ICIC Express Letters
Part B: Applications
Volume **8**, Number **9**, September **2017**

ICIC International ©2017 ISSN 2185-2766
pp. 1287–1297

PAnDE: AVERAGED n-DEPENDENCE ESTIMATORS FOR POSITIVE UNLABELED LEARNING

Fuyi Li[1,2,3], Jiangning Song[1,2,4,*], Chen Li[1,2], Tatsuya Akutsu[5]
and Yang Zhang[3,*]

[1]Department of Biochemistry and Molecular Biology
[2]Biomedicine Discovery Institute
[4]Monash Centre for Data Science
Faculty of Information Technology
Monash University
Melbourne, VIC 3800, Australia
{ Fuyi.Li1; Chen.Li }@monash.edu; *Corresponding author: Jiangning.Song@monash.edu

[3]College of Information Engineering
Northwest A&F University
No. 22, Xinong Road, Yangling 712100, P. R. China
LiFuyi@nwsuaf.edu.cn; *Corresponding author: ZhangYang@nwsuaf.edu.cn

[5]Bioinformatics Center
Institute for Chemical Research
Kyoto University
Gokasho, Uji, Kyoto 611-0011, Japan
takutsu@kuicr.kyoto-u.ac.jp

Received April 2017; accepted June 2017

Abstract. *Traditional data mining algorithms are commonly based on fully labeled data, which is often practically difficult to obtain. In recent years, positive unlabeled (PU) learning has emerged as a useful technique to address this issue, which allows algorithms to learn from only positive and unlabeled data by relaxing the requirement for obtaining fully labeled data. Existing PU learning algorithms based on Bayesian classifiers, including PNB and PAODE, have been successfully applied to multiple classification problems. However, their empirical performance is affected by the attribute independence assumption. With the goal of effectively tackling positive unlabeled learning tasks with higher-level attribute dependence, we propose a novel PU learning algorithm in this study, termed PAnDE, which extends the AnDE (Averaged n-Dependence Estimators) algorithm based on the 'selected completely at random' assumption. We performed benchmarking tests to compare the performance of PAnDE with PNB (based on Naive Bayes) and PAODE (based on the Averaged One-Dependence Estimators) on 20 UCI datasets and three other real-world (human protein glycosylation) datasets. The results demonstrate that PAnDE outperformed PNB and PAODE, highlighting the predictive power of PAnDE and its scalability in a range of real-world applications.*

Keywords: Positive unlabeled learning, Averaged n-dependence estimators, Bayesian classification, PAODE, PNB

1. **Introduction.** In many real-world applications of data mining, it is often difficult to obtain fully labeled data, as labeling data is time-consuming and labor-intensive. The positive unlabeled (PU) learning scheme is a useful strategy to address this problem by learning from only positive and unlabeled data, which is relatively easy to obtain in real-world applications. This learning scheme has a capacity to significantly save human efforts of labeling samples and achieve competitive performance compared with supervised learning algorithms [1,2]. In view of the merits of PU learning and its comparable performance,

it is desirable to evolve traditional supervised algorithms to PU learning algorithms in order to efficiently learn from only positive and unlabeled data.

The Naive Bayesian (NB) classifier has been widely applied in many classification tasks. It assumes that all attributes are independent of each other within each given class label, which is known as the attribute independence assumption. This assumption makes NB simple and easy to implement, but at the same time sacrifices the classification accuracy [3]. Furthermore, this assumption is often violated in practice, since attributes are commonly dependent on each other in many real-world applications. To tackle this, AODE [4] was proposed based on a relaxed attribute independence assumption, by isolating one parent-attribute from other assumed independent attributes within the given class label. AnDE, an updated version of AODE, generalizes AODE to higher-level dependence to further relax the attribute independence assumption [5]. In AnDE, the observed frequencies of each combination of the $(n + 1)$ attribute values and class labels are used to form an $(n+2)$-dimensional probability model. Different from the majority of other algorithms that seek to build a model to directly estimate a high-dimensional probability, AnDE learns a high-dimensional probability by extrapolating from lower-dimensional probabilities. Experimental studies have shown that AnDE could achieve improved classification performance compared with other state-of-the-art algorithms [5].

To date, a number of Bayesian classifiers have been modified to perform PU learning, including PNB [14] and PAODE [18]. However, the performance of these classifiers is affected by the attribute independence assumption to various extents. To facilitate learning from positive and unlabeled data with a relaxed attribute independence assumption, we propose a novel algorithm based on AnDE in this study, termed as PAnDE ($\underline{\text{P}}$ositive $\underline{\text{A}}$veraged $\underline{n}$-$\underline{\text{D}}$ependence $\underline{\text{E}}$stimators). We performed empirical studies to compare the performance of PAnDE, PNB and PAODE. The results demonstrate that PAnDE achieves a better performance compared with the other two algorithms on a test suit of 20 UCI datasets and 3 human protein glycosylation datasets, highlighting the predictive power of PAnDE in real-world applications.

The rest of this paper is organized as follows. Section 2 reviews related work on PU learning based on different strategies. The proposed algorithm, PAnDE, is discussed in detail in Section 3. Prediction performance of PAnDE, PNB and PAODE is evaluated and discussed in Section 4 followed by conclusions and future work in Section 5.

