Extreme Re-balancing for SVMs: a case study

Bhavani Raskutti
Telstra Corporation
770 Blackburn Road, Clayton
Victoria, Australia
Bhavani.Raskutti@team.telstra.com

Adam Kowalczyk
Telstra Corporation
770 Blackburn Road, Clayton
Victoria, Australia
Adam.Kowalczyk@team.telstra.com

ABSTRACT

There are many practical applications where learning from single class examples is either, the only possible solution, or has a distinct performance advantage. The first case occurs when obtaining examples of a second class is difficult, e.g., classifying sites of "interest" based on web accesses. The second situation is exemplified by 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, where minority one-class SVMs significantly outperform models learnt using examples from both classes.

This paper explores the limits of supervised learning of a two class discrimination from data with heavily unbalanced class 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 extreme situations when one of the classes is ignored completely and the learning is accomplished using examples from a single class.

Our investigation with the data for KDD 2002 Cup as well as text benchmarks such as Reuters Newswire shows that there is a consistent pattern of performance differences between one and two-class learning for all SVMs investigated, and these patterns persist even with aggressive dimensionality reduction through automated feature selection. Using insight gained from the above analysis, we generate synthetic data showing similar pattern of performance.

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, e.g., classifying sites of "interest" to a web surfer where the sole information that is available are the positive examples or sites that are of interest to the user. In such a case, learning from examples of one class is the only possible solution.

In addition, there are situations when the data has heavily unbalanced representatives of the two classes of interest, e.g., fraud detection and information filtering. A supervised algorithm applied to such a problem has to implement some form of balancing. In some situations, it may be beneficial to design re-balancing even more radically than warranted by unequal proportions, and ignore the large pool of negative examples and learn from positive examples only. A real life learning problem that has benefited from such an approach is the second task of the KDD Cup 2002 [6], where the winning submission learnt using just the positive examples which consisted of $< 3\%$ of the training data [16].

This paper explores the limits of two-class learning and analyses situations when this discrimination learning may break down. We focus on supervised learning with support vector machines (Section 3) and investigate two forms of imbalance compensation techniques (Section 4). Our experiments with two real world collections, namely, the KDD 2002 Cup data and the Reuters Newswire benchmark, show that in some real life learning problems, SVMs do benefit from extreme re-balancing, i.e., ignoring all of the majority class examples (Section 6). We investigate this surprising result with a systematic study using synthetic data (Section 7). This study coupled with our earlier experiments with real world data sets leads us to conclude that data with a certain combination of properties, e.g., the presence of label noise, sparsity of features and low proportion of minority class, lends itself to better performance with one-class learners (Section 8).

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 [17], information retrieval and filtering [18], fraud detection [3] and biological domains [6; 16]. Many solutions have been proposed to address the imbalance problem including sampling and weighting examples, cf. [14] for a thorough survey. Typically, these methods focus on cases when the imbalance ratio of minority to majority class is around 10:90. For instance, in [3], though the number of fraudulent cases is much lower than 1% in the raw data, the actual data used in learning is pre-filtered so that the ratio of minority to majority class is 20:80; in [4] where more extreme imbalance ratios are considered, tens of thousands of training examples were used in input spaces of unspecified dimensionality (most likely of order of tens to hundreds of attributes). In contrast, in this paper, we focus on extreme imbalance in very high dimensional input spaces, where at the learning stage the minority class consists of around 1-3%

of the data and the learning sample size is much below the dimensionality of the input space exceeding 10,000 features. In this context, we explore different sampling strategies. In particular, we extend the sampling to situations when one of the classes is ignored completely and learning is accomplished using examples from a single class only.

