Fast Nonparametric Estimation of Class Proportions in the Positive-Unlabeled Classification Setting

Daniel Zeiberg, Shantanu Jain, Predrag Radivojac
Khoury College of Computer Sciences
Northeastern University, Boston, MA, U.S.A.

Abstract

Estimating class proportions has emerged as an important direction in positive-unlabeled learning. Well-estimated class priors are key to accurate approximation of posterior distributions and are necessary for the recovery of true classification performance. While significant progress has been made in the past decade, there remains a need for accurate strategies that scale to big data. Motivated by this need, we propose an intuitive and fast nonparametric algorithm to estimate class proportions. Unlike any of the previous methods, our algorithm uses a sampling strategy to repeatedly (1) draw an example from the set of positives, (2) record the minimum distance to any of the unlabeled examples, and (3) remove the nearest unlabeled example. We show that the point of sharp increase in the recorded distances corresponds to the desired proportion of positives in the unlabeled set and train a deep neural network to identify that point. Our distance-based algorithm is evaluated on forty datasets and compared to all currently available methods. We provide evidence that this new approach results in the most accurate performance and can be readily used on large datasets.

Introduction

Positive-unlabeled, or PU learning, has emerged as an active and important area of machine learning research (Denis 1998; Denis, Gilleron, and Letouzey 2005; du Plessis, Niu, and Sugiyama 2014; Hsieh, Natarajan, and Dhillon 2015; Chang et al. 2016). It generally refers to a binary classification setting where learning discriminators between positive and negative examples is based on the data that contain positively labeled examples and a set of unlabeled examples that contain an unknown mix of positives and negatives. The PU framework is well-suited to open world problems and is common in science and commerce. Typical applications are found in molecular biology, medicine, social networks, text mining, online advertising, etc. (Liu et al. 2003; Lee and Liu 2003; Yu, Han, and Chang 2004; Elkan and Noto 2008; Ward et al. 2009).

The focus of this work is on the positive-unlabeled setting and the problem of estimating fractions of positive and negative examples, or class priors, in unlabeled data. Such a task is often considered in sciences to understand the prevalence of natural, physical or social phenomena or in business to characterize the userbase and can therefore be an integral part of knowledge generation and decision support. However, class prior estimation also has instrumental value in machine learning research as a key component in the development and evaluation of classification models in the PU setting (Elkan and Noto 2008; Ward et al. 2009; Menon et al. 2015; Jain, White, and Radivojac 2016; 2017).

Following Elkan and Noto (2008), we will refer to the models developed to discriminate between positive and negative examples as *traditional classifiers* and the models that discriminate between positive and unlabeled examples as *nontraditional classifiers*. Surprisingly, an optimal nontraditional model is simultaneously an optimal traditional model for a broad class of loss functions and performance measures, including classification accuracy and the area under the ROC curve (Reid and Williamson 2010). However, the task of approximating posterior distributions is substantially more difficult and requires estimation of class priors in the set of unlabeled examples in order to transform a nontraditional classifier into a traditional classifier (Ward et al. 2009; Jain, White, and Radivojac 2016).

Another task requiring knowledge of class priors is accurate estimation of classification performance. Even in the cases where optimal traditional and nontraditional classifiers are equivalent, the true traditional performance can wildly differ from its nontraditional estimates (Jain, White, and Radivojac 2017; Ramola, Jain, and Radivojac 2019). It has been recently shown that many true performance measures can be recovered if one has well-estimated class priors (Menon et al. 2015; Jain, White, and Radivojac 2017; Ramola, Jain, and Radivojac 2019). Furthermore, certain performance metrics, such as balanced error rate and the Matthews correlation, permit optimal thresholding of the raw prediction scores based solely on nontraditional evaluation, whereas metrics such as error rate and F_1 score do not guarantee optimal performance (Ramola, Jain, and Radivojac 2019). The latter group of metrics, together with sensitivity, specificity, and precision, require class prior estimation for the thresholding task.

Nonparametric estimation of mixing proportions in the PU setting has been actively researched. Important directions include (non)identifiability results (Ward et al. 2009; Scott and Blanchard 2009; Blanchard, Lee, and Scott 2010;

Jain et al. 2016) and estimation algorithms (Elkan and Noto 2008; du Plessis and Sugiyama 2014; Sanderson and Scott 2014; Jain et al. 2016; Ramaswamy, Scott, and Tewari 2016; du Plessis, Niu, and Sugiyama 2017; Bekker and Davis 2018), as reviewed in the Background section. Although estimation algorithms have advanced over the past decade, there remain issues in both accuracy and scalability. To address these problems, we propose an intuitive nonparametric algorithm based on sampling that only requires finding nearest neighbors between positive and unlabeled examples. We evaluate all currently available algorithms on the largest set of datasets thus far, ranging from low-dimensional to high-dimensional. We provide evidence that the new algorithm performs very well against the best alternatives.

The remainder of this paper is structured as follows. We introduce all important concepts, provide problem specification and review related work in the Background section. We then present our algorithms in the Methodology section. We describe our evaluation strategy and provide results of all empirical evaluations in Experiments and Results. Finally, we offer closing remarks in the Conclusions section.

