Border Sampling Through Coupling Markov Chain Monte Carlo

Guichong Li[1], Nathalie Japkowicz[1], Trevor J. Stocki[2], and R. Kurt Ungar[2]
[1]Computer Science of University of Ottawa
{jli136, nat}@site.uottawa.ca
[2]Radiation Protection Bureau, Health Canada, Ottawa, ON, Canada
{trevor_stocki, kurt_ungar}@hc-sc.gc.ca

Abstract

Recently, Progressive Border Sampling (PBS) was proposed for sample selection in supervised learning by progressively learning an augmented full border from small labeled datasets. However, this quadratic learning algorithm is inapplicable to large datasets. In this paper, we incorporate the PBS to a state of the art technique called Coupling Markov Chain Monte Carlo (CMCMC) in an attempt to scale the original algorithm up on large labeled datasets. The CMCMC can produce an exact sample while a naive strategy for Markov Chain Monte Carlo cannot guarantee the convergence to a stationary distribution. The resulting CMCMC-PBS algorithm is thus proposed for border sampling on large datasets. CMCMC-PBS exhibits several remarkable characteristics: linear time complexity, learner-independence, and a consistent convergence to an optimal sample from the original training sets by learning from their subsamples. Our experimental results on the 33 either small or large labeled datasets from the UCIKDD repository and a nuclear security application show that our new approach outperforms many previous sampling techniques for sample selection.

1. Introduction

In recent research, a new approach called Progressive Border Sampling (PBS) was proposed to learn a small sample from original populations by incorporating a novel method called the Border Identification in Two Stages (BI_2) algorithm with progressive learning techniques [12]. Border sampling by PBS tends to find a theoretically minimally sufficient training set given any training set. Its main advantage is that it is learner-independent. As a result, many classic learning algorithms can build successful classifiers on the resulting sample without any loss of performance with respect to the classifiers built on the full training sets [12]. On the other hand, despite this advantage, the algorithm is still inapplicable to large datasets due to its quadratic learning time. For example, the algorithm could not efficiently process the UCIKDD [2] Letter datasets (with twenty thousand data points) or even the UCIKDD Splice datasets (with three thousand data points).

Recent research has focused on learning tasks on large datasets [9][15]. However, there exist some vital drawbacks to this research. For example, within the classification branch of machine learning, Progressive Sampling techniques (PS) [9][15][24] are subject to a failure in converging to an optimal sample due to the bias of a base learner. Active learning techniques or semi-supervised learning whose goal is to reduce learning cost for labeling [3][22] suffer from the same difficulty as PS to converge to an optimal sample due to the high bias of the selected learner. Conversely, we believe that reducing the variance of the data caused by redundancies can help reduce the learning cost without loss of performance.

In this paper, we first review and discuss the theoretical issues related to the previous PBS technique. In terms of this theoretical foundation, we investigate and discuss the scalability of PBS to large datasets. A natural way is to adopt the standard Markov Chain Monte Carlo (MCMC) [1] for border sampling on large datasets. Among the variant MCMC techniques, the state of the art Coupling Markov Chain Monte Carlo (CMCMC) can produce an exact sample while the standard MCMC cannot guarantee to converge to the stationary distribution [1][14].

As a result, we incorporate PBS with CMCMC for border sampling on large datasets, and propose a new approach, called Coupling Markov Chain Monte Carlo-based PBS (CMCMC-PBS), in which two interactive Markov chains, called the B chain for border points and the R chain for redundant data points are evolved

by using PBS as an oracle. Correspondingly, the convergence detection for the B chain and the collapsing condition for the R chain are heuristically defined.

There are three main advantages to the CMCMC-PBS. First, it is independent of inductive algorithms as PBS itself. Therefore, it can unlimitedly learn an optimal sample by reducing the variance of the data due to redundancies from the original large population. Second, the CMCMC-PBS is a linear algorithm and can efficiently converge to a perfect sample by calling many small subsamples with a rapid mixing time related to the CMCMC techniques. Therefore, it is feasible to use in practical applications. Third, CMCMC-PBS is not restricted for use in either small or large datasets because it is not sensitive to the sampling window. At the extreme case, the whole training set is fitted in the sampling window.

We compare the proposed CMCMC-PBS algorithm with previous sampling techniques for reducing learning cost in large labeled training sets [9][15] by conducting experiments on benchmark datasets from UCIKDD repository [2], as well as on a problem of nuclear explosion detection through the monitoring of radioxenon levels in the atmosphere [20]. This work is conducted in the context of the Comprehensive Nuclear-Test-Ban Treaty (CTBT) whose purpose is to ban the testing of nuclear explosive devices worldwide.

The remainder of this paper is organized as follows. In Section 2, we review the theoretical foundation for Border Identification (BI) and Markov Chain Monte Carlo techniques proposed in previous research. Our main work for border sampling on large datasets is described in Section 3. In Section 4, we describe our experimental design and results. We conclude and suggest future work in Section 5.

2. Theoretical Foundation

Recently, a new method, called Border Identification in Two Stages, denoted as BI_2, was proposed for full border identification by avoiding the limitation of the traditional BI, which only discusses partial borders [12]. We give definitions and a formal description of BI_2 and PBS in the following sections followed by a review of MCMC.

2.1. Formal Definitions

Several functions are described as follows.

1NN(p): nearest neighbor function, which returns the nearest neighbor of a data point p among all data points (the entire domain);

1NN(p, D): extended nearest neighbor function, which returns the nearest neighbor or the *informative data point* of a data point p in the given domain D;

l(p)*:* label function, which returns the label of the given data point p;

C(p): a set of data points with the same category as p, denoted as C_p.

Redundant data points can be defined as follows.

Definition 1. Given a labeled dataset D and its subset $B \subseteq D$, any point $p \in D - B$ is a *redundant data point* with respect to B if $p' = 1NN(p, B)$ and $l(p) = l(p')$. A set of redundant points R with respect to B, denoted as

$R(B, D) = \{p \mid \forall p \in D - B, \exists p' \in B, p' = 1NN(p, B) \text{ and } l(p) = l(p')\}$, denoted as R(B), without any confusion.