2. **Related Work.**

A number of PU learning algorithms have been published in the last decade. Existing PU learning algorithms can be divided into two main categories according to the strategies employed [6,7]. The first category is referred to as the two-step strategy. Its main idea is to first identify reliable negative examples from the unlabeled dataset. Then the positive examples and the identified reliable negative examples are used to train a classifier. Representative algorithms of this category include M-C [8], S-EM [9], PUDI (PU learning for disease gene identification) [10,11] and LELC (PU learning by extracting likely positive and negative micro-clusters) [12]. The M-C algorithm was proposed for document classification and its classification. The S-EM algorithm was presented using the Spy extraction technique to identify reliable negative examples [8]. PUDI was developed to identify disease-associated genes from human genome data [9,10]. LELC [11] was further developed to cope with the task of data stream classification. The Spy [8] and Rocchio [8] extraction techniques were used to identify the two datasets including the likely positive set LP and the likely negative set LN.

The second category focuses on devising traditional classification models (e.g., decision tree and Naive Bayesian classifier) in order to enable them to learn from positive and unlabeled data directly. For example, POSC4.5 [13] was proposed based on the standard C4.5 decision tree algorithm. PNB was devised based on the multinomial model of NB for text classification [14]. This algorithm requires users to input the prior probability of the positive class (p) and uses this input to estimate the probability of each attribute for each class. A PU learning method was proposed by applying Biased Support Vector Machine (BSVM) to detecting electric heat pumps from coarse-grained smart meter data, and was shown to achieve better performance than the previous algorithms [15]. More recently, a PU learning method based on '*selected completely at random*' assumption and SVM was proposed for protein complex prediction of the protein-protein interaction networks [16]. Another PU learning algorithm, termed puNet [17], was proposed for networked text data classification based on NMF (Non-negative Matrix Factorization, an unsupervised learning algorithm).

The PU learning has also been achieved using ensemble learning. For example, a method was proposed by combining Bagging and BSVM [18]. A subset of unlabeled examples was selected randomly from the unlabeled dataset in each bagging iteration, and then BSVM was used to train the classifier. PEBL [19] built a set of classifiers by iteratively applying the SVM and using weighted votes of all classifiers generated in the iteration steps to construct the final classifier. A bagging method called ProDiGe, was proposed to perform disease-associated gene prediction [20].

In this paper, we focus on solving the PU learning task with Bayesian algorithms to handle general classification tasks using the second strategy. In [21], PNB was extended to address general classification tasks, which did not require users to provide the prior probability of a positive class. In addition, [21] used the '*selected completely at random*' assumption [1] to create PAODE by extending AODE for PU learning.

3. Positive Averaged *n*-Dependence Estimators (PA*n*DE).

In this section, we firstly describe the classification problem, followed by a brief introduction of AnDE. Then we describe the proposed PAnDE algorithm based on the '*selected completely at random*' assumption [1].

3.1. Problem description.

Let T be a given training set ($T = P \cup U$), where P is the set of positive examples and U denotes the set of unlabeled examples. An example can be represented by $< \mathbf{x}, y, l >$, where $\mathbf{x} =< x_1, \ldots, x_k >$ denotes the attribute vector with k attributes. The class variable is defined as y, $y \in \{0, 1\}$. $y = 1$ represents positive examples, while $y = 0$ indicates negative examples. The label variable is defined as l, $l \in \{0, 1\}$, $l = 1$ for labeled examples and $l = 0$ for unlabeled examples [1], respectively.

We adopt the '*selected completely at random'* assumption proposed in the PU learning model in [1]. It assumes that the labeled positive examples are selected completely at random from all positive examples. Based on this assumption, and an assumption about the proportion of positive examples that are accordingly selected, we can estimate the prior probability of the positive class, p, from positive and unlabeled examples.

3.2. A*n*DE.

The main task of Bayesian classifiers is to estimate the conditional probability for a given example. According to the definition of conditional probability, for an example, $\mathbf{x}$, its probability can be estimated as:

$$P(y|\mathbf{x}) = P(\mathbf{x}, y)/P(\mathbf{x}) \tag{1}$$

where $P(\mathbf{x}) = P(\mathbf{x}, y = 1) + P(\mathbf{x}, y = 0)$. Therefore, in order to estimate $P(y|\mathbf{x})$, the major problem considered in this paper is to estimate $P(\mathbf{x}, y)$. For brevity, we denote $P(\mathbf{x}, y = 1)$ and $P(\mathbf{x}, y = 0)$ as $P(\mathbf{x}, 1)$ and $P(\mathbf{x}, 0)$, respectively.

According to the definition of conditional probability, we have:

$$P(\mathbf{x}, y) = P(y)P(\mathbf{x}|y) \tag{2}$$

NB is based on the attribute independence assumption, which assumes that all attributes are independent of each other given the class label. Thus, $P(\mathbf{x}|y)$ is extrapolated from each two-dimensional probability estimate $P(x_i|y)$:

$$P_{\mathrm{NB}}(\mathbf{x}, y) = P(y) \prod_{i=1}^{k} P(x_i|y) \tag{3}$$

AODE relaxes the attribute independence assumption by averaging over all Bayesian Network Classifiers in which one attribute is selected as a super-parent and all other attributes are considered conditionally independent given this super-parent and the class:

$$P_{\mathrm{AODE}}(\mathbf{x}, y) = \sum_{\alpha=1}^{k} P(y, x_\alpha) P(\mathbf{x}|y, x_\alpha)/k \tag{4}$$

Furthermore, AnDE extends AODE by further relaxing the attribute independence assumption. Each sub-model selects n parent-attributes and assumes that all other attributes are independent for the given class label. For brevity, we define: $x_{\{i,j,\dots,q\}} =< x_i, x_j, \dots, x_q >$.

