Bootstrap Sampling Based Data Cleaning and Maximum Entropy SVMs for Large Datasets

Conference Paper · November 2012

DOI: 10.1109/ICTAI.2012.164

3 authors, including:

Senzhang Wang
Nanjing University of Aeronautics & Astronautics
107 PUBLICATIONS **773** CITATIONS
SEE PROFILE

Xiaoming Zhang
Beihang University (BUAA)
63 PUBLICATIONS **479** CITATIONS
SEE PROFILE

2012 IEEE 24th International Conference on Tools with Artificial Intelligence

Bootstrap Sampling based Data Cleaning and Maximum Entropy SVMs for large Datasets

Senzhang Wang, Zhoujun Li* and Xiaoming Zhang
State Key Laboratory of Software Development Environment
Beijing Key Laboratory of Network Technology
Beihang University, Beijing, China
Email: wangsenzhang@126.com, {lizj@,yolixs@cse.}buaa.edu.cn

***Abstract*—Support Vector Machines (SVMs) is a popular machine learning algorithm based on Statistical Learning Theory (SLT). However, traditional solutions suffer from $O(n^2)$ time complexity. In this paper, a novel two-stage informative pattern abstraction algorithm is proposed. The first stage of the algorithm is data cleaning based on bootstrap sampling. A bundle of weak SVM classifiers are trained based on the sampled small datasets. Training data correctly classified by all the weak classifiers are cleaned. In the second stage, to further improve performance of final classifier and reduce training time, two novel informative pattern extraction algorithms based on entropy maximization SVMs are proposed. Empirical studies show our approach is effective in reducing size of training datasets and the computational cost, outperforming the state-of-the-art SVM training algorithms PEGASOS, RSVM and LIBLINEAR SVM with comparable classification accuracy.**

***Keywords*—SVMs, bootstrap sampling, entropy maximization**

I. Introduction

Support Vector Machines (SVMs) is a popular and effective classification and regression algorithm which is widely used in many fields like data mining [1], information retrieval [2] and social network [3]. However, traditional SVMs solution need $O(n^2)$ time complexity. Though many effective algorithms are proposed, SVMs suffer a lot from its high time complexity when dealing with large datasets.

[4] showed that the final SVMs classifier only depends on a few samples which are called support vectors, and removing non-support vectors will not significantly affect performance of classifier. [5] only selected the instances close to the boundary between classes and instances likely to be non-support vectors are eliminated. Similarly, a problem addressed in this paper is: *Can a minimal subset of the training datasets be found and the generalization performance of the SVMs trained on the subset is comparable with the whole dataset?* To address this problem, a novel informative pattern abstraction algorithm which consists of two stages: *data cleaning based on bootstrap sampling* and *informative pattern extraction* is proposed.

In the data cleaning stage, training data are sampled using the bootstrap sampling method from initial training datasets, and weak SVMs classifiers used to pre-classify training datasets are trained on sampled small datasets. Data correctly classified by all the weak classifiers will be eliminated and only the data misclassified by at least one weak classifier will be retained to the next stage. This will be described in section III. In the informative pattern extraction stage, to further extract the most controversial data points, two novel informative pattern extraction algorithms based on information entropy maximization are proposed. The information entropy of each data from the first stage is calculated according to its empirical misclassification probability. This will be described in section IV.

Contributions of this paper are as follows:

- A novel bootstrap sampling based data cleaning method is proposed. Compared with other data-processing methods like concept boundary detection method [5] and Cascade SVMs [6], our method is more intuitive and effective. Moreover, parallelization can be used in this stage.
- To further extract informative patterns and eliminate outliers, two maximum entropy algorithms are proposed. These algorithms can further extract informative patterns according to their probability of misclassification. Therefore, training time can be significantly reduced.

The rest of the paper is organized as follows: Section II reviews related work. Section III details the *bootstrap sampling based data cleaning* stage. Section IV details the *informative pattern extraction* process. Two training data extraction algorithms based on information entropy maximization are proposed. Section V presents experimental results comparing our approach with other approaches LIBSVM, EGASOS, RSVM and LIBLINEAR SVM. Section VI concludes this paper.

II. Related Work

Prior work related to speed up SVMs training can be categorized data-reduction level and quadratic programming (QP) algorithm level. The data-reduction level approaches mainly focus on how to reduce the training data quantity while not to loss much useful information. The QP-algorithms focus on make the QP solver faster [9,10,12,13]. Our approach falls in the data-reduction level methods, and hence we mainly discuss related work in this level.

