In Proceedings of the European Workshop on Data Mining and Text Mining for Bioinformatics, held in conjunction with ECML/PKDD, 2003, Dubrovnik, Croatia

Fringe SVM Settings and Aggressive Feature Reduction

Adam Kowalczyk
Bhavani Raskutti
Telstra Corporation, 770 Blackburn Road, Clayton, Victoria, Australia

Abstract

Statistical techniques for aggressive feature reduction are studied on data obtained in a gene knock-out experiment. The essential part of the process is automatic assessment of the quality of various feature selection methods. This is done by comparison of the performance of discriminating models built on candidate subsets of features. Experiments show that typical settings of popular 2-class discriminators, support vector machines (SVM), cannot be used as they produce models of very poor quality. The proposed way around is to use “fringe classifiers” such as SVMs trained on positive class data only or class centroids. Additionally, we also use models generated by such algorithms directly for identification of most discriminating features. We recommend that such simple machine learning techniques should be included into a repertoire of discriminators used on such occasions. We show that such relatively superior performance of fringe SVMs can also be observed on regular text-mining data bases, such as Reuters newswire benchmark, if only the less frequent features (words) are used.

1. Introduction

As our knowledge of genome of various organisms increases, the number of potentially relevant scientific articles proliferates, making it impossible for an individual to keep up. This creates a need for statistical tools capable of pre-filtering and identifying useful information prior to human inspection, thus alleviating the information overload.

Traditional statistical tools typically deal with situations when there are more “training” examples than potentially predictive features to consider. However, in biological domains, it can be expensive to perform laboratory tests to obtain training data, while it may be easy to obtain millions of potentially relevant features. Solving problems in such domains would therefore require methods that robustly deal with (1) extremely high dimensional spaces with only a handful of “training” examples and (2) “low” relevance of available features to any particular problem.

As a first step towards development of such methods, we explore, in this paper, the use of support vector machines (SVM) in many different modes. We consider, in particular, fringe classifiers that correspond to some extreme settings of parameters, e.g., learning from examples of a single class and settings that provide solutions that are equivalent to learning from data centroids. We show that for the 2002 KDD cup data (AHR-data) [4], these fringe classifiers provide robust discriminating models, which are far more accurate than typical 2-class SVMs. Additionally, we show that the phenomenon of domination of fringe classifiers is not unique to AHR-data, but is observable in the popular Reuters Newswires text mining benchmark.

We investigate the utility of such classifiers along with other discriminating models for feature selection. We use these classifiers, firstly for the evaluation of feature selection methods based on their accuracy of prediction, and secondly, for directly selecting informative features using the generated models. Our investigation shows that such model-based selection can both provide accurate classifiers and select a small subset of features which may be used in a variety of ways: (1) inspection by researchers for identification of biological mechanisms underlying an investigated phenomenon, (2) generation of key words for retrieval of relevant scientific articles, and (3) development of a more refined and simpler to interpret discriminating models.

The paper is organised as follows. Section 2 introduces the machine learning algorithms and the performance measure used in this research. Sections 3 and 4 present our experiments with AHR-data and Reuters data, respectively. In Section 5, we discuss the implications of our results and present some intuitive explanations of
where the observed phenomena.

2. Classifiers and Metrics

In this section we introduce the basic machine learning algorithms used in this paper. We focus on linear classifiers, in particular, on Support Vector Machines (SVM). There are a number of reasons for this focus. Firstly, SVMs have been top performers in text [9, 16] and biomining tasks. Secondly, they have been top performers on AHR-data: 3 out of the top 5 submissions to KDD Cup 2002 were based on SVMs [6, 13, 11]. Finally and most importantly, SVMs are well suited to process sparse, high dimensional data.

Our classification problem is formulated as follows. 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 $i = 1, \ldots, m$. The case of prime interest here is when the target class, labelled $+1$, is much smaller than the background class (labelled $-1$), $\approx 1\%$ of the data. Our aim is to find a “good” discriminating linear function

$$f(x) := w \cdot x + b$$

(1)

that scores the target class instances higher than the background class instances. Here $x \in \mathbb{R}^n$. $w$ denotes the dot product in $\mathbb{R}^n$ and $(w, b) \in \mathbb{R}^n \times \mathbb{R}$ is defined by one of the five learning algorithms described below.