Definition 2. Given a labeled dataset D, the *full border* B of D can be defined recursively as follows:

1) $B = B \cup B_n$, where $B_n = \{q \mid \forall p \in D, \exists q \in D, q = 1NN(p, C_q) \text{ and } l(p) \neq l(q)\}$, called *near border*.
2) $B = B \cup B_f$, where $B_f = \{q \mid \forall p \in B, \exists q \in D, q = 1NN(p, C_q - B - R(B))\}$ and $l(p) \neq l(q)$, called *far border*.

According to these definitions, we can show that a redundant data point with respect to the full border B is always near data points of the same category and far from data points of different categories. The related proofs are omitted due to space limitation.

2.2. Progressive Border Sampling (PBS)

The BI_2 is used for identifying a full border [12], i.e., in the first stage, the BI_2 identifies the near border between any two categories. In the second stage, the BI_2 will iteratively identify new far borders in the two categories. For example, a simple XOR function can be visualized by 4 labeled data points in 2D. The BI_2 can identify two near border points and two far border points from the XOR domain while the depth of the recursion for far border points is 1. Empirically, the maximum depth of the recursion is shown with a bound ($<< n$) in many practical applications [12] (see §3.2.3).

Because a full border identified by the BI_2 is insufficient for statistical learning, Progressive Border Sampling (PBS) [12] has been proposed to progressively learn an augmented full border in the pairwise strategy [7][21] for multiclass domains such that the resulting border points can be used for training in supervised learning tasks [12].

Clearly, referring to previous research [12], we emphasize that PBS can be equivalent to the BI_2 only for full border identification by ignoring convergence detection for an augmented full border, and it can be regarded as a *forward selection* for border sampling.

However, this quadratic algorithm is infeasible for border sampling on large datasets. In this paper, we use the PBS as an *oracle* for border sampling on large datasets by adopting the CMCMC.

2.3. Markov Chain Monte Carlo

The standard Markov Chain Monte Carlo (MCMC) is a sampling technique such that selecting sample $x^{(i+1)}$ only depends on sample $x^{(i)}$, where the superscript i is a nonnegative integer, and the chain is expected to converge to a stationary distribution π with two properties: irreducible and aperiodic.

The initial convergence time is called the burn-in time, which measures how quickly much a Markov chain takes to eliminate the bias of the starting point $x^{(0)}$. The mixing rate measures how fast a Markov chain converges. Ideally, the stationary distribution of a good chain is reached quickly starting from an arbitrary position, i.e., low burn-in time and mixing rate.

Given a target distribution, we can heuristically design a transition matrix, e.g., a Markov chain transition graph for webpages and links [1], for guiding the evolution of a chain only if the transition matrix follows the two properties. Essentially, we are required to design a function or algorithm to establish the transition matrix for the evolution of a chain.

Besides those characteristics defined in the standard MCMC, the Coupling From The Past (CFTP) is an exact sampling technique, which consists of the following three main components [14]: oracle, which is a random map procedure which generates a subsample from the original population; the composition of maps, which can be used to simulate the flow for many time-steps; the convergence detection, which is used for ascertaining whether total coalescence has occurred.

The oracle can be used to produce maps f_{-1}, f_{-2}, f_{-3}…, f_{-N}, where N is how far we have to go into the past, and is determined at run-time. We can define a composite map by $F_{-N}^{0} \stackrel{\text{def}}{=} f_{-1} \circ f_{-2} \circ f_{-3} \circ ... \circ f_{-N}$, and the composite map must bring in *collapsing* with respect to some N.

3. Border Sampling Through CMCMC

A *naïve* strategy for the scalability of the oracle, i.e., the PBS, on large datasets can be depicted as follows. Given a large training set D and the specified size N of a partition, called the *sampling window*, we can obtain M subsamples, S_i, i = 1,…, M, where M = |D| / N, by stratified sampling. The PBS can be executed as BI_2 to identify each local full border B_i on each subsample S_i, and the resulting border is given by $B = \bigcup_{i=1}^{M} B_i$. Standard stratified sampling techniques are used for reducing the variance in estimation of the Monte Carlo analysis. The MCMC technique iteratively produces successive samples containing border points from the previously identified borders. The successive samples evolve from a large population with many redundant data points to a small population with few redundant data points. As a result, the MCMC can be regarded to as a *backward elimination* method for border sampling.

3.1. Coupling MCMC

The naïve strategy for the MCMC, as described above, is insufficient to converge to the stationary distribution π because we cannot guarantee the monotonicity of the states, i.e., $P(B^{(i)}) \leq P(B^{(i+1)})$, where i represents the i^{th} state B of the Markov Chain.

Given a labeled dataset D, we can obtain the composite B containing border points identified by the oracle from subsamples defined by a specified sampling window in the naïve strategy. B does not contain sufficient border points while D′ = D – B does not purely contain redundant points. The naïve strategy can be used again on D′ for new border points. The B can be augmented by adding the new border points while D′ is reduced by removing the new border points identified from it. As a result, the iterative procedure can cause D′ *collapsing* to some star convex graph with possible fewer border points. For example, a simple XOR domain is thought to satisfy the collapsing condition for a star convex graph with 4 data points.

Heuristically, the collapsing condition can be defined by c(c – 1) + 2 by assuming that each pair of classes has at least two border points and a simple XOR structure, where c is the number of classes.

Therefore, we construct two Markov Chains: the sequence of B for the sets of border points, called the *B chain*, and the sequence of D′ for the sets of redundant points, called the *R chain*. The iterative procedure can be regarded to as a Coupling MCMC (CMCMC) consisting of the B chain and multiple R chains.

For illustration, a complicated XOR binary domain [12] is shown in Fig. 1, where some states of the B chain and the R chains of the CMCMC generated by the CMCMC-PBS algorithm (see §3.2) are depicted.