Lee et al. [11] found that the nonlinear separating surface just depends on a small randomly selected portion of the dataset. The Reduced SVMs (RSVM) they proposed randomly selects a subset $\overline{n}$ of the entire dataset n to be training dataset.

*Corresponding author.

1082-3409/12 $26.00 © 2012 IEEE
DOI 10.1109/ICTAI.2012.164

Han Peter et al. [6] proposed a novel SVMs algorithm which can be executed in parallel: Cascade SVM. In this method, the dataset are split into subsets and optimized separately with multiple SVMs. The partial results are combined and filtered again in a 'Cascade SVMs', until the global optimum is reached. Navneet Panda et al. [5] proposed a concept boundary detection method to speed up SVMs. In this paper, training data which are likely to be non-support vectors are eliminated in the concept-independence preprocessing stage. And in the concept-specific sampling stage, they effectively select useful training data for each target concept. Yu et al. [8] uses a hierarchical micro-clustering technique to capture the training data that are close to the decision boundary. Lawrence et al. [7] proposed an Information Vector Machines (IVMs). A framework for sparse Gaussian process methods which uses forward selection with criteria based on information theoretic principles was proposed in this paper.

SVM-Perf [14] is an optimization method that uses a cutting planes approach for training linear SVMs. [14] showed that SVM-Perf can find a solution with accuracy ε in time $O(nd/(\lambda\varepsilon))$. The extended training SVMs with Kernels method in [15] represents the learned rule using arbitrary basis vectors, not just the support vectors from the training set. The algorithm's efficiency and effectiveness is characterized both theoretically and experimentally, especially on large datasets with sparse feature vectors. Shai et al. proposed PEGASOS [18], a simple and effective iterative algorithm for solving the optimization problem cast of SVMs. The number of iterations required to obtain a solution of accuracy ε is $O(1/\varepsilon)$, compared with other methods which requires $O(1/\varepsilon^2)$. LIBLINEAR [17] is a novel SVMs algorithm to large-scale sparse data. This method is especially effective for text classification. [17] showed that LIBLINEAR can reach an ε-accurate solution in $O(log(1/\varepsilon))$ iterations. The run time of PEGASOS and LIBLINEAR do not increase with the sample size, so the training process runs very fast than tradition method mentioned above. The flip side is that these methods have much worse dependence on the optimization accuracy.

III. Bootstrap Sampling based Data Cleaning

In this section, we will present our data cleaning method based on bootstrap sampling in detail.

A. SVMs Introduction

Before proposing our method, we briefly introduce SVMs and give some notations used in this paper. Given the training data $X = \{(x_1, y_1)...(x_n, y_n)\}$ in space $X \subseteq R^d$; where $x_i \subseteq R^d$ represents the d-dimensional pattern and $y_i = \pm 1$ represents the class label. The goal of SVMs training is to find a linear predictor $g(x) = w^T x + w_0$ with small empirical loss relative to a large classification "margin". With this predictor, the testing samples are separated according to the inequalities:

$$y_i = \begin{cases} 1 & \text{if} \quad (w^T \cdot x_i + w_0) \geq 1; \\ -1 & \text{if} \quad (w^T \cdot x_i + w_0) \leq -1. \end{cases} \tag{1}$$

The SVMs solution for this problem consists in maximizing the following quadratic optimization function (dual formulation):

$$\begin{aligned} W(\alpha) &= -1/2 * \sum_i^n \sum_j^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) + \sum_i^n \alpha_i \\ Subject \quad to: &\sum_i^n y_i \alpha_i = 0 \quad 0 \leqslant \alpha_i \leqslant C, i = 1, ..., n \end{aligned} \tag{2}$$

Where $\alpha = (\alpha_1, \ldots, \alpha_n)$ is nonnegative Lagrange multipliers associated with the constrain (1). $K(x_i, x_j)$ is the matrix of kernel values between pattern and the Lagrange multiplier α_i.

B. Bootstrap Sampling based data Cleaning with Weak SVMs

Many sampling methods have been used to facilitate speeding up SVMs training [6,7,11,13]. However, these methods aim to train an approximate SVMs classier instead of data cleaning. To address this problem, we propose the bootstrap sampling based method for data cleaning.

