $w_{\vec{\mathbf{x}y}^m}$ is defined as *a homogeneous support vector machine (hSVM)* which minimises the regularised risk (Cristianini & Shawe-Taylor, 2000; Schölkopf & Smola, 2001; Vapnik, 1998)

$$w \mapsto ||w||^2 + \sum_{i=1}^{m} C_i[\max(0, 1 - y_i w \cdot x_i)]^p, \qquad (2)$$

where $C_i \geq 0$ are regularisation constants and $p = 1$ (*linear penalty*, $hSVM^1$) or $p = 2$ (*quadratic penalty*, $hSVM^2$). Both values of p were used in our experiments with marginal differences in performance.

Without further comments, we shall assume from now on the usage of the "homogenised" or augmented data, i.e., $x_i = (x'_i, 1)$ where x'_i is the original feature vector.

One thing to stress is that (2) provides a unique solution in "regular" cases of interest, in particular, if at least some $C_i \neq 0$ and $0 \in \mathbb{R}^n$ does not belong to the convex shell spanned by all vectors $y_i x_i$. This means in particular that such a solution is provided also if all data points belong to a single class. In fact, we can always absorb the signum y_i by substituting $x_i \leftarrow y_i x_i$, which formally reduces the two class problem (2) to a single class optimisation.

The geometrical meaning of the solution (2) can be most clearly illustrated in the limiting case of "hard margin", i.e. $C_i = C \to \infty$. In such a case, the optimal solution $w_{\vec{\mathbf{x}y}^m}$ of (2) is the direction of the shortest vector from the origin to the convex shell spanned by all vectors $y_i x_i$.

4. Re-balancing of the data

We investigate two forms of imbalance compensation in this paper.

4.1. Sample Balancing

This method "re-balances" data by neglecting some examples from the training set. It selects m'_- and m'_+ examples out of the total m_- and m_+ examples from the negative and the positive label classes in the training set, respectively. The regularisation constant is the same for all instances, i.e., $C_i = C > 0$ for all i. In this case we will be reporting the class *proportion ratio* $\frac{m'_-}{m_-} : \frac{m'_+}{m_+}$ directly. In particular, the proportion ratios 1:0, 1:1 and 0:1 represent the case of 1-class learner using all of the negative examples, 2-class learner using all training examples, and 1-class learner using all of the positive examples, respectively.

This form of sample balancing is a generalisation of the techniques used in (Elkan, 2001), where all minority cases are used while the majority cases are sampled so as to take into account the relative cost of mis-classification of the two classes. In this specialised $MajorityOnly$ sampling, since all minority cases are used, i.e. $m'_+ = m_+$, we can use a single number to describe the proportion ratio uniquely. We shall call this number $B_{-/+} := m'_-/m_+$, the class *mixture ratio*, and it varies from 0 to $\top := m_-/m_+$. The value $B_{-/+} = 0$ is the case when only minority class examples are used (equivalent to the proportion ratio 1:0) and $B_{-/+} = \top$ represents the situation when all training instances are used (equivalent to the proportion ratio 1:1).

The sample balancing has speed advantages since a smaller number of examples are actually used for training, hence it has been used in most of our experiments.

4.2. Weight Balancing

In this case all training examples are used, but we use different values of the regularisation constants for the minority and majority class data:

$$C_i = \begin{cases} (1+B)C/2m_+ & \text{if } y_i = +1, \\ (1-B)C/2m_- & \text{if } y_i = -1, \end{cases} \tag{3}$$

for $i = 1, ..., m$, where $C > 0$ and $-1 \leq B \leq 1$ is a parameter called *a balance factor*. In the above formulae, the denominators do compensate for unequal class proportions in the training set while the parameter B introduces an additional compensation. For instance, the case of "balanced proportions" achieved for $B = 0$ discounts the majority class by the ratio of the two class sizes in training, $\frac{m_+}{m_-}$. Further discounting of the majority class occurs in the range $0 < B \leq +1$, with $B = +1$ representing the case of learning from positive examples only. Similarly, learning from negative class only is achieved for $B = -1$, with discounting of positive examples in the range $-1 \leq B < 0$.