Ordering Fig. 1 from top to bottom and from left to right by *a*, *b*, *c*, *d*, *e*, *f*, *g*, and *h*, we have (*a*) a complicated XOR binary domain with 640 data points, the sampling window for stratified sampling is set to 100. The original dataset is the beginning state of the B chain, and it becomes the beginning state of the first R

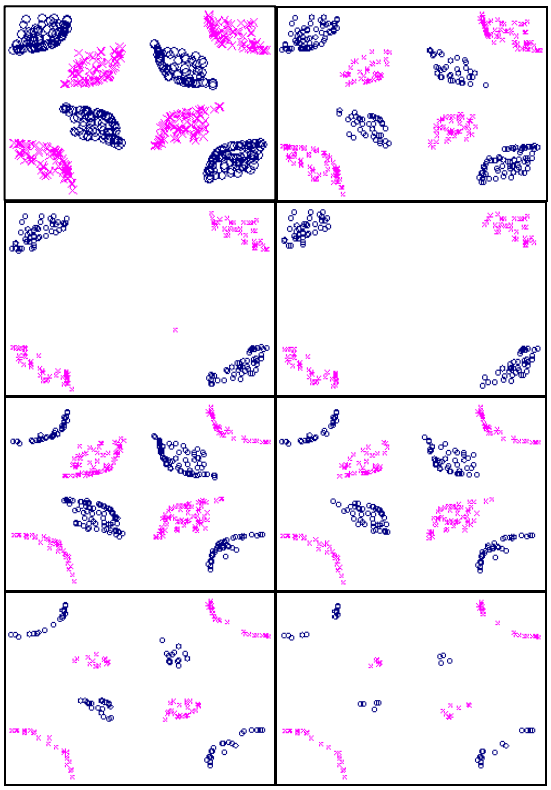

Fig. 1. A B chain and two successive R chains of the CMCMC from *a* to *h*.

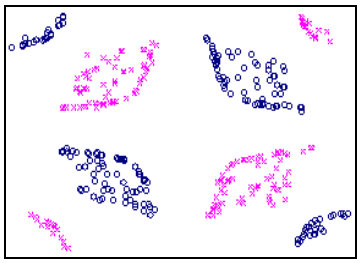

Fig. 2. The third state of the B chain obtained by removing data points in *e* from *h* in Fig. 1.

chain; (*b*) a state of the first R chain; (*c*) the state of the first R chain ahead of the collapsing state; (*d*) the collapsing state, which is nearly a star convex graph with many redundant data points and fewer border points; the ending state through 7 states of the first R chain; (*e*) the new (second) state of the B chain, which has 485 data points fewer than the original 640 data points. It was obtained by removing redundancy in (*d*) from the first state of the first R, i.e., (*a*), and becomes the beginning state of the second R chain; (*f*) a state of the second R chain; (*g*) the state of the second R chain ahead of the next collapsing state; (*h*) collapsing due to the presence of fewer redundant data points fitting in the sample window; the ending state through 4 states of the second R chain. Correspondingly, the third state of the B chain is obtained by retaining border points after redundant data points identified in (*h*) are removed from (*e*), as shown in Fig. 2, which is coalescent to *e*, and leads to the occurrence of convergence. The resulting sample in *e* is returned through 3 states in B chain and 11 states in two R chains of the CMCMC.

3.2. CMCMC-PBS Algorithm

According to the above discussion, we first propose a new method, called Coupling Markov Chain Monte Carlo-based PBS (CMCMC-PBS), see below, for border sampling on large datasets. Given the two inputs of the algorithm, the training set D and the sampling window W, it returns the result in Border. The algorithm produces the B chain in the while loop from Step 2 to Step 6 while the R chain is generated in the while loop from Step 11 to 21.

A linear machine, Naïve Bayes (NB), is used as a base learner for convergence detection of the B chain at Steps 4 and 5. NB has been successfully used for progressive learning for convergence detection [9][12], i.e., ValidateNBModel() is used for building a NB classifier for estimating the current sample D′, and the downside or the beginning points of the plateau (not fringe) of the generated adaptive learning curve saved in LearningCurve is used as a convergent point.

The Coupling procedure generates a R chain in the while loop beginning at Step 11 while initially the condition of the forward selection of the PBS is defined at Step 10 (see §3.2.4). The floor function is used for specifying a sampling window at Step 12, and the stratified sampling technique helps reduce the variance of MCMC at Step 13. The PBS is used as an oracle for identifying local full borders at Step 17, and the algorithm tests collapsing at Step 19 according to the collapsing condition. Initially, cg is false at Step 9. It leads to the oracle performing as BI_2 only for full border identification at Step 17. Because S will be shrunken at Step 18, when S is fitted in the sampling window W, PBS searches for an augmented full border in forward selection. The while loop exits at Step 20 if the conditions are met. The result is returned at Step 22 by removing redundancies in S′ from the input D.

3.2.1. Relation to CFTP. We propose CMCMC by adapting CFTP to border sampling. PBS is suggested as our oracle, and a state in the B chain corresponds to a composition map defined by those states in the related R chain. Essentially, the oracle establishes the transition matrices related to the B chain and the R chain for their evolutions. In a sense, these transition matrices obey the laws of irreducibility and aperiodicity. Collapsing can be observed by approximately testing the occurrence of some star

convex graph from the states of the R chain. The NB is assumed for convergence detection for the B chain.

3.2.2. Similarity measures. Generally, any distance metric or similarity measure can be used in the oracle of the CMCMC-PBS for searching for border points, e.g., Radial-Based Function (RBF), Cosine, Euclidean distance or normalized Euclidean distance, Pearson Coefficient, Mahalanobis distance, and Extended Jaccard similarity [18], etc. However, different effects have been observed in the oracle, e.g., RBF is bias to the class contour while Cosine is bias to the class core [12]. Instead of developing an ideal similarity, we empirically show that RBF, e.g., in Fig. 1, has an asymptotical effect for Monte Carlo integration in CMCMC-PBS in most cases.