Background

We consider a binary classification problem of mapping an input space $\mathcal{X}$ to an output space $\mathcal{Y} = \{0, 1\}$. Let $p(x)$ be the true distribution of the inputs $x \in \mathcal{X}$, $p(x|y)$ be the class-conditional distribution, and $p(y)$ be the prior distribution for $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Given a set of positive examples from $p(x|y = 1)$ and a set of unlabeled examples from $p(x)$, the problem of estimating class priors $p(y)$ can be seen as estimating the mixing proportion in the following two-component mixture

$$f(x) = \alpha f_1(x) + (1 - \alpha) f_0(x), \tag{1}$$

where $\alpha = p(y = 1)$ is the mixing proportion we seek to estimate, $f(x) = p(x)$ and $f_y(x) = p(x|y)$.

Identifiability

Mixing proportion (α) estimation is an ill-posed problem due to unidentifiability (Blanchard, Lee, and Scott 2010; Jain et al. 2016). This means that there exist multiple values of α that lead to the same $f(x)$ for a given $f_1(x)$ because $f_0(x)$ itself can be a mixture containing $f_1(x)$. Furthermore, using $\mathcal{P}_\mathcal{X}$ to denote the set of all densities (except f_1) on $\mathcal{X}$, the set of all valid α values is an interval of the form $[0, \alpha^*]$, where

$$\alpha^* = a_f^{f_1} = \sup\{a : f = af_1 + (1 - a)h_0, h_0 \in \mathcal{P}_\mathcal{X}\}. \tag{2}$$

The value of 0 from $[0, \alpha^*]$ corresponds to the case where f_0 contains the same amount of f_1 as f does (i.e., there is no f_1 in f), whereas the value of α^* corresponds to the "irreducible" case where f_0 cannot be expressed as a mixture of f_1 and any other distribution from $\mathcal{P}_\mathcal{X}$. Therefore, the irreducibility assumption; that is, that f_0 is not a mixture containing f_1, makes the problem identifiable with α taking its largest value (Blanchard, Lee, and Scott 2010; Jain et al. 2016). More formally,

$$a_{f_0}^{f_1} = 0 \Rightarrow \alpha = \alpha^*. \tag{3}$$

Estimation

Several algorithms for nonparametric class-prior estimation have been proposed in the literature (Elkan and Noto 2008; Ward et al. 2009; du Plessis and Sugiyama 2014; Sanderson and Scott 2014; Jain et al. 2016; Ramaswamy, Scott, and Tewari 2016; du Plessis, Niu, and Sugiyama 2017; Bekker and Davis 2018).

The first algorithm was given by Elkan and Noto (2008) under the assumptions of non-overlapping supports between class-conditional distributions and availability of a nontraditional classifier that estimates posterior distributions. Their work, however, contains six different estimators that often give different values and it does not provide resolution as to which one to choose. du Plessis and Sugiyama (2014) showed that the e_1 estimator from Elkan and Noto (2008) is equivalent to a partial distribution matching formulation that minimizes Pearson divergence. They have subsequently generalized this approach to a broad family of f-divergences (du Plessis, Niu, and Sugiyama 2017).

Jain et al. (2016) derived a nonparametric class-prior estimation algorithm called AlphaMax. AlphaMax maximizes the likelihood of the positive-unlabeled data at multiple values of α in $(0, 1)$ using convex optimization. It then estimates α^* as the x-coordinate of the elbow on the maximum log-likelihood versus α curve. Although this approach can be implemented directly on multidimensional data using kernel density estimation, it can be computationally prohibitive and lead to sub-optimal performance when run on high-dimensional data. As a default option, its practical implementation first transforms the data into a single dimension using the α^*-preserving transform. We will review this transform in the next subsection because our new algorithm can be run both on the original as well as on transformed data.

Ramaswamy, Scott, and Tewari (2016) derive an objective function based on representing distributions by functions in a Reproducing Kernel Hilbert space (RKHS) on $\mathcal{X}$. For a given $\lambda \in [0, \infty)$, a $\mathcal{C}$-distance function is defined as the distance of $\lambda f + (1 - \lambda) f_1$ to its closest distribution in the RKHS. Theoretically, $\mathcal{C}$-distance is 0 if $\lambda f + (1 - \lambda) f_1$ represents a probability distribution. This is only true for an interval of λ values between 0 to some maximum λ^*, beyond which the distance is expected to increase. Practically, $\mathcal{C}$-distance is estimated by minimizing a quadratic form defined by a kernel matrix giving pairwise similarity between all the points (labeled and unlabeled) in the dataset and an $(|\mathbb{M}| + |\mathbb{C}|)$ dimensional weight vector. λ^* is detected from the distance curve via a thresholding (KM-1) or a gradient thresholding estimator (KM-2) and the class prior is estimated as $1 - 1/\lambda^*$. This method has a relatively high time complexity due to the computation of an $(|\mathbb{M}| + |\mathbb{C}|) \times (|\mathbb{M}| + |\mathbb{C}|)$ kernel matrix and the quadratic programming in $|\mathbb{M}| + |\mathbb{C}|$ variables, where $\mathbb{C}$ is the component sample and $\mathbb{M}$ is the mixture sample.

Bekker and Davis (2018) derive an estimation algorithm by first identifying feature subspaces with large proportions of labeled data. Practically, the subspaces are determined by the high purity nodes of a nontraditional decision tree trained between the labeled and unlabeled datasets. Because

such regions have fewer negative examples, the ratio of the labeled example to the total number of points gives a tight lower bound of the label frequency; i.e., the probability that a positive example is in the labeled set. The class prior is then estimated from the lower bound and the proportion of labeled examples.