The possibility of single class learning with support vector machines (SVM) has been noticed previously. In particular, [23] have suggested a method of adapting the SVM methodology to *one-class* learning by treating the origin as the only member of the second class. This methodology has been used for image retrieval [5] and for document classification [19]. 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 *kernel machines* in a form suitable for this paper. Given a training sequence (x_i, y_i) of binary n-vectors $x_i \in \{0,1\}^n \subset \mathbb{R}^n$ and bipolar labels $y_i \in \{\pm 1\}$ for $i = 1, ..., m$. The case of prime interest here is when the target class, labelled $+1$, is much smaller than the background class (labelled -1), consisting of a minute fraction, $\approx 1 - 3\%$, of the data. Our aim is to find a "good" discriminating function $f : \{0,1\}^n \to \mathbb{R}$ that scores the target class instances higher than the background class instances. The solution will be given in a form of a kernel machine

$$f(x) = f^H(x) + b := \sum_{i=1}^{m} \beta_i k(x, x_i) + b \qquad (1)$$

where $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a kernel function of one of the forms specified below and $\beta_i, b \in \mathbb{R}$ are parameters to be defined for the given training set as the minimiser of the regularised risk of the form as follows.

$$(\beta_i, b) \mapsto ||f^H, b||^2 + \sum_{i=1}^{m} C_{y_i} \phi\big(1 - y_i(f^H(x_i) + b)\big), \qquad (2)$$

where $C_{+1}, C_{-1} \geq 0$ are class dependent regularisation constants, $\phi : \mathbb{R} \to \mathbb{R}_+$ is a convex loss function penalising deviations of scores from allocated labels and $\| \cdot \|$ is a norm as specified below. Now we specify variations of the regularised risk (2) leading to two different cases of kernel machines used in this paper.

- $hSVM^1$: For *the (homogeneous) support vector machine with linear penalty*, we use the norm,

$$\|f^H, b\|^2 := \sum_{i,j=1}^{m} \beta_i \beta_j k(x_i, x_j) + b^2 \qquad (3)$$

  and the "hinge loss" $\phi(\theta) := \max(0, \theta)$, $\theta \in \mathbb{R}$ [7; 24; 25];

- $hSVM^2$: For *the (homogeneous) support vector machine with quadratic penalty* [7] we use norm (3) and the squared hinge loss $\phi(\theta) := (\max(0, \theta))^2$ for $\theta \in \mathbb{R}$;

If the kernel k satisfies the Mercer theorem assumptions [7; 24; 25] then for the minimiser of (2) we have $\beta_i = y_i \alpha_i$, where $\alpha_i \geq 0$ for $i = 1, ..., m$.

In our investigations we shall be using the popular polynomial kernel

$$k(x, x') = (x \cdot x')^d = (\sum_{i=1}^{n} \xi_i \xi_i')^d$$

for $x = (\xi_i)$ and $x' = (\xi_i')$ from $\{0,1\}^n$ and degree $d = 1, 2, 3$ and 4.

Note that $hSVM^1$ and $hSVM^2$ implement classifiers that correspond to separation of the data $(z_i, y_i) := (\Phi(x_i), 1, y_i) \in \mathbb{R}^N \times \mathbb{R} \times \{\pm 1\}$ by a hyperplane in the extended feature space passing through the $(0, 0) \in \mathbb{R}^N \times \mathbb{R}$, $\Phi : \mathbb{R}^n \to \mathbb{R}^N$ is a *feature mapping* of *the observation space* $\mathbb{R}^n$ into an appropriate Euclidean space $\mathbb{R}^N$ (*the features* space). In particular such a solution is provided also if all data points belong to a single class, i.e. if $y_i = const$.

The geometrical meaning of the solution (2) can be most clearly illustrated in the limiting case of "hard margin", i.e. $C \to \infty$. In such a case, the optimal solution of (2) corresponds to the direction of the shortest vector to the convex shell spanned by all vectors $y_i z_i \in \mathbb{R}^N \times \mathbb{R}$, $i = 1, ..., m$.

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 [9], 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:

$$C_i = \begin{cases} (1+B)C/2m_+ & \text{if } y_i = +1, \\ (1-B)C/2m_- & \text{if } y_i = -1, \end{cases} \quad (4)$$

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 web-sites "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 Real World 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. [6] 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 [16]. 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 which is 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 [2; 10]). 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 [1; 11].