CMCMC-PBS algorithm
Input D, W
Output Border
begin
1 Border = ∅, i = 0..K, D′ = D, LearningCurve[0] = 0;
2 **while**(**true**)
3 D′ = **Coupling**(D′, W)
4 LearningCurve[*i*+1] = **ValidateNBModel**(D′, D)
5 **if**(LearningCurve[i+1] ≤ LearningCurve[i]) **break**;
6 i++
7 Border = D′
8 **return** Border
end
Proceduce Coupling(D, W**)**
9 c = Number of class in D, cg = false, D′ = D
10 **if**(c ≤ 5) cg′ = **true else** cg′ = **false**
11 **while**(**true**)
12 B = ∅, N = ⌊|D′| / W⌋
13 S = **StratifiedSampling**(D′, W), |S| = N
14 **if**(N = 1) cg = cg′**;**
15 S′ = ∅, collapsing = **true**
16 **for**(k = 0; k < N; k++)
17 B_k = **PBS**(S(k), cg)
18 S′ = S′ ∪ (S(k) – B_k)
19 **if**(|B_k| > c(c – 1) + 2)) collapsing = **false**
20 **if**(collapsing ∨ cg) **break**;
21 D′ = S′
22 **return** D – S′

3.2.3. Linear time complexity. Clearly, the space complexity of the CMCMC-PBS is a linear increase. We analyze its time complexity as follows. Considering the two while loops in the CMCMC-PBS, the time complexity can be first simply given by $O(T\times K\times N/W\times C_0)$, where T is the number of tries for convergence detection in the while loop beginning at Step 2; K is the number of iteration in the while loop beginning at Step 11 in Coupling; N is the size of a given training set D and W is the sampling window; C_0 is the time complexity of the oracle with the sampling window W, and $C_0 = O(T_0K_0FW^2)$, where F is the number of features; K_0 is the depth of the recursive far borders; T_0 is the number of tries for convergence detection [12]. Therefore, the time complexity of the CMCMC-PBS is given by

$$O(TT_0KK_0FWN + TFN) = O(TT_0KK_0FWN) \quad (1)$$

where the term O(TFN) [4][13] is the time complexity for learning a NB in the CMCMC-PBS. T_0 and K_0 are empirically analyzed in previous research by a bound with a small number («n) given a domain [12]. Especially, $T_0 = 1$ if the oracle runs as BI_2.

Because BI_2 assumes a pairwise or one against all strategy for border identification on multi-domains, K_0 has nothing to do with the number of classes.

As a result, the extended non-redundant XOR with 8 data points in 2D is constructed as the worst case, as shown in Fig. 3. The depth of the recursive far borders is 3. Further, we obtain an upper bound, i.e., $K_0 \le 2F - 1$, by a constructed XOR of dimension F. It is just equal to the size of the boundary of an F-cube minus one. Empirically, K_0 is much smaller than F [12].

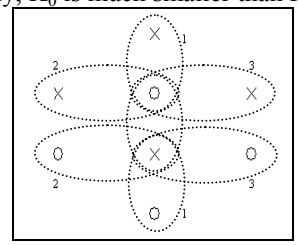

Fig. 3. The extended XOR with 8 data points in 2D.

On the other hand, Coupling searches for redundant points on the entire dataset by the oracle until a collapsing occurs. K is a little domain-related, e.g., redundancies, while it is related to collapsing test. But its small value («n) has been observed.

According to Eq. (1), the CMCMC-PBS is an efficient learning method in linear time complexity with respect to the sample size N for border sampling.

3.2.4. Convergence detections and collapsing. The NB is used for convergence detection in the B chain from Step 2 to Step 6 of the CMCMC-PBS algorithm. Empirically, it is not always effective to track the adaptive learning curve of this linear machine for convergence detection if random sampling is used. However, it has been shown that the effectiveness can be obtained precisely by border sampling [12].

If the oracle runs for forward selection with convergence detection by setting cg = true at Step 14, $T_0 > 1$. As a result, the CMCMC-PBS performs two convergence detections (T times for the CMCMC in the B chain and T_0 times in the oracle [12]). On the other hand, collapsing detection (K times for the R chain) is heuristically defined by assuming the occurrence of a star convex with less border points.

T and T_0 can be thought of as converging rates of linear functions approaching to the class boundary in the CMCMC-PBS. In some cases, the oracle's convergence detection can lead to the reduction of the CMCMC's convergence detection. We emphasize that the CMCMC's backward elimination is more efficient for convergence than the oracle's forward selection in multiclass domains.

Empirically, if a state in the R chain is fitted in the sampling window and the number of classes ≤ 5, the oracle performs convergence detection for progressive border sampling, e.g., (*h*) in Fig. 1.

3.3.5. Learning measures. The ROC curve is drawn for selecting an optimal classifier [5] and assessing the ranking in terms of separation of the classes by Area under ROC curve (AUC). The AUC has been used for evaluating the performance of classifiers on class imbalanced domains because it is more discriminate than other learning measures, e.g., accuracy, and is not sensitive to imbalance [5]. The AUC is suggested as a learning measure for border sampling. As a result, the adaptive learning curve of NB is an AUC curve.

3.3. Related work

This paper follows previous research on PBS [12], and attempts to address its scalability on large datasets. The original BI_2 for full border identification and the original PBS for an augmented full border [12] have been adapted according to the formalization in Def2.

Given a labeled dataset D, based on 1NN(.,.) and 1NN(.), respectively, $B_1 = \{q \mid \forall p \in D, \exists q \in D, q = 1NN(p, C_q)$ and $l(p) \neq l(q)\}$ and $B_2 = \{q \mid \forall p \in D, \exists q \in D, q = 1NN(p)$ and $l(p) \neq l(q)\}$ are not equivalent. As a result, 1NN(.) is subject to failure for defining a border while 1NN(.,.) should be used to define the near border. This observation is used for explaining the main difference between the border sampling techniques shown in the BI_2 and the techniques for the reduction of training sets in those algorithms by the nearest neighbor editing rule [23], whose purpose is the reduction of training sets for Instance-Based Learning. We emphasize that the reduction of training sets should be one of the tasks of border sampling.

Furthermore, because some learners, e.g., Naive Bayes and Decision Trees, etc, can be very fast at learning a good classifier even with a large training set of, say, ten thousand examples, reducing the sample size by simply selecting a small sample as per previous research is not expected to reduce the learning cost without loss of performance, e.g., active learning for sample selection [22]. On the other hand, we claim that our current research for border sampling in supervised learning can be easily migrated to active learning or semi-supervised learning.

Another related work is incremental learning, which can be regarded as a learning method that builds a theory from examples available over time [6][19]0. For example, incremental SVM for online application has been studied in [11]. Clearly, the CMCMC-PBS can be easily used for incremental learning by suggesting and focusing on incremental sampling.

