ARTIFICIAL DATA GENERATION FOR ONE-CLASS CLASSIFICATION

A Case Study of Dimensionality Reduction for Text and Biological Data

Santiago D. Villalba and Pádraig Cunningham
School of Computer Science and Informatics, University College Dublin, Ireland

Keywords: Dimensionality reduction, One-class classification, Novelty detection, Locality preserving projections, Text classification, Functional genomics.

Abstract: Artificial negatives have been employed in a variety of contexts in machine learning to overcome data availability problems. In this paper we explore the use of artificial negatives for dimension reduction in one-class classification, that is classification problems where only positive examples are available for training. We present four different strategies for generating artificial negatives and show that two of these strategies are very effective for discovering discriminating projections on the data, *i.e.*, low dimension projections for discriminating between positive and real negative examples. The paper concludes with an assessment of the selection bias of this approach to dimension reduction for one-class classification.

1 INTRODUCTION

Sometimes in practical classification problems we are given a sample in which only one of the classes, typically called the "positive" or "target" class, is well represented, while the examples for the other classes are not statistically representative or simply do not exist. That can be the case when the negatives space is too broad (*e.g.*, the writings of Cervantes against any other possible writing), when it is expensive to label the negatives (*e.g.*, multimedia annotation) or when negative examples have not yet arisen (*e.g.*, industrial process monitoring). In these cases building a discriminative model using the ill-defined negatives sample will lead to very poor generalization performance and therefore conventional supervised techniques are not appropriate (when usable).

One-class classification (OCC) techniques (Tax, 2001), designed to construct discriminative models when the training sample is representative of only one of the classes, emerge as a solution to this kind of problem. The difference is operational, while the task is still to accept or reject unseen examples, this can be done only based on their similarity to the known positives. Consequently OCC approaches can operate with no or very few negative training examples, handling the "no-counter-example" and "imbalanced-data" problems by considering only positive data.

Many of the domains where one-class classification is appealing are characterized by high dimensional datasets. This high dimensionality poses several challenges to the learning system and so dimensionality reduction becomes desirable. In this paper we propose a simple technique that aims to introduce a discriminative bias in dimensionality reduction for one-class classification. The algorithm is as follows: 1- enrich the training set by creating a second sample that will act as a contrast for the actual positives 2- apply dimensionality reduction in the enriched dataset and 3- use the low-dimensional representation found to train a one-class classifier. This idea follows a recent trend in the relevant literature where OCC is cast as a conventional supervised problem by sampling artificial negatives from a reference distribution (see section 3). In this way we try to bridge the gap between supervised classification and one-class classification

However, the gap is wide. Formally when tackling the classification task in a supervised way we are given a training set $Z = \{z^{(1)}, \ldots, z^{(n)}\}$ where $z^{(i)} = (\mathbf{x}^{(i)}, y^{(i)})$ is an input-output pair, $\mathbf{x}^{(i)} \in \mathcal{X}$ is an input example and $y^{(i)} \in \mathcal{Y}$ is its associated output from a set of classes. Usually $\mathcal{X} \subseteq \mathbb{R}^m$ so $\mathbf{x}^{(i)} = (\mathbf{x}^i_1, \mathbf{x}^i_2, \ldots, \mathbf{x}^i_m)$ is an m-dimensional real vector. Using Z we infer a classification rule $h \in \mathcal{H} : \mathcal{X} \rightarrow \mathcal{Y}$ which maps inputs $\mathbf{x}$ to predicted outputs $h(\mathbf{x}) = \hat{y} \in \mathcal{Y}$. Given the usual 01 loss $L_{01}(h,x) = I(f(x) \neq y)$ we are endeavor to find $\hat{h}$ that minimizes the risk

D. Villalba S. and Cunningham P. (2009).
ARTIFICIAL DATA GENERATION FOR ONE-CLASS CLASSIFICATION - A Case Study of Dimensionality Reduction for Text and Biological Data.
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, pages 202-210
DOI: 10.5220/0002310202020210

functional $R(h) = \int L_{01}(h,\mathbf{x})p(\mathbf{x})d\mathbf{x}$. The primary assumption in this learning setting is that Z is representative of the concept to be learnt, in this case the classification rule, which means that both the distribution of the inputs $p(\mathbf{x})$ and the conditional distribution of the classes given the inputs $p(y|\mathbf{x})$ can be estimated from Z.

