Instance Selection by Border Sampling in Multi-Class Domains

Guichong Li[1], Nathalie Japkowicz[1], Trevor J. Stocki[2], and R. Kurt Ungar[2].

[1]School of Information Technology and Engineering, University of Ottawa
800 King Edward Ave, Ottawa, ON, Canada, K1N 6N5
{jli136, nat}@site.uottawa.ca
[2]Radiation Protection Bureau, Health Canada, Ottawa, ON, Canada, K1A 1C1
{trevor_stocki, kurt_ungar}@hc-sc.gc.ca

Abstract. Instance selection is a pre-processing technique for machine learning and data mining. The main problem is that previous approaches still suffer from the difficulty to produce effective samples for training classifiers. In recent research, a new sampling technique, called Progressive Border Sampling (PBS), has been proposed to produce a small sample from the original labelled training set by identifying and augmenting border points. However, border sampling on multi-class domains is not a trivial issue. Training sets contain much redundancy and noise in practical applications. In this work, we discuss several issues related to PBS and show that PBS can be used to produce effective samples by removing redundancies and noise from training sets for training classifiers. We compare this new technique with previous instance selection techniques for learning classifiers, especially, for learning Naïve Bayes-like classifiers, on multi-class domains except for one binary case which was for a practical application.

Keywords: Instance Selection, Border Sampling, Multi-class Domains, Class Binarization method.

1 Introduction

It has been realized that the redundancies and noise in data hinder data mining and machine learning algorithms to achieve their goals [11]. Practitioners have made a lasting effort on developing effective pre-processing techniques in recent decades [8][13][17][20]. Instance selection (IS) is a pre-processing technique that selects a consistent subset of the original training set for a supervised learning [11][20]. As a result, IS brings in two benefits: reducing the learning cost with respect to computational cost and helping learners build successful classifiers.

However, the previously proposed IS techniques still suffer from ineffectiveness of the resulting samples for learning any classifier. For example, Condensed Nearest Neighbour rule (CNN) and Editing Nearest Neighbour rule (ENN) tend to be used for Instance-Based Learning (IBL) [2][20]. Instead of those issues for relieving the learning cost, in this paper, we emphasize the effectiveness of IS to help induction algorithms learn successful classifiers if a training set reduction becomes possible

because mining an effective sample from the original training set, especially on a multi-class domain, is not a trivial issue.

Recently proposed Progressive Border Sampling (PBS) [9][10] can overcome the drawback encountered in previously proposed approaches by borrowing the idea of Progressive Sampling techniques (PS) [13]. PBS can produce a small sample from the original training set by identifying and augmenting the border points for training any classifier. In this paper, we discuss how to use PBS to produce effective samples by removing redundancies and noise on multi-class domains.

As we know, the previously proposed Repeated Nearest Neighbour rule (RENN) [20] can be used for removing noise by repeatedly applying ENN. The main problem of RENN is that the repeated process is subject to a loss of information. In this paper, we first improve RENN by incorporating the Progressive Learning (PL) technique of PS with ENN for algorithmic convergence. After the noise is removed by avoiding the loss of information, PBS identifies and augments border points by assuming a pairwise border sampling strategy on multi-class domains. We show that the new method by incorporating the strategies for noise on multi-class domains with PBS outperforms the previously proposed IS techniques for training Naïve Bayes-like classifiers such as Aggregating One-Dependence Estimators (AODE) [19], etc.

The remainder of this paper is organized as follows. In Section 2, we introduce the two related works to give some background. In Section 3, we discuss the method for border sampling on multi-class domains. We discuss a new strategy for removing noise, and then incorporate the new strategy with PBS for effective samples in Section 4. The experimental design and results are reported in Section 5. Finally, we draw our conclusion and suggest future work in Section 6.

2 Preliminary

We introduce the methodology related to instance selection on multi-class domains.

2.1 Instance selection by border sampling

Instance selection techniques (IS) focus on selecting a consistent subset of the training set for Instance-Based Learning [11][20]. The Condensed Nearest Neighbour rule (CNN) [17][20], a pioneer of the IS, finds a minimally consistent subset S of the training sets T. Editing Nearest Neighbour rule (ENN) reduces training sets by removing noise, which cannot be correctly classified by their k nearest neighbours, e.g., $k = 3$ in this paper. Because the removal of a noisy data point might lead to a new source of noise, Repeated Editing Nearest Neighbour rule (RENN) repeatedly removes noisy data until no noise of this kind is found [20]. Further, a variant of Decremental Reduction Optimization Procedure 3 (DROP3) [20] and Iterative Case Filtering (ICF) [2], denoted as DROP3.1, can be used for removing redundant data points. DROP3.1 first executes ENN to remove noise from the original training set T, and sort the resulting instances S by distances to their nearest neighbours belonging to the other classes in S, and then remove redundant points, which can be classified by their k nearest neighbours, e.g., $k = 5$ in this paper, in S with a high probability p, e.g., $p \geq 0.8$ in this paper, without the redundant points.