AnDE seeks to use:

$$P_{\mathrm{A}n\mathrm{DE}}(\mathbf{x}, y) = \sum_{S \in \binom{A}{n}} P(y, x_S) P(\mathbf{x}|y, x_S) / \binom{k}{n} \tag{5}$$

In Equation (5), $\binom{A}{n}$ denotes the set of all size-n subsets of the attribute set $A = \{1, \dots, k\}$, and x_S denotes a tuple of parent-attributes with n attributes.

3.3. **Averaged n-dependence estimators for PU learning.** Owing to the higher-level dependence, AnDE is able to achieve a better classification performance than NB and AODE when there are sufficient data to prevent overfitting. Hence, we present PAnDE to extend AnDE for the PU learning scenario and improve the classification performance of PU Bayesian learning. The classification algorithm of PAnDE is described as follows:

$$P_{\mathrm{PA}n\mathrm{DE}}(\mathbf{x}) = \begin{cases} \arg\max\limits_{y} \sum\limits_{S \in \binom{A}{n}} \delta(x_S) P(y, x_S) \prod\limits_{m=1}^{n} P(x_{ml}|y, x_S) : & \sum\limits_{S \in \binom{A}{n}} \delta(x_S) > 0 \\ f_{\mathrm{PA}(n-1)\mathrm{DE}}(\mathbf{x}) : & \text{otherwise} \end{cases} \tag{6}$$

where x_{ml} denotes the l-th value of the attribute x_m. We use function $\delta(x_S)$ to avoid using parent-attributes whose values do not occur in the training data [5], $\delta(x_S) = 1$ if x_S occurs in the training dataset; otherwise, $\delta(x_S) = 0$. Note that PAnDE becomes PA$(n-1)$DE when $\delta(x_S) = 0$ [5].

To solve Equation (6), we need to estimate the probability of $P(1, x_S)$, $P(0, x_S)$, $P(x_{ml}|0, x_S)$ and $P(x_{ml}|1, x_S)$ from the training dataset. In the PU learning scenario, $P(1, x_S)$ and $P(0, x_S)$ can be estimated by Equations (7) and (8) respectively:

$$P(1, x_S) = P(x_S|1)p \tag{7}$$

$$P(0, x_S) = P(x_S|0)(1 - p) \tag{8}$$

where p is the prior probability of the positive class mentioned above, which can be estimated using the following method proposed in [1]:

$$p = P(y = 1) = \frac{P(l = 1, y = 1)}{P(l = 1|y = 1)} \tag{9}$$

where l is the label variable and y is the class variable.

$P(x_{ml}|1, x_S)$ can be estimated using a maximum likelihood estimator with the Laplace correction on the positive set [21]. $P(x_{ml}|0, x_S)$ can be estimated by:

$$P(x_{ml}|0, x_S) = \frac{P(x_{ml}, x_S|0)}{P(x_S|0)} \tag{10}$$

$P(x_{ml}, x_S|0)$ can be estimated using the method described in [22]:

$$P(x_{ml}, x_S|0) = \frac{1 + \max\left(0; N_{S,ml} - P(x_{ml}, x_S|1)p|U|\right) \frac{1}{Z_{S,m}}}{r_S r_m + (1-p)|U|} \tag{11}$$

where $N_{S,ml}$ denotes the number of unlabeled examples for the combination of x_{ml} and x_S, r_m denotes the number of attribute values of x_m, while r_S denotes the product of the number of each parent-attribute x_S value. $Z_{S,m}$ can be calculated using the following equation proposed in [22]:

$$Z_{S,m} = \sum_{S \in \binom{A}{n}} \sum_{l=1}^{r_m} \frac{\max\left(0; N_{S,ml} - P(x_{ml}, x_S|1)p|U|\right)}{(1-p)|U|} \tag{12}$$

4. Empirical Study.

We compared the classification performance of PAnDE with PAODE and PNB on 20 UCI[1] datasets and 3 real-world (human protein glycosylation) datasets. In each group of experiments, 10 trials of experiments were conducted, and the averaged performance results of the trials were reported. Here we used the reformulated PNB proposed in [21], which has a similar performance with PNB proposed in [14] and does not require users to provide p, and the same approach is also used by PAODE and PAnDE to release users from providing parameter p. Considering that the computational requirements of A3DE defeat the WEKA[2] implementation [5], we thus used PA2DE throughout the experiments in this paper.

We assume that the number of positive examples affects the performance of the algorithm in the PU learning scenario. When the positive set is relatively small compared to the negative set, classifiers trained with these sets would usually perform poorly on the positive class. Therefore, we used the F_1 score, which is extensively used in the PU learning scenario [7,9,21], as a measure to evaluate the performance of different classifiers.

All the experiments were conducted on a PC with quad-core CPU and 2.0 GB memory. All the algorithms were implemented using the WEKA software package.

4.1. Datasets.

A total of 20 UCI datasets were used for conducting the performance evaluation experiments. Table 1 provides a summary of these datasets. In particular, for datasets that contain more than two classes, we defined one of these classes as the positive class, and combined the remaining as the negative class.