Conceptually one-class classification is very attractive. In practice it is very hard. Due to the absence of a well-sampled second class, the learning system cannot get comprehensive feedback and therefore uncertainty governs the whole process. The fundamental machine learning assumption that the training set is representative of the concept to be learnt does not hold and by definition neither $p(\mathbf{x})$ nor $p(y|\mathbf{x})$ can be estimated. From an OCC perspective we call the incomplete information on $p(\mathbf{x})$ the lack of Knowledge of the Inputs Distribution (KID). The incomplete information on $p(y|\mathbf{x})$ means we lack an Estimatable Loss Function (ELF) that might be used in parameter setting or model selection.

The rest of the paper is organized as follows. In section 2 we introduce the problem of dimensionality reduction in OCC and describe Locality Preserving Projections, the dimension reduction technique we will use in combination with our artificial samples. In section 3 we present a brief review of the relevant literature for artificial negative generation and describe four simple strategies for generating artificial negatives. In section 4 we show the promising results of our approach in a comprehensive set of text classification problems and a biological dataset. We bring the paper to a conclusion in section 5.

2 DIMENSIONALITY REDUCTION

2.1 Dimensionality Reduction for One-Class Classification

The curse of dimensionality poses several challenges for data analysis tools (François, 2008). In practice, one-class problems are typically of high dimension so dimensionality reduction (DR) is an important preprocessing step. In fact, the evaluation on text classification presented by Manevitz and Yousef (Manevitz and Yousef, 2001) shows that one-class Support Vector Machine (SVM) performance is quite sensitive to the number of features used. This contrasts with two-class SVMs which are generally considered to be robust to high data dimensionality. Although the literature on the topic is quite sparse, it is necessary to study methods for combating high dimensionally in the one-class setting.

Using dimensionality reduction prior to one-class classification should follow this rationale: find a *discriminative* representation (by feature selection or transformation) that will improve the classification performance of the model *describing* the positive class. Due to the ELF problem, conventional supervised and semi-supervised DR techniques cannot be used for one-class classification. This is unfortunate because, clearly, supervision is more effective at discovering discriminative representations. On the other hand, unsupervised alternatives, relying on assumptions like locality or variance preservation, can be irrelevant or even harmful for classification, especially in the absence of actual negatives.

Unsupervised techniques can prove very useful when their bias are correct for the problem at hand and are well synchronized with the classifier being used (Villalba and Cunningham, 2007). Conventional techniques for unsupervised dimensionality reduction can do so as a byproduct of the underlying assumptions, but the KID problem has an important impact in their application. For example, consider the case of principal components analysis (PCA), perhaps the most popular feature transformation technique. PCA finds decorrelated dimensions in which the data variance is large, that is, where the data has a large spread. Theoretically spreading the data has nothing to do with finding discriminative directions, yet there are numerous scenarios where PCA enhances classification accuracy. However, based on geometrical intuitions and an assumption of solvability for the classification problem, we can distinguish two different scenarios when predicting the effectiveness of PCA for classification – if it has access to positives only or if it can see both positives and negatives.

This conjecture of solvability is based on this observation: in the real world, we will usually face types of classification problems where there will be class separability in at least some subspace. Often separability comes together with high variability between the classes and so, with a large spread in the whole data. If projecting into those discriminative subspaces will spread the data as a side effect, in practice we can take the reverse path and find high variability subspaces with the hope that they will lead to class separability. See figure 1 for a toy example.

On the other hand, in the pure one-class setting, with no negatives at all at training time, spreading the data can be regarded as a bad idea. Because of our total ignorance of the negatives, the approach should be to maximize the chance that, whatever is their distribution, we will accept as few of them as possible.

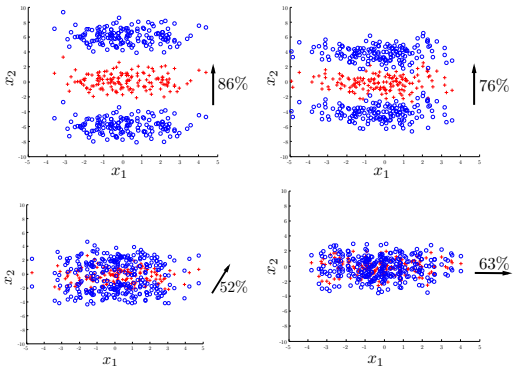

Figure 1: PCA over an artificial 2-dimensional example. We generate three mirroring data clouds by sampling from Gaussian distributions with diagonal covariance, the variance in x_1 ("horizontal dimension") is three times that in x_2 ("vertical dimension"), and the means differ only in x_2. We label the central cloud as the positives examples and the upper and lower clouds as the negatives, where the total number of positives and negatives is the same. When computing PCA only with the positive data, the first principal component is x_1, accounting for a 75% of the variance. This is clearly a bad option. On the right side of each plot we indicate the direction of the first principal component found by using both positives and negatives, labelled with the amount of variance it accounts for. We move the negative clouds so that they get closer and, eventually, overlap the positive cloud. In this case PCA finds "the right direction" until it is no longer possible to do so because both classes overlap.