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 [1] we implement in this paper the definition

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

expressing *AROC* in terms of conditional probabilities.

The trivial uniform random predictor has *AROC* of 0.5, while a perfect predictor has an AROC of 1.

We note that AROC is a metric that evaluates the classifier's performance across the entire range of decision thresholds and is especially useful when the operating condition for the classifier is unknown or the classifier is expected to be used in situations with significantly different class distributions. If the operating point is known, then point metrics such as break-even point and F-measure are appropriate when the target class is a tiny fraction of the data [15].

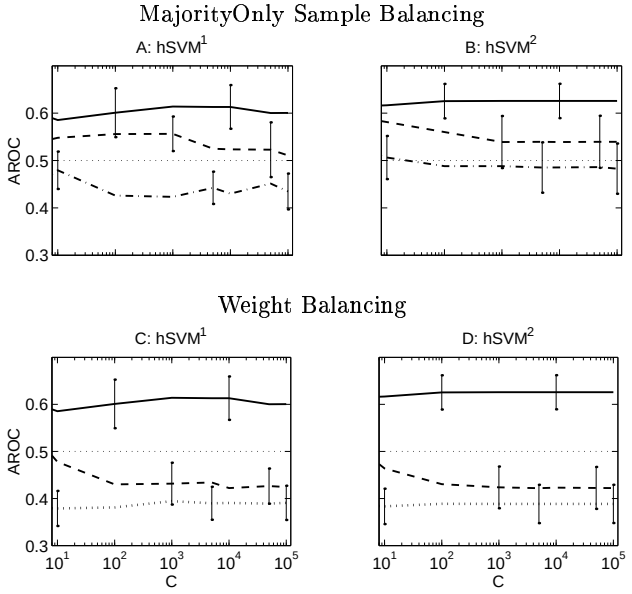

Figure 1: Results for AHR-data, the either-task. We plot mean AROC as a function of the regularisation constant C for $hSVM^1$ (Figs. A & C) and $hSVM^2$ (Figs. B & D). We use two balancing techniques: MajorityOnly sample balancing (Figs. A & B) and weight balancing (Figs. C& D). Plots are shown for 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).

6. EXPERIMENTS WITH REAL WORLD DATA

For our experiments with the AHR-data and Reuters, we have concentrated on the simplest linear kernel SVMs 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 [8; 21], and (iii) the non-linear kernel case viewed from the feature space level reduces to the linear one anyway [25].

For each experiment reported, 20 random splits of the data into the training and test sets were implemented. These splits were generated with stratified sampling (proportional sampling without replacement) from the positive and the negative classes in the pooled set. For the majority of experiments, 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).

An inspection of plots brings a number of interesting observations:

- 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.
- 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).
- 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].
- 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-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.
- 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.

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

[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!

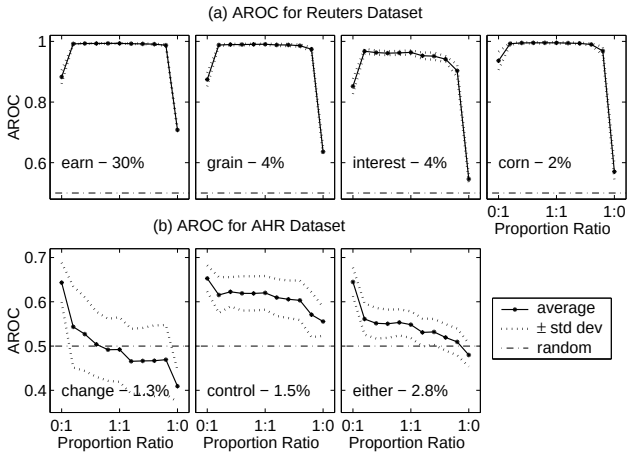

Figure 2: Average AROC ± standard deviation of test set as a function of the proportion ratio $\frac{m'_-}{m_-} : \frac{m'_+}{m_+} \in \{0{:}1,\ 0.2{:}1,\ 0.4{:}1,\ 0.6{:}1,\ 0.8{:}1,\ 1{:}1,\ 1{:}0.8,\ 1{:}0.6,\ 1{:}0.4,\ 1{:}0.2,\ 1{:}0\}$. Results are presented for $hSVM^2$ trained for four Reuters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

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.