4.3. Balancing Modes

When balancing the data, we consider two modes: *similarity detector* which learns a discriminator based predominantly on positive examples (e.g., $B_{-/+} \approx 0$, $B \approx 1$), and *novelty detector* which is trained using primarily negative examples or majority class examples (e.g., $B_{-/+} >> 1$, $B \approx -1$). In practice both modes have applications. For instance, classification of websites "attractiveness" based on history of user's activities is an application where negative examples (i.e. the sites of no interest) are difficult to obtain. On the other hand, for network intrusion detection, we have few (if any) examples of the target class we want to identify, i.e. of successful intrusion episodes.

5. Experimental Setup

In our experiments, we first pre-process the data in a manner appropriate for the data set, and create a sparse matrix representing the data set. For the textual data set, this matrix is the word presence matrix while for the AHR data this is some property of the gene associated with that instance.

5.1. Data Collections

AHR-data. Our primary corpus is the AHR-data set which is the combined training and test data sets used for task 2 of KDD Cup 2002. The data set is based on experiments by Guang Yao and Chris Bradfield of McArdle Laboratory for Cancer Research, University of Wisconsin. These experiments aimed at identification of yeast genes that, when knocked out, cause a significant change in the level of activity of the Aryl Hydrocarbon Receptor signalling pathway, cf. (Craven, 2002) for more details. Each training instance is labelled with one of three class labels: "nc", "control", or "change". Each of the 4507 instances in the data set is described by a variety of information that characterises the gene associated with the instance, e.g., associated abstracts from scientific articles, genes whose encoded proteins physically interact with one another, information about the subcellular localisation and functional classes of the proteins encoded by various genes. For the experiments described in this paper, we convert all of the information from the different files to a sparse matrix containing 18330 features (Kowalczyk & Raskutti, 2002). Following the KDD Cup requirements we experiment with three tasks: *change-task* discriminating "change" class instances from the rest, *control-task* discriminating "control" class instances from the rest and *either-task* discriminating instances in either "change" or "control" classes from the rest, i.e. "nc". The class sizes vary considerably with 57 instances of "change" 70 instances of "control" and the rest 4380 instances labelled "nc".

Reuters data. Our second corpus is the popular text mining benchmark, Reuters-21578 news-wires. Here we used a collection of 12902 documents (combined test and training sets of so called modApte split available from http://www.research.att.com/lewis) which are categorised into 115 overlapping categories. Each document in the collection has been converted to a vector of 20,197 dimensional word-presence feature space using a standard stop-list and after stemming all of the words using a standard Porter stemmer.

5.2. Performance Measures

We have used *AROC*, the Area under the Receiver Operating Characteristic (ROC) curve as our main performance measure. In that, we follow the steps of KDD 2002 Cup, but also, we see it as the natural metric of general goodness of classifier (as corroborated below) capable of meaningful results even if the target class is a tiny fraction of the data.

We recall that the ROC curve is a plot of the *true positive rate* or precision, $P(f(x_i) > \theta | y_i = 1)$, against the *false positive rate*, $P(f(x_i) > \theta | y_i = -1)$, as a decision threshold θ is varied. The concept of ROC curve originates in signal detection but these days it is widely used in many other areas, including data mining, psychophysics and medical diagnosis (cf. review (Centor, 1991)). In the latter case, *AROC* is viewed as a measure of general "goodness" of a test, formalised as a predictive model f in our context, with a clear statistical meaning as follows. $AROC(f)$ is equal to the probability of correctly answering the two-alternative-forced-choice problem: given two cases, one x_i from the negative and the other x_j from the positive class, allocate scores in the right order, i.e. $f(x_i) < f(x_j)$. Additional attraction of $AROC$ 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 (Bamber, 1975).

There are some ambiguities in the case of $AROC$ estimated from a discrete set in the case of ties, i.e. when multiple instances from different classes receive the same score. Following (Bamber, 1975) we implement in this paper the definition

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

expressing $AROC$ in terms of conditional probabilities.

Note that trivial uniform random predictor has $AROC$ of 0.5.

6. Experiments

For each experiment, 20 random splits of the data into the training and test sets were implemented. These splits were generated with proportional sampling (without replacement) from the positive and the negative classes in the pooled set. The sizes of the data split training:test were 50%:50% for the Reuters data and 70%:30% for the AHR-data. Other splits produce similar results and are not shown here for brevity.

We first study the impact of regularisation constant C on SVM solutions, and choose a restricted range of C for further experimentation. We then experiment with different forms of balancing with these values of C.