The first four algorithms, requiring dedicated solvers, are versions of the popular SVMs. For all of them the solution $(w, b) \in \mathbb{R}^n \times \mathbb{R}$ is defined as a minimiser of the regularised risk functional of the following form.

$$\|w, b\|^2_* + \sum_{i=1}^m C_i \phi(1 - y_i(w \cdot x_i + b)),$$

(2)

where $\| \|_*^2$ is a squared “norm” penalising for the “complexity of the classifier”, $\phi: \mathbb{R} \rightarrow \mathbb{R}_+$ is a convex loss function penalising for deviations of the machine from allocated labels and the regularisation constants are defined as follows:

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

(3)

for $i = 1, \ldots, m$, where $C > 0$, $m_+$ and $m_-$ denote the numbers of examples with labels $y_i = +1$ and $y_i = -1$, respectively. Here $-1 \leq B \leq 1$ is a balance parameter designed to balance the impact of instances from the positive and the negative class, respectively.

Now we specify four variations of the regularised risk (2) leading to four different machines to be used in this paper.

**Algorithm 1, SVM$_{BC}$**: This is the popular SVM with linear penalty. Here, we use the norm $\|w, b\|^2_* := \|w\|^2 = w \cdot w$ and the “hinge loss” $\phi(\theta) := \max(0, \theta)$, $\theta \in \mathbb{R}$ [5, 19, 20];

**Algorithm 2, hSVM$_{BC}$**: Replacing the norm in the above definition by

$$\|w, b\|^2_* := \|w\|^2 + b^2$$

we obtain the homogeneous SVM with linear penalty.

**Algorithm 3, hSVM$_{2BC}$**: For the (homogeneous) SVM with quadratic penalty [5] we use norm (4) and the squared hinge loss $\phi(\theta) := (\max(0, \theta))^2$ for $\theta \in \mathbb{R}$;

**Algorithm 4, hRN$_{2BC}$**: For the homogeneous regularisation network [7, 21] or the ridge regression [5, 7, 21] we use norm (4) and ordinary square loss $\phi(\theta) := \theta^2$ for $\theta \in \mathbb{R}$.

**Algorithm 5, Cntr$_B$**: This is the simplest of the five algorithms. Here, we set $b := 0$ and

$$w := \frac{1 + B}{2m_+} \sum_{y_i = +1} x_i = \frac{1 - B}{2m_-} \sum_{y_i = -1} x_i.$$  

For $B = +1$ vector $w$ is exactly the centroid of the minority (the target) class, for $B = -1$ it is the centroid of the majority (the background) class while for $B = 0$ it is half of the difference between the centroids of the two classes.

Note that Cntr$_B$ can be viewed as the “limit” case of the SVM, in the sense that Cntr$_B = \lim_{C \rightarrow 0^+} \frac{\text{SVM}_{2BC}}{C}$, for $p = 1, 2$. We refer to [12] for further discussion of this link.

2.1 1-class SVMs

Note that hSVM$_{1BC}$, hSVM$_{12BC}$ and hRN$_{1BC}$ implement classifiers that correspond to separation of the data $(x_i, 1, y_i) \in \mathbb{R}^n \times \mathbb{R} \times \pm 1$ by a hyperplane $\langle (w, b), (x, 1) \rangle = 0$ passing through the origin $(0, 0) \in \mathbb{R}^n \times \mathbb{R}$. One thing to stress is that (2) provides a non-trivial (i.e. $\neq$constant), unique classifier (1) in all “regular” cases of interest, in particular, for $B = \pm 1$ if at least one $C_i \neq 0$ and $(0, 0) \in \mathbb{R}^n \times \mathbb{R}$ does not belong to the convex shell spanned by all vectors $y_i(x_i, 1) \in \mathbb{R}^N \times \mathbb{R}$. These cases of $B$ are equivalent to learning from data belonging to a single class label, the target class $y_i = +1$ for $B = +1$ and the background class $y_i = -1$ for $B = -1$. We shall call such machines the 1-class SVM’s.