On the other hand, border points potentially exist in a labelled dataset [9]. A full border consists of near borders and far borders, and it can be identified by a recent technique called Border Identification in Two Stages (BI_2) [9]. Because initial border points have high uncertainty, which is insufficient for adequate learning [5][12], Progressive Border Sampling (PBS) [9] has been proposed to augment border points for an effective sample the context of supervised learning by borrowing the basic idea behind Progressive Sampling technique (PS) [13], which progressively learns a small sample from the original training set with an acceptable accuracy by defining a sampling schedule and convergence condition [8][13].

2.2 Pairwise Naïve Bayes

Given a training set with a probability distribution P, Naïve Bayes assumes the probabilities of attributes $a_1,a_2,\ldots,a_n$ to be conditionally independent given the class $c_i \in C$ [5][12], and is given by

$$y_i = \arg\max_{c_i \in C} \prod_{j=1}^{n} P(a_j \mid c_i) P(c_i)$$

Because the conditional independence is not expected to be satisfied in practical applications [4], previous research has proposed Naïve Bayes-like classifiers for the enhancement of Naïve Bayes by relieving the restriction of conditional independence. Aggregating One-Dependence Estimators (AODE) [19] achieves higher accuracy by averaging over a constrained group of 1-dependence Naive-Bayes models built on a small space. AODE with Subsumption Resolution (AODEsr) [22] augments AODE by detecting the specialization-generalization relationship between two attribute values at classification time and deleting the generalization attribute value. Hidden Naïve Bayes (HNB) [21] constructs a hidden parent for each attribute. Weightily Averaged One-Dependence Estimators (WAODE) [7] weights the averaged 1-dependence classifiers by the conditional mutual information.

On the other hand, learning a Naïve Bayes is different from learning a Support Vector Machine (SVM) because SVM is originally designed as a binary classifier while other classifiers, e.g., Naïve Bayes and Decision Tree, are directly designed on either binary or multi-class domains. The class binarization methods [3][18], e.g., the one-against-one (oo) and one-against-all (oa), are used for enhancing binary classifiers on multi-class domains.

In general, the pairwise classification or the oo method transforms a multi-class domain with m class into m(m-1)/2 binary domains. Each binary domain consists of all examples from a pair of classes. A binary classifier is trained on each binary domain. For classification, an observation x is input to all binary classifiers, and the predictions of the binary classifiers are combined to yield a final prediction.

There is a theoretical discussion about the pairwise Naïve Bayes classifiers, which is related to the pairwise Bayes classifier [16]. A pairwise probabilistic classifier is trained on a binary domains consisting of all examples in either c_i or c_j, denoted as c_{ij}, to estimate probabilities $p_{ij} = P(c_i|x, c_{ij})$ and $p_{ji} = P(c_j|x,c_{ij}) = 1 - p_{ij}$ for voting. It has been shown that the resulting prediction from all binary classifiers by a linear combination of votes is equivalent to regular Bayes classification for class ranking.

The oa classification splits a multi-class domain into m binary domains consisting of one class c_i, i = 1…m, from all other classes, and train these binary classifiers using all examples of class c_i as positive examples and the examples of the union of all other classes $c_j = D - c_i$ as negative examples.

It has been realized that pairwise Naïve Bayes built on each pair of classes of a multi-class domain is reduced to a standard Naïve Bayes directly built on the multi-class domain [16]. Although the oa classification can be reduced to a regular Bayes classification, a Naïve Bayes classifier with the oa is not consistent with a regular Naïve Bayes because the related probability estimates are not equivalent [16].

3 Pairwise Border Sampling

We discuss two main issues related to border sampling on multi-class domains.

3.1 Class binarization method

Border sampling on multi-class domains is not a trivial issue. The previous class binarization methods for classification provide a direct venue for the border sampling on multi-class domains. As a result, two kinds of class binarization methods, i.e., one-against-one (oo) and one-against-all (oa), for border sampling on multi-class domains can be described as follows.