We first train a bundle of weak SVMs classifiers on some subsets of initial training datasets using the bootstrap sampling method. These weak SVMs classifiers are then used to pre-classify training datasets. Data correctly classified by all the weak classifiers will be eliminated. A definition of weak SVMs classifier used in this paper is as follows:

Definition III.1. ε-Gross-granularity weak SVMs: Assume $X = \{x_1, ..., x_n\}$ is the training dataset, $\widetilde{X}$ is a subset of X and the cardinality of $\widetilde{X}$ is far less than X. That is $\widetilde{X} \subset X$, and $Card(\widetilde{X}) \ll Card(X)$. Assume $g(\widetilde{x}) = \widetilde{w}^T \cdot \widetilde{x} + \widetilde{w_0} = 0$ is the SVMs predictor of $\widetilde{X}$ and $g(x) = w^T \cdot x + w_0$ is the SVMs predictor of X. $f(W; X) = \frac{1}{n}\sum_i l(w; (x_i, y_i)) + \frac{\lambda}{2}|w||^2 = \frac{1}{n}\sum_i max\{0, 1 - y_i < w, x_i >\} + \frac{\lambda}{2}|w||^2$ is the empirical loss function. The SVMs of $\widetilde{X}$ is called ε-Gross-granularity weak SVMs of X if the empirical loss function meet the following equation:

$$f(\widetilde{W}; \widetilde{X}) \leqslant f(W; X) + \varepsilon \tag{3}$$

where ε is a predefined constant.

According to above definition, if we have a bundle of ε-Gross-granularity weak SVMs with ε-accuracy solutions, we can pre-classify the training dataset X using these weak classifiers and eliminate useless training data. The framework is shown in Fig. 1.

In the data cleaning process, a difficult, yet important problem is how to find the proper ε-Gross-granularity weak SVMs. Prior studies have shown that the empirical error of SVMs is directly bounded up with the size of training dataset. Bartlett and Mendelson [20] proved that the maximum discrepancy of kernel classifications between hypothesis F and real probability distribution on training dataset X meets the following conditions.

$$G_n(F) \leq \frac{2B}{n} \sqrt{\sum_{i=1}^n K(x_i, x_j)} = 2B \sqrt{\frac{E(K(X, X))}{n}} \tag{4}$$

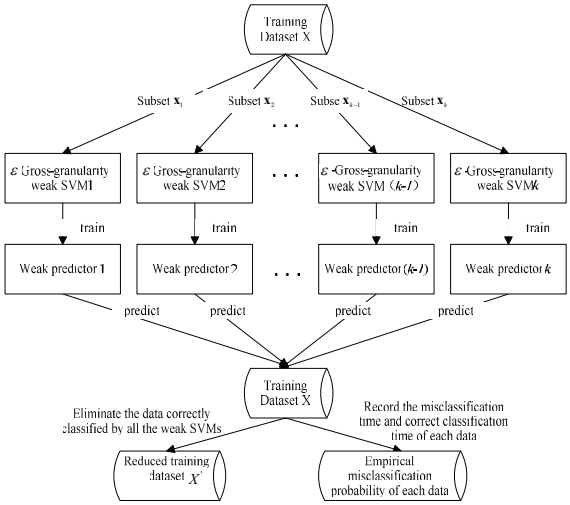

Fig. 1. Bootstrap sampling based data cleaning with weak SVMs

B is a nonnegative const. $K(X, X)$ is kernel matrix. Inequality (4) shows performance of hypothesis predictor depends on the size of training dataset, but they are not linear related. Shai et al. [18] proved that in low-norm and linear condition, to get a desired generalization error of $f(W; X) + \varepsilon$, $m = O(\frac{||w||^2}{\varepsilon^2}) = O(\frac{E(K(X,X))}{\varepsilon^2})$ samples are needed. Though the relation between training dataset size and error bound is clear, we are still unclear how weak the classifiers should be (the value of ε) and how many training data are needed to train a ε-Gross-granularity weak SVMs.

To address this problem, a heuristic method is proposed. We define a common phenomenon in classification using SVMs classifier.

Definition III.2. Sampling Gelling Point: Assume $X = \{x_1, ..., x_n\}$ is training dataset, $\widetilde{X}$ is the random subset of X, and P is the sampling percentage. P is defined as sampling Gelling Point of a SVMs classifier $f(W; X)$ if the testing error will not decrease significantly with the increasing of sampling percentage P.

Many empirical results indicate that the generalization performance is not linear related to sampling percentage P. When the sampling percentage P increases to some extent, assuming $P^{'}$, the empirical error will not decrease significantly and testing accuracy tends to stable. To illustrate Gelling Point phenomenon, 11 UCI datasets are used in our experiments. Fig. 2 shows Gelling Point phenomenon on these datasets.