- **DocFreq** (Document frequency thresholding): This method has its origins in information retrieval [22] 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.

- **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 [26]. The final score for a feature is the maximum of these class scores.

- **MutInfo**: (Mutual Information): This method prioritises the features of the basis of the joint and marginal probabilities of their usage estimated from the training data [26]. The score allocated to jth feature is calculated as follows:

$$MutInfo(j) = \max_{y=\pm 1} \log \frac{P(x_{i,j} = 1, y_i = y)}{P(x_{i,j} = 1)P(y_i = y)}$$

  where the joint and marginal probabilities are estimated from the training set, i.e. with respect to index i, $1 \leq i \leq m$ [26].

- **InfGain**: (Information gain): This is frequently employed as a term goodness measure in machine learning [20], 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 [22] 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.

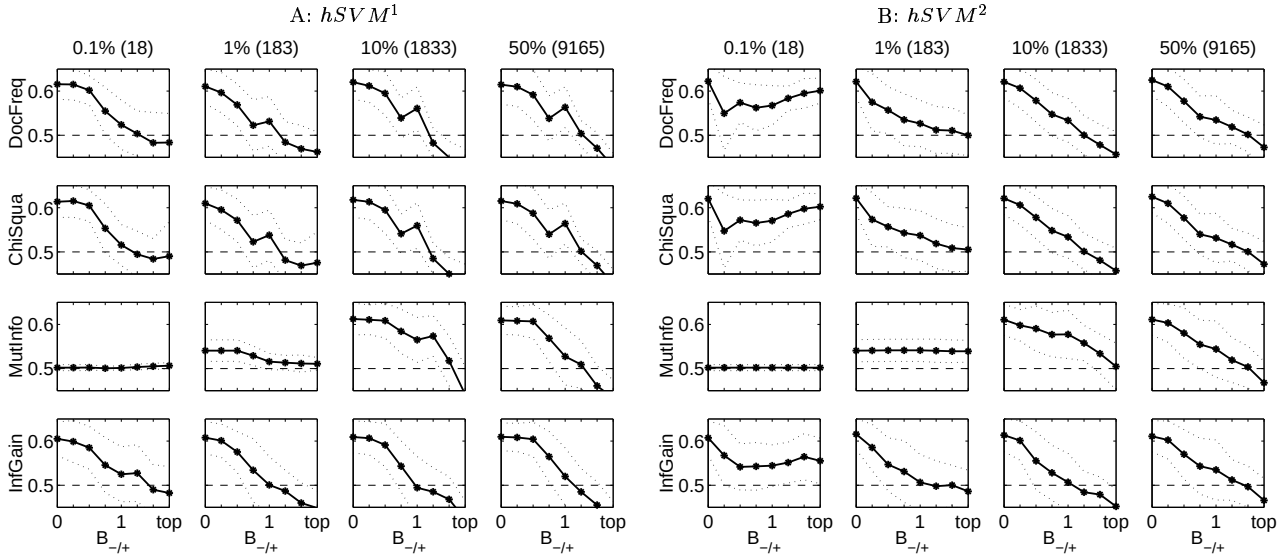

Figure 3: Mean AROC for the KDD subtask "either" as a function of the mixture ratio $B_{-/+}$ for four different fractions of the original feature set (0.1%, 1%, 10% and 50%), for three linear kernel machines with $C = 100$: (A) $hSVM^1$ and (B) $hSVM^2$. $B_{-/+}$=[0, 0.01, 0.1, 0.5, 1, 5, 10, $\top$].