- **oo method**

It is also called the pairwise method. Border sampling with the oo strategy identifies the pairwise borders on each pair of classes. All obtained c(c – 1)/2, where c is the number of classes, pairwise borders are combined together by a simple union as the resulting sample.

- **oa method**

Border sampling with the oa strategy identifies individual borders b_i in each class by identifying a pairwise border b'_i between the class and the rest of classes such that b_i can be obtained by retaining border points in class i out of b'_i. All obtained individual borders b_i, i = 1,…, k are combined together by a simple union as the resulting border.

3.2 Naïve Bayes Validation

Initially identified border points have high uncertainty, which might be improper for sufficiently learning. Uncertainty can be overcome by progressively learning new border points on the remaining data obtained by removing the previously identified border points for an augmented border until this augmented border is sufficient for Bayesian learning [5].

The paiwise border sampling identifies and augments border points on each pair of classes by assuming the oo strategy. Heuristically, the augmentation on each pair can be validated by building a Naïve Bayes model and testing on the pair until the performance of the Naïve Bayes model does no longer ascend [9]. As a result, a pairwise Naïve Bayes are built from all pairs of classes. According to the early

discussion, it is believed that this pairwise Naïve Bayes can be reduced to the standard Naïve Bayes built on the resulting sample.

However, a Naïve Bayes with the oa is not equivalent to a standard Naïve Bayes due to the probability estimation [16]. Moreover, because the oo is applied on each pair of classes, it requires less data access than the oa. As a result, the pairwise PBS is preferable to the PBS with the oa.

4 Instance selection by Border Sampling in Multi-Class Domains

Noise removal is an important issue for instance selection. In general, there are two kinds of methods: the Tomek Link based method and the RENN based method, for noise removal. A Tomek Link is a pair of adjunct data points belonging to different categories, and one in the pair is identified as a noise if it is farther from its nearest neighbour in the same category than the adjunct point in the Tomek Link [17][20].

As we know in Section 2.1, ENN is used for removing noise, which cannot be classified correctly by its nearest neighbours. RENN is a method to repeatedly remove noise of this kind by applying ENN. Therefore, this RENN based method for noise removal appears preferable to the Tomek Link based method because it has a more direct effect for classification than the latter. The main problem of RENN is that it suffers from the loss of information because some border points are also removed as noise while they are useful for training classifiers [20].

4.1 PENN

```
PENN algorithm
Input       D: a sample for training with c classes
Output      D'
begin
1     D' = D, oD = D, LCurve[k], k = 0..K(100),k = 1
2     while(true)
3         LCurve[k] = LearningNB(D', D)
4         if(LCurve[k] < LCurve[k-1])
5            D' = oD, break
6         H_k = kNN(D', 3), D" = ∅, isFinished = true
7         for(each p ∈ D')
8            if(H_k(p))
9              D" = D" ∪ p
10           else
11             isFinished = false
12        if(isFinished)
13           break;
14        oD = D', D' = D", k++
15    return D'
end
```

Figure 1. PENN algorithm.

We propose a new algorithm for improving the original Repeatedly Editing Nearest Neighbour rule by assuming PL technique. The new algorithm is called Progressively

Editing Nearest Neighbour (PENN), as shown in Figure 1. PENN has only input: D, which is the original training set. It outputs D′ as the reduced training set. The algorithm initializes its variables at Step 1. LCurve is used for describing the learning curve of Naïve Bayes. From Step 2 to Step 14, the algorithm progressively learns the resulting sample D′ by removing noise in the previously generated D′. A Naïve Bayes classifier is built on D′ and tested on the original data D during Step 3. If the learning curve descends at Step 4, the algorithmic convergence is detected, and the previously learned result oD is returned as D′ at Step 5. Otherwise, the algorithm builds a k-Nearest Neighbour classifier (kNN) on D′ with its parameter of 3 (the number of nearest neighbours) at Step 6, and the kNN model classifies each data point in D′ at Step 8. Actually, the algorithm from Step 6 to Step 11 corresponds to the original ENN. If all data are correctly classified, then the while loop exits at Step 13. Otherwise, the algorithm continues in the while loop.