On the level of the extended feature space $\mathbb{R}^n \times \mathbb{R}$, any hSVM$_{1BC}$, hSVM$_{12BC}$ or hRN$_{1BC}$ can be reduced to a single class machine. In fact, we can always absorb the signum $y_i$ by considering the data $(\tilde{x}_i, \tilde{y}_i) := (x_i, y_i)$.
\((y_i, x_i, y_i, 1)\) rather than \((x_i, 1, y_i)\) and then minimising the following functional equivalent to (2):

\[
\langle \tilde{w}, \tilde{w} \rangle + \sum_{i=1}^{m} C_i \phi (1 - \langle \tilde{w}, \tilde{z}_i \rangle)
\]

(5)

where \(\langle \ , \ \rangle\) stands for the dot product in \(\mathbb{R}^n \times \mathbb{R}\). This formally reduces the two class problem (2) to “single class learning”. In the case of \(h SVM^1\), the solution to (5) can be found using \(SVM^1\) if an extra point, namely \((0,0) \in \mathbb{R}^N \times \mathbb{R}\), with the opposite label \(-1\), is added to the data. Such a method for one-class learning has been considered previously in [14, 18].

We shall refer to any \(h SVM^P \) or \(h SVM^2\) as fringe SVMs. We include here \(Cntr_{\pm 1}\) since the SVM solution for low \(C\) approaches the solution of \(Cntr_{\pm 1}\) (cf. [12]). This equivalence also motivates our setting of \(C\) to 5000 (the “hard margin” case) for our feature selection experiments, since the low values of \(C\) are covered by the centroid solution.

2.2 Performance measures

We have used \(A ROC\), the Area under the Receiver Operating Characteristic (ROC) curve as our main performance measure. In that, we follow the steps of KDD Cup 2002, 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 \(\theta\) 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]). In the latter case, \(A ROC\) 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. \(A ROC(f)\) 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 \(A ROC\) 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].

There are some ambiguities in the case of \(A ROC\) 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

\[
A ROC(f) = P(f(x_i) < f(x_j)) - y_i = y_j = 1
\]

\[+0.5P(f(x_i) = f(x_j)) - y_i = y_j = 1\]

expressing \(A ROC\) in terms of conditional probabilities.

3. Analysis of AHR-data

In our main experiments we have used 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. [4] for more details). In this paper we follow the setting of the “broad task” of the KDD Cup: the discrimination between 127 ‘positive’ genes from the combined class encompassing the labels “change” and “control” and the remaining 4380 genes forming the ‘negative’ class. In our experiments this set has been repeatedly split into 70% for training and 30% for testing. All averages and standard deviations reported are for independent tests on 20 such random splits.

3.1 Data representation

Each training and test gene was represented by a vector of binary attributes extracted from the very rich data sources provided: function/localisation annotations, protein-protein interactions and Medline abstracts.

Hierarchical information about function, protein classes and localisation was converted to a vector per gene. For instance, the following two entries in the file function.txt

YGR072W CYTOPLASM SUBCELLULAR LOCALISATION
YGR072W NUCLEUS SUBCELLULAR LOCALISATION

yielded three function attributes: “cytoplasm”, “subcellular localisation” and “nucleus” each with a value of 1 for the gene “YGR072W”. This processing created 409 attributes: 213 for gene function, 154 for protein classes and 42 for localisation.

Textual information from all abstracts associated with a gene was converted to ‘word token’ presence vectors (‘a bag of words’). A ‘word token’, in this context, is any string of alphanumeric characters, which may or may not correspond to an ordinary word. Word tokens corresponding to words in a standard list of stop words, such as “the”, “a” and “in”, have been
The gene-gene interaction file is symmetric. Hence, the abstracts.

The above abstract processing resulted in 48,089 word token attributes. Around 3/4th of these attributes were subsequently eliminated by discarding all those that occurred in only one gene, and by discarding all those which had a total frequency that was greater than one standard deviation from the mean. After this processing, we were left with 16,474 attributes from the abstracts.

