Fuzzy Systems and Data Mining IX
A.J. Tallón-Ballesteros and R. Beltrán-Barba (Eds.)

doi:10.3233/FAIA231021

Scalable Semi-Supervised Support Vector Machine Based on Adaptive Sampling

XinYue ZHAO [a], Xiaoyang SUN [a], and Jing ZHANG [a] Yunsheng SONG [a,1]
[a] *School of Information Science and Engineering, Shandong Agricultural University, Taian 271018, China*

Abstract. Semi-supervised support vector machine (S3VM) algorithms can effectively deal with the problem of a few labeled instance and a large number of unlabeled instances due to its good performance. The solution of the existing semi-supervised support vector machine algorithms requires the use of many types of optimization strategies because it takes all the training data as parameters to participate in iterative optimization, which makes it difficult to efficiently process large-scale data. Although simple random sampling is an effective means to consider efficient modeling from the perspective of data preprocessing, the problem that it determines the sample size in advance is difficult to process for the existence of sampling randomness and sample difference. To fully characterize the original unlabeled data and ensure the robustness of the model, we have proposed an adaptive sampling to train the model on the labeled set and the sampled unlabeled set. The fixed size unlabeled instances are continually sampled from the original unlabeled set until the proposed statistics on the obtained sample meet the stopping condition, where the statistics and stopping condition are generated by the density estimation. This method solves the problem of subjectively determining the sample size in advance, the robustness of the proposed algorithm has been proved with the probably approximately correct learning theory.

Keywords. Semi-supervised classification, support vector machine, sample size, random sampling, large-scale unlabeled data

1. Introduction

Semi-supervised learning is a paradigm that incorporates both labeled and unlabeled data during the learning process. Semi-supervised support vector machines (S3VMs) extend the SVM framework by incorporating both labeled and unlabeled data during the training process. The idea is that the unlabeled data can provide additional information about the underlying structure of the data, potentially improving the classifier's performance [1,2,3,4,3]. In recent years, advances in hardware and software have enabled researchers to develop scalable algorithms for large-scale semi-supervised SVMs. There exist popular algorithms for large-scale semi-supervised SVMs, including transductive SVMs (TSVMs)[5], Laplacian SVMs (LapSVMs)[6], and online SVMs (OS^{3}VMs)[7].

Transductive support vector machines (TSVMs) were introduced by Vapnik as a semi-supervised extension of SVMs. The core idea of TSVMs is to find a decision bound-

[1] Corresponding Author: Yunsheng Song ; E-mail: songys@sdau.edu.cn

ary that separates not only the labeled data but also the unlabeled data. This can be achieved by minimizing a cost function that accounts for both labeled and unlabeled data. The most popular TSVM algorithm is the S3VM by Joachims [5], which scales well to large datasets. Laplacian support vector machines (LapSVMs) were introduced by Belkin et al. [6] as another approach to semi-supervised SVMs. LapSVMs utilize a graph-based representation of the data to incorporate unlabeled data into the training process. The algorithm minimizes a cost function that includes a regularization term based on the graph Laplacian, which encourages the smoothness of the decision function over the data manifold. Facing large-scale stream data, Liu et al. [7] propose an online least squares support vector machine based on flow pattern regularization, which uses some key samples to construct regularization factors to achieve the decomposition of higher-order matrices, thereby greatly improving the solving efficiency of the algorithm.

Moreover, several optimization techniques have been proposed to improve the scalability of semi-supervised SVM algorithms. Some of the most popular techniques include the decomposition method [8], stochastic gradient descent [9], and parallelization [10]. These techniques have been shown to significantly improve the computational efficiency of semi-supervised SVMs, making them more suitable for large-scale applications. Large-scale semi-supervised SVMs have been applied in various domains, including text classification, image recognition, bioinformatics, and speech recognition. These applications have demonstrated the potential of semi-supervised SVMs to effectively handle large datasets and improve classification performance when labeled data are scarce.

The current research work simply considers the efficient approximation solution of discriminant semi-supervised classification algorithm from the perspective of optimization, but it is difficult to efficiently process massive data because it requires multiple alternating iterative optimizations and all data as parameters to participate in optimization. In addition, label-free data not only provides parametric optimization information, but also contains rich feature information, and how to use this information to consider efficient modeling from the perspective of data granulation is the key means to solve large-scale complex problems. In this paper, we have proposed a scalable learning dreamwork for semi-supervised SVM. The proposed method can efficiently deal with large-scale data with comparable classification performance.

2. Main Content

2.1. Basic Description

For the given training set T which is the union of the labeled instance set $L = \{x_1, x_2, \cdots, x_l\}$ and the unlabeled instance set $U = \{x_{l+1}, x_{l+2}, \cdots, x_{l+u}\}$, where each instance $x_i = \{x_{i1}, x_{i2}, \cdots, x_{im}\}^\top$ is expressed by a m-dimension vector, l and u are the number of labeled instances and unlabeled instances, and $i = 1, 2, \cdots, l+u$. Semi-supervised classification algorithms simultaneously use the labeled instance set L and unlabeled instance set U to train a classifier $f(x)$ with good performance.