4.2 PEBS algorithm: A hybrid of PENN and PBS

We can combine PENN and PBS for instance selection. The combination of PENN and PBS is a hybrid algorithm, called PEBS, as shown in Figure 2. First, PENN is used for removing noise. Second, PBS is used for removing redundancy. Arguably, PENN is not suggested to be invoked after PBS is invoked in PEBS because there is no chance to add new border points after noise is removed.

```
PEBS algorithm
Input      D: a sample for training with c classes
Output     B
begin
1    D = PENN(D) , B = ∅;
2    C = getClassset(D), C = {C_i | i = 0, …, c}
3    for ∀i, j, where i < j, C_i ≠ ∅, and C_j ≠ ∅
4        B_ij = ∅, C'_i = C_i, C'_j = C_j; C_ij = C_i ∪ C_j
5        Acc[k] = 0, k = 0, 1, …, K, K = 100
6        while(true)
7            B'_ij = BI_2(C'_i, C'_j, B_ij)
8            B_ij = B_ij ∪ B'_ij,
9            C'_i = C'_i – B'_ij, C'_j = C'_j - B'_ij
10           Acc[k] = ValidateNBModel(B_ij, C_ij)
11           if(Acc[k] ≤ Acc[k-1])
12              B_ij = old; break;
13              continue
14           old = B_ij, k++
15       B = B ∪ B_ij
```

Figure 2. PEBS: The hybrid of PENN and PBS.

In Figure 2, PEBS applies pairwise border sampling on each pair of classes from Step 3 to Step 15. In the while loop from Step 6 to Step 14, PEBS identifies at Step 7, augments border points at Step 8, and validates a Naïve Bayes built at Step 10 on the current border points and tested on the pair of classes for convergence detection at Step 11. All augmented border points from all pairs of classes are unified together at Step 15 as a resulting sample.

The number of iterations of the while loop from Step 2 to Step 14 in PENN is expected to be bounded with a small number. However, PENN has a quadratic time complexity due to kNN is quadratic for classification [2][20]. PEBS is also quadratic due to PENN and the original PBS [9] although PBS can be scaled up [10].

5 Experiments

Our experimental design and results are reported as follows.

5.1 Datasets for Experiments

We conducted experiments on 10 benchmark multi-class datasets chosen from the UCIKDD repository [1] and one binary dataset obtained from a nuclear security application, as shown in Table 1, where the columns #attr, #ins, and #c are the number of attributes, instances, and classes in training sets, respectively; #PEN, #PEBS, #CNN, #ENN, #RENN, and #DROP3.1 are the sample sizes generated by PENN, PEBS, CNN, ENN, RENN, and DROP3.1, respectively; %PEN, %PEBS, and %RENN are the percents of #PEN, #PEBS, and #RENN to #ins, respectively.

For the application, a possible method of explosion detection for the Comprehensive nuclear-Test-Ban-Treaty [15] consists of monitoring the amount of radioxenon in the atmosphere by measuring and sampling the activity concentration of Xe-131m, Xe-133, Xe-133m, and Xe-135 [14]. Several samples are synthesized under different circumstances of nuclear explosions, and combined with various levels of normal concentration backgrounds so as to synthesize a binary training dataset, called XenonT2D1.

In our experiments, PEBS ran with the Radial-based function [12] as a similarity measure for computing the nearest neighbours. Several inductive algorithms are used for training Naïve Bayes-like classifiers on either the resulting samples generated by PEBS or the full training sets (Full), or those generated by previous approaches, i.e., CNN, ENN, RENN, and DROP3.1, which is implemented for experiments in this paper. The performances of these classifiers with respect to the Area under ROC curve (AUC) [6], based on averages obtained within 20 runs of the 10 cross validation, are used for comparison between PEBS and the other algorithms.

The software tools for Naïve Bayes and three Naïve Bayes-like learners: AODE, AODEsr, and HNB (WAODE is omitted due the limitation of space) are chosen from the Waikato Environment for Knowledge Analysis (Weka) [23]. The datasets have been pre-processed by using the ReplaceMissingValue tool in Weka for missing values and the unsupervised Discretize tool in Weka for discretizing continuous values. The classifiers are built with their default settings, with no loss of generality, e.g., NB with Maximum Likelihood estimator, and AODE with a frequencyLimit of 1, i.e., any attribute with values below this limit cannot be use as a parent, etc.

5.2 Experimental Results

Our initial results in Table 1 show that PEBS can produce much smaller samples, e.g., on average, 303 samples and 653 samples from Anneal and Hypothyroid, respectively, than other approaches, i.e., CNN, ENN, and RENN except DROP3.1,

while it can retain most instances, e.g., in Vowel, if few redundancies can be found. The comparison between PENN and RENN is discussed later. On average, PEBS produces smaller samples than other approaches except for DROP3.1, which intends to produces the smallest samples among all approaches.