Examples in Fig.2 demonstrate that many datasets have Gelling Point phenomenon. Consequently, it is possible to train an approximate SVMs classifier with a small subset of the initial training dataset.

Based on the above fact, we propose to sample the training data at sampling Gelling Point. The advantages are as follows:

- Generalization performance of weak SVMs at the sampling Gelling Point is close to SVMs. Therefore, ε value can be very small.

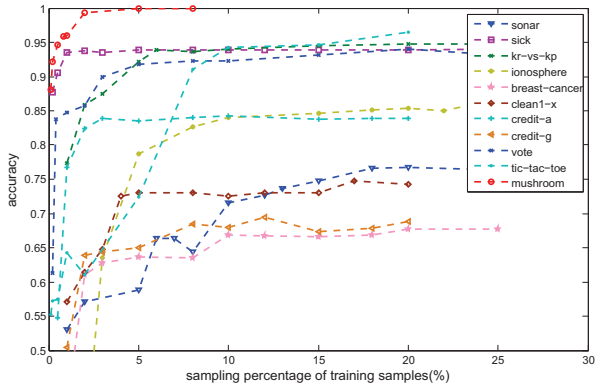

Fig. 2. Examples of sampling Gelling Point phenomenon. Horizontal axis is sampling percentage of training data and vertical axis is testing accuracy. The figure shows the experimental results of 11 toy datasets from UCI repository. And for most toy datasets, the Gelling Point is less than 5%.

- Experimental results show sampling Gelling Point is usually very small, in most cases less that 10%. For most large datasets, it is even lesser. Consequently, weak SVMs can be trained with small computational complexity.

IV. Informative Pattern Extraction based on Maximum Information Entropy

Learning theory defines informative patterns as: *given a model trained on a sequence of patterns, a new pattern is informative if it is difficult to predict by a model trained on previously seen data*[21,22]. In this paper, we define informative patterns as: *a pattern is informative if it is difficult to predict by a bulk of ε-Gross-granularity weak SVMs*. In other words, if the number of ε-Gross-granularity weak SVMs predicting a pattern label as -1 is close to the number of ε-Gross-granularity weak SVMs predicting a pattern label as 1, this pattern is very informative. Pattern labels predicted as -1 or 1 by all the ε-Gross-granularity weak SVMs is not informative at all.

To extract the informative patterns, we propose a maximum information entropy algorithm. This method is detailed as follows:

Assuming there are n training samples $X = \{x_1, ..., x_n\}$, m classes $C = \{c_1, ..., c_m\}$ and the probability of samples x_i classified to class c_j is a random variable. The information entropy of sample x_i can be defined as follows:

$$H(x_i) = E(I(x_i)) \tag{5}$$

$I(x_i)$ is the amount of information of x_i. The amount of information of x_i is defined as follows:

$$I(x_i) = -log_2 p(x_i) \tag{6}$$

$p(x_i)$ is the probability mass function, so the equation (5) can be rewritten as follows:

$$\begin{aligned} H(x_i) &= \sum_{j=1}^{m} p(c_{x_i} = c_j) \bullet I(c_{x_i} = c_j) \\ &= -\sum_{j=1}^{m} p(c_{x_i} = c_j) \bullet log_2 p(c_{x_i} = c_j) \end{aligned} \tag{7}$$

Assuming there are N ε-Gross-granularity weak SVMs. We use $k_i, i = 1, ..., n$ to denote the number of weak SVMs predictors misclassifying the training sample x_i. And the number of weak predictors correctly classifying x_i is denoted as $N - k_i, i = 1, ..., n$. Therefore, the empirical probabilities of misclassification and correct classification can be calculated as follows:

$$\begin{aligned} p(x_i\_correct) &\approx p(pre_{x_i} = c_{x_i}) = \frac{N - k_i}{N} \\ p(x_i\_wrong) &\approx p(pre_{x_i} \neq c_{x_i}) = \frac{k_i}{N} \end{aligned} \tag{8}$$

In (8), pre_{x_i} represents the predicted class label of sample x_i and c_{x_i} represents the true class label of sample x_i. Using equation (6) and (8), the amount of information of sample x_i can be approximately calculated by equations as follows:

$$\begin{aligned} I(x_i\_correct) &\approx -log_2(pre_{x_i} = c_{x_i}) = -log_2(\frac{N - k_i}{N}) \\ I(x_i\_wrong) &\approx -log_2(pre_{x_i} \neq c_{x_i}) = -log_2(\frac{k_i}{N}) \end{aligned} \tag{9}$$