6.1. Impact of Regularisation Constant

We plot in Figure 1 mean AROC (with standard deviation bars) as a function of C for the two linear kernel machines: $hSVM^1$ (Figures 1A and 1C) and $hSVM^2$ (Figures 1B and 1D). We use two balancing techniques: MajorityOnly sample balancing (Figures 1A and 1B) and the weight balancing (Figures 1C and 1D). For this test, we focus on the either-task for the AHR-data, and means are computed over 20 random splits of the pooled set into 70%:30%, learning:test. Plots are shown for four different modes: (*i*) *positive 1-class* ($B_{-/+} = 0$ and $B = +1$, solid line); (*ii*) *negative 1-class* ($B = -1$, dotted line); (*iii*) *balanced 2-class* ($B_{-/+} = 1$ and $B = 0$, dashed line); (*iv*) *un-balanced 2-class* ($B_{-/+} = \top$, the dash-dot line).

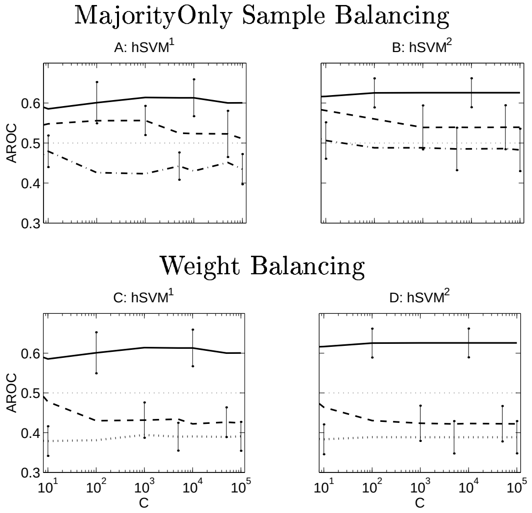

Figure 1. Results for AHR-data, the either-task. We plot mean AROC as a function of the regularisation constant C for $hSVM^1$ (Figs. A & C) and $hSVM^2$ (Figs. B & D). We use two balancing techniques: MajorityOnly sample balancing (Figs. A & B) and weight balancing (Figs. C& D). Plots are shown for four different modes: (*i*) positive 1-class ($B_{-/+} = 0$ and $B = +1$, solid line); (*ii*) negative 1-class ($B = -1$, dotted line); (*iii*) balanced 2-class ($B_{-/+} = 1$ and $B = 0$, dashed line); (*iv*) un-balanced 2-class ($B_{-/+} = \top$, the dash-dot line).

An inspection of plots brings a number of interesting observations:

(1) the AROC values for the positive one-class classifier is consistently above that for the two-class classifier for all values of C, and this is irrespective of the machine that is used for training.

(2) The performance of the positive one-class learner is not sensitive to the value of C, although the performance is slightly better at higher values of C (the "hard margin" case).

(3) As expected, the performance of the negative one-class learner is consistently worse than both the positive one-class and the balanced and un-balanced two-class learners for the two machines, performing worse than random for all values of C[1].

(4) There are differences in performance based on whether sample or weight balancing is used particularly for the balanced two-class learner, and weight-balanced two-class learners (dashed line in Figures 1C and 1D) perform significantly worse than sample-

[1] In fact, for $hSVM^1$, a better classifier may be obtained by using the negative one-class learner and inverting the labels than by using any other $hSVM^1$ learner! This intriguing phenomenon is the subject of our current research.

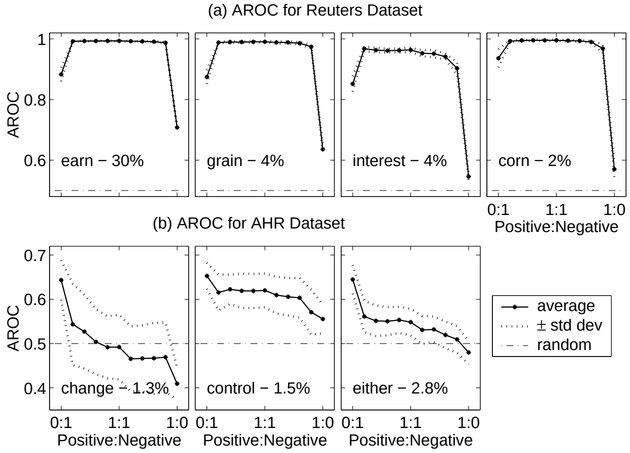

Figure 2. Average AROC ± standard deviation of test set as a function of the proportion ratio $\frac{m'_-}{m_+} : \frac{m'_+}{m_+} \in \{$0:1, 0.2:1, 0.4:1, 0.6:1, 0.8:1, 1:1, 1:0.8, 1:0.6, 1:0.4, 1:0.2, 1:0$\}$. Results are presented for $hSVM^2$ trained for four Reuters categories and three AHR-tasks for $hSVM^2$ trained with sample balancing method (Section 4).