This is achieved by projections that make the positive data occupy as little space as possible (collapsing), which in PCA corresponds to those explaining less variance (Tax and Muller, 2003).

In previous experiments with a wide range of high dimensional datasets, PCA was found, indeed, not as useful in a setting without actual negatives (Villalba and Cunningham, 2007). It can still help when the aim is to reduce the dimensionality while keeping as much information as possible, but the discriminative aspect that emanates from class separability completely disappears when training with just one class. Related to the KID problem, the usefulness of unlabeled data in classification is one of the central questions of the semi-supervised approach to learning (Chapelle et al., 2006, sect. 1.2); while in semi-supervised classification the effect of unlabeled data can be negligible from a theoretical point of view, unlabeled data plays a principal role in semi-supervised one-class classification (Scott and Blanchard, 2009).

2.2 Locality Preserving Projections

In this paper we focus on the interactions between one-class classification and Locality Preserving Projections (LPP) (He and Niyogi, 2003). LPP belongs to the family of spectral methods, where the low dimensional representations are derived from the eigenvectors of specially constructed matrices. The idea behind LPP is that of finding subspaces which preserve the *local structure* in the data. LPP has its roots in spectral graph theory (Chung, 1997), and the algorithmic details along with the specific setup used in our experiments are as follows:

1. **Construct the Adjacency Graph:** let X be the training set and G denote a graph with n nodes. We put an edge between nodes i and j if $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ are "close". When mixing with artificial generation techniques (sec. 3) we use a supervised k-nearest neighbors approach, where nodes i and j are connected if i is among the k-nearest neighbors of j or vice-versa and $y^{(i)} = y^{(j)}$, that is, we only allow links between examples of the same class. We also use self-connected graphs.
2. **Choose the Weights for the Graph Edges.** W is the adjacency matrix of G, a symmetric $n \times n$ matrix with W_{ij} having the weights of the edge joining vertices i and j, and 0 if there is no such edge. In this paper we use the simple approach of putting $W_{ij} = 1$ when nodes i and j are connected.
3. **Eigenmaps.** Compute the eigenvectors and eigenvalues for the generalized eigenvector problem:
$$XLX^T\mathbf{e} = \lambda XDX^T\mathbf{e} \qquad (1)$$
where D is a the degree matrix and L is the Laplacian matrix (Chung, 1997). The embedding is defined by the bottom eigenvectors in the solution of Equation 1.

It can be shown that by solving 1 we find the direction $\mathbf{e}$ that minimizes $\sum_{i,j}(\mathbf{e}^T\mathbf{x^{(i)}} - \mathbf{e}^T\mathbf{x^{(j)}})^2 W_{ij}$. This objective function incurs a high penalty if neighbor points $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ are mapped far apart. Therefore the bias of LPP is that of *collapsing* neighbor points. This seems appropriate for one-class classification, where collapsing the target class so that it occupies as little space as possible should account for many "attacking" distributions. LPP can prove effective for one-class classification in domains with high redundancy and low irrelevancy between dimensions, for example chemical spectra or data coming from multiple sensors. However, when using LPP with one-class classification we still miss the discriminative aspect, so we will be collapsing neighborhoods *inside* the target class without necessarily creating a discriminating representation.

3 ARTIFICIAL NEGATIVES GENERATION

A possible solution to incorporate a discriminative bias into OCC is to constrain the nature of the negatives by studying what are the relevant negative distributions that can appear in practice. In this way, we could generate artificial negatives (ANG) that could be used to help train the system. Our data would then come from a mixture distribution Q:

$$X \sim Q = (1-\pi)\mathcal{P} + \pi\mathcal{A} \quad (2)$$

where $\mathcal{P}$ is the distribution for the positives, $\mathcal{A}$ is the assumed distribution for the negatives and $\pi \in [0,1]$ controls their proportion. Paradoxically, with this approach the distribution we know is that of the negatives while $\mathcal{P}$ is to be estimated from data.