Univariate Transforms

Mixing proportion estimation is often formulated as a density estimation problem, which may be problematic in high-dimensional spaces (Jain et al. 2016; Ramaswamy, Scott, and Tewari 2016). Fortunately, there exist α^*-preserving transforms that can be used to reduce the data to a single dimension, while still preserving α^* in the transformed space. Formally, for an α^*-preserving transform, $\tau : \mathcal{X} \to \mathbb{R}$, it holds that

$$a_{f_\tau}^{f_{\tau,1}} = a_f^{f_1}, \tag{4}$$

where f_τ and $f_{\tau,1}$ are density functions on $\mathbb{R}$ that are obtained as counterparts of f and f_1 after transforming the inputs using τ. It can be shown that a nontraditional classifier, a classifier trained to discriminate positive examples against the unlabeled examples treated as negatives, can serve as an α^*-preserving transform (Jain et al. 2016).

Methodology

Let $\mathbb{M} = \{m_i\}$ be a mixture (unlabeled) sample drawn from $f(x)$ and $\mathbb{C} = \{c_i\}$ be the component (positive) sample drawn from $f_1(x)$. The distance-based algorithm for class prior estimation takes the following two steps.

1. **Construction of the distance curve** involves repeatedly (1) sampling with replacement an example from $\mathbb{C}$, (2) recording the distance to its nearest neighbor from $\mathbb{M}$, and (3) removing the nearest neighbor from $\mathbb{M}$. Since $\mathbb{M}$ contains approximately $\alpha^*|\mathbb{M}|$ positives, distances recorded until removal of $\alpha^*|\mathbb{M}|$ examples are expected to be smaller compared to those recorded after (Figure 1).
2. **Estimating the class prior from the distance curve** involves training a prediction model on many simulated distance curves with known values of α^* and applying it to the distance curve obtained from the input samples.

Constructing the Distance Curve

To construct the distance curve, $|\mathbb{M}|$ number of examples are sampled from $\mathbb{C}$ with replacement. At each step k, an example c_k is sampled from $\mathbb{C}$ and its nearest neighbor from $\mathbb{M}$ is then removed from $\mathbb{M}$. The distance of the nearest neighbor is computed using some distance function, $\mathcal{D}$, on $\mathcal{X} \times \mathcal{X}$. The sampling can be repeated several times and the recorded distances are averaged for each of the $|\mathbb{M}|$ steps to produce the smoothed distance curve. More specifically, let d_k be the observed smoothed distance value at step k. The graph of d_k against $k/|\mathbb{M}|$ gives a distance curve on the $[0, 1]$ interval and is used in the next step (Figure 1).

Although $\mathcal{D}$ can be an arbitrary distance function, we will generally work with Minkowski distances (Deza and Deza 2013). We will additionally use univariate transforms discussed before and define $\mathcal{D}$ to be the distance on the one-dimensional space obtained by transforming the inputs using the scoring function, τ, of a nontraditional classifier trained to separate the labeled positives from the unlabeled data (Elkan and Noto 2008); i.e., $\mathcal{D}(x, y) = |\tau(x) - \tau(y)|$. The choice of τ as a transform is special since, unlike any arbitrary transform, it ensures that the class prior is preserved after the transformation. Finally, to exploit multiple transforms, we will also work with the Cityblock distance between vectors whose dimensions represent outputs of each univariate transform.

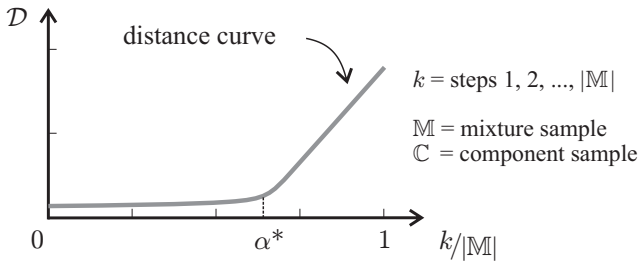

Figure 1: An idealized smoothed distance curve. The x-axis shows the $[0, 1]$ range for mixing proportions and the y-axis shows the expected distance between the k-th drawn positive example from $\mathbb{C}$ and its nearest unlabeled example from $\mathbb{M}$. The expected distance between a labeled and its closest unlabeled example slightly grows until α^* and then grows rapidly until 1.

We record the classification performance for each univariate transform by computing the area under the Receiver Operating Characteristic (ROC) curve (Fawcett 2006). Since the performance is calculated in the nontraditional sense we will refer to it as AUC^{pu}. When choosing the optimal transform, we generally select the classifier with the largest AUC^{pu}, as described in Experiments and Results.

A pseudo code to generate the distance curve is given in Algorithm 1. The curve was smoothed with $L = 10$ repetitions of the sampling process in all experiments.

Estimating the Class Prior from Distance Curves

A multi-layer feed-forward neural network is trained to predict the class priors using 100 features derived from the smoothed distance curve. To generate a fixed-length feature vector from the curve containing the number of points equal to the size of the unlabeled sample $\mathbb{M}$, we use the 100-quantiles of the distance values (y-coordinates of the curve). Precisely, if d_k denote the distances sorted in ascending order, then the i-th 100-quantile is given by $d_{\lceil i \cdot |\mathbb{M}|/100 \rceil}$, for $i = 1, \ldots, 99$ and d_1 for $i = 0$. These 100 values are subsequently normalized such that the total length of the vector is equal to 1.