The information entropy of sample x_i can be represented by equation (10) according to equation (5) and (9).

$$\begin{aligned} H(x_i) &= \sum p(x_i) \bullet I(x_i) \\ &\approx -(p(pre_{x_i} = c_{x_i}) \bullet log_2(pre_{x_i} = c_i) \\ &+ p(pre_{x_i} \neq c_{x_i}) \bullet log_2(pre_{x_i} \neq c_i)) \\ &= -((\frac{N - k_i}{N}) \bullet log_2(\frac{N - k_i}{N}) + (\frac{k_i}{N}) \bullet log_2(\frac{k_i}{N})). \end{aligned} \tag{10}$$

The information entropy of the whole training dataset can be calculated by:

$$H(X) = \sum_{i=1}^{n} H(x_i) \tag{11}$$

A. Informative Pattern Extraction Algorithm A

The intuitive idea of informative pattern extraction algorithm A is to select a subset $\widetilde{X}$ of the dataset $X^{'}$ from the first stage.The information entropy of $\widetilde{X}$ is very close to $X^{'}$ while the size of $\widetilde{X}$ is much smaller than $X^{'}$. Assuming given training dataset $X^{'} = \{x_1, ..., x_{n^{'}}\}$ and sampling percentage $q\%$ ($n^{'} \bullet q\% = \widetilde{n}$), problem is how to select a subset to get the maximum information entropy. This problem can be solved by optimizing an equation as follows:

$$\widetilde{X} = \{x_{t1}, ..., x_{t\widetilde{n}}\} = arg\ max(H(\widetilde{X})) \tag{12}$$

According to above deductions, this optimization problem can be further represented as follows:

$$\begin{aligned} \widetilde{X} &= arg\ max(H(\widetilde{X})) \\ &= arg\ max((p(pre_{x_{ti}} = c_{x_{ti}}) \bullet log_2(pre_{x_{ti}} = c_{x_{ti}}) \\ &\quad + p(pre_{x_{ti}} \neq c_{x_{ti}}) \bullet log_2(pre_{x_{ti}} \neq c_{x_{ti}}))) \\ &\approx arg\ max \sum_{i=1}^{\widetilde{n}} \left[-(\frac{N - k_{x_{ti}}}{N}) \bullet log_2(\frac{N - k_{x_{ti}}}{N}) \right. \\ &\quad \left. \frac{k_{x_{ti}}}{N} \bullet log_2(\frac{k_{x_{ti}}}{N}) \right] \end{aligned} \tag{13}$$

B. Informative Pattern Extraction Algorithm B

Algorithm A tries to maximize the information entropy under the condition of fixing the sampling percentage of misclassified samples. On the other hand, algorithm B tries to select a minimal subset of misclassified samples under the condition of fixing the percentage of all the information entropy.

The intuitive idea of algorithm B is to fix the percentage of all the information entropy and to minimize the size of misclassified samples. For example, how many samples we need at least to get 95% information entropy of all the training samples?

According to definitions above, given training dataset $X^{'}$ from the first stage and percentage $m\%$ of all the information entropy, try to find a subset $\widetilde{X}$ with the minimum cardinality. This problem can be represented as follows:

$$\begin{aligned} \widetilde{X} = \{x_{t1}, ..., x_{t\widetilde{m}}\} &= arg\ min(Card(\widetilde{X})) \\ Subject\ to: H(\widetilde{X}) &\geq m\% \bullet H(X^{'}) \end{aligned} \tag{14}$$

where $Card(\widetilde{X})$ is the cardinality of subset $\widetilde{X}$.

Pseudo code of algorithm A and algorithm B are as follows:

Precedure Infor_Extra_A&B($X^{'}$,$q_A\%$&$q_B\%$,$m\%$,$\widetilde{X}$,N,K)

Input: the dataset $X^{'}$ from the first stage, number of ε-Gross-granularity weak SVMs N, misclassification times of each training data K (vector). A) sampling percentage $q_A\%$. B) entropy percentage $q_B\%$.

Output: A) $q\%$ data selected from the initial dataset $X^{'}$ with the maximum information entropy. B) selected minimal data size with $m\%$ entropy of dataset $X^{'}$ entropy.