Thus, the total number of binary attributes used by the learning algorithm was 18,330 (= 409 + 16,474 + 1,447).

3.2 Feature selection

We now explore the utility of the classifiers for feature selection as (i) techniques for evaluation of various selected feature sets and (ii) tools for selection of such sets.

We investigate several strategies for scoring features: the first four score features based on their distribution in the training set (the filter approach as defined in [10], while the others are based on the SVM models generated (w). In all cases, the computed score is used to sort the features so that the most informative features may be selected.

A: DocFreq (Document frequency thresholding): This method has its origins in information retrieval [17] and is based on the supposition 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 has been equal to 1.

B: ChiSqua ($\chi^2$): The $\chi^2$ 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 [22].

C: MutInfo: (Mutual Information): The score is allocated to a feature on the basis of the joint and marginal probabilities of its usage estimated from the training set [22].

D: InfGain: (Information gain): This is frequently employed as a term goodness measure in machine learning [15], and measures the number of bits of information obtained for class prediction by knowing the presence or absence of a term in an instance.

E-J: Model-based feature selection: In this case, the score of a feature is simply the magnitude of the weight allocated to it by a linear model generated to discriminate two classes of interest, i.e. corresponding entries in the vector $w = (w_1, ..., w_n)$. Thus, our method is much simpler and easier to implement than the sophisticated SVM-model-based selection described in [8].

Our model-based approach is also different from the wrapper approach used in [10] in that we do not restrict the use of the selected features for the particular learning algorithm. Instead, the chosen features are then used as features for evaluating all algorithms.

In our experiments we employ a small variation: we average $w$ over 20 models generated for 20 random stratified splits of the data into 70% training and 30% test sets (instances of both classes were split independently in the indicated proportions). We have used three learning machines, $h_{SVM}^{B,5000}(E,F)$, $h_{SVM}^{B,5000}(G,H)$ and $Cntr_B(I,J)$ in two modes: positive 1-class mode $B = +1$ (E,G,I) and balanced 2-class mode, $B = 0$ (F,H,I), thus yielding six additional methods.

Figure 1 shows the results of evaluation of the ten feature selection techniques (columns A-J) by four different algorithms (rows 1-4). Each evaluation technique has been used in two modes: positive 1-class and balanced 2-class. (We have not shown evaluation results for $SVM^1$ as they are very similar to those for 2-class $h_{SVM}^{1}$.) The results can be summarised as follows:

1. As a general rule, for all SVMs, 1-class models perform much better than 2-class models when using the same set of features. In addition, these two modes of SVM often give quite opposite evaluation of the utility of selected features (the notable exception being column H). While 1-class finds them informative ($AROC > 0.5$), 2-class finds them detrimental with $AROC < 0.5$, i.e. below that of the random classifier.

2. DocFreq and MutInfo both provide very poor results for low number of features, although they use completely different metrics for scoring. MutInfo is strongly influenced by the marginal probability of terms and tends to favour rare terms, while DocFreq selects the most common terms.
Figure 1. Evaluation of ten feature selection methods (columns A-J) by four classification algorithms (rows 1 - 4). Plots show mean AROC ± Std as an envelope as a function of the % of features selected out of the total 18,330. Curves are plotted for two modes: the positive 1-class ($B = +1$, solid lines) and balanced 2-class ($B = 0$, dashed lines). All SVMs used $C = 5000$.

3. $Cntr_0$ (row 4, dashed line) performs the best of all 2-class algorithms, generally matching 1-class centroid classifier, $Cntr_{+1}$.

4. 2-class h$\text{SVM}^+_2,5000$ (Columns H) provides very good features for 2-class mode classifiers, allowing them to perform above random $AROC > 0.5$. In fact, this feature selection in combination with ridge regression learning, $h\text{RN}^+_0,5000$, provides the best performance for around 2% ($\approx 300$) features.

5. As a general trend, features selected by models allow development of better discriminating models than features selected by the evaluated feature selection algorithms (A-D), provided positive 1-class mode is used for learning.