For each dataset, in the PU learning scenario, firstly, we randomly selected 50% of the examples as the training set and 20% of the examples as the validation set (for estimation of p), respectively. Then we combined the training and validation sets to re-train the classifiers. Finally, the remaining 30% of the examples were used to test the trained classifiers.

4.2. Experiment with the parameter *UnLevel*.

We defined the parameter *UnLevel* as the proportion of unlabeled examples in the training set, and simulated PU learning scenario with different *UnLevel* values. We experimented with different *UnLevel* values ranging from 40% to 80% with a step size of 10% and then compared the performance of PA2DE with PAODE and PNB.

Figures 1 and 2 show the performance comparison results of PA2DE versus PNB, and PA2DE versus PAODE, respectively. The x-axis represents the F_1 score of PNB (Figure 1) or PAODE (Figure 2), and the y-axis indicates the F_1 score of PA2DE. In each figure, there are totally 20 dots representing the experimental results on the 20 datasets. If a dot is located within the triangle area above the diagonal line, it means that PA2DE performs better on the corresponding dataset; or if it is located within the triangle area below the

[1] http://archive.ics.uci.edu/ml/

[2] http://www.cs.waikato.ac.nz/ml/weka/

TABLE 1. A summary of experimental datasets used to benchmark the performance of different algorithms

Dataset	Size	#Attributes	#Class	Pos Class	Pos/Neg
Adult	48842	14	2	<= 50K	76.1%/23.9%
Balance-scale	625	4	3	L	36.5%/63.5%
Blood	748	4	2	0	76.2%/23.8%
Breast-cancer	286	9	2	no-recurrence-events	70.3%/29.7%
Car evaluation	1728	6	4	unacc	70.0%/30.0%
Colic	368	22	2	no	37.0%/63.0%
Contraceptive Method Choice	1473	9	3	2,3	57.3%/42.7%
Credit Approval	690	15	2	+	44.5%/55.5%
German	1000	20	2	1	70.0%/30.0%
Heart Disease	294	13	2	< 50	63.9%/36.1%
Ionosphere	351	34	2	g	64.1%/35.9%
Iris Classification	150	4	3	Iris-versicolor	33.3%/66.7%
King-rook-vs-king-pawn	3196	36	2	won	52.2%/47.8%
Liver Disorders	345	6	2	1	42.0%/58.0%
Mushroom	8124	22	2	e	51.8%/48.2%
Nursery	12960	8	5	priority	32.9%/67.1%
Primary-tumor	339	27	21	lung	24.8%/75.2%
Tic-tac-toe	958	9	2	positive	65.3%/34.7%
Vote	435	16	2	democrat	61.4%/38.6%
Wine	178	13	7	1	33.1%/66.9%

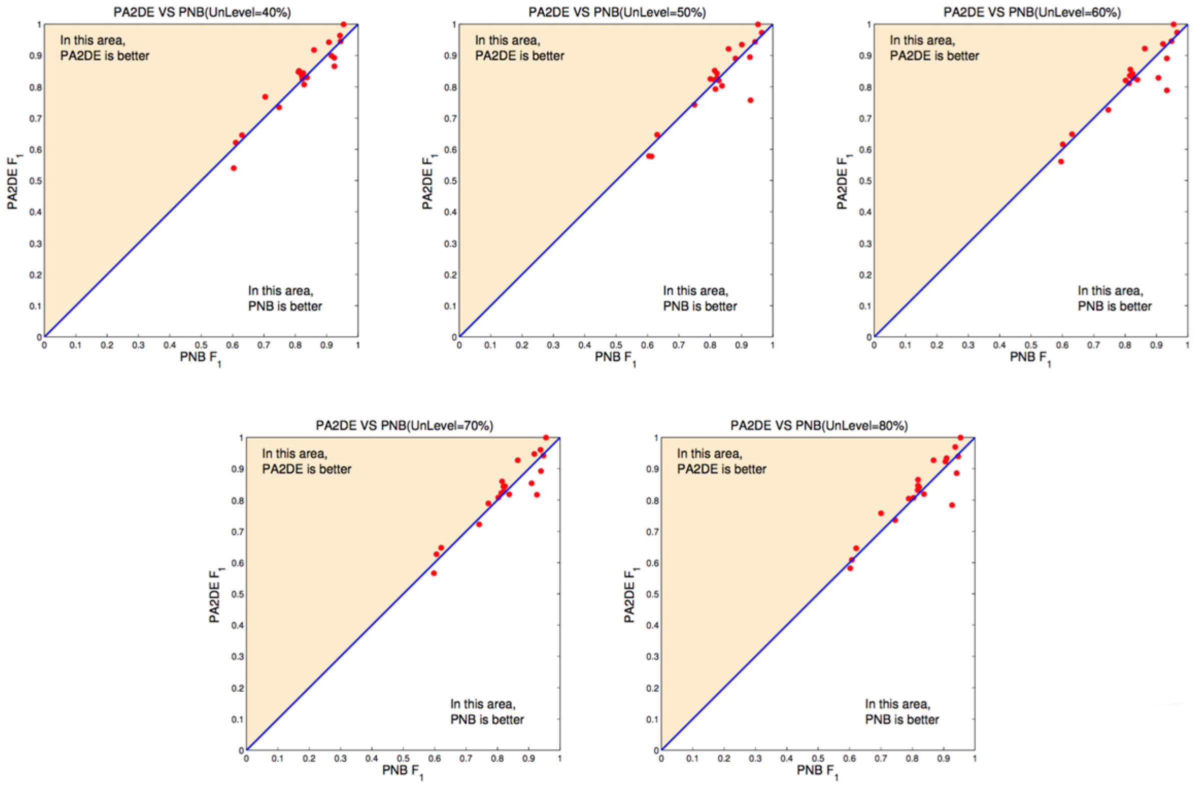

FIGURE 1. Performance comparison between PA2DE and PNB with different *UnLevel* values