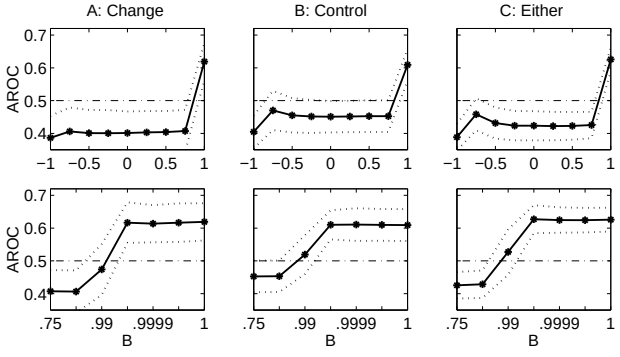

Figure 4: Mean AROC with standard deviation envelope as a function of balance factor B for the AHR dataset. We use here $hSVM^2$ with $C = 10$.

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 (4), we investigate the performance of classifiers using weight balancing as described in Section 4.

6.3.1 Tests on AHR data

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 C show similar trends [16]. 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.

6.3.2 Tests on Reuters

Experiments reported in the previous sections show that 1-class SVMs tend to perform better than traditional 2-class SVMs on AHR-data. On the other hand, using both classes always produces better results with Reuters categories. In order to understand the reason for this difference in behaviour, we performed additional experiments on the Reuters data set to observe the performance of 1-class and 2-class SVMs when the most frequent features are removed. To this end, we removed the most frequent x% of the 20,197 features (highest $DocFreq$ scores) and trained classifiers on random 5% (stratified sample) and then tested on remaining 95% of the data. As usual, the average AROC $\pm Std$ for 20 such tests is shown in Figure 5. Four different target cases were used: the 3rd, the 6th, the 10th and the combined 11th-15th largest categories. The sizes of target classes are shown in the sub-figure titles.

An inspection of plots highlights a few observations:

1. The accuracy of all classifiers is very high when all features are used. As the most frequent features are removed, all SVM models start degenerating, however, the drop in performance for 2-class SVM models is much larger, and 1-class SVM models start outperforming the 2-class models. This behaviour is also present in other categories not shown in Figure 5, so long as the target class is less than 10% of the total data. This trend of better performance with 1-class models is most apparent in $hSVM^1$, although $hSVM^2$ also shows similar trends. Thus, when there are many weakly informative features, and the target class is a small fraction of the data set, 1-class SVMs outperform the traditional 2-class SVM models.

2. The mean AROC is always > 0.5 indicating that even after feature removal, this data set does not quite have all the properties of AHR-data where balanced 2-class models

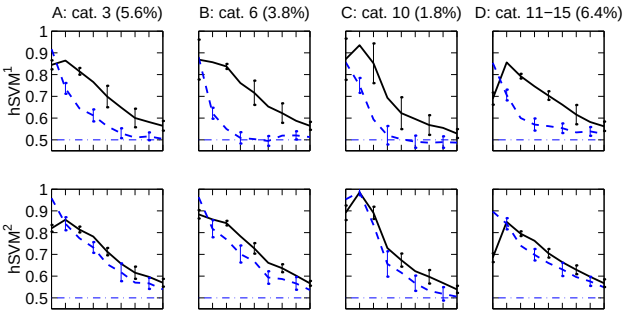

Figure 5: Mean AROC as a function of the % of features removed (with standard deviation envelope). Four different target cases were used: the 3rd, the 6th, the 10th and the combined 11th-15th largest categories. Results are presented for two machines: (1) $hSVM^1$ and (2) $hSVM^2$ Plots are shown for the positive 1-class ($B = +1$) (solid line) and the balanced 2-class ($B = 0$) (dashed line) modes.