To train the neural network, a rich set of distance curves is obtained from simulated positive-unlabeled datasets sampled using univariate parametric distributions at many different values of α. Specifically, the data is simulated using Beta(a_1, b_1) and Beta(a_0, b_0) as the positive and negative

Algorithm 1 DistCurve algorithm for class prior estimation.

Require: $\mathbb{M}$, $\mathbb{C}$, number of repetitions L
Ensure: α^*
// Transform the data as scores from a nontraditional classifier.
$[sMix, sComp] \leftarrow \texttt{transform}(\mathbb{M}, \mathbb{C})$
// Calculate AUC^{pu} using bootstrapped samples
// from $[sMix, sComp]$ if comparing multiple
// nontraditional classifiers.
$aucpu \leftarrow \texttt{AUC}(sMix, sComp)$
for $i = 1, \ldots, L$ **do**
  $sm \leftarrow sMix$
  for $j = 1, \ldots, \texttt{length}(sMix)$ **do**
    // Sample an example randomly from the
    // component sample with replacement.
    $c \leftarrow \texttt{sampleWithReplacement}(sComp)$
    // Get the closest example from the mixture sample.
    $m \leftarrow \texttt{nearestNeighbor}(c, sm)$
    // Remove m from the mixture sample.
    $sm \leftarrow \texttt{remove}(m, sm)$
    // Compute distance between c and m.
    $dist[i, j] \leftarrow \texttt{distance}(c, m)$
  end for
end for
// Take an average of the distances across multiple runs.
$d \leftarrow \texttt{columnAverage}(dist)$
// Calculate [0,100) 100-quantiles.
$q \leftarrow \texttt{quantiles100(d)}$
// Predict α^* by applying a neural network trained on
// simulated data.
$\alpha^* \leftarrow \texttt{applyTrainedNN}(q)$

component densities. A random sampling of

$$c_i \sim \text{Beta}(a_1, b_1)$$
$$m_i \sim \alpha\, \text{Beta}(a_1, b_1) + (1 - \alpha)\, \text{Beta}(a_0, b_0)$$

is used to generate positive and unlabeled samples $\mathbb{C}$ and $\mathbb{M}$, respectively. A total of 100,000 pairs of positive and negative densities are constructed to capture a diverse range of overlaps between the class-conditional densities. The density overlap is measured in terms of the normalized distance between functions defined in Yang et al. (2019). The range of the distance values, $[0, 1]$, is divided in 100 equal-width bins defined by bin boundaries $0, 0.01, \ldots 0.99, 1$. Density parameters (a_0, b_0, a_1, b_1) are repeatedly sampled as follows

$$a_0 \sim \text{Uniform}(2, 100)$$
$$b_0 \sim a_0 \cdot \text{Uniform}(1, 10)$$
$$a_1 \sim a_0 + a_0 \cdot \text{Beta}(0.5, 0.5)$$
$$b_1 \sim b_0 + b_0 \cdot \text{Beta}(0.5, 0.5)$$

and added to their respective bin, until each of the bins contained 1000 parameter sets. The extra parameters are thrown away. This ensures that different overlap values are uniformly represented in the dataset. For each of the 100,000 density pairs selected, 10 mixture distributions are constructed by picking 10 different values of α randomly from Uniform$(0.01, 0.99)$ giving a total of 1,000,000 positive-unlabeled datasets. The unlabeled and positive sample sizes are also picked randomly as $|\mathbb{M}| \sim \text{Uniform}(1000, 10000)$ and $|\mathbb{C}| \sim \text{Uniform}(100, 5000)$. We note that choices for the hyperparameters do not play a significant role as long as a broad selection of the underlying distributions is maintained.

The network was trained as a regression model. It contained 100 input nodes, three hidden layers, with sizes 2048, 1024, and 512, respectively, with each layer followed by a rectified linear unit activation layer, batch normalization layer and dropout layer with probability 0.5. The output was constrained to the range $[0, 1]$. The model was trained for 100 epochs using batch size 32, minimizing the mean absolute error (MAE) on the class prior prediction. Early stopping was used in the training process, monitoring the loss on 200,000 synthetic instances held out as the validation set.

Experiments and Results

Datasets

Two groups of real-world datasets were collected for experimental evaluation: (1) thirty multi-dimensional real-valued datasets, generally of low-to-medium dimension, and (2) ten high-dimensional text mining datasets each with tf-idf features. Most datasets were downloaded from the UCI Machine Learning Repository (Dua and Graff 2017), except for three text mining datasets that were found in the literature; *Inauthentic* (Dalkilic et al. 2006), *Webkb* (Cardoso-Cachopo 2007), and *LifeSci* (Yang et al. 2019). All datasets are listed in the Results subsection, alongside the estimated performance of different algorithms.

To create binary classification problems, we broadly followed the protocols from previous work (Jain et al. 2016; Ramaswamy, Scott, and Tewari 2016; du Plessis, Niu, and Sugiyama 2017; Bekker and Davis 2018). That is, categorical features were transformed into one-hot binary features, regression target values were transformed to class labels using the mean target value, and multi-class classification problems were binarized. Overall, these datasets contained confidently established positive and negative examples that allowed us to evaluate the performance of all algorithms by selecting a sample of positive examples as the positive (component) sample while the remaining examples comprised the unlabeled (mixture) set.