XenonT2D1 is a distinct case that the synthesized data contains much redundancy. PEBS can produce a much smaller sample from XenonT2D1 than other approaches while other approaches reduce a little redundancy except for DROP3.1.

We show the effectiveness of PEBS by comparing PEBS with CNN, ENN, RENN, and DROP3.1 for training the classifiers, i.e., NB, AODE, AODEsr, and HNB, as shown from Table 2 to Table 5. We use 'w' and 'l' to represent PEBS's wins and losses, respectively, against the corresponding methods in terms of the paired t-test (first) and Wilcoxon signed rank test (second) at significance levels of 0.05.

Table 1. The 11 datasets

Datasets	#attr	#ins	#c	#PEN	%PEN	#PEBS	%PEBS	#CNN	#ENN	#RENN	%RENN	#DROP3.1
Anneal	39	898	5	808	100	303	37	793	800	797	99	148
Audiology	70	226	24	203	100	164	80	179	154	140	69	88
Autos	26	205	6	185	100	161	87	179	155	143	78	107
Balance-s	5	625	3	528	94	459	82	458	500	499	89	306
Hypothyroid	30	3772	4	3395	100	653	19	3071	3185	3170	93	111
P-tumor	18	339	21	305	100	302	99	293	167	128	42	124
Soybean	36	683	18	615	100	541	88	519	582	573	93	162
Vehicle	19	846	4	720	95	697	91	753	624	592	78	336
Vowel	14	990	11	891	100	891	100	845	843	828	93	570
Zoo	18	101	7	88	97	39	43	71	88	87	96	23
XenonT2D1	5	640	2	572	99	26	5	578	567	567	98	30
Average		848		756	99	385	67	703	697	684	84	182

PEBS can help learn better NB and other three Naïve Bayes-like classifiers, as shown from Table 2 to Table 5, in most cases in terms of the paired t-test and Wilcoxon signed rank test as compared with Full, and other approaches. Especially, it is consistently superior to DROP3.1 in all cases for training classifiers.

The averaged AUC are shown at the bottoms of Table 2 to Table 5. We summarized the results for statistical test in Table 6. The results clearly show that PEBS consistently outperforms previously proposed instance selection approaches for training set reduction, and helps learn successful classifiers as compared with Full.

PENN is an improved method for noise removal by incorporating PL technique with ENN. As we can see in Table 1, PENN is not expected to reduce much noise from the original datasets. There are only four cases, i.e., Balance-s, Vehicle, Zoo, and XenonT2D1, where PENN can remove noise, which is less than that removed by RENN. We emphasize that PENN can guarantee few loss of information such that PEBS can produce effective samples for training classifiers as compared with Full and other instance selection approaches.

We compare PENN with RENN by training NB and other three Naïve Bayes-like classifiers on either the resulting samples generated by PENN and RENN or the full training sets (Full), as shown in Table 7, where the names of datasets are omitted, and

the rows correspond to the datasets in Table 1 in order without any confusion. The bottom row shows the average values.

As we can see, there is only case, i.e., Balance-s, where PENN is inferior to RENN for training NB and other three Naïve Bayes –like classifiers in terms of the paired t-test and Wilcoxon signed rank test. PENN is superior to RENN in all other cases by avoiding loss of information, and PENN consistently helps learn NB and other three Naïve Bayes-like classifiers without any loss of information as compared with Full.

Balance-s is also a case that PENN enhances PBS in PEBS. We conducted the related experiments in that PBS without PENN is inferior to other approaches for training Naïve Bayes and other three Naïve Bayes-like classifiers although it does not intend to degrade the performance of these classifiers built on the resulting sample as compared with learning on the original training set. In addition, the maximum tries of PEBS for pairwise border sampling is 16 on P-tumor case. Empirically, it is bound by a small number, as discussed in the previous research for PBS [9].

The results on XenonT2D1 surprise us that PEBS consistently outperforms other approaches for training successful classifiers by producing a much small sample.

Table 2. Training NB.