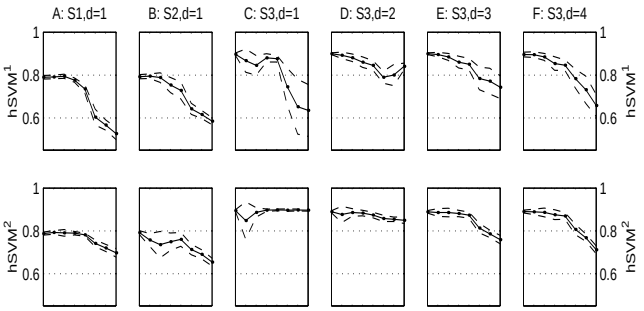

Figure 6: Mean AROC with ± standard deviation envelopes as a function of the mixture ratio $B_{-/+}$ for four machines with $C = 1000$: $hSVM^1$ and $hSVM^2$. Plots A, B, C show results for linear kernels with S_1, S_2 and S_3, respectively. Results with higher degree polynomial kernels ($d = 2, 3, 4$) are shown for S_3 in plots D, E and F, respectively. $B_{-/+}$=[0, 0.01, 0.1, 0.5, 1, 5, 10, 35].

performed worse than random for many settings of the regularisation constant.

7. EXPERIMENTS WITH SYNTHETIC DATA

We observed in Section *6.2.1* that even in low dimensional space, the 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, hence once the background instances are added, the SVM solution becomes suboptimal. Now we explore this intuition using synthetic data.

We use three data sets of instances of similar structure. The observation vectors in these synthetic data sets contain a small number n_{inf} of *informative attributes* and the remaining, larger number, n_{noise}, of *noise attributes*. These attributes are binary, generated according to uniform random distribution with probabilities P_{inf} and P_{noise} of value $= +1$, respectively. The informative attributes determine the labels modulo the additional *label noise* which is the random reversal of certain proportions of labels, namely the proportions LN_+ of the positive and LN_- of the negative labels. In all sets, we generate $m = 9000$ instances of which $p_{y=+1} = 3\%$ have labels $y = +1$.

- S_1: For this data set we use $n = n_{inf} + n_{noise} = 1 + 999$ dimensions and $P_{noise} = 2\%$. The labels are generated as a random bipolar label vector $y \in \{\pm 1\}^{9000}$ with the proportion $p_{y=+1} = 3\%$ of positive examples. For the informative dimension we set $x_{inf} = (y+1)/2 \in \{0, 1\}$ and then change randomly the proportion $LN_- = 20\%$ of 0s to 1s.

- S_2: In this case $n_{inf} = 10$, $n_{noise} = 990$, $P_{inf} = 5\%$, $P_{noise} = 2\%$. Having defined informative attributes $x_{inf,i} \in \mathbb{R}^{10}$ for $i = 1, ..., 9000$, we have randomly generated a vector $v \in \mathbb{R}^{10}$, then chosen a bias $b \in \mathbb{R}$ such that for 2004 ($\approx 22\%$) instances i we got the scores $x_{inf,i} \cdot v > b$. Of these 2004 instances, we randomly select 270 instances (= 3% of 9000) and label them $+1$ and the remaining 8730 instances we labelled -1.

- S_3: This set was designed to test the impact of non-linear kernels. It is generated as S_1 with the difference that only $n = n_{inf} + n_{noise} = 1 + 19 = 20$ dimensions are used and the random proportions LN_+ and LN_- of the both $+1$ and of 0 entries, respectively, are reversed in the second phase of the generation of the informative attribute x_{inf}.

In experiments, each set of 9000 instances generated as described above, was split randomly into 3000 training and 6000 test instances, with proportional sampling (without replacement) from both classes. All results reported are averages of 20 such random splits.

Figure 6 presents the results of experiments evaluating AROC as a function of the mixture ratio $B_{-/+}$, for the two kernel machines. For all three data sets, we show the results for the linear kernel (Figures 6A-6C), and for S_3 we show the impact of higher degree polynomial kernels (Figures 6D-6F). The results, especially for $hSVM^1$, strikingly resemble those obtained for the AHR data (c.f Figure 3), with the consistent pattern of decreasing performance with increasing proportion of negative class instances. As kernel degree increases we observe the familiar pattern of decreasing performance with increasing dominance of negative class instances (Figures 6C-6F). Thus, the relatively low dimensional S_3 data set when used with higher degree polynomial kernels behaves in a way similar to that of the high dimensional datasets S_1 and S_2 with linear kernels.