This notion of arbitrarily generating negative data to enable the application of supervised techniques in unsupervised problems seems very naïve, but it is advocated by well respected statisticians (Hastie et al., 2001, pg. 449). Recent theoretical studies in one-class classification also provide justification for this approach. El-Yaniv and Nisenson study the decision aspect of one-class classification, when to accept a new example, in an hypothetical setting where $\mathcal{P}$ is fully known (El-Yaniv and Nisenson, 2006). Using a game-theoretic, "foiling the adversary" analysis, they conclude that the optimal strategy to deal with an unknown "attacking distribution" is to use randomization at the decision level (*i.e.*, incorporate a random element in the classifier outputs). They also justify the common heuristic of using the uniform for $\mathcal{A}$, when defining negatives as examples in low density areas of positives, as a worst-case attacking distribution in this scenario.

Estimating density level sets has been cast as supervised problems with contrasting examples sampled from a reference distribution (Scott and Nowak, 2006; Steinwart et al., 2005). Again, these are applied to one-class classification by defining negatives as examples in low density areas. Related heuristics have been used in one-class classification for tasks such as model selection by volume estimation (Tax and Duin, 2002). These use the volume as a proxy to estimate the error. Other fully supervised approaches for one-class classification by the generation of artificial negative samples and the use of supervised classifiers can also be found in the literature (Fan et al., 2004; Abe et al., 2006).

3.1 Non-parametric Artificial Negatives Generation

Actual negatives could live anywhere in the input space, thus the space of actual classification problems for a given set of positive data samples is very large. In high dimensional spaces, we can generate negatives anywhere and the generation method chosen will bias the resulting classifier. So the question is, what are appropriate principles to drive the generation process?

Ultimately we want to train a classifier that will be prepared for mischievous and adversarial attacking distributions of negatives. A principled way to do that is to try to generate negatives that resemble the positives - mimicking some aspects found in P - as that will create *hard* but *solvable* problems. By solvable we mean that there should be a way of discriminating P from A, while by hard we mean, for example, looking for boundary cases or for negative samples in which the correlations between the features present in the positives are kept. In layman terms, our motivation is to generate artificial negatives that look like the positives without being positives so that the discriminating dimensions that are chosen stress the real essence of the positives.

In fact, for the ANG based technique proposed in (Hempstalk et al., 2008) it is shown that an ideal solution is to generate negatives by sampling from the very same distribution of the positives. Parametric models fitted to the positives (*e.g.*, a Multivariate Gaussian) could be a useful ANG. However, in high-dimensional spaces fitting a parametric model seems futile. Therefore we turn our attention towards non-parametric and geometrically motivated ANG techniques. The following are four simple methods for generating artificial negatives:

Uniform. Negatives coming from the uniform distribution are commonly used in the literature. As indicated previously, the rationale for sampling the negatives from the uniform is that of low-density rejection. This method can perform poorly when the distribution of the actual negatives is far from uniform while still having a big overlap with $\mathcal{P}$(Scott and Blanchard, 2009), and it has important computational problems when trying to cover high dimensional spaces.

Marginal. Generating negatives by random sampling from the empirical marginal distribution of the positives, that is, to randomly permute the values within each feature, breaks the correlation between the features while maintaining the artificial negatives in dense areas of positives (Francois et al., 2007). Breiman and Cutler, in their random forest implemen-

tation (Breiman, 2001), apply this method to allow the construction of forests which, as a byproduct, produce an emergent measure of proximities between examples and a ranking of features (Shi and Horvath, 2006).

Left-right. This method simple translates each example in one of two directions, "left" or "right". The translation in each dimension depends on the observed range of that dimension and is scaled by a parameter $\rho \in \mathbb{R}$, chosen *a priori*. Formally $\mathbf{a}^{(i)} = \mathbf{p}^{(i)} + \rho^{(i)}\mathbf{r}$, where $\mathbf{r} = (r_1, r_2, \ldots, r_m)$ is the vector of features ranges ($r_k = |max(x_i) - min(x_i)|$)and $\rho^{(i)}$ is selected at random from $-\rho$ ("to the left") and ρ ("to the right "). See figure 2.