Experimental Protocol

Estimation experiments were run 50 times on each dataset with randomized selection of positive and unlabeled examples. More specifically, n_1 positive examples were randomly selected to form the component sample $\mathbb{C}$ and the remaining positive and negative examples constituted the mixture sample $\mathbb{M}$. The size of the mixture samples was further limited to n = 10,000, with positives and negatives proportionally reduced. The size of the component sample $\mathbb{C}$ was kept at n_1 = 1,000 except in smaller datasets where n_1 = 100 was set because the total number of positives was not sufficiently large.

The performance of each estimator was measured using the Mean Absolute Error (MAE) between the true class prior

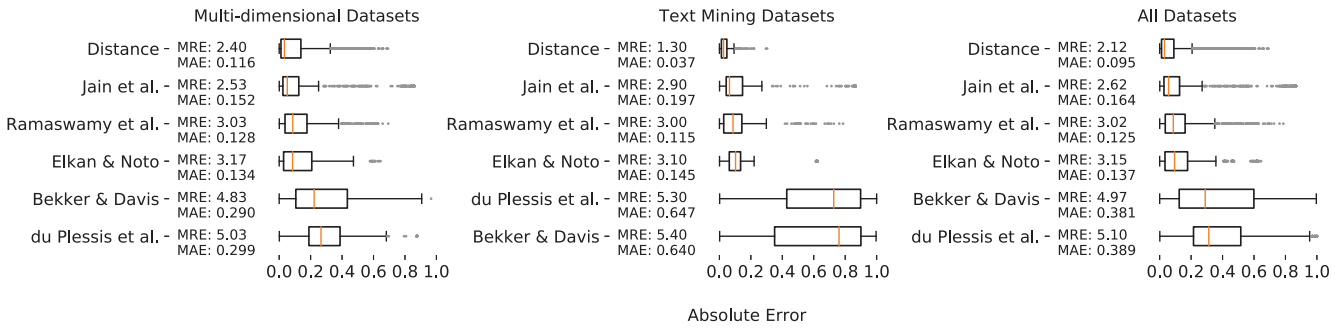

Figure 2: Distributions of absolute errors over different datasets, grouped into multi-dimensional real-valued data and text mining tf-idf data. Alongside each boxplot is the method's Mean Absolute Error (MAE) over all instances in the subset and Mean Rank Error (MRE) calculated using the mean absolute error on each dataset and then averaged over all datasets. The order of methods was selected according the the best ranking.

and the estimated class prior (Jain et al. 2016); i.e., the absolute errors for each dataset were averaged over the 50 random selections of the positive and unlabeled samples. Following Bekker and Davis (2018), we further used MAE to rank each algorithm's performance on each dataset and report its average rank amongst six competitors; Mean Rank Error (MRE). Finally, we measured the time it took each algorithm to complete its estimation on each dataset and report Mean Time (MT) and Mean Rank Time (MRT). All algorithms were run on identical datasets and computers with similar hardware, although we cannot guarantee that the available software implementations are equally optimized.

Selection of Univariate Transforms

Selection of the univariate transform for multi-dimensional datasets was carried out using the following nontraditional classification models: (1) a bagged ensemble of 100 two-layer feed-forward neural networks with $h \in \{1, 5, 25\}$ hidden units, (2) a bagged ensemble of 1000 regression trees, and (3) a polynomial kernel support vector machine (SVM) with a degree $d \in \{1, 2\}$, followed by the correction of Platt (1999). The data was z-score normalized prior to training (Tan, Steinbach, and Kumar 2006). In the cases of ensemble models, the transform was generated using an out-of-bag approach, whereas the SVM transforms were created using 10-fold cross-validation where the predictions on the 10 test sets were combined to ultimately comprise the univariate transform on the entire dataset. Because text mining datasets were sparse and high-dimensional, we only used linear SVMs as univariate transforms on these data without any normalization. As before, SVM scores were transformed to posterior probabilities using Platt's correction.

For each experiment in the multi-dimensional datasets, the optimal transform was selected by sequentially performing pairwise model comparisons based on the estimated AUC^{pu}. A more complex model was preferred only when its AUC^{pu} was higher with statistical significance ($P < 0.05$) than that of a previously selected simpler model, as estimated through 1000 bootstrap samples. Over all iterations, linear SVMs performed best in 43.9% of the cases, neural network ensembles with $h = 1$ in 3.8%, neural network ensembles with $h = 5$ in 25.9%, neural network ensembles with $h = 25$ in 6.6%, quadratic SVMs in 6.2%, and ensembles of regression trees in 13.6% of the cases.

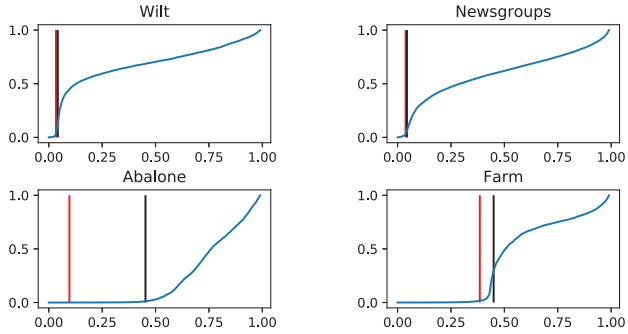

Figure 3: Four example distance curves with true mixing proportion marked in red and the model's predicted mixing proportion marked in black.