balanced two-class learners (dashed line in Figures 1A and 1B). The performance of sample-balanced two-class learners is close to random for all but very low values of C, while that of weight-balanced two-class learners is closer to the negative one-class learner.

(5) There are noticeable differences between the performance of different SVMs (e.g. the differences between unbalanced two-class $hSVM^1$ and $hSVM^2$ represented by the dash-dot lines in Figures 1B and 1D). However, observations (1)-(4) hold for both classifiers over the whole range of values for the regularisation constant C.

6.2. Experiments with Sample Balancing

The sample balancing has an obvious advantage in speed since in training we use only a part of the data set. For this reason it has been used in our main experiments requiring multiple generations of SVMs. For the results reported in this section we have used several class proportion ratios starting from $0 : 1$ (100% of positive class and 0% of negative class), through $1 : 1$ (100% of examples of both classes) to $1 : 0$ (0% positive and 100% of negative examples). In experiments we have used all three categories of the AHR-data as described above and selected four Reuters categories: “earn”, “grain”, “interest” and “corn”.

Figure 2 presents the averages and standard deviations of test set AROC for different values of class proportion ratio $\frac{m'_-}{m_+} : \frac{m'_+}{m_+}$. Plots are shown for four Reuters categories and the three categories of the AHR dataset. Due to space considerations, results are shown only for the $hSVM^2$ classifier.

The results for the Reuters dataset are as expected, with positive and negative examples on their own providing sufficient information to perform better than random predictor (Figure 2(a)). However, both are outperformed by the two-class model even if the model includes only 20% of the other class data. Further, the AROC with 2-class learners is close to 1 for all categories indicating that this categorisation problem is reasonably easy to learn.

The AROC for the AHR dataset, on the other hand, has a maximum mean value of around 0.64 for all three categories (Figure 2(b)). For all three categories, the AROC starts off at the highest point when positive examples alone are used, and then drops as negative examples are added, indicating that the knowledge of negative examples in this problem is detrimental to learning. Further, the standard deviations are the lowest when only positive examples are used. Once again, the balanced two-class learner performs close to a random classifier (mean AROC ≈ 0.5). The negative one-class learner performs much worse than random (mean AROC ≈ 0.4), in effect, providing better discrimination than balanced two-class learner (cf. footnote 1).

6.2.1. Impact of feature selection

In order to determine if the better performance of the single class learner is due to the sparse high dimensional input space, we explore the same KDD cup 2002 data, but this time with aggressive dimensionality reduction of the input space using automatic feature selection, or more precisely feature ordering methods. The ordering is done via sorting the features in decreasing order of scores calculated by one of the following methods.

A: DocFreq (Document frequency thresholding): This method has its origins in information retrieval (Salton & McGill, 1983) and is based on the notion that rare features are not informative for predicting classes. In this case the score of a feature is simply the number of instances where it is equal to 1.

B: ChiSqua (χ^2): The χ^2 measures the lack of independence between a feature and a class of interest. First, for each feature and each class, i.e. $y = \pm 1$, a score is computed on the basis of the two-way contingency table (Yang & Pedersen, 1997). The final score for a feature is the maximum of these class scores.

C: MutInfo: (Mutual Information): This method prioritises the features of the basis of the joint and