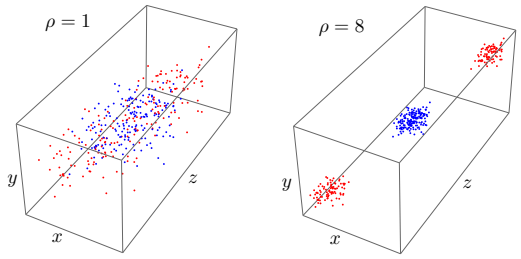

Figure 2: From the many directions possible, the LeftRight generator displaces each positive point to the left (translate each coordinate by a negative amount) or to the right (translate each coordinate by a positive amount). By choosing to translate in these two unique directions, we are generating two clouds of points that are arbitrarily far from the original sample of positives. Translation is an affine transformation and so all the distances ratios get preserved in each of the two clouds, so each cloud accounts for a different stochastic view of the neighborhoods present in the positives. This gives different related goals for LPP and also forces it to "collapse" the positives, as the graph W is made up of at least three connected components that arise from analogous clouds of points in the original Euclidean space. Our arbitrary choice to scale up the translation by the range in each dimension makes the distances between clouds larger in dimensions with high variance, in this case z.

Normalizer. This another simple transformation is based on normalization. It projects the positives onto the surface of the unit-L1 "sphere" to produce the negatives ($\mathbf{a}^{(i)} = \|\mathbf{x}^{(i)}\|_1^{-1}\mathbf{x}^{(i)}$) and then projects them again onto the surface of the unit-L2 sphere ($\mathbf{p}^{(i)} = \|\mathbf{x}^{(i)}\|_2^{-1}\mathbf{x}^{(i)}$) to produce the normalized positives. See figure 3.

4 RESULTS

In this section we study the behaviour of LPP applied over samples of positives enriched with the negatives

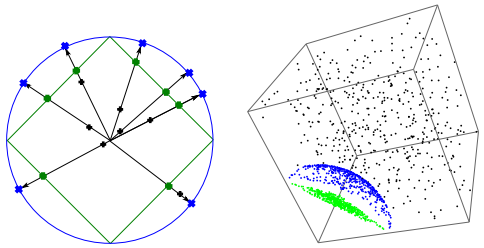

Figure 3: Effect of the normalizer generator in two and three dimensions. The *normalizer* ANG maps the positives (black) onto the unit-L1-sphere to produce the artificial negatives (internal simplex, green) and then maps them again into the unit-L2-sphere (external circle, blue) to generate the normalized positives. This transformation keeps in-class neighborhood relations and feature correlations between the two samples. It also generates two close clouds of points, as it is easy to show that the Euclidean distance between $\mathbf{p}^{(i)}$ and $\mathbf{a}^{(i)}$ is bounded by 1 and likely to be close to 1. In high dimensions this usually generates interesting contrasting distributions where the negatives are closer to negatives and the positives are also closer to the negatives than to other positives. It is difficult to illustrate this last effect in two or three dimensions, but it happens consistently, for example, in the experiments described in section 4.

generated in the four ways explained in the previous section. We do so over a suite of datasets, for which it is known that dimensionality reduction is possible and desirable, in two domains: text classification and functional genomics.

4.1 Experimental Setup

In order to avoid complex experimental setups, we will consider only reducing to one-dimension through a linear transformation, represented by $\mathbf{e}$. It is known that this kind of dramatic dimensionality reduction is possible for the text classification task (Kim et al., 2005). In this way we avoid the problem of selecting the optimal dimensionality and the threat of reporting overoptimistic results due to multiple testing effects.

For the ANG we set the proportion π to 0.5, generating the same number of artificial negatives as training positives. For the left-right ρ parameter we use 20 that always generates well separated clouds of points. As baseline for dimension reduction techniques we apply the standard unsupervised LPP (using 5 as the number of nearest neighbors), PCA and a Gaussian random projection. Apart from those we also project over the direction defined by the standard deviation of each feature, that is, $\mathbf{e} = (\sigma_1, \sigma_2, \ldots, \sigma_m)$ where σ_k is the standard deviation of feature k. The rationale for this last technique, that we call StdDevPr, will become clear when reading the experiments with text classification. We also report the results got when applying OCC without dimensionality reduction. The

following are the two one-class classifiers we use:

Gaussian Model (Tax, 2001). Fit a unimodal multivariate normal distribution to the positives. When applied to 1-dimensional data, this classifier simply returns the distance to the mean.

One-class SVM. We use the one-class ν-SVM (Schölkopf et al., 2001) method, that computes hypersurfaces enclosing (most of) the positive data. We set ν, the regularization parameter that controls how much we expect our training data to be contaminated with outliers, to 0.05. As it is common practice in OCC we use the Gaussian kernel, initializing the width of the kernel to the average pairwise Euclidean distance in the training set.