Assuming some optimal probability distribution, a Bayesian decision rule can define a Bayesian decision boundary by discriminating functions $g_i(x) = p(x|w_i)p(w_i)$ [4]. Our research suggests that the new method tends to learn optimal class conditional probability distributions $p(x|w_i)$ by border sampling such that the related prior probability distribution $p(w_i)$ and Bayesian decision boundary can be obtained.

Because the CMCMC-PBS has a nice bias towards border data points lying close to the class boundary, it can produce an optimal sample for training. Thus, it provides a promising treatment for the class imbalance problem as compared with previous techniques for under-sampling and oversampling [8][10].

4. Experiments

In this section, we discuss our experimental design and results as follows.

4.1. Datasets and settings

We conducted experiments on 33 datasets including one obtained from a nuclear security application and 32 chosen from the UCIKDD repository [2]. For the application, a possible method of explosion detection for the Comprehensive nuclear-Test-Ban-Treaty [20] 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 by radionuclide monitoring. Several samples are synthesized under different circumstances of nuclear explosions, and combined with various levels of normal concentration backgrounds so as to synthesize a training dataset, called Explosion, for use with machine learning methods.

The characteristics of these datasets are described in Table 1, where the columns are the names of the datasets, the number of attributes (#attr), the number of instances (#ins), the number of classes (#c), the number of data points selected from training sets by CMCMC-PBS (#CPBS), the percent (%) of data selected by CMCMC-PBS over the training sets, the average number of trials (T) in CMCMC-PBS for convergence

detection in the B chain, the average number of iterations (K) for the collapsing test in the R chain.

In our experiments, CMCMC-PBS with the RBF similarity measure selects samples from the training sets with a specified sampling window. Several inductive algorithms are used for training classifiers on either the resulting samples generated by CMCMC-PBS, or the full training sets (Full), or those generated by previous approaches, i.e., static (Static), arithmetic (Arith), and geometric PS with LRLS (Geo) [9][15]. The performances of these classifiers with respect to the AUC are used for evaluation between the CMCMC-PBS and the other algorithms.

To test the performance of the CMCMC-PBS on either small datasets or large datasets with different sampling windows, these datasets are divided into three groups. The sizes of the sampling window for the datasets in the first, second, and third group are set to 10, 100, and 1000, respectively.

We selected the four learners: Naïve Bayes (NB), Decision Tree (DT, i.e., J48), Support Vector Machine (SVM, i.e., SMO [16]), and IB1 for Instance-Based Learning (IBL) from the Weka data mining package [25]. They have been widely used for many practical applications. The classifiers are built with their default settings, e.g., NB with Gaussian estimator, DT with no reduced error pruning and no C4.5 pruning [17] and no Laplace smoothing, SVM with polynomial of 1 for kernel function and constant C of 1 for soft margins, and IB1 with normalized Euclidean distance for IBL.

Table 1. The characteristics of 33 datasets.

datasets	#attr	#ins	#c	#CPBS	%	T	K
Anneal	39	898	5	418	51.72	3	2
Audiology	70	226	24	183	89.97	3	1
Autos	26	205	6	170	92.14	3	1
Balance-s	5	625	3	540	96.00	3	6
Breast-w	10	699	2	173	27.50	2	3
Colic	23	368	2	245	73.97	5	4
Credit-a	16	690	2	543	87.44	3	5
Diabetes	9	768	2	531	76.82	2	4
Glass	10	214	6	166	86.19	2	1
Heart-s	14	270	2	219	90.12	3	5
Hepatitis	20	155	2	66	47.31	3	3
Ionosphere	35	351	2	222	70.28	2	4
Iris	5	150	3	62	45.93	2	1
Labor	17	57	2	40	77.97	2	2
Lymph	19	148	4	117	87.84	4	3
P-tumor	18	339	21	295	96.69	3	1
Sonar	61	208	2	160	85.47	5	3
Soybean	36	683	18	603	98.10	3	1
Vehicle	19	846	4	690	90.62	3	1
Vote	17	435	2	162	41.38	16	5
Vowel	14	990	11	891	100.00	2	1
Zoo	18	101	7	75	82.51	2	1
Hypothyroid	30	3772	4	558	16.44	4	3
kr-vs-kp	37	3196	2	2434	84.62	10	6
Segment	20	2310	7	1883	90.57	3	5
Sick	30	3772	2	528	15.54	3	4
Splice	62	3190	3	2847	99.16	6	3
Letter	17	20000	26	16627	92.37	13	3
Mushroom	23	8124	2	3440	47.05	3	12
Waveform	41	5000	3	4256	94.58	22	7
Adult	15	48842	2	24665	56.11	2	13
Shuttle	10	58000	7	18254	34.97	2	70
Explosion	5	92630	2	620	0.74	2	24

4.2. Experimental Results

As shown in Table 1, our initial results show that CMCMC-PBS can select a small sample from the original training set after redundancies are removed, e.g., samples with only 15.54 and 0.74 percents of the original training sets are selected in the Sick and Explosion datasets, respectively, while it can retain most instances in the original training sets if little redundancies can be found, e.g., on Vowel and Splice. The average number of trials T can be 2 in Shuttle or 22 in Waveform. The average number of iterations K for the collapsing test in Coupling can be 1 in Audiology or 70 in Shuttle. The Ks are much smaller than the #ins, though.

Further, we show the efficiency and effectiveness of CMCMC-PBS for sample selection on large datasets, e.g., Adult and Shuttle, by comparing CMCMC-PBS with Arith, Geo, and Full for NB and DT while SVM and IB1 are ignored due to their intractability on large datasets. In more detail, we employed 10 fold cross validation. Both the average elapsed time and the AUC over the 10 runs on the large datasets are computed. Static is executed by resampling with replacement using the same sample size as that of the resulting sample identified by CMCMC-PBS from the training sets with the same class distribution. Arith and Geo, on the other hand, are executed according to their specified schedules on each run, and the curves of the elapsed times and the AUCs obtained on 10 runs are averaged.