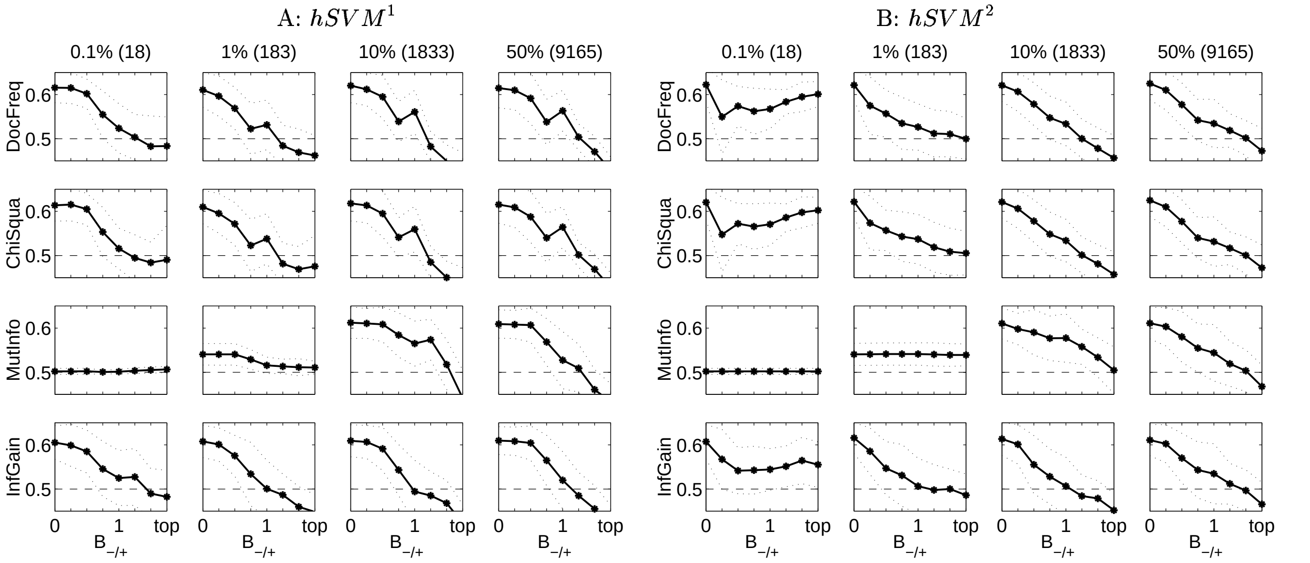

Figure 3. Mean AROC for the KDD subtask “either” as a function of the mixture ratio $B_{-/+}$ for four different fractions of the original feature set (0.1%, 1%, 10% and 50%), for three linear kernel machines with $C = 100$: (A) $hSVM^1$ and (B) $hSVM^2$. $B_{-/+}$=$[0, 0.01, 0.1, 0.5, 1, 5, 10, \top]$.

marginal probabilities of their usage estimated from the training data (Yang & Pedersen, 1997).

D: InfGain: (Information gain): This is frequently employed as a term goodness measure in machine learning (Quinlan, 1986), and measures the number of bits of information obtained for class prediction by knowing the presence or absence of a term in an instance.

Given the worse than random performance of negative one-class learners, for these experiments, we have used all of the minority cases and sampled the majority cases at different mixture ratios (MajorityOnly sample balancing). Figure 3 shows mean AROC (with standard deviation as an envelope) as a function of the mixture ratio $B_{-/+}$ for different fractions of the original feature set (0.1%, 1%, 10% and 50%). Results are shown for KDD either-task, for two linear kernel machines: (A) $hSVM^1$ and (B) $hSVM^2$. For both machines, results are presented for $C = 100$, although results for $C = 10$ and $C = 1000$ show similar trends. Results are presented for the four different feature selection methods listed above. The other feature selection methods such as Idf-tf (inverse document frequency – term frequency) and average discrimination scoring (Salton & McGill, 1983) showed similar behaviour.

As seen from Figure 3, all feature selection methods select informative features that allow learning at some mixture ratio. This is the case even at very low fraction of features (0.1% or just 18 features) for all methods except $MutInfo$. The poor performance of $MutInfo$ at low fractions is not surprising given that this measure is strongly influenced by the marginal probability of terms and tends to favour rare terms rather than common terms. Hence, at low fractions most of the instances have all of their attributes set to 0, and very little learning is accomplished. This is in contrast to the performance of $DocFreq$ which simply selects the most common terms.

The drop in performance as negative class examples are added is consistently visible for $hSVM^1$ (Figure 3(A)) and $hSVM^2$ (Figure 3(B)). Interestingly, with $hSVM^2$, $DocFreq$ and $ChiSqua$ with just 18 features (first column, rows 1-2 of Figure 3(B)), the unbalanced 2-class learner using all training examples performs surprisingly well indicating that feature selection can indeed combat the destructive influence of the negative class examples.