In order to select an operating point for the classifiers we compute a threshold by assuming that a 5% of the training data are outliers. This is a common choice in the one-class literature. The role of threshold selection by train-rejection lies in one or both of these two assumptions (a) the presence of noise and some counterexamples in the train data, (b) our classifier is not powerful enough as to accommodate all positive examples. Another underlying assumption is that in the training data we have boundary cases, so that the threshold will not be too tight as for rejecting too many positives. A more practical view is that, probably, this is the most straightforward way of selecting the operating point.

Threshold selection is directly related to the robustness and capital for one-class classifiers generalization capabilities. If it is too tight the number of false negatives will be increased; this can happen if the noise level specified by the user is too high. If it is too loose, the number of false positives will increase; this will happen if the noise level specified is too low. In either case one-class classifiers become reject-all or accept-all machines, which is a very common and undesirable effect.

For each target class we perform a 10-fold cross-validation, except for those classes with less than 10 examples, which we ignore, and those with sample sizes between 10 and 15, for which we perform a leave-one-out cross validation (in OCC this means constructing a model using all positives to classify all negatives, and constructing a model leaving out each of the positives). Of course, the ANG sampling and DR computations are also included in the cross-validation loop, only granting them access to the train data in each fold. We report the area under the ROC curve (AUC) and the Balanced Accuracy Rate (BAR) defined as the average of the True Positive (sensitivity) and True Negative (specificity) Rates.

4.2 Text Classification

We use a suite of text classification problems provided by Forman (Forman, 2003)[1]. Those come from several well-known text classification corpora (ohsumed, reuters, trec...). In total this accounts for 265 different classification tasks. These are high dimensional (from 2000 to 26832 features) low sample size datasets, therefore the data is sparse. We use the Bag-of-Words (BoW) representation that embodies a simplistic assumption of word independence, and normalize each document to unit-L2 norm, as is usual practice in information retrieval.

There is a fundamental trap when working with dimensionality reduction for text classification in OCC. Due to the sparsity, many of the words do not appear at all in any of the documents of the class. These words are *unobserved features*, features that are constant zero in the training set of a class. Unobserved features are highly discriminative, but cannot be used in a principled way for training one-class classifiers. This phenomenon is pervasive, with unobserved ratios per class ranging between 5% and 95% of the features in the datasets evaluated. Unobserved features can make a big difference in performance. For example, using the Gaussian classifier the average AUC varies from 0.9 when allowing unobserved features in the training set to 0.68 when using only observed features. In the present experiments we only use observed features.

The results are shown in figure 4. The baseline AUC for no dimension reduction is a poor 0.68. Neither PCA nor LPP provide useful projections when trained with positive examples only. They are even harmful performing worse than random projection, which also performs poorly in this evaluation. In the ANG realm we realize that both the Uniform and the Marginal, while still improving over the baseline of LPP, does not provide the best performance. Therefore we focus on the three best techniques: Normalizer and LeftRight + LPP and the StdDevPr.

The StdDevPr is the best technique in our test-bench. Its computation is extremely efficient ($O(mn)$), requiring only a single pass over the positive examples. To the best of our knowledge it is novel and have not been used before, although related biases can be found in the literature (*e.g.*, the term frequency variance, where in a feature selection context each word is scored by its variance in the whole cor-

[1] Available for download at http://jmlr.csail.mit.edu/papers/v3/forman03a.html. We used an extra dataset, new3s, also supplied by Forman and available at http://prdownloads.sourceforge.net/weka/19MclassText-Wc.zip?download

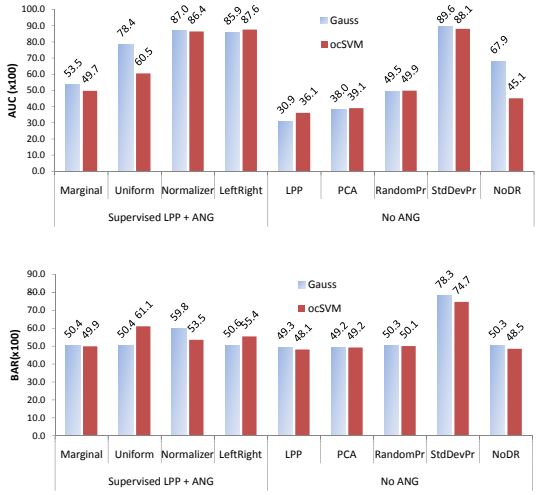

Figure 4: Cross-validation AUCs (top) and BARs (bottom) averaged over 265 datasets/ classes in text classification.

pus of positives and negatives (Dhillon et al., 2003)). It accounts for a very simple rationale: a dimension (word) is promoted inside a class when it is used a lot in several of the training documents (modelling phenomena such as word burstiness (Madsen et al., 2005)), always in relation to the size of these documents (recall that we work with normalized documents).