We also experimented with different values of the regularisation constant C, but found that this had marginal impact on AROC in the above settings.

In addition, our experiments with different label noise settings (LN_+ and LN_-) show that the pattern of decreasing performance with increasing amounts of negative class instances persists with different levels of label noise.

8. DISCUSSION

Deterioration of 2-class SVMs. The degradation of learning performance in the presence of abundant negative examples has been noted in [17]. Their solution of focusing on the best positive region works in low dimensional input

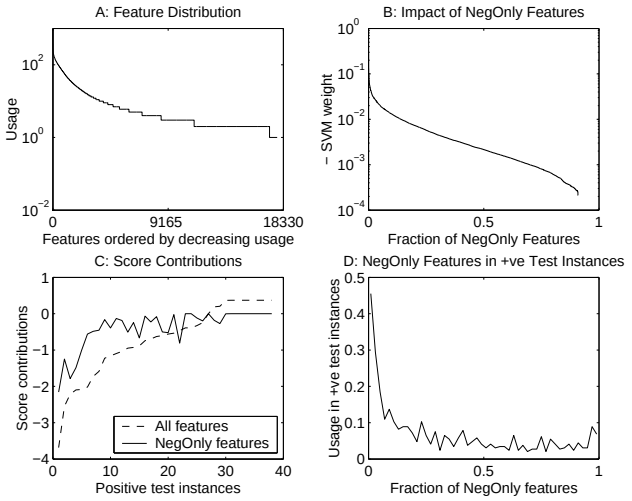

Figure 7: Understanding the influence of sparse high dimensional space on the SVM solution of two-class learner. (A) Usage of features in decreasing order of usage. (B) Magnitude of SVM weights for two-class model for the $NegOnly$ (features used only in the negative class in the training set) features in decreasing order of magnitude. (C) Contribution of $NegOnly$ features to SVM score. (D) Usage of $NegOnly$ features in the positive test set.

space when there is a single region to be labelled as positive and the minority class is around 10% of the data. For our situation of very high dimensional input space with around 3% minority class data in the case of AHR, the more drastic solution of totally ignoring negative class examples seems to work better for all machines.

Such learning from positive examples alone has been shown to be of benefit for some real world balanced data sets with low dimensional input space [12; 13]. In there, it is shown (using synthetic datasets in two dimensional input space) that negative examples may be counter-productive in domains where the negative class data are not all located in a few localised spots, but rather, they wrap around the positive data.

In order to gain some insight into why this phenomenon occurs with high dimensional input space, we first explore the feature space for one particular randomisation of the KDD cup 2002 pooled data. Figure 7(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 7(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 our hypothesis is that for many of these features balanced 2-class $hSVM^2$ allocates excessively low (highly negative) weights, which is an 'easy way' to minimise the margin errors. However, when some of these features occur in positive test examples, they push the scores of these examples excessively into negative direction, which causes a deterioration in the overall performance.

Figure 7 shows results corroborating this hypothesis. In Figure 7(B), we plot 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 7(D) their usage in the test instances contributes large negative scores for the positive instances in the test set, cf. Figure 7(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).

Persistent dominance of 1-class SVMs. The above analysis is applicable to a high dimensional feature set. However, we have 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 concurs with that provided in [12; 13]. 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 diffuse 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.

Weakly informative features. An alternative explanation for the relatively good performance of 1-class SVMs is implied by experiments with Reuters data. We hypothesise that one factor is the relatively "weak" connection between the labels and the features in the case of AHR-data. Since the contrary is true for topic-based classification in Reuters, the superior performance with fringe classifiers is not evident until the most frequent features, which tend to be strongly indicative of the labels for this dataset, are removed (Figure 5). Thus, we may expect 1-class SVMs to work well in other real world applications with weak connection between labels and attributes.