6.3. Experiments with Weight Balancing

In order to understand if the impact of negative examples may be reduced using the balance factor B in Equation (3), we investigate the performance for the AHR dataset using weight balancing as described in Section 4.

Figure 4 plots the mean and standard deviation of the test set AROC as a function of the balance factor B. Plots are shown for the three categories of the AHR dataset for the $hSVM^2$ classifier with regularisation constant $C = 10$, although results for other values of

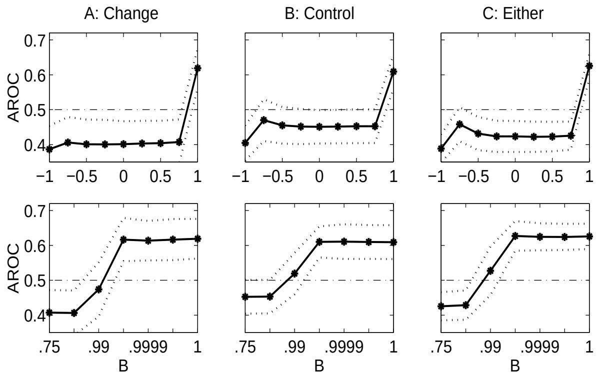

Figure 4. Mean AROC with standard deviation envelope as a function of balance factor B for the AHR dataset. We use here $hSVM^2$ with $C = 10$.

C show similar trends (Kowalczyk & Raskutti, 2002). The first row explores the whole range -1.0 to +1.0, while the second row expands the range 0.75 to 1.0 where sudden rises in AROC occur. We note that the best AROC values for all three learning tasks are obtained for $B \geq 0.99$, and the worst for $B = -1.0$.

Thus, both the weight balancing and the sample balancing techniques yield the conclusion that for the AHR dataset, extreme re-balancing by ignoring all of the negative examples produces the best AROC.

7. Discussion and Conclusions

In order to understand why one can obtain better results using examples from a single class rather than both classes, we first explore the feature space for one particular randomisation of the KDD cup 2002 pooled data. Figure 5(A) plots the number of instances when a particular feature is used in the pooled set versus the number of features, where these features are ordered in the decreasing order of their usage in the pooled set. As seen from Figure 5(A), the high dimensional space consisting of 18,330 features is hardly sampled. Furthermore, for this particular split, there are around 14,610 features that occur only in the negative examples of the training set. We call these features *NegOnly* features, and explore how these *NegOnly* features affect the two-class models.

To this end, we plot in Figure 5(B), the magnitude of the SVM weights for the same split, for the balanced two-class $hSVM^2$ model created with the setting $C = 5000$, $B = 0$. The x-axis is the fraction of *NegOnly* features, where these features are sorted by decreasing order of magnitude of the SVM weights for the features. The usage of these features in the positive class of the training set is 0. Hence, during the training (minimisation of regularised risk (2)), these features may have relatively large negative weights so as to minimise the error penalty. However, as shown in Figure 5(D) their usage in the test instances contributes large negative scores for the positive instances in the test set, cf. Figure 5(C) which plots the contribution of these features to the SVM scores. Effectively, *NegOnly* features are "confusing" the two-class classifier, while leaving the one-class learner unaffected (since one-class solution vector has entries corresponding to these features set to zero).

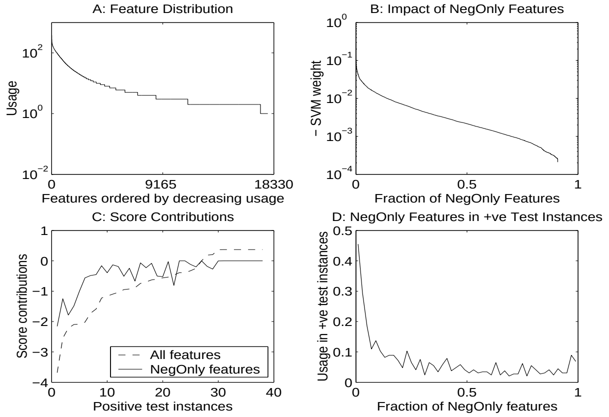

Figure 5. Understanding the influence of sparse high dimensional space on the SVM solution of two-class learner. (A) Usage of features in decreasing order of usage. (B) Magnitude of SVM weights for two-class model for the *NegOnly* (features used only in the negative class in the training set) features in decreasing order of magnitude. (C) Contribution of *NegOnly* features to SVM score. (D) Usage of *NegOnly* features in the positive test set.