	PEBS	Full	CNN	ENN	RENN	DROP3.1
Anneal	0.9587	0.9601 -l	0.96 -l	0.9593 -l	0.9592	0.9501 -w
Audiology	0.6984	0.7012 -l	0.7002	0.6904 ww	0.6843 ww	0.6868 ww
Autos	0.9096	0.9119	0.9122	0.8736 ww	0.8602 ww	0.8712 ww
Balance-s	0.8942	0.8307 ww	0.8989	0.9074 -l	0.9075 -l	0.8442 ww
Hypothyroid	0.8995	0.8802 ww	0.8805 -w	0.8805 -w	0.7863 ww	0.8141 ww
P-tumor	0.7543	0.7544	0.7543	0.73 ww	0.7049 ww	0.7308 ww
Soybean	0.9983	0.9983 -w	0.9983	0.9981 -w	0.998 -w	0.9981
Vehicle	0.8109	0.8077 -w	0.8079 -w	0.8079 -w	0.7951 ww	0.7812 ww
Vowel	0.9591	0.9591	0.9574 -w	0.9493 ww	0.9416 ww	0.9572
Zoo	0.894	0.894	0.8917	0.8917	0.894	0.894
XenonT2D1	0.9873	0.9919	0.9919	0.9919	0.9919	0.955 ww
Average	0.8777	0.8698	0.8761	0.8688	0.8531	0.8528

Table 3. Training AODE.

	PEBS	Full	CNN	ENN	RENN	DROP3.1
Anneal	0.9596	0.961	0.961	0.9602	0.9601	0.9515 -w
Audiology	0.6987	0.7015 -l	0.7008 -l	0.6907 ww	0.6844 ww	0.6872 ww
Autos	0.9326	0.9349	0.9352	0.8933 ww	0.8772 ww	0.8897 ww
Balance-s	0.8641	0.798 ww	0.8678	0.8877 -l	0.8856 -l	0.7699 ww
Hypothyroid	0.8952	0.8733 ww	0.8735 ww	0.8735 ww	0.7893 ww	0.8115 ww
P-tumor	0.7546	0.7547	0.7541	0.7305 ww	0.705 ww	0.7315 ww
Soybean	0.9986	0.9986	0.9985	0.9983 ww	0.9982 ww	0.9983
Vehicle	0.8994	0.9013 -l	0.9019 -l	0.9019 -l	0.877 ww	0.8615 ww
Vowel	0.994	0.994	0.9938	0.987 ww	0.9818 ww	0.9902 ww
Zoo	0.894	0.894	0.8917	0.8917	0.894	0.894
XenonT2D1	0.9878	0.9917	0.9917	0.9915	0.9915	0.9579 ww

Average	0.8891	0.8811	0.8878	0.8815	0.8653	0.8585

Table 4. Trianing AODEsr.

	PEBS	Full	CNN	ENN	RENN	DROP3.1
Anneal	0.9647	0.9651	0.9651	0.9639 -w	0.9636 -w	0.9597 -w
Audiology	0.7082	0.7069	0.7075	0.6993 ww	0.6918 ww	0.6962 ww
Autos	0.9403	0.9419	0.9424 -l	0.8954 ww	0.8774 ww	0.8998 ww
Balance-s	0.8665	0.7073 ww	0.8691	0.8858 -l	0.8842 -l	0.7825 ww
Hypothyroid	0.9103	0.8916 -w	0.892 -w	0.892 -w	0.8048 ww	0.8525 ww
P-tumor	0.7576	0.758	0.7577	0.7305 ww	0.7044 ww	0.7343 ww
Soybean	0.9988	0.9989 -l	0.9989	0.9986	0.9986 -w	0.9987
Vehicle	0.8983	0.8979	0.8981	0.8981	0.873 ww	0.8714 ww
Vowel	0.9971	0.9971	0.9971	0.9929 ww	0.987 ww	0.9935 ww
Zoo	0.894	0.894	0.894	0.894	0.894	0.894
XenonT2D1	0.9891	0.9919	0.9919	0.9917	0.9917	0.9773
Average	0.8936	0.8759	0.8922	0.8851	0.8679	0.8683

Table 5. Training HNB.

	PEBS	Full	CNN	ENN	RENN	DROP3.1
Anneal	0.9644	0.9641	0.9638	0.9635	0.9633	0.9583 -w
Audiology	0.7029	0.7044 -l	0.7029	0.6939 ww	0.6878 ww	0.6938 ww
Autos	0.9458	0.9451	0.945	0.8978 ww	0.8769 ww	0.8966 ww
Balance-s	0.8485	0.8808 -l	0.8536	0.8727 -l	0.8727 -l	0.7507 ww
Hypothyroid	0.9066	0.8864 -w	0.8848 -w	0.8848 -w	0.7842 ww	0.8448 ww
P-tumor	0.7557	0.7557	0.7556	0.727 ww	0.7016 ww	0.7273 ww
Soybean	0.999	0.999 -w	0.999	0.9988 -w	0.9987 ww	0.9988 -w
Vehicle	0.9075	0.9078	0.9077	0.9077	0.8794 ww	0.8742 ww
Vowel	0.9974	0.9974	0.9973	0.9931 ww	0.9861 ww	0.9939 ww
Zoo	0.894	0.894	0.894	0.894	0.8893	0.894
XenonT2D1	0.9816	0.9921	0.9921	0.9915	0.9915	0.977
Average	0.8922	0.8935	0.8904	0.8833	0.8640	0.8632

Table 6. Summary of statistical tests.