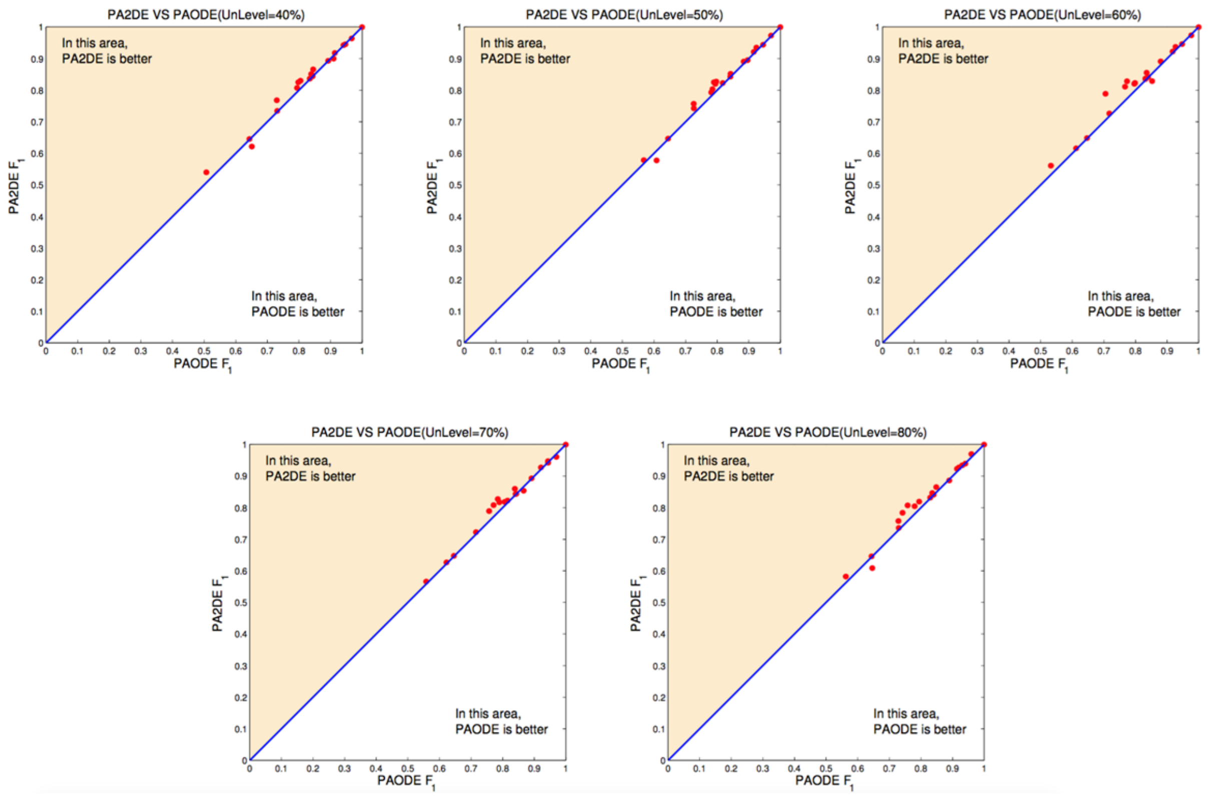

FIGURE 2. Performance comparison between PA2DE and PAODE with different *UnLevel* values

diagonal line, it means PNB (Figure 1) or PAODE (Figure 2) performs better; otherwise (i.e., located on the diagonal line), it means that the classification performance of PA2DE and PNB (or PAODE) is similar or close to each other.

From Figure 2, it is apparent that PA2DE outperformed PAODE on the majority of datasets. However, the dots were mostly located close to the diagonal line, suggesting that the performance improvement of PA2DE was not substantial. Meanwhile, we found that PA2DE achieved a better classification performance on 14 datasets (Figure 1).

4.3. **Experiment with the parameter $\boldsymbol{a}$.** According to the '*select completely at random*' assumption, positive examples are randomly labeled as positives with a constant probability $1 - \boldsymbol{a}$ in the positive dataset [1] and labeled as unlabeled with the probability $\boldsymbol{a}$. In this paper, we experimented with $\boldsymbol{a} = 0.2$, 0.4, 0.6. The experimental results are shown in Figures 3 and 4 for PA2DE versus PAODE and PA2DE versus PNB, respectively.

From Figure 3 and Figure 4, we notice that PA2DE generally performed better than PAODE and PNB with different $\boldsymbol{a}$ values. In particular, PA2DE outperformed PAODE on 19 datasets, while PA2DE outperformed PNB on 18 datasets when $\boldsymbol{a} = 0.4$.