Distance Curves

Sample distance curves, illustrating the performance of our new algorithm are shown in Figure 3. The estimates are relatively straightforward on *Wilt* (MAE = 0.008, AUC^{pu} = 0.980) and *Newsgroups* (MAE = 0.007, AUC^{pu} = 0.875) datasets where the estimated and true class priors are very close. On the other hand, performance on the *Abalone* dataset (MAE = 0.380; AUC^{pu} = 0.688) was poor. Since other algorithms similarly overestimated the class prior here, we hypothesize that either the feature set is not discriminative enough to enable good classification accuracy or there may exist a considerable number of unknown positive examples among the negatively labeled examples. Finally, performance on the *Farm* dataset (MAE = 0.039, AUC^{pu} = 0.774) was moderate. Regardless of the performance, visual inspection can provide an additional layer of interpretation.

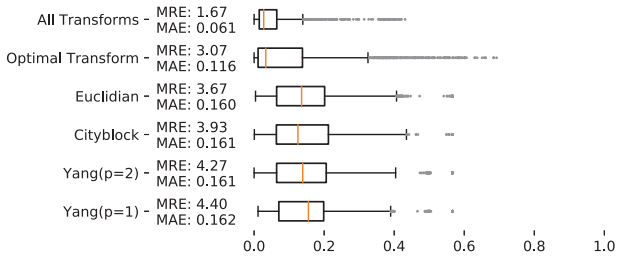

Figure 4: Distribution of absolute errors for transformed and nontransformed multi-dimensional data, together with mean rank error (MRE) and mean absolute error (MAE) for the distance-based algorithm.

Comparative Experiments

We compare the performance of our distance-based algorithm to five estimators available in the literature (Elkan and Noto 2008; Jain et al. 2016; Ramaswamy, Scott, and Tewari 2016; du Plessis, Niu, and Sugiyama 2017; Bekker and Davis 2018). We implemented the Elkan-Noto algorithm by following its description from the publication and used downloadable code for the remaining four procedures. The AlphaMax and Elkan & Noto estimators used the optimal model as described above. Both KM-1 and KM-2 kernels were used for the method of Ramaswamy, Scott, and Tewari (2016) and the one with the better performance was included in the comparisons. Finally, PE-DR (du Plessis, Niu, and Sugiyama 2017) and TIcE (Bekker and Davis 2018) algorithms were run with default parameters.

We note that Elkan and Noto (2008) and Bekker and Davis (2018) report the class prior that is the fraction of positive examples in the total dataset (including both positive and unlabeled examples). The remaining algorithms report the class prior as the fraction of positive examples in unlabeled data. The conversion has been made for the former two algorithms as $\alpha^{\text{new}} = ((n + n_1) \cdot \alpha - n_1)/n$, where n and n_1 are the dataset sizes for the unlabeled and labeled data, respectively.

Figure 2 and Table 2 summarize the performance accuracy of all algorithms. The box plots from Figure 2 show the distribution of absolute error over all 50 iterations of each dataset with summarized MAE and error rank averaged over all datasets. The results provide evidence that the new algorithm performs well on a broad class of problems. Table 2 gives MAE for each algorithm on each dataset. We also timed each experiment in order to evaluate the speed of evaluated estimators. The average ranking of each algorithm is reported in Table 1, together with the average ranking based on MAE.

Utility of Univariate Transforms

To understand the utility of univariate transforms, we evaluated our algorithm with and without transforms. The results "with transform" included two versions: (1) one-dimensional data using an optimal transform, selected based on the estimated performance of a nontraditional classifier, and (2) six-dimensional transform data using all evaluated transforms, where each example was mapped into a six-dimensional vector (one dimension per transform), from which the Euclidean distances were computed to obtain the distance curve. The results "without transform" included four distance metrics: (1) Minkowski distance with $p = 1$, or the cityblock distance, (2) Minkowski distance with $p = 2$, or the Euclidean distance, (3) normalized Yang's distance with $p = 1$, and (4) normalized Yang's distance with $p = 2$. Prior to computing distances on multi-dimensional datasets, all features were normalized using z-scores.

Table 1: Mean Ranking Error (MRE), Mean Absolute Error (MAE), Mean Ranking Time (MRT), and Mean Time (MT) measured for all estimators. MRE and MAE calculated on all datasets; MRT and MT (in seconds) calculated on all multi-dimensional datasets with a neural network nontraditional classifier ($h = 5$) used for Elkan-Noto, AlphaMax and Distance estimators. The main ranking is based on MRE.

Method	MRE	MAE	MRT	MT (*sec*)
Distance	1.7	.10	4.0	66.4
Jain et al.	2.8	.17	5.0	70.1
Ramaswamy et al.	3.0	.13	5.9	2818.1
Elkan & Noto	3.1	.15	3.0	65.6
Bekker & Davis	5.2	.38	1.0	1.1
du Plessis et al.	5.2	.39	2.1	28.3

Although the sampling algorithm on the original (untransformed) data led to excellent performance, univariate transforms are clearly beneficial. Combining the transforms provides additional improvements and suggests that there exist effective ensembling techniques that will result in superior performance when used on small and medium sized data. Interestingly, the evaluated distance functions show only minor variation in performance.

Conclusions

In this paper, we propose an intuitive and fast algorithm for estimating class priors from positive and unlabeled data. The algorithm is based on repeated sampling and nearest neighbor calculation to generate a distance curve, which is subsequently used as an input to approximate the class prior via a regression model. The obtained distance curve can also be used for visual inspection. Owing to its simplicity, this procedure is appealing as it can be used with any distance or kernel functions, including the ones learned from the data at hand; see, for example, Weinberger and Saul (2009) or Ting et al. (2016). The algorithm can also rely on one or more univariate transforms in order to exploit a rich set of classification models and overcome learning challenges such as high dimensionality, correlated features, irrelevant features, etc.