		Full	CNN	ENN	RENN	DROP3.1
Paired t-test	PEBS	5\39\0	1\43\0	18\26\0	26\18\0	29\15\0
Wilcoxon signed rank test	PEBS	10\27\7	6\34\4	25\13\6	29\11\4	34\10\0

6 Conclusion and Future Work

Instance selection by PBS on multi-class domains is not a trivial issue. As a result, we argue that PBS prefers the pairwise border sampling to the one-against-all method on multi-class domains by borrowing class binarization methods for classification on multi-class domains. We show an improved PENN algorithm, which incorporates Progressive Learning (PL) technique with Editing Nearest Neighbour rule (ENN), for

noise removal without any loss of information. Finally, we design a new hybrid method, called Progressively Editing Nearest Neighbour rule for Progressive Border Sampling (PEBS), for instance selection by incorporating PENN with PBS. PENN is used for noise removal first, and then PBS is used for removing redundancies.

The experimental results show that PEBS can produce much smaller samples than other instance selection approaches in some cases while it produces little larger samples than these approaches in other cases. On average, PBS can produce smaller samples than other approaches except DROP3.1. On the other hand, PEBS consistently outperforms other approaches to produce effective samples in all cases in terms of the paired t-test and in most cases in terms of the Wilcoxon signed rank test. Especially, PEBS consistently outperforms DROP3.1 in all cases. In addition, PENN is not expected to remove much noise as compared with RENN by avoiding loss of information. PENN produces a small sample consistent with the full training set by removing noise if possible. PENN outperforms RENN in most cases except for one case, where it is inferior to RENN. Especially, we show that PENN enhances PBS in the worse case as compared with the full training set.

PENN is not efficient due to its quadratic time complexity, and PEBS for border sampling is still subject to small failures in some case in terms of the Wilcoxon signed rank test. These drawbacks are expected to be overcome in future work.

Table 7. The comparison between PENN and RENN.

NB			AODE			AODEsr			HNB		
PENN	Full	RENN	PENN	Full	RENN	PENN	Full	RENN	PENN	Full	RENN
0.9601	0.9601	0.9592 -w	0.961	0.961	0.9601 -w	0.9651	0.9651	0.9636 -w	0.9641	0.9641	0.9633 -w
0.7012	0.7012	0.6843ww	0.7015	0.7015	0.6844ww	0.7069	0.7069	0.6918ww	0.7044	0.7044	0.6878ww
0.9119	0.9119	0.8602ww	0.9349	0.9349	0.8772ww	0.9419	0.9419	0.8774ww	0.9451	0.9451	0.8769ww
0.8797	0.8307-w	0.9075 -l	0.8421	0.798 ww	0.8856 ll	0.8034	0.7073ww	0.8842 ll	0.8724	0.8808	0.8727
0.8802	0.8802	0.7863ww	0.8733	0.8733	0.7893ww	0.8916	0.8916	0.8048ww	0.8864	0.8864	0.7842ww
0.7544	0.7544	0.7049ww	0.7547	0.7547	0.705 ww	0.758	0.758	0.7044ww	0.7557	0.7557	0.7016ww
0.9983	0.9983	0.998 -w	0.9986	0.9986	0.9982 -w	0.9989	0.9989	0.9986 -w	0.999	0.999	0.9987 -w
0.809	0.8077	0.7951ww	0.8976	0.9013	0.877 ww	0.8948	0.8979	0.873 ww	0.903	0.9078	0.8794ww
0.9591	0.9591	0.9416ww	0.994	0.994	0.9818ww	0.9971	0.9971	0.987 ww	0.9974	0.9974	0.9861ww
0.9919	0.9919	0.9919	0.9916	0.9917	0.9915	0.9917	0.9919	0.9917	0.9915	0.9921	0.9915
0.8846	0.8796	0.8629	0.8949	0.8909	0.8750	0.8949	0.8857	0.8777	0.9019	0.9033	0.8742

References

[1] Bay, S. D.: The UCI KDD archive, 1999. http://kdd.ics.uci.edu (1999).