We came to consider using StdDevPr almost by accident and only after carefully analyzing the actual reasons behind the success of LeftRight and the Normalizer ANGs. We realized that the embeddings found by LPP using these two ANG techniques where highly correlated, so it became obvious that LPP was responding to the same characteristic of our positives in both cases. It was obvious too that because of the range-based scaling on the translation part of the LeftRight ANG, we were artificially stretching dimensions with high variance. These embeddings are also highly correlated with those found by StdDevPr, so the the success when applying these ANGs techniques is mainly attributed to their similarity to the StdDevPr technique.

In the bottom part of Figure 4 the performance of the simple threshold selection technique used is shown. It is clear that only the StdDevPr enhanced AUC is well used while both the ranking enhancements provided by LeftRight and Normalizer, in spite of having the same potential, are lost because of a poor threshold selection strategy. The target dimensionality (the dimensionality of the data after the application of the DR technique) can be regarded as a regularization parameter (Mosci et al., 2007). In classification, when fixing the thresholding policy, it controls the trade-off between sensitivity and specificity; overfitting and underfitting can be easily provoked by a wrong selection of the target dimensionality. Studying the interactions between the threshold and target dimension selection and the DR and classification techniques is essential, but lies beyond the aims of this paper.

4.3 Translation Initiation Site Prediction

We applied the same experimental setup to an important biological problem: recognizing translation initiation sites (TIS) in a genomic sequence. We used the dataset described in (Liu and Wong, 2003)[2]. It has 3312 positive examples and 10063 negatives. These examples have 927 features that represent counts (repetitions) of k-grams in the DNA sequence. In this case we do not normalize to unit-L2 norm, but instead normalize each feature to be in $[0, 1]$ in our training set. Therefore this time the LeftRight ANG will not promote high variance directions using the range as a proxy, since all ranges are the same.

The results are shown in figure 5. In these results we see two dominant techniques: using the original feature set (AUC = 0.82) and the LeftRight + LPP (AUC=0.92). That accounts for an increase of a 10% by reducing the dimensionality to 1. Surprisingly,as shown in the bottom part of the figure, by using our simple thresholding technique we get classification accuracies that are competitive with most of the results in the literature got by using supervised techniques (Liu and Wong, 2003).

We still don't have conclusive answers for why LeftRight works so well in this case. Our hypothesis is that our motivation when we designed the simple LeftRight ANG to collaborate with LPP in order to "collapse the class" works. Referring to the distances of the embedded points, LPP does a good job on getting them very close to zero in the training sets, and getting similar effects in the test sets. The Uniform generator has an analogous effect on the training sets but the embeddings are not so good at test time (as reflected by its performance in figure 4), which is probably due to LPP responding to specific stochastic interactions in the artificial uniform sample.

5 CONCLUSIONS

We have explored the feasibility of artificial negative generation techniques in the context of dimensional-

[2]Available for download at http://datam.i2r.a-star.edu.sg/datasets/krbd/SequenceData/TIS.html

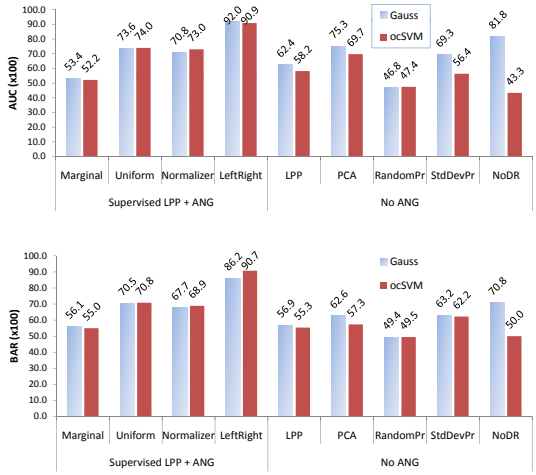

Figure 5: Cross-validation AUCs (top) and BARs (bottom) for the TIS dataset.

ity reduction for one-class classification. Applying very simple artificial negative generation techniques working together with a locality preserving dimension reduction has shown promising results in a experiment with a comprehensive set of text classification datasets and a genomics dataset. This area of research is by its very nature speculative, as ultimately one always needs to rely on the relations between the artificial sample and the actual negatives, the latter being unknown. It is also the case that for each ANG mechanism we can find the corresponding unsupervised bias. In the case of text classification we found via this indirect approach that stretching up the directions - words - which account for more variance within the class once the documents are normalized is a fast, reliable and class-dependent bias for dimension reduction in one-class classification. For the genomics dataset one of our proposed techniques excels at finding discriminative representations and all seems to indicate that this is due to our algorithm-design rationale working as expected.