Comprehensive experiments on low-dimensional and high-dimensional data provide evidence that the new algorithm has excellent performance in a wide range of classification problems. It achieves this performance at a competitive speed and can therefore be run on big data. We also in-

Table 2: Mean Absolute Error (MAE) for each estimator on each dataset, along with dataset size (n) and dimension (d)

Dataset	n	d	Jain et al.	du Plessis et al.	Bekker & Davis	Ramaswamy et al.	Elkan & Noto	Distance
Multi-dimensional Datasets								
Abalone	4,177	8	0.504	0.509	0.672	0.423	0.316	0.380
Activity S_1	524,282	8	0.011	0.238	0.278	0.071	0.031	0.022
Activity S_2	22,646	8	0.857	0.024	0.090	0.002	0.011	0.073
Adult	48,842	119	0.114	0.407	0.120	0.096	0.263	0.062
Airfoil	1,503	5	0.117	0.347	0.456	0.209	0.444	0.086
Anuran	7,195	22	0.084	0.072	0.018	0.039	0.057	0.008
Bank	45,000	13	0.116	0.196	0.270	0.041	0.156	0.066
CASP	45,730	9	0.143	0.375	0.525	0.192	0.260	0.127
Concrete	1,030	8	0.088	0.303	0.418	0.156	0.337	0.061
Covertype	581,010	54	0.040	0.099	0.173	0.005	0.039	0.076
Epileptic	11,500	178	0.009	0.876	0.089	0.110	0.044	0.338
Gas	5,574	127	0.017	0.243	0.172	0.056	0.227	0.011
H1B	6,590	160	0.036	0.269	0.154	0.093	0.012	0.071
Housing	506	13	0.064	0.420	0.345	0.285	0.136	0.088
Landsat	6,435	36	0.008	0.119	0.907	0.013	0.056	0.062
Molbio	3,190	287	0.073	0.328	0.083	0.335	0.169	0.076
Mushroom	8,124	126	0.015	0.240	0.058	0.037	0.318	0.010
Pageblock	5,473	10	0.019	0.426	0.085	0.059	0.047	0.177
Parkinsons	5,875	20	0.060	0.220	0.096	0.158	0.109	0.028
Pendigit	10,992	16	0.021	0.216	0.350	0.042	0.036	0.003
Pima	768	8	0.143	0.321	0.441	0.208	0.149	0.077
Shuttle	58,000	9	0.031	0.163	0.510	0.010	0.015	0.035
Smartphone	10,929	561	0.027	0.236	0.117	0.015	0.045	0.002
Spambase	4,601	57	0.063	0.365	0.134	0.081	0.106	0.027
Thyroid	7,200	21	0.708	0.355	0.053	0.120	0.004	0.393
Transfusion	748	4	0.252	0.393	0.874	0.552	0.023	0.223
Waveform	5,000	21	0.057	0.285	0.270	0.090	0.095	0.438
Waveform (n)	5,000	40	0.055	0.344	0.224	0.103	0.090	0.390
Wilt	4,837	5	0.710	0.251	0.298	0.061	0.032	0.008
Wine	6,497	11	0.113	0.341	0.434	0.189	0.385	0.058
Text Mining Datasets								
BBC	2,225	9,635	0.042	0.575	0.837	0.130	0.090	0.031
Farm	4,143	54,877	0.057	0.235	0.561	0.155	0.134	0.039
Inauthentic	930	99,899	0.694	0.564	0.985	0.045	0.080	0.017
LifeSci	9,000	52,516	0.152	0.844	0.785	0.168	0.053	0.073
Movie	2,000	39,659	0.555	0.981	0.291	0.013	0.619	0.014
NIPS	5,811	11,463	0.136	0.769	0.827	0.132	0.106	0.068
Newsgroups	18,846	130,110	0.175	0.782	0.708	0.236	0.027	0.007
Reuters	2,065	8,943	0.044	0.310	0.380	0.099	0.134	0.033
TTC3600	3,600	5,692	0.033	0.668	0.207	0.069	0.098	0.047
Webkb	2,803	7,288	0.102	0.716	0.825	0.083	0.134	0.043

vestigated the utility of univariate transforms and show that the estimator is the most accurate when paired with a single or multiple carefully selected univariate transforms through training of nontraditional classifiers (Elkan and Noto 2008). Finally, we note that the size of the neural network for the identification of the elbow point on the distance curve did not play a significant role on the transformed data and so a reduction to a few dozen hidden neurons resulted in a very similar performance. We found, however, that larger networks had a significant impact when the algorithm was used on the original data with Minkowski's and Yang's distances, and we therefore opted to use the larger network as the final model.

Although future work remains to provide theoretical guarantees for our procedure, we believe that, overall, the new algorithm is simple to understand, implement and run. It thus provides an attractive tool in positive-unlabeled learning.

Acknowledgements

The last two authors should be regarded as joint senior authors. Funding: NSF grant DBI-1458477 (PR). Code Availability: github.ccs.neu.edu/dzeiberg/ClassPriorEstimation.

References

Bekker, J., and Davis, J. 2018. Estimating the class prior in positive and unlabeled data through decision tree induction. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, AAAI 2018, 2712–2719.

Blanchard, G.; Lee, G.; and Scott, C. 2010. Semi-supervised novelty detection. *J Mach Learn Res* 11:2973–3009.