[2] Brighton, H., & Mellish, C.: Advances in instance selection for instance-based learning algorithms. *Data Mining Knowledge Discovery*, *6(2)*, 153–172 (2002).

[3] Debnath, R., Takahide, N., and Takahashi, H.: A decision based one-against-one method for multi-class support vector machine. Pattern Anal Applic (2004) 7: 164-175. DOI 10.1007/s10044-004-0213-6 (2004).

[4] Domingos, P. & Pazzani, M.: Beyond independence: Conditions for the optimality of the sample Bayesian classifier. Machine Learning 29: 103-130 (1997).

[5] Duda, R.O. & Hart, P.E.: Pattern Classification and Scene Analysis. A Wiley Intersience Publication (1973).
[6] Fawcett, T.: ROC graphs: Notes and practical considerations for researchers. http://www.hpl.hp.com/-personal/TomFawcett/papers/ index.html (2003).
[7] Jiang, L., Zhang, H.: Weightily Averaged One-Dependence Estimators. In: Proceedings of the 9th Biennial Pacific Rim International Conference on Artificial Intelligence, PRICAI 2006, 970-974 (2006).
[8] John, G. & Langley, P.: Static versus dynamic sampling for data mining. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press, pp. 367-370 (1996).
[9] Li, G., Japkowicz, N, Stocki, T.J., and Ungar, R. K.: Full Border Identification for reduction of training sets. Proceedings of the 21st Canadian Artificial Intelligence. Winsor, Canada. Pages: 203-215 (2008).
[10] Li, G., Japkowicz, N, Stocki, T.J., and Ungar, R. K.: Border sampling through Markov chain Monte Carlo. Proceedings of the 2008 Eighth IEEE International Conference on Data Mining. Pisa. Pages 393-402 (2008).
[11] Liu, H. & Motoda, H.: On issues of instance selection. Data Mining and Knowledge Discovery, 6:115–130 (2002).
[12] Mitchell, T.: *Machine Learning*. McGraw-Hill Companies, Inc (1997).
[13] Provost, F., Jensen, D., and Oates, T.: Efficient Progressive Sampling. Proc. of the fifth ACM SIGKDD. San Diego, California, US. Pages: 23 – 32 (1999).
[14] Stocki, T.J. Blanchard, X., D'Amours, R., Ungar, R.K., Fontaine, J.P., Sohier, M., Bean, M., Taffary, T., Racine, J., Tracy, B.L., Brachet, G., Jean M., Meyerhof, D.: Automated radioxenon monitoring for the comprehensive nuclear-test-ban treaty in two distinctive locations: Ottawa and Tahiti. J. Environ. Radioactivity 80:305-326 (2005).
[15] Sullivan, J.D.: The comprehensive test ban treaty. Physics Today 151, 23 (1998).
[16] Sulzmann, J., Fürnkranz, J., and Hüllermeier, E.: On Pairwise Naive Bayes Classifiers. In Proceedings of the 18th European Conference on Machine Learning. (ECML-07), pp. 658-665, Warsawa, Poland. Springer-Verlag (2007).
[17] Tomek, I.: Two modifications of CNN. IEEE Transactions on Systems, Man and Cybernetics, vol.6, no.6, pp.769–772 (1976).
[18] Tsujinishi, D., Koshiba, Y., and Abe, S.: Why pairwise is better than one-against-all or all-at-once. In Proceedings of IEEE International Conference on Neural Networks, volume 1, pages 693–698. IEEE Press (2004).
[19] Webb, G. I., Boughton, J., Wang, Z.: Not So Naive Bayes: Aggregating One-Dependence Estimators. Machine Learning. 58(1):5-24 (2005).
[20] Wilson, D. R., and Martinez, T. R.: Reduction Techniques for Instance-Based Learning Algorithms. Machine Learning, 38, 257–286 (2000). Kluwer Academic Publishers. Printed in The Netherlands (2000).
[21] Zhang, H., Jiang, L., Su, J.: Hidden Naive Bayes. In: Twentieth National Conference on Artificial Intelligence, 919-924 (2005).
[22] Zheng, F., Webb, G. I.: Efficient lazy elimination for averaged-one dependence estimators. In: Proc. 23th International Conference on Machine Learning (ICML2006), 1113-1120 (2006).
[23] WEKA Software, v3.5.2. University of Waikato. http://www.cs.waikato.ac.nz-/ml/weka/index_datasets.html.