For example, we compare CMCMC-PBS with Arith by NB and DT on Adult, as shown in Fig. 4. Arith has a higher sampling cost than CMCMC-PBS (at 24665) after the queried sample size by NB or DT exceeds 6300 or 9400, respectively. No matter how a sample is

queried, however, Arith degrades the performance of DT as compared with CMCMC-PBS since the AUC of CMCMC-PBS is Arith's upper bound. On the other hand, Arith can approximately obtain the same performance with NB as CMCMC-PBS by selecting a small sample, i.e., 3200, in less time.

We also compared CMCMC-PBS with Geo. Geo can sample data efficiently with unavoidable failures in selecting an optimal sample as compared to CMCMC-PBS. We omit the details due to space limitation.

Instead, we summarize the results obtained by CMCMC-PBS (CPBS), Full, and Static for NB and DT on averages of the results of 20 runs on the datasets in the second group, and 10 runs on the datasets in the third group. These results are listed in Table 2, where 'w' and 'l' represent the corresponding methods in terms of the paired t-test and the Wilcoxon signed rank test at significance levels of 0.05 the CMCMC-PBS wins and losses, respectively, and two statistical test results are possible pairs separated by a comma. We can see that CMCMC-PBS consistently outperforms Static for NB and DT, and outperforms Full for NB. It is very competitive with Full for DT in terms of the resulting AUC, the paired t-test, and the Wilcoxon signed rank test except in the case of Adult for DT.

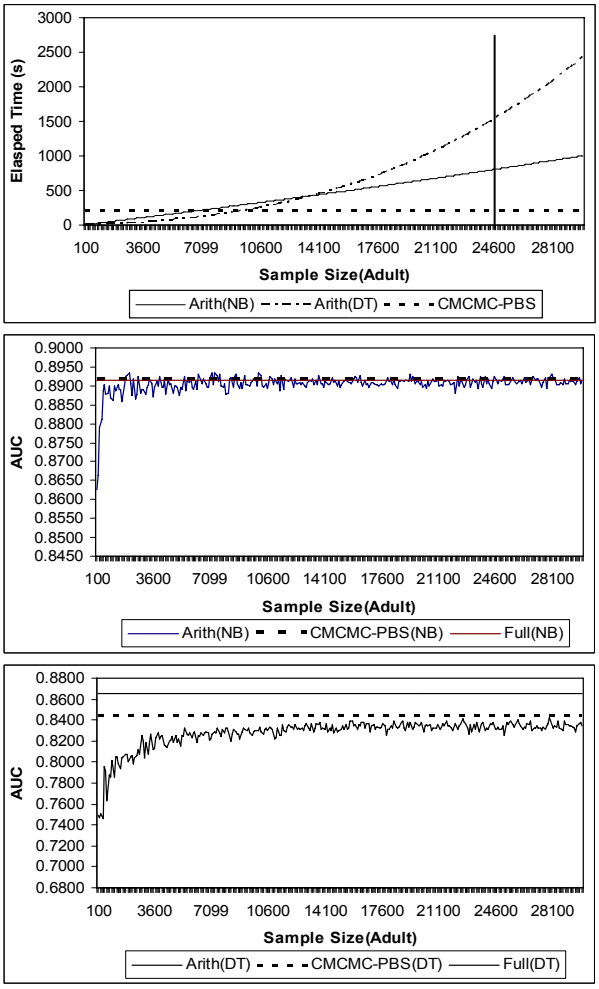

Fig. 4. The comparison between the CMCMC-PBS and Arith on Adult in the third group for NB and DT with respect to the elapsed time and the AUC.

The same results with respect to AUC, the paired t-test, and Wilcoxon signed rank test are also obtained by averaging of 20 runs on the datasets in the first group, as shown in Table 3. Moreover, the average AUCs are shown at the bottom of Tables 2 or 3, respectively. According to these results, CMCMC-PBS (CPBS) outperforms Static by overall upgrading the performance of all selected classifiers, and even outperforms Full by upgrading the performance of NB and SVM and by reducing the training set size without degrading the performance of either DT on the datasets in the first and second group or IB1 on the datasets in the first group.

There are some exceptions, however. For example, CMCMC-PBS degrades the performance of DT on Adult as compared with Full, and degrades the performance of NB on Ionosphere as compared with Full and Static. This suggests that the proposed algorithm suffers a failure on these domains.

In addition, on Vowel, due to little redundancy the CMCMC-PBS still wins the Static, which performs resampling with 100% of the training set with the replacement and same class distribution. The Explosion is a synthesized domain. The experimental results on Explosion in Table 2 reveal that CMCMC-PBS is superior to Static with respect to the AUC of NB and DT while it is competitive with Full with respect to the AUC of NB and DT. In all cases, it requires quite a small sample for training.

Table 2. The comparison between the CMCMC-PBS and Full, Static for NB and DT with respect to AUC on the datasets in the first and second groups.

Datasets	NB			DT		
	CPBS	Full	Static	CPBS	Full	Static
Hypothyroid	.9378	.9399	.9267	.945	.9623	.9521
kr-vs-kp	.9812	$.9521^{w,w}$	$.9482^{w,w}$	.9987	.9983	$.9925^{w,w}$
Letter	.9572	$.9552^{w,w}$	$.9546^{w,w}$	.9498	.9509	$.9336^{w,w}$
Mushroom	.9994	$.9981^{w,w}$	$.9973^{w,w}$	.9999	1	1
Segment	.9779	.9779	.9764w	.9836	.9836	.9809
Sick	.922	.9271	.9238	.967	.9525	$.9137^{w,w}$
Splice	.9947	$.9944^{,w}$	$.9939^{w,w}$	.9515	.9531	.9444
Waveform	.9619	$.9567^{w}$	$.9551^{w}$	.828	.8255	$.8156^{w}$
Adult	.8915	.8914	.8916	.849	$.8649^{l,l}$	$.8307^{w,w}$
Shuttle	.9895	$.9782^{w,w}$	$.9383^{w,w}$	.9804	.9798	$.9322^{,w}$
Explosion	$.5427^{w,w}$	.4172	.5272	.68	.6953	$.5^{w,w}$
Average	**.9233**	.9080	.9121	.9212	.9242	.8905

We repeated our experiments with incremental window sizes on the same datasets. The increments for sample selection by CMCMC-PBS on the datasets in the first, second, and third groups are set to 10, 100, and 100, respectively. We obtained 9 additional results related to the AUC, the resulting sample size, and the number of iterations K during the Coupling. For example, as shown in Fig. 5, we plotted the curves of the AUC as a function of window size for DT on Anneal and Shuttle. We observed only negligible impacts of the sampling windows on AUC on these two datasets while no evident result shows any significant effect of the sampling window on the performance of CMCMC-PBS in other cases. Similarly, we observed an insignificant effect of the sampling window on K. The related results are omitted due to space limitation.