Cardoso-Cachopo, A. 2007. Improving methods for single-label text categorization. *Ph.D. thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa*.

Chang, S.; Zhang, Y.; Tang, J.; Yin, D.; Chang, Y.; Hasegawa-Johnson, M. A.; and Huang, T. S. 2016. Positive-unlabeled learning in streaming networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 2016, 755–764.

Dalkilic, M. M.; Clark, W. T.; Costello, J. C.; and Radivojac, P. 2006. Using compression to identify classes of inauthentic papers. In *Proceedings of the 6th SIAM International Conference on Data Mining*, SDM 2006, 604–608.

Denis, F.; Gilleron, R.; and Letouzey, F. 2005. Learning from positive and unlabeled examples. *Theor Comput Sci* 348(16):70–83.

Denis, F. 1998. PAC learning from positive statistical queries. In *Proceedings of the 9th International Conference on Algorithmic Learning Theory*, ALT 1998, 112–126.

Deza, M. M., and Deza, E. 2013. *Encyclopedia of distances*. Springer.

du Plessis, M. C., and Sugiyama, M. 2014. Class prior estimation from positive and unlabeled data. *IEICE Trans Inf & Syst* E97-D(5):1358–1362.

du Plessis, M. C.; Niu, G.; and Sugiyama, M. 2014. Analysis of learning from positive and unlabeled data. In *Advances in Neural Information Processing Systems*, NIPS 2014, 703–711.

du Plessis, M. C.; Niu, G.; and Sugiyama, M. 2017. Class-prior estimation for learning from positive and unlabeled data. *Mach Learn* 106(4):463–492.

Dua, D., and Graff, C. 2017. UCI Machine Learning Repository. *University of California, Irvine, School of Information and Computer Science*. http://archive.ics.uci.edu/ml.

Elkan, C., and Noto, K. 2008. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 2008, 213–220.

Fawcett, T. 2006. An introduction to ROC analysis. *Pattern Recogn Lett* 27:861–874.

Hsieh, C. J.; Natarajan, N.; and Dhillon, I. S. 2015. PU learning for matrix completion. In *Proceedings of the 32nd International Conference on Machine Learning*, ICML 2015, 2445–2453.

Jain, S.; White, M.; Trosset, M. W.; and Radivojac, P. 2016. Nonparametric semi-supervised learning of class proportions. *arXiv preprint arXiv:1601.01944*.

Jain, S.; White, M.; and Radivojac, P. 2016. Estimating the class prior and posterior from noisy positives and unlabeled data. In *Advances in Neural Information Processing Systems*, NIPS 2016, 2693–2701.

Jain, S.; White, M.; and Radivojac, P. 2017. Recovering true classifier performance in positive-unlabeled learning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, AAAI 2017, 2066–2072.

Lee, W. S., and Liu, B. 2003. Learning with positive and unlabeled examples using weighted logistic regression. In *Proceedings of the 20th International Conference on Machine Learning*, ICML 2003, 448–455.

Liu, B.; Dai, Y.; Li, X.; Lee, W.; and Yu, P. S. 2003. Building text classifiers using positive and unlabeled examples. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, ICDM 2003, 179–186.

Menon, A. K.; van Rooyen, B.; Ong, C. S.; and Williamson, R. C. 2015. Learning from corrupted binary labels via class-probability estimation. In *Proceedings of the 32nd International Conference on Machine Learning*, ICML 2015, 125–134.

Platt, J. C. 1999. *Probabilistic outputs for support vector machines and comparison to regularized likelihood methods*. MIT Press. 61–74.

Ramaswamy, H. G.; Scott, C.; and Tewari, A. 2016. Mixture proportion estimation via kernel embedding of distributions. In *Proceedings of the 33rd International Conference on Machine Learning*, ICML 2016, 2996–3004.

Ramola, R.; Jain, S.; and Radivojac, P. 2019. Estimating classification accuracy in positive-unlabeled learning: characterization and correction strategies. *Pac Symp Biocomput* 24:124–135.

Reid, M. D., and Williamson, R. C. 2010. Composite binary losses. *J Mach Learn Res* 11:2387–2422.

Sanderson, T., and Scott, C. 2014. Class proportion estimation with application to multiclass anomaly rejection. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, AISTATS 2014, 850–858.

Scott, C., and Blanchard, G. 2009. Novelty detection: unlabeled data definitely help. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, AISTATS 2009, 464–471.

Tan, P. N.; Steinbach, M.; and Kumar, V. 2006. *Introduction to data mining*. Pearson.

Ting, K. M.; Zhu, Y.; Carman, M.; Y., Z.; and Zhou, Z. H. 2016. Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 2016, 1205–1214.

Ward, G.; Hastie, T.; Barry, S.; Elith, J.; and Leathwick, J. 2009. Presence-only data and the EM algorithm. *Biometrics* 65(2):554–563.

Weinberger, K. Q., and Saul, L. K. 2009. Distance metric learning for large margin nearest neighbor classification. *J Mach Learn Res* 10:207–244.

Yang, R.; Jiang, Y.; Mathews, S.; Housworth, E. A.; Hahn, M. W.; and Radivojac, P. 2019. A new class of metrics for learning on real-valued and structured data. *Data Min Knowl Disc* 33(4):995–1016.

Yu, H.; Han, J.; and Chang, K. C. C. 2004. PEBL: web page classification without negative examples. *IEEE Trans Knowl Data Eng* 16(1):70–81.