1. Calculate the information entropy of each training data

$$\begin{aligned} H(x_i) &= E(I(x_i)) \\ &\approx -((\frac{N - k_i}{N}) \bullet log_2(\frac{N - k_i}{N}) + (\frac{k_i}{N}) \bullet log_2(\frac{k_i}{N})) \end{aligned}$$

2. Calculate the information entropy of the whole training data

$$H(X^{'}) = \sum_{i=1}^{n} H(x_i)$$

3. A) select a subset $\widetilde{X}$ to get the maximum information entropy

$$\widetilde{X} = \{x_{t1}, ..., x_{t\widetilde{n}}\} = arg\ max(H(\widetilde{X}))$$

B) Select a minimum subset X and the entropy of X is no less than $m\% \bullet H(X^{'})$

$$\widetilde{X} = \{x_{t1}, ..., x_{t\widetilde{m}}\} = arg\ min(Card(\widetilde{X}))$$

4) **return** $\widetilde{X}$

end Precedure

V. Experimental Results

LIBSVM is used as SVMs training package in all our experiments for its perfect generalization performance [23]. We use the RBF kernel $exp(-\gamma||x_i - x_j||^2)$. The parameter γ is set to 1 and the penalty cost C in (2) is set to 100000.

10 weak SVMs are used in the first stage. We report average result on 10 runs. The experiments are parallelized executed on 5 WindowsXP machines with four 2.2GHz processor and 4GB DRAM.

A. Datasets

Four large datasets are used in our experiments. Table 1 shows information of the four datasets.

KDD99 is the KDDCUP-99 Intrusion Detection dataset used for the Third International Knowledge Discovery and Data Mining Tools Competition. Extended USPS dataset is the extended US Postal Service handwritten digits recognition corpus. IJCNN1 is IJCNN 2001 challenge (task1) dataset. Web is the web page dataset[1].

TABLE I
EXPERIMENT DATASETS

Dataset	KDD99	extended USPS	IJCNN1	Web
Attribute	127	676	22	300
Training	4,898,431	266,079	49,990	49,749
Testing	311,029	75,383	91,701	14,951
Gelling Point	0.2%	2%	8%	0.5%

B. Speedup and Generalization Performance Experiments

To evaluate the speedup values and generalization performance of our algorithms, we compare the two algorithms with regular LIBSVM.

We evaluate the speedup values, test accuracy and AUC values of our methods with different entropy values. As comparison, the test accuracy and AUC value of regular LIBSVM is used to be baseline result. Fig.3 shows the experimental results. The upper figures show the speedup values and the lower figures show the accuracy and AUC values under different entropy values. Baseline-1 and baseline-2 represent the test accuracy and AUC values of regular LIBSVM respectively.

Fig.3 shows that the improvement on speedup values is significant on all the four datasets. For datasets KDD99, web and USPS, speedup value can be 100 to 200 while the testing accuracy and AUC value are very close to regular LIBSVM. Although speedup value for IJCNN1 dataset is not so significant as that on the other three datasets, both algorithms are 20 times faster than regular LIBSVM. Experimental results on IJCNN1 and web datasets also show that our methods can achieve higher AUC values than regular LIBSVM when sampling misclassified samples to enough high entropy value.

Experiment also shows test accuracy of SVMs with 50% to 60% information entropy is comparable to that on the whole dataset. Therefore, our methods effectively extract the most informative pattern and significantly reduce the training time with little performance loss.

C. Comparative Experiments with PEGASOS, RSVM and LIBLINEAR SVM

To further compare the performance of our algorithms with other fast SVMs algorithms, we make comparative experiments between our algorithms and three state-of-the-art algorithms PEGASOS, RSVM and LIBLINEAR SVM.

[1]All the datasets can be downloaded from website http://c2inet.sce.ntu.edu.sg/ivor/cvm.html.

PEGASOS and LIBLINEAR are two very effective and fast SVMs algorithms which have shown excellent performance on very large datasets, especially datasets with very sparse, linear kernels. Besides three state-of-the-art SVMs algorithms, LIBSVM with random sampling is also used as a comparative method. Fig.4 shows our experimental results.

Some conclusions can be drawn from the experiment results. First, from the training time perspective, there is no significant difference between our algorithms and the three state-of-the-art methods. For web and USPS datasets, our algorithms need less time to converge than PEGASOS and LIBLINEAR. Second, the testing accuracy of our methods outperform other algorithms. For web, IJCNN1 and USPS datasets, PEGASOS and LIBLINEAR even inferior to LIBSVM with random sampling. It was reported in [17,18] that PEGASOS and LIBLINEAR are linear SVM which are effective for large, sparse datasets like text classification with linear kernels. However, our experiments indicate that PEGASOS and LIBLINEAR might not be so effective for large yet not sparse datasets. Comparison between our methods and RSVM demonstrates that both methods we proposed outperform RSVM in test accuracy. Meanwhile, our methods need less training time than RSVM. Experimental results on four datasets show that our method are more effective to find informative training data than RSVM.