5. Conclusion and Future Work

This paper discusses the scalability of the previously proposed PBS algorithm for border sampling on large labeled datasets. This scalability is achieved by a novel method incorporating PBS with the CMCMC technique. The CMCMC-PBS algorithm is proposed for border sampling on either small datasets or large datasets. It can efficiently and effectively converge to an exact sample by learning on many subsamples in linear time complexity. For example, CMCMC-PBS speeds up PBS by 60% and 40% for border sampling and helps improve the performance of NB on Letter and Splice, respectively.

We conducted experiments on 33 datasets and showed that CMCMC-PBS consistently outperforms three earlier methods, Static, Arith, and Geo for sample selection although it needs a little more time expenses only by a little bit than Arith and Geo. It helps train NB in most cases as compared with Full while it is consistent with Full for training SVM, DT and IB1 in all cases. No evident result shows an impact of sampling window on the resulting samples for training. Therefore, CMCMC-PBS is efficient and effective for sample selection in many practical applications.

Two exceptions in our experiments show that CMCMC-PBS needs further improvement in the direction of performing a more precise geometric computation for collapsing test than the current method for collapsing. These are issues for either reducing the learning cost of active learning [3][22] or for dealing with the class imbalance problem [5][10] by using CMCMC-PBS. These are natural focuses in our future.

References

[1] C. Andrieu, N. D. Freitas, A. Doucet, M. I. Jordan. An Introduction to MCMC for Machine Learning. Machine Learning, 50, 5–43, 2003. Kluwer Academic Publishers. Manufactured in The Netherlands.

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

[3] D. Cohn, Z. Ghahramani, and M. Jordan. Active learning with statistical models. Journal of Artificial Intelligence Research 4: 129-145, 1996.

[4] R.O. Duda and P.E. Hart. Pattern Classification and Scene Analysis. A Wiley Intersience Publication, 1973.

[5] T. Fawcett. ROC graphs: Notes and practical considerations for researchers. http://www.hpl.hp.com/-personal/TomFawcett/papers/ index.html, 2003.

[6] C. Giraud-Carrier. A Note on the Utility of Incremental Learning. AI Communications. Volume 13, Issue 4 (January 2000). Pages: 215 – 223. ISSN:0921-7126.

[7] T. Hastie and R. Tibshirani. Classification by pairwise coupling. The Annals of Statistics, 26(1):451–471, 1998.

[8] N. Japkowicz and S. Stephen. The Class Imbalance Problem: A Systematic Study. Intelligent Data Analysis Journal, Volume 6, Number 5, November 2002.

[9] G. John and P. Langley. Static versus dynamic sampling for data mining. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (1996), AAAI Press, pp. 367-370.

[10] M. Kubat, S. Matwin. Addressing the Curse of Imbalanced Training Sets: One-Sided Selection (1997). *Proc. 14th International Conference on Machine Learning*, pp. 179-186, 1997.

[11] P. Laskov, C. Gehl, S. Kr¨uger, K. M¨uller. Incremental Support Vector Learning: Analysis, Implementation and Applications. Journal of Machine Learning Research 7 (2006) 1909– 1936.

[12] G. Li, N. Japkowicz, T. J. Stocki, and R. K. Ungar. Full Border Identification for Reduction of Training Sets. In *Proceedings of the 21st Canadian Conference in Artificial Intelligence* (AI'2008), S.Bergler (Ed.) Canadian AI 2008, LNAI 5032, pp.203-215, 2008.

[13] T. Mitchell. *Machine Learning*. McGraw-Hill Companies, Inc, 1997.

[14] J. G. Propp and D. B. Wilson. Coupling from the past: a user's guide. Aldous, D. and Propp, J. G., editors, Microsurveys in Discrete Probability, vol. 41 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Amer. Math. Soc., Providence, RI, 1998, pp. 181-192.

[15] F. Provost, D. Jensen, and T. Oates. Efficient Progressive Sampling. In *Proceedings of KDD'99*, AAAI/MIT Press, 1999.

[16] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in kernel methods - support vector learning*. MIT Press, 1998.

[17] J. R. Quinlan: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993).

[18] A. Strehl and J. Ghosh. Value-based customer grouping

Table 3. The comparison between the CMCMC-PBS, Full and Static on the datasets in the first group for NB, DT, SVM, and IB1 with respect to the AUC, the paired t-test, and the Wilcoxon signed rank test.