Figures 1, 2, 3 and 4 also demonstrate that the classification performance of PA2DE was considerably improved compared with that of PAODE and PNB. For example, when $\boldsymbol{a} = 0.2$ on the Colic dataset, PA2DE achieved a much better performance improvement compared with PAODE (0.8235 vs. 0.6353). In addition, when $\boldsymbol{a} = 0.4$, the performance improvement of PA2DE was also significant (0.7993 vs. 0.5001) compared with PAODE.

4.4. **Evaluation on time and space.** In this section, we evaluated the time and space usage for building PA2DE, PAODE and PNB models. Here we set $\boldsymbol{a} = 0.4$. The running time and memory usage of these three classifiers were shown and compared in Figure 5. Due to the page limitation, here we only report experimental results on 10 UCI datasets with different sizes and numbers of attributes.

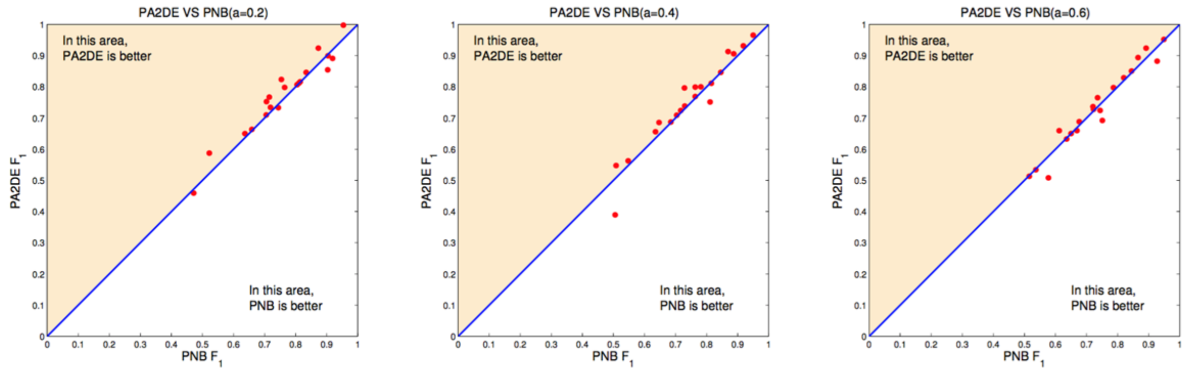

Figure 3. Performance comparison between PA2DE and PNB with different $\boldsymbol{a}$ values

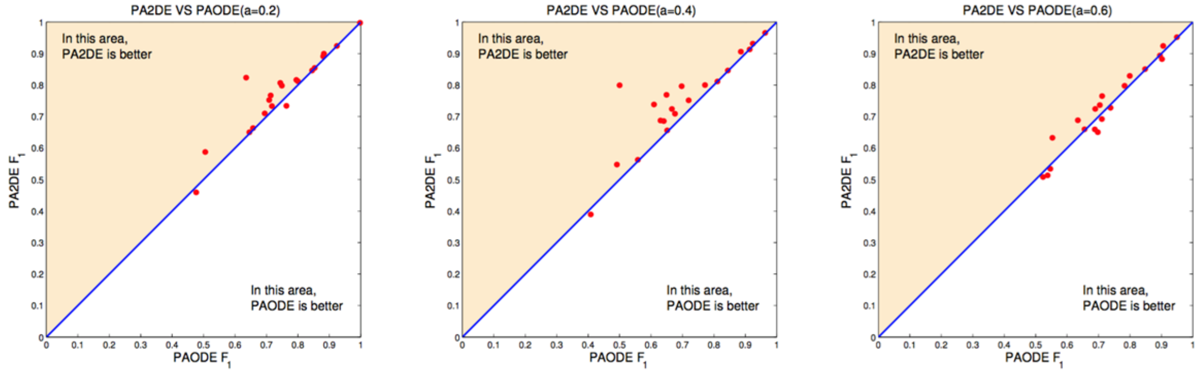

Figure 4. Performance comparison between PA2DE and PAODE with different $\boldsymbol{a}$ values

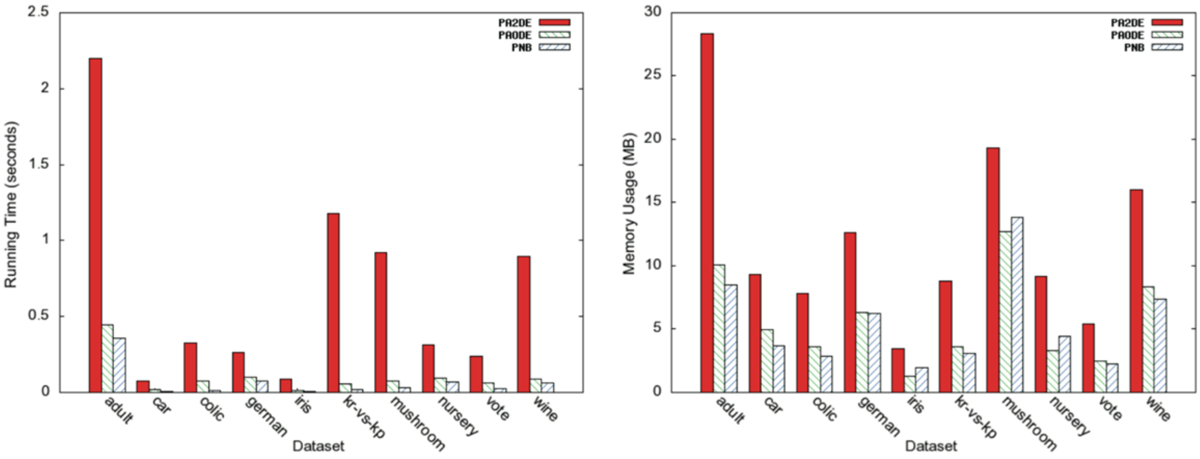

Figure 5. Time and space required by the three algorithms PA2DE, PNB and PAODE on 10 UCI datasets