3.2.1 Selected Features

Table 1 lists the top 20 features selected according to different methods. We observe that there is a large overlap between features selected by $\chi^2$ and positive 1-class methods: $h\text{SV}M^{\chi}_1,5000$, $p = 1,2$ and $Cntr_{+1}$. The features selected are primarily from function and localisation data. The 10 common features in the top 20 are: 1: $F_4$ - subcellular localisation, 2: $F_7$ - cell cycle and dna processing, 3: $F_{10}$ - metabolism, 4: $F_{17}$ - cell fate, 5: $F_{27}$ - mrna transcription, 6: $F_{28}$ - transcription, 7: $F_{29}$ - unclassified proteins, 8: $F_{32}$ - cellular transport and transport mechanisms, 9: $F_{58}$ - protein fate (folding, modification, destination), 10: $L_4$ - cytoplasm.

Similarly, there are overlaps between $\text{InfGain}$ and the 2-class centroid, $Cntr_0$. However, these sets include many features from the abstracts, and thus are different from those selected by the positive 1-class methods. The 15 common features are: 1: $F_4$ - subcellular localisation, 2: $F_{22}$ - nucleus, 3: $L_3$ - nucleus, 4: $A_{419}$ - redund, 5: $A_{426}$ - much, 6: $A_{543}$ - abnorm, 7: $A_{613}$ - comprom, 8: $A_{639}$ - despit, 9: $A_{711}$ - harbor, 10: $A_{973}$ - surprisingli, 11: $A_{1002}$ - subset, 12: $A_{1291}$ - carboxi, 13: $A_{1609}$ - green, 14: $A_{2104}$ - taken, 15: $A_{4290}$ - inviabl.

Interestingly, there are no overlaps in the top 20 features between the 2-class centroid ($Cntr_0$) and the 2-class SVMs ($B = 0$) or between $\text{InfGain}$ and the 2-class SVMs.

4. Tests on Reuters

Experiments reported in the previous section show that fringe SVMs tend to perform better than traditional 2-class SVMs on AHR-data. In this section we report some experiments with popular text mining benchmark, Reuters-21578 news-wires, which show similar tendency. For these experiments we used a col-
Table 1. Top twenty features selected by ten feature selection methods. We use the following convention: the letters stand for the data source (A-abstracts, F-function class, P-protein class, I-gene interactions, and L-localisation) and the subscript is the number of the feature. The last row gives mean AROC ± Std of the models used for the model-selection method. We put "-" in front of features with negative weights (2-class SVMs).