	NB			DT			SVM			IB1		
Datasets	CPBS	Full	Static	CPBS	Full	Static	CPBS	Full	Static	CPBS	Full	Static
Anneal	0.9597	0.9599$^{\cdot l}$	0.9514$^{\cdot w}$	0.8292	0.819	0.7899$^{\cdot w}$	0.8484	0.8408$^{\cdot w}$	0.8371$^{\cdot w}$	0.8213	0.7975^{w,w}	0.8216
Audiology	0.7024	0.7017$^{\cdot w}$	0.6987$^{\cdot w}$	0.6212	0.6196	0.6141	0.6436	0.6433	0.6243^{w,w}	0.5993	0.6039	0.594
Autos	0.7225	0.7251	0.7025	0.735	0.7369	0.7159	0.7724	0.7733	0.7619	0.6933	0.6942	0.6618^{w,w}
Balance-s	0.8738	0.8789^{l}	0.8445	0.6852	0.6721	0.7121	0.6648	0.6633	0.6642	0.675	0.6969	0.6897
Breast-w	0.9903	0.9879	0.9892	0.9421	0.9483	0.9372	0.963	0.9635	0.9625	0.9417	0.9485	0.9593$^{\cdot l}$
Colic	0.8532	0.8372^{w}	0.8342	0.8487	0.8533	0.8171	0.7951	0.8095	0.7669	0.7673	0.7795	0.7518
Credit-a	0.8997	0.8982	0.8959	0.8294	0.8479	0.8427	0.8599	0.8572	0.8504	0.7998	0.8075	0.7986
Diabetes	0.8168	0.8174	0.8135	0.7574	0.7697	0.6868^{w,w}	0.7131	0.7114	0.7143	0.6694	0.6637	0.6509
Glass	0.8111	0.8116	0.8083	0.7959	0.7924	0.7253^{w,w}	0.7208	0.7305	0.7272	0.7331	0.7353	0.7302
Heart-s	0.8972	0.8981	0.8931	0.7947	0.759	0.7486	0.835	0.8313	0.8183	0.7588	0.7596	0.77
Hepatitis	0.8878	0.8797	0.8465	0.7276	0.7176	0.7223	0.7705	0.7474	0.7159	0.6671	0.658	0.6712
Ionosphere	0.9159	0.9390^{l}	0.9360^{l}	0.8803	0.8902	0.8664	0.8421	0.8464	0.8238	0.8311	0.8246	0.7800^{w,w}
Iris	0.99	0.9893	0.9907	0.9667	0.9713	0.9647	0.96	0.9833	0.965	0.9667	0.9783	0.975
Labor	0.9646	0.9771	0.9771	0.8125	0.7854	0.8354	0.8792	0.8917	0.7958$^{\cdot w}$	0.8417	0.8479	0.8417
Lymph	0.8921	0.8922	0.8818	0.7303	0.7083	0.7064	0.8128	0.8105	0.7893	0.6927	0.6884	0.6697$^{\cdot w}$
P-tumor	0.7613	0.7613	0.7581	0.6452	0.6469	0.6233$^{\cdot w}$	0.715	0.7132	0.7077	0.5827	0.587	0.591
Sonar	0.8484	0.7984$^{\cdot w}$	0.7862$^{\cdot w}$	0.7325	0.7631	0.6994	0.801	0.7721	0.7501$^{\cdot w}$	0.8692	0.8595	0.8130$^{\cdot w}$
Soybean	0.9983	0.9983	0.9982	0.9743	0.9722	0.9525^{w,w}	0.988	0.9881	0.9876	0.9674	0.968	0.9648
Vehicle	0.7498	0.7462	0.7393$^{\cdot w}$	0.7933	0.813	0.7663	0.8306	0.833	0.8243	0.7419	0.7404	0.7265
Vote	0.9887	0.974^{w,w}	0.9741^{w,w}	0.9745	0.9785	0.9577$^{\cdot w}$	0.9571	0.9567	0.955	0.8938	0.9226	0.917
Vowel	0.9547	0.9547	0.9351^{w,w}	0.9269	0.9269	0.8928^{w,w}	0.9484	0.9483	0.9320^{w,w}	0.9956	0.9956	0.9766^{w,w}
Zoo	0.8917	0.8917	0.8821	0.7976	0.7976	0.7917	0.8048	0.8048	0.8095	0.8024	0.8024	0.803
Average	**0.8805**	0.8781	0.8698	**0.8091**	0.8086	0.7895	**0.8239**	0.8236	0.8083	0.7869	0.7891	0.7799

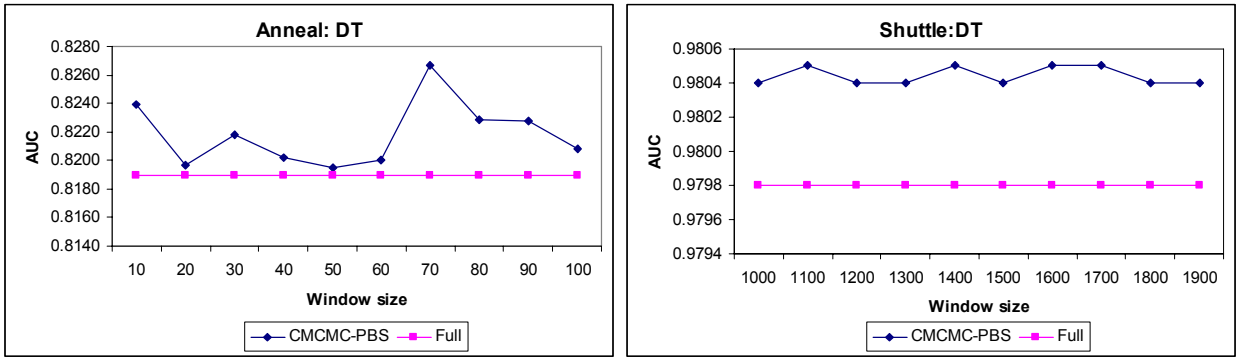

Fig. 5. The effect of window size on the performance of CMCMC-PBS.

from large retail data-sets. In *Proc. SPIE Conference on Data Mining and Knowledge Discovery, Orlando,* volume 4057, pages 33-42. SPIE, April 2000.

[19] M. J. A. Strens. Evolutionary MCMC sampling and optimization in discrete spaces. In *Proceedings of the 20th International Conference on Machine Learning* ICML-2003, 2003.

[20] J. D. Sullivan. The comprehensive test ban treaty. Physics Today 151, 23. 1998.

[21] J. Sulzmann, J. Fürnkranz, and E. Hüllermeier. On Pairwise Naive Bayes Classifiers. In *Proceedings of the 18th European Conference on Machine Learning*. (ECML-07), pp. 658-665, Warsawa, Poland, 2007. Springer-Verlag.

[22] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. Journal of Machine Learning Research, 2:45–66, 2001.

[23] D. R. Wilson and T. R. Martinez. Reduction Techniques for Instance-Based Learning Algorithms. Machine Learning, Kluwer Academic Publishers. Printed in The Netherlands, 38:257–286, 2000.

[24] G. M. Weiss and F. Provost. Learning when training data are costly: the effect of class distribution on tree induction. Journal of Artificial Intelligence Research 19 (2003) 315-354.

[25] WEKA Software, v3.5.2. University of Waikato. http://-www.cs.waikato.ac.nz/ml/weka/index_datasets.html