From Figure 5, we can see that PA2DE requires more time and memory than the other two algorithms. This is not only because the time and space complexity of AnDE increases with n, but also because PA2DE has to update each combination of $n + 1$ attributes and values for each example [5]. However, the gap between PA2DE and the other two algorithms would be relatively small when the dimensionality of the test dataset is smaller, such as the Car evaluation and Iris Classification datasets.

4.5. **Experiments on real-world datasets.** In order to illustrate the scalability of PAnDE in real-world applications, in this subsection, we further performed a group of experiments using three human protein glycosylation datasets: C-linked glycosylation, N-linked glycosylation and O-linked glycosylation derived from [23,24]. Glycosylation is an important type of protein post-translational modification (PTM) in eukaryotic cells. There were two different benchmark datasets for each type of glycosylation due to different feature selection techniques (i.e., IG+ IFS and mRMR+IFS, which means two types of two-step feature selection strategy based on Information Gain (IG)+Incremental Feature Selection (IFS) and minimum Redundancy Maximum Relevance (mRMR)+IFS respectively) [23,24]. The datasets with fewer attributes were chosen for each type of glycosylation, since for PAnDE, larger attribute dimensionality will require more computational resources, which is intractable for WEKA implementation. Accordingly, three datasets were selected and their summary is provided in Table 2.

We experimented with different *UnLevel* values ranging from 40% to 80% with a step size of 10% and compared the classification performance of PA2DE with PAODE and PNB. The results are presented in Table 3. We can see that PA2DE generally performed well across all three types of glycosylation datasets in terms of the F_1 measure.

For the C-linked glycosylation dataset, when *UnLevel* was set as 50%, 60% and 80%, PA2DE outperformed PAODE and PNB, especially when *UnLevel* = 50% ($F_1 = 0.9997$). When *UnLevel* = 40%, all the three algorithms achieved the F_1 score of 1. On the N-linked glycosylation dataset, PA2DE achieved a competitive performance on all these five groups of experiments, except that PA2DE had the same F_1 score with PAODE when *UnLevel* = 40% and 60%. On the O-linked glycosylation dataset, PNB outperformed PA2DE when *UnLevel* = 40%, 60% and 80%, but the performance of PA2DE was competitive with these two algorithms when *UnLevel* = 50% and 70%.

Table 2. A summary of the three types of glycosylation datasets used to evaluate the performance of the algorithms in real-world applications

Datasets	**Size**	**#Attributes**	**#Class**	**Pos Class**	**Pos/Neg**
C-linked (IG + IFS)	163	8	2	1	33.7%/66.3%
N-linked (IG + IFS)	1000	4	2	1	33.3%/66.7%
O-linked (mRMR + IFS)	1538	4	2	1	33.8%/66.2%

Table 3. The classification performance of three algorithms on three types of glycosylation datasets, in terms of the F_1 scores

UnLevel	**Datasets**	**PA2DE**	**PAODE**	**PNB**
40%	C-linked	**1**	**1**	**1**
	N-linked	**0.9224**	**0.9224**	0.8603
	O-linked	0.8310	0.831	**0.8331**
50%	C-linked	**0.9997**	0.9667	0.9667
	N-linked	**0.9109**	0.9032	0.8856
	O-linked	**0.8321**	0.8117	0.8227
60%	C-linked	**0.9667**	0.9574	0.9556
	N-linked	**0.9241**	**0.9241**	0.8851
	O-linked	0.8274	0.8022	**0.8292**
70%	C-linked	0.9556	0.9634	**0.9667**
	N-linked	**0.8595**	0.8440	0.856
	O-linked	**0.8215**	0.8112	0.8193
80%	C-linked	**0.9333**	0.9226	0.9216
	N-linked	**0.8851**	0.8851	0.8777
	O-linked	0.8253	0.8115	**0.8331**

5. **Conclusion and Future Work.** Based on the '*selected completely at random*' assumption, we developed a new positive-unlabeled learning algorithm, termed as PAnDE, by relaxing the attribute independence assumption. Empirical results on the 20 UCI datasets show that PA2DE ($n = 2$) achieved better classification performance compared with two other popular NB-based PU learning algorithms, namely PNB and PAODE. We also evaluated the performance of PAnDE with PNB and PAODE on the human protein glycosylation datasets. The results were consistent with those on the UCI datasets and show that PA2DE outperformed the other two algorithms, suggesting that PAnDE is a useful algorithm in real-world applications. It is expected that PAnDE will be applied in many real-world applications to facilitate effective learning from positive and unlabeled data in the future. The current version of PA2DE can only handle discrete values, which will be extended to deal with continuous values in our future work and be more suitable to deal with PU classification tasks in real-world scenarios. Finally, subsumption resolution [25] and weighted averaging [26] have both been demonstrated to substantially improve the accuracy of AnDE [27]. They may also prove effective when coupled with PAnDE.