<table>
<thead>
<tr>
<th>Rank</th>
<th>Doc-Freq</th>
<th>Chi-Squa</th>
<th>Mut-Info</th>
<th>Inf-Gain</th>
<th>hSVM_B,5000 B=1</th>
<th>hSVM_B,5000 B=0</th>
<th>hSVM_B,5000 B=0</th>
<th>CntB B=0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>F95</td>
<td>F4</td>
<td>P48</td>
<td>P4</td>
<td>F29</td>
<td>F2</td>
<td>F29</td>
<td>L4</td>
</tr>
<tr>
<td>2</td>
<td>I150</td>
<td>F29</td>
<td>P32</td>
<td>P22</td>
<td>F4</td>
<td>F46</td>
<td>F4</td>
<td>L5</td>
</tr>
<tr>
<td>3</td>
<td>A1045</td>
<td>F10</td>
<td>P110</td>
<td>L3</td>
<td>F20</td>
<td>F150</td>
<td>F58</td>
<td>L104</td>
</tr>
<tr>
<td>4</td>
<td>I1136</td>
<td>L3</td>
<td>I61</td>
<td>A2104</td>
<td>P7</td>
<td>F10</td>
<td>F19</td>
<td>F22</td>
</tr>
<tr>
<td>5</td>
<td>F72</td>
<td>F32</td>
<td>I66</td>
<td>A2460</td>
<td>F58</td>
<td>F58</td>
<td>L4</td>
<td>F16</td>
</tr>
<tr>
<td>6</td>
<td>F39</td>
<td>F28</td>
<td>I43</td>
<td>A343</td>
<td>L4</td>
<td>F104</td>
<td>L1</td>
<td>F38</td>
</tr>
<tr>
<td>7</td>
<td>F89</td>
<td>F18</td>
<td>I55</td>
<td>A426</td>
<td>F31</td>
<td>F42</td>
<td>F2</td>
<td>I43</td>
</tr>
<tr>
<td>8</td>
<td>A925</td>
<td>F7</td>
<td>I95</td>
<td>A11</td>
<td>F10</td>
<td>F38</td>
<td>F22</td>
<td>F210</td>
</tr>
<tr>
<td>9</td>
<td>A970</td>
<td>L4</td>
<td>J33</td>
<td>A1669</td>
<td>F75</td>
<td>F28</td>
<td>F28</td>
<td>F64</td>
</tr>
<tr>
<td>10</td>
<td>A1098</td>
<td>F37</td>
<td>J34</td>
<td>A1002</td>
<td>F86</td>
<td>-L4</td>
<td>F23</td>
<td>F30</td>
</tr>
<tr>
<td>11</td>
<td>A426</td>
<td>F31</td>
<td>J35</td>
<td>A973</td>
<td>F67</td>
<td>-I835</td>
<td>F37</td>
<td>F28</td>
</tr>
<tr>
<td>12</td>
<td>A430</td>
<td>F17</td>
<td>I645</td>
<td>A461</td>
<td>F32</td>
<td>-I451</td>
<td>F27</td>
<td>F22</td>
</tr>
<tr>
<td>13</td>
<td>I448</td>
<td>F34</td>
<td>I68</td>
<td>A1291</td>
<td>F7</td>
<td>F2</td>
<td>F9</td>
<td>I309</td>
</tr>
<tr>
<td>14</td>
<td>I1209</td>
<td>F32</td>
<td>I666</td>
<td>A349</td>
<td>F55</td>
<td>F40</td>
<td>P17</td>
<td>F5</td>
</tr>
<tr>
<td>15</td>
<td>A660</td>
<td>F26</td>
<td>J32</td>
<td>A52</td>
<td>F66</td>
<td>F20</td>
<td>F6</td>
<td>F38</td>
</tr>
<tr>
<td>16</td>
<td>A586</td>
<td>F15</td>
<td>I38</td>
<td>F27</td>
<td>F64</td>
<td>F32</td>
<td>F21</td>
<td>F30</td>
</tr>
<tr>
<td>17</td>
<td>I1213</td>
<td>F16</td>
<td>I870</td>
<td>A1699</td>
<td>F17</td>
<td>-I420</td>
<td>F23</td>
<td>F32</td>
</tr>
<tr>
<td>18</td>
<td>I3299</td>
<td>F6</td>
<td>I671</td>
<td>A348</td>
<td>F28</td>
<td>F59</td>
<td>F5</td>
<td>I351</td>
</tr>
<tr>
<td>19</td>
<td>I1212</td>
<td>A426</td>
<td>I1050</td>
<td>A1345</td>
<td>F27</td>
<td>F18</td>
<td>F6</td>
<td>I43</td>
</tr>
<tr>
<td>20</td>
<td>P17</td>
<td>F25</td>
<td>I1600</td>
<td>A419</td>
<td>F6</td>
<td>F39</td>
<td>F26</td>
<td>I43</td>
</tr>
</tbody>
</table>

AROC | .61 ± .05 | .43 ± .05 | .63 ± .04 | .42 ± .05 | .62 ± .03 | .62 ± .04

lecion 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 (full analogy to the preprocessing of Abstract for AHR-data). Then we gradually removed the most frequent 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 ± Std for 20 such tests is shown in Figure 2. Four different target cases were used: the 3rd, the 6th, the 9th 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 2, 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_{B,5000}, although hSVM_{B,5000} also shows similar trends. Thus, when there are many weakly informative features, and the target class is a small fraction of the data set, the fringe classifiers outperform 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 2-class models performed worse than random for many settings of the regularisation constant.

5. Discussion

Related Research. A possibility of single class learning with support vector machines (SVM) has been noticed previously. In particular, Scholkopf et al. [18] have suggested a method of adapting the SVM methodology to 1-class learning by treating the origin as the only member of the second class. This methodology has been used for image retrieval [3] and for document classification [14]. In both cases, modelling was performed using examples from the positive class only, and the 1-class models perform reasonably, although much worse than the 2-class models learned using examples from both classes.