VI. CONCLUSION

To make SVMs practical to large datasets, an algorithm trying to clean data and extract informative patterns for the SVMs classifier is proposed. This algorithm includes two stages, data cleaning stage and informative patterns extracting stage. A bootstrap sampling based data cleaning method is proposed in the first stage, and two maximum entropy based informative patterns extraction methods are proposed in the second stage. In most cases, the training data size will be reduced significantly after the two stages. Therefore, training time complexity will be reduced significantly. Empirical results show that our approach can significantly speedup the training process while testing accuracy is comparable to the classifier with the whole dataset using LIBSVM. Comparisons between our approach and three state-of-the-art methods PEGASOS, RSVM and LIBLINEA demonstrate that our approach is more efficient for large yet not sparse datasets.

Although the proposed algorithm is effective, there are some problems need to be solved in future work. First, in the data cleaning stage, the sampling percentage selection is very empirical. The Gelling Point phenomenon may be not prominent for all the datasets. In such cases, appropriate sampling percentage determination might depend on specific classification task and priori knowledge. Second, how many weak classifiers are needed? Theoretically, the more the better. In our experiments, only 10 weak SVMs classifiers are used. However, more weak classifiers need more training time and more parallel computers. Tradeoff between computing resource and testing accuracy must be carefully considered.

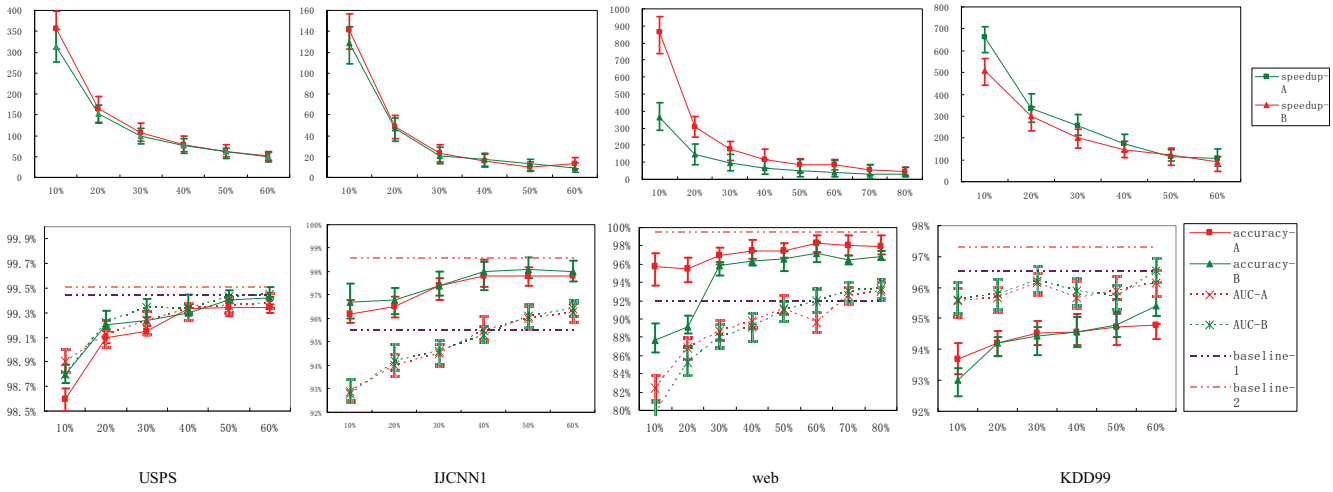

Fig. 3. testing accuracy versus training time speedup for four datasets

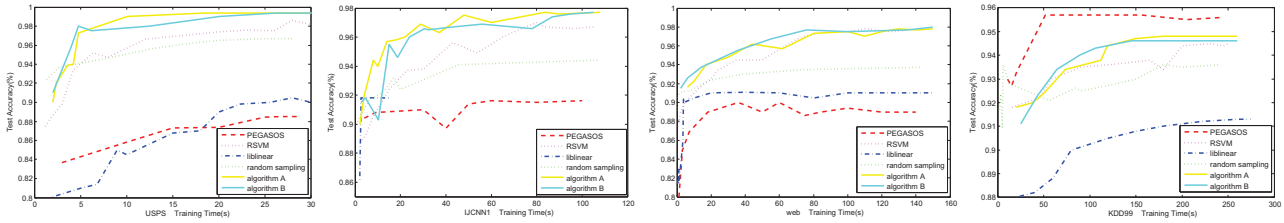

Fig. 4. testing accuracy versus training time for four datasets

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China [grant number 60973105, 90718017, 61170189, 61202239], the Research Fund for the Doctoral Program of Higher Education [grant number 20111102130003] and the Fund of the State Key Laboratory of Software Development Environment [grant number KLSDE-2011ZX-03].