Interaction of learning algorithm with feature selection methods. An additional point regarding selection of features is that the performance of any dedicated statistical system for feature selection is a function of both, the selection method and the learning strategy for evaluation of the selection. For instance, all 1-class SVMs in Figure 3 perform very well with features selected by *ChiSqua* while all 2-class learners perform poorly with the same features. Thus, evaluation of feature selection methods cannot be performed in isolation from the learning algorithm.

Impact of kernels. Experiments with the polynomial kernels seem to indicate that interactions between the 19 noisy attributes in the set S_3 are equivalent to explicit addition of hundreds of extra noise attributes in the datasets S_1 and S_2. The higher the degree of the kernel, the more such 'noisy' virtual attributes are added (on the level of the feature space) and the more pronounced is the difference between one-class and two-class learning. Note that in this

case, in contrast to the case of AHR-data case, the range of AROC values is around 60-90% and never drops below 50%.

9. CONCLUSION

In this paper we have explored imbalance compensation techniques for data with heavily unequal priors using two real world 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. Experiments with synthetic data indicates that favourable conditions for such performance can naturally arise in many other situations, in particular when popular support vector machines with non-linear kernels are used.

Our investigation suggests that one-class learning from positive class examples can be a very robust classification technique when dealing with very unbalanced data and high dimensional noisy feature space. It can be used as an alternative to aggressive feature selection usually used in such situations and can be very attractive for learning with non-linear kernels, when direct feature selection on the feature space level cannot be implemented.

Acknowledgements

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

10. REFERENCES

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

[2] R. Centor. The use of ROC curves and their analysis. *Med. Decis. Making*, 11:102 – 106, 1991.

[3] P. K. Chan and S. J. Stolfo. Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In *Knowledge Discovery and Data Mining, KDD-98*, pages 164–168, 1998.

[4] P. K. Chan and S. J. Stolfo. Toward scalable learning with non-uniform distributions: Effects and a multi-classifier approach. In *http://www1.cs.columbia.edu/ sal/recent-papers.html*, 1999.

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

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

[7] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, Cambridge, 2000.

[8] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive Learning Algorithms and Representations for Text Categorization. In *Seventh International Conference on Information and Knowledge Management*, 1998.

[9] C. Elkan. The foundations of cost-sensitive learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 973–978, 2001.

[10] T. Fawcett. Roc graphs: Notes and practical considerations for researchers. In *HP Labs Tech Report HPL-2003-4*, 2003.

[11] D. Hand and R. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45:171 – 186, 2001.

[12] N. Japkowicz. Are we better off without counter examples? In *Proceedings of the First International ICSC Congress on Computational Intelligence Methods and Applications (CIMA-99)*, pages 242–248, 1999.

[13] N. Japkowicz, C. Myers, and M. Gluck. A novelty detection approach to classification. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 518–523, 1995.

[14] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis Journal*, 6(5), 2002.

[15] M. Joshi. On Evaluating Performance of Classifiers for Rare Classes. In *Proceedings of the Second IEEE International Conference on Data Mining (ICDM'02)*, 2002.

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

[17] M. Kubat, H. R., and S. Matwin. Learning when negative examples abound. In *Proceedings of the Ninth European Conference on Machine Learning ECML97*, 1997.

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

[19] L. M. Maneivitz and M. Yousef. One-class SVMs for Document Classification. *Journal of Machine Learning Research*, 2:139–154, 2002.

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

[21] B. Raskutti, H. Ferrá, and A. Kowalczyk. Second Order Features for Maximising Text Classification Performance. In *Proceedings of the Twelfth European Conference on Machine Learning ECML01*, 2001.

[22] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw Hill, 1983.

[23] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution. In *Technical Report 99-87, Microsoft Research, 1999.*, 1999.

[24] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond.* MIT Press, 2001.

[25] V. Vapnik. *Statistical Learning Theory.* Wiley, New York, 1998.

[26] Y. Yang and J. O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997.