In contrast, in this paper, we show that for certain problems such as AHR-data, positive 1-class SVMs significantly outperform models learned using examples
from both classes.

**Impact of regularisation constant.** In this paper, we have restricted our investigation to $C = 5000$ (the “hard margin” case) and the very low values of $C$ through the $Cntr$ classifier. However, other experiments with different values of $C$ show that the performance of 1-class SVMs, unlike that of 2-class SVMs, is very robust across the whole range of $C$ values [12]. The performance of 2-class SVMs improves as the regularisation constant decreases, and for very low values of $C$, its performance is roughly equal to that of 1-class SVMs.

**Deterioration of 2-class SVMs.** We have observed that for AHR-data, fringe SVMs tend to have systematically better AROC than the traditional 2-class SVMs. Typically, the latter deteriorates with increase in the number of features used. In order to gain some insight into this phenomenon, we have compared two 2-class models, a $hSVM_{B,5000}$ (test $AROC = 0.39$) and $Cntr_0$ (test $AROC = 0.63$) trained on the same data split. For this training set, we found that there were 14,610 features occurring only in the negative class training instances ($NegOnly$ features). Both models allocate non-positive (all negative for $Cntr_0$) weights to such features. Our hypothesis is that for many of these features $hSVM_{B,5000}$ 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 3 shows results corroborating this hypothesis. In Figure 3A we plot weights allocated to the $NegOnly$ features, sorted in the reverse order of their magnitude, for each model separately and for each weight vector normalised to the unit length. Figure 3B shows probability of usage of these features in the positive class test examples (the curves are in fact 50-bin histograms). For both models the usage distribution is very similar and the most popular $NegOnly$ features have the most negative weights. However, the weights from $SVM$ for those most popular $NegOnly$ features are about twice as large in magnitude as those for $Cntr_0$ model. These weights result in excessive decreasing of scores of some positive test instances leading to deterioration in the overall performance.

**Persistent dominance of 1-class SVMs.** The above analysis is applicable to a high dimensional feature set. However, we have also observed in Section 3.2 that even in low dimensional spaces, this phenomenon of better performance with one-class learners 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, when data from both classes is
used, the minority class is “swamped” by the background class. In such a case the SVM solver seeking a “maximal margin” separation between classes, chooses a direction which is suboptimal in terms of AROC. The strange thing is that heavy discounting of the majority class by a factor $B = 0.99999$ does not rectify this impact completely [11].

**Weakly informative features.** An alternative explanation for the relatively good performance of fringe classifiers 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 2). Thus, we may expect fringe classifiers to work well in other real world applications with weak connection between labels and attributes.

**Importance of evaluation algorithm for feature selection.** An additional point regarding feature selection is that the performance of any dedicated statistical system for that purpose is a function of both, the feature selection method and the learning strategy for evaluation of the selection. For instance, all 1-class SVMs and all centroid learners in Figure 1 perform very well with features selected by ChiSqua and MutInfo, while all 2-class learners, other than the 2-class centroid, perform poorly with the same features.

### 6. Conclusion

We have shown that SVMs even for a single kernel can split into a number of different modes, with dramatically different performance. Thus this popular class of learning machines cannot be treated as a monolithic black box, but should be viewed as a rich family of classifiers that need to be carefully tuned if top performance is required.

Further, some easy to implement fringe classifiers, such as centroids and positive 1-class SVMs, often outperform complicated 2-class SVMs. The very good performance of the fringe classifiers is related to sparsity of data and weak links between labels and features, and persists even after aggressive feature reduction. Thus, these classifiers could be used as baseline methods for biomining in general and for machine evaluation of utility of various feature selections. In particular, we recommend the use of centroid classifiers, which are trivial to implement.

Finally, model-based feature selection techniques can provide better results than dedicated feature pre-selection algorithm, facilitating development of more accurate discriminating models.

**Acknowledgements**

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

**References**