REFERENCES

[1] C. Elkan and K. Noto, Learning classifiers from only positive and unlabeled data, *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.213-230, 2008.

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

[3] G. I. Webb and M. J. Pazzani, Adjusted probability Naive Bayesian induction, *Advanced Topics in Artificial Intelligence*, pp.285-295, 1998.

[4] G. I. Webb, J. R. Boughton and Z. Wang, Not so Naive Bayes: Aggregating one-dependence estimators, *Machine Learning*, pp.5-24, 2005.

[5] G. I. Webb, J. R. Boughton, F. Zheng, K. M. Ting and H. Salem, Learning by extrapolation from marginal to full-multivariate probability distributions: Decreasingly naive Bayesian classification, *Machine Learning*, pp.233-272, 2012.

[6] B. Zhang and W. Zuo, Learning from positive and unlabeled examples: A survey, *International Symposiums on Information Processing (ISIP)*, pp.650-654, 2008.

[7] C. Li, Y. Zhang and X. Li, OcVFDT: One-class very fast decision tree for one-class classification of data streams, *Proc. of the 3rd International Workshop on Knowledge Discovery from Sensor Data*, pp.79-86, 2009.

[8] H. Yu, J. Han and K. C.-C. Chang, PEBL: Positive example based learning for web page classification using SVM, *Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.239-248, 2002.

[9] B. Liu, W. S. Lee, P. S. Yu and X. Li, Partially supervised classification of text documents, *Proc. of International Conference on Machine Learning*, pp.387-394, 2002.

[10] P. Yang, X. Li, J. Mei, C. K. Kwoh and S. K. Ng, Positive-unlabeled learning for disease gene identification, *Bioinformatics*, vol.28, no.20, pp.2640-2647, 2012.

[11] P. Yang, X. Li, H. Chua, C. K. Kwoh and S. K. Ng, Ensemble positive unlabeled learning for disease gene identification, *PLOS ONE*, 2014.

[12] X. Li, S. Y. Philip, B. Liu and S. K. Ng, Positive unlabeled learning for data stream classification, *Proc. of SIAM International Conference on Data Mining*, pp.257-268, 2009.

[13] F. Denis, R. Gilleron and F. Letouzey, Learning from positive and unlabeled examples, *Theoretical Computer Science*, vol.348, pp.70-83, 2005.

[14] F. Denis, R. Gilleron, A. Laurent and M. Tommasi, Text classification and co-training from positive and unlabeled examples, *Proc. of the ICML 2003: The Continuum from Labeled to Unlabeled Data*, pp.80-87, 2003.

[15] H. Fei, Y. Kim, S. Sahu, M. Naphade, S. K. Mamidipalli and J. Hutchinson, Heat pump detection from coarse grained smart meter data with positive and unlabeled learning, *Proc. of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.1330-1338, 2013.

[16] J. Zhao, X. Liang, Y. Wang, Z. Xu and Y. Liu, Protein complexes prediction via positive and unlabeled learning of the PPI networks, *The 13th International Conference on Service Systems and Service Management (ICSSSM)*, 2016.

[17] M. Li, S. Pan, Y. Zhang and X. Cai, Classifying networked text data with positive and unlabeled examples, *Pattern Recognition Letters*, vol.77, pp.1-7, 2016.

[18] F. Mordelet and J. P. Vert, A bagging SVM to learn from positive and unlabeled examples, *Pattern Recognition Letters*, vol.37, pp.201-209, 2014.

[19] T. Peng, W. Zuo and F. He, SVM based adaptive learning method for text classification from positive and unlabeled documents, *Knowledge and Information Systems*, vol.16, pp.281-301, 2008.

[20] F. Mordelet and J. P. Vert, ProDiGe: Prioritization of disease genes with multitask machine learning from positive and unlabeled examples, *BMC Bioinformatics*, vol.12, 2011.

[21] J. He, Y. Zhang, X. Li and Y. Wang, Bayesian classifiers for positive unlabeled learning, *Web-Age Information Management*, pp.81-93, 2011.

[22] B. Calvo, P. Larranaga and J. A. Lozano, Learning Bayesian classifiers from positive and unlabeled examples, *Pattern Recognition Letters*, vol.28, no.16, pp.2375-2384, 2007.

[23] F. Li, C. Li, M. Wang, G. I. Webb, Y. Zhang, J. C. Whisstock and J. Song, GlycoMine: A machine learning-based approach for predicting N-, C- and O-linked glycosylation in the human proteome, *Bioinformatics*, vol.31, no.9, pp.1411-1419, 2015.

[24] F. Li, C. Li, J. Revote, Y. Zhang, G. I. Webb, J. Li, J. Song and T. Lithgow, GlycoMine^{struct}: A new bioinformatics tool for highly accurate mapping of the human N-linked and O-linked glycoproteomes by incorporating structural features, *Scientific Report*, 2016.

[25] F. Zheng, G. I. Webb, P. Suraweera and L. Zhu, Subsumption resolution: An efficient and effective technique for semi-Naive Bayesian learning, *Machine Learning*, vol.87, no.1, pp.93-125, 2012.

[26] L. Jiang and H. Zhang, Weightily averaged one-dependence estimators, *Pacific Rim International Conference on Artificial Intelligence*, pp.970-974, 2006.

[27] N. Zaidi and G. I. Webb, Fast and effective single pass bayesian learning, *Proc. of the 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp.149-160, 2013.