This work can be extended by studying synergies between ANG and corresponding supervised techniques. For example, for text classification, applying Linear Discriminant Analysis together with parametric ANG techniques has shown consistent good performance. We are also exploring the potential to incorporate the other bit of information we have in OCC, the testing point, to guide the creation of our artificial negatives. Artificial negatives could also lead to data-driven techniques for other tasks in the classification system, like the threshold or target dimensionality selections.

REFERENCES

Abe, N., Zadrozny, B., and Langford, J. (2006). Outlier detection by active learning. In *KDD: International Conference on Knowledge Discovery and Data Mining*, pages 767–772.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Chapelle, O., Schölkopf, B., and Zien, A., editors (2006). *Semi-Supervised Learning*. The MIT Press, Cambridge, MA.

Chung, F. R. K. (1997). *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society.

Dhillon, I., Kogan, J., and Nicholas, C. (2003). Feature selection and document clustering. In *A Comprehensive Survey of Text Mining*, pages 73–100. Springer.

El-Yaniv, R. and Nisenson, M. (2006). Optimal single-class classification strategies. In *NIPS: Advances in Neural Information Processing Systems*.

Fan, W., Miller, M., Stolfo, S., Lee, W., and Chan, P. (2004). Using artificial anomalies to detect unknown and known network intrusions. *Knowledge and Information Systems*, 6(5):507–527.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305.

François, D. (2008). *High-dimensional Data Analysis: From Optimal Metrics to Feature Selection*. VDM Verlag.

Francois, D., Wertz, V., and Verleysen, M. (2007). The concentration of fractional distances. *IEEE Trans. on Knowl. and Data Eng.*, 19(7):873–886.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2001). *The Elements of Statistical Learning*. Springer.

He, X. and Niyogi, P. (2003). Locality preserving projections. In *NIPS: Advances in Neural Information Processing Systems*.

Hempstalk, K., Frank, E., and Witten, I. H. (2008). One-class classification by combining density and class probability estimation. In *ECML: European Conference of Machine Learning*.

Kim, H., Howland, P., and Park, H. (2005). Dimension reduction in text classification with support vector machines. *Journal of Machine Learning Research*, 6:37–53.

Liu, H. and Wong, L. (2003). Data mining tools for biological sequences. *Journal of Bioinformatics and Computational Biology*, 1(1):139–167.

Madsen, R. E., Kauchak, D., and Elkan, C. (2005). Modeling word burstiness using the dirichlet distribution. In *ICML: International Conference on Machine Learning*, pages 545–552.

Manevitz, L. M. and Yousef, M. (2001). One-class SVMs for document classification. *Journal of Machine Learning Research*, 2:139–154.

Mosci, S., Rosasco, L., and Verri, A. (2007). Dimensionality reduction and generalization. In *ICML: International Conference on Machine Learning*, pages 657–664.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J. C., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471.

Scott, C. and Blanchard, G. (2009). Novelty detection: Unlabeled data definitely help. In *AISTATS: Artificial Intelligence and Statistics, JMLR: W&CP 5*.

Scott, C. D. and Nowak, R. D. (2006). Learning minimum volume sets. *Journal of Machine Learning Research*, 7:665–704.

Shi, T. and Horvath, S. (2006). Unsupervised learning with random forest predictors. *Journal of Computational & Graphical Statistics*, 15:118–138.

Steinwart, I., Hush, D., and Scovel, C. (2005). A classification framework for anomaly detection. *Journal of Machine Learning Research*, 6:211–232.

Tax, D. M. J. (2001). *One-class classification. Concept learning in the absence of counterexamples*. PhD thesis, Delft University of Technology.

Tax, D. M. J. and Duin, R. P. W. (2002). Uniform object generation for optimizing one-class classifiers. *Journal of Machine Learning Research*, 2:155–173.

Tax, D. M. J. and Muller, K.-R. (2003). Feature extraction for one-class classification. In *ICANN/ICONIP: Joint International Conference on Artificial Neural Networks and Neural Information Processing*.

Villalba, S. D. and Cunningham, P. (2007). An evaluation of dimension reduction techniques for one-class classification. *Artificial Intelligence Review*, 27(4):273–294.