While the above analysis is for the whole feature set, we also observed in Section 6.2.1 that even in low dimensional dense space, this phenomenon of better performance with one-class learner persists. Our intuitive explanation here is that if the learner uses the minority class examples only, the "corner" (the half space) where minority data resides is properly determined. However, the minority class is "swamped" by the background class. Once the background instances are added, the SVM solution is determined by the need to minimise the margin errors for this class at the expense of the target class and the resulting solution becomes suboptimal in terms of the resulting ROC curve. The strange thing is that the heavy discounting of the majority class does not rectify this impact completely, cf. $B = 0.99$ in Figure 4.

In this research we have concentrated on the simplest linear kernel case only. There are three reason for such a choice: (i) simplicity, (ii) from past experience, on

Reuters data, non-linear kernels improve performance only marginally (Dumais et al., 1998; Raskutti et al., 2001), and (*iii*) the non-linear kernel case viewed from the feature space level reduces to the linear one anyway (Vapnik, 1998). Some preliminary experiments with synthetic datasets using polynomial kernels show that our finding with linear kernels do indeed carry over to non-linear kernels (Kowalczyk & Raskutti, 2003).

In the paper we have analysed two data sets, Reuters and AHR data set. The Reuters dataset is an example of a 'regular data set', where extreme re-balancing, provides quite good results but using both classes always produces better results. On the other hand, the AHR data set behaves differently, with the positive one-class learners performing significantly better than two-class learners. Further, for this dataset, negative one-class learner performs worse than random. This anomalous behaviour and its potential for discrimination is the subject of our current research.

ACKNOWLEDGEMENTS

The permission of the Managing Director, Telstra Research Laboratories, to publish this paper is gratefully acknowledged.

References

Bamber, D. (1975). The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. *J. Math. Psych.*, *12*, 387 – 415.

Centor, R. (1991). The use of ROC curves and their analysis. *Med. Decis. Making*, *11*, 102 – 106.

Chen, Y., Zhou, X., & Huang, T. (2001). One-class svm for learning in image retrieval. *Proceedings of IEEE International Conference on Image Processing (ICIP'01 Oral).*

Craven, M. (2002). The Genomics of a Signaling Pathway: A KDD Cup Challenge Task. *SIGKDD Explorations*, **4(2)**.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods.* Cambridge: Cambridge University Press.

Dumais, S., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive Learning Algorithms and Representations for Text Categorization. *Seventh International Conference on Information and Knowledge Management.*

Elkan, C. (2001). The foundations of cost-sensitive learning. *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence* (pp. 973–978).

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis Journal*, *6*.

Kowalczyk, A., & Raskutti, B. (2002). One Class SVM for Yeast Regulation Prediction. *SIGKDD Explorations*, **4(2)**.

Kowalczyk, A., & Raskutti, B. (2003). Exploring Fringe Settings of SVMs for Classification. *Proceedings of the Fourteenth European Conference on Machine Learning ECML03 (to appear).*

Kubat, M., R., H., & Matwin, S. (1997). Learning when negative examples abound. *Proceedings of the Ninth European Conference on Machine Learning ECML97.*

Lewis, D., & Catlett, J. (1994). Training Text Classifiers by Uncertainty Sampling. *Proceedings of the Seventeenth International ACM SIGIR Conference on Research and Development in Information Retrieval.*

Maneivitz, L. M., & Yousef, M. (2002). One-class SVMs for Document Classification. *Journal of Machine Learning Research*, *2*, 139–154.

Quinlan, J. R. ((1986)). Induction of Decision Trees. *Machine Learning*, **1***(1).*

Raskutti, B., Ferrá, H., & Kowalczyk, A. (2001). Second Order Features for Maximising Text Classification Performance. *Proceedings of the Twelfth European Conference on Machine Learning ECML01.*

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval.* McGraw Hill.

Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., & Williamson, R. (1999). Estimating the support of a high-dimensional distribution.

Schölkopf, B., & Smola, A. J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond.* MIT Press.

Vapnik, V. (1998). *Statistical learning theory.* New York: Wiley.

Yang, Y., & Pedersen, J. O. (1997). A Comparative Study on Feature Selection in Text Categorization. *Proceedings of the Fourteenth International Conference on Machine Learning.*