REFERENCES

[1] S.Z. Wang, Z.J. Li, W.H. Chao and Q.H. Cao: Applying Adaptive Oversampling Technique Based on Data Density and Cost-Sensitive SVM to Imbalanced Learning, In: 2012 IEEE World Congress on Computational Intelligence (IJCNN 2012).

[2] X. Hu and H. Liu: Text Analytics in Social Media, Mining Text Data, pp: 385-414 (2012).

[3] X. Hu, L. Tang and H. Liu: Enhancing accessibility of microblogging messages using semantic knowledge, In: CIKM (2011).

[4] C. Burges: Geometry and invariance in kernel based methods, In Advances in Kernel Methods: Support Vector Learning, MIT Press.

[5] N. Panda, Y.C. Edward, G. Wu: Concept Boundary Detection for Speeding up SVMs, In: ICML, pp:681-688 (2006).

[6] H.P. Graf, E. Cosatto, L. Bottou, I. Durdanovic, V. Vapnik: Parallel Support Vector Machines: The Cascade SVM, Advances in neural information processing system, 17 pp: 521-528, MIT Press (2006).

[7] N.D. Lawrence, M. Seeger, R. Herbrich: Fast sparse gaussian process methods: the informative vector machine, Advances in Neural Information Processing Systems. MIT Press (2003).

[8] H. Yu, J. Yang, J. Han: Classifying large datasets using svm with hierarchical clusters, In: KDD (2003).

[9] J.C. Platt: Fast training of support vector machines using sequential minimal optimization, Advances in Kernel Methods - Support Vector Learning, pp: 185-208. MIT Press (1999).

[10] I.W. Tsang, T.K. James, P.-M. Cheung: Core Vector Machines: Fase SVM Training on Very Large Data Sets, Journal of Machine Learning Research, 6 pp: 363-392 (2005).

[11] Y.-J. Lee, O.L. Mangasarian: RSVM: Reduced support vector machines. In Proceedings of the First SIAM International Conference on Data Mining (2001).

[12] A. Smola, B. Scholkopf: Sparse greedy matrix approximation for machine learning, In: ICML, pp: 911-918 (2000).

[13] D. Achlioptas, F. McSherry, B. Scholkopf: Sampling techniques for kernel methods, In Advances in Neural Information Processing Systems 14, MIT Press (2002).

[14] T. Joachims: Training linear svms in linear time, In: KDD, (2006).

[15] A. Smola, S. Vishwanathan, Q. Le: Bundle methods for machine learning, Advances in Neural Information Processing Systems 20 (2008).

[16] T. Joachims, C.-N.J. Yu.: Sparse kernel SVMs via cutting-plane training, In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part I (ECML PKDD '09)

[17] R.E. Fan, K.-W. Chang, C.J. Hsieh, X.-R. Wang, C.-J. Lin: LIBLINEAR: A Library for Large Linear Classification, Journal of Machine Learning Research 9, pp: 1871-1874, (2008).

[18] S.S. Shai, Y. Singer, N. Srebro: Pegasos: Primal Estimated sub-GrAdient Solver for SVM, In: ICML (2007).

[19] L.B. Peter, S. Mendelson: Rademacher and Gaussian Complexities: Risk Bounds and Structural Results, Journal of Machine Learning Research 3, pp: 463-482 (2002).

[20] S.S. Shai, N. Srebro: SVM Optimization: Inverse Dependence on Training Set Size, In: ICML (2008).

[21] I. Guyon, N. Matic, V. Vapnik: Discovering Informative Patterns and Data Cleaning, In: AAAI Workshop on Knowledge Discovery in Databases (1994).

[22] D. MacKay: Information-based objective functions for active data selection. Neural Computation, 4 (4) pp:590-604 (1992).

[23] C.-C. Chang, C.-J. Lin: LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, pp: 2:27:1–27:27 (2011).