2.2. Learning Curve

Traditional learning curve mainly is widely used for supervised classification problem, it describes the relationship between the classification performance of the learner and the training set size. Generally speaking, the classification performance of the learner gradually improves as the increasing training set at the beginning stage, and it then reaches the top and then no longer increases significantly. Tradition SVM algorithm is fit for the learning curve, while the adaptability of semi-supervised SVM for it needs to be verified. Different from supervised methods, we use the labeled set L and the sampled labeled instances from U to construct the increasing training set. Therefore, we have used a real dataset with LapSVM algorithm to study this problem.

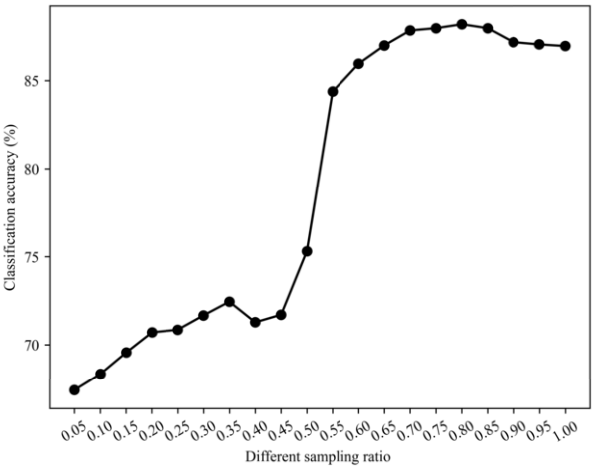

Figure 1. The learning curve under different sampling ratio

Fig. 1 shows the classification accuracy of LapSVM algorithm under the fixed labeled set and different sampled ratio of unlabeled instances on the real dataset. It can be found that classification accuracy increases until the sampled ratio is 0.7, and then it tends to stabilize. This phenomenon can be also verified for other kinds of semi-supervised SVM algorithms, because they have the same assumption that the hyperplane passes through a low-density region in the input space, where the instances close to the decision surface contribute more than instances farther from the decision boundary. This validates the feasibility of sampling several unlabeled instances with the labeled instances to obtain an SVM classifier with good performance.

Therefore, we have proposed a scalable learning framework for semi-supervised SVM to deal with large-scale unlabeled instances. In this framework, the subset S of the original unlabeled instance set U combined with the labeled set L are used to train the classifier. Owning to the fact that the execution efficiency is positively correlated to the size of the training set, then the obtained classifier with a smaller training set has high execution efficiency. On the other hand, the learning curve has confirmed that the obtained classifier can achieve a similar classification performance as the one trained on the original set as long as the sample is enough. The pseudocode of the proposed framework is listed in Algorithm 1.

Algorithm 1 Scalable learning framework under sampling for semi-supervised SVM

```
Input: The training set $T$ which is the union of the labeled instance set $L = \{x_1, x_2, \cdots, x_l\}$ and the unlabeled instance set $U = \{x_{l+1}, x_{l+2}, \cdots, x_{l+u}\}$, semi-supervised SVM algorithm $A$.
Output: The classifier $f$.
Obtain a subset $S$ form the set $U$ using sampling, where $S$ keeps the most distribution information of $U$;
Train the classifier $f$ on the union of $L$ and $S$ using algorithm $A$ ;
return $f$.
```

2.3. Adaptive Sampling

The sample size is critical for the random sampling method, and it decides the equality of the sample. As random sampling exists randomness, the obtained sample of the fixed size has significantly different instances. Moreover, different unlabeled instances take different degrees of contribution to the classifier and the randomness of sampling, then the same amount of sampled unlabeled instances also have larger differences. So the sample size cannot be pre-determined before sampling, it should be chosen by the features of the data and task.

For semi-supervised classification task, there exists a hypothesis that the probability distribution $P(x)$ of the input instance x takes valuable information to the posterior probability distribution $P(y|x)$. If we want to reduce the unlabeled instance set U to be a smaller S while keeping the information, the difference in probability distribution between U and S is smaller. Therefore, it needs a measurement to evaluate the difference. The kernel function is widely used to estimate the density function, where the density estimation is the average value of the kernel function on the given set. Therefore, we use the difference in the density estimation between U and S to obtain the subset S of high quality.

Let $K(x)$ be the kernel function. Then $\overline{g} = \sum\limits_{x_i \in U} K(x_i)/u$ and $\widehat{g} = \sum\limits_{x_i \in S} K(x_i)/s$ are the density estimation for the set U and its subset $S \subseteq U$ using simple random sampling without replacement, where s is the size of the set S. In this way, the difference in density estimation between the set U and its subset S can be measured by $|\overline{g} - \widehat{g}|$. If the sample S is a good subset of U with high equality, the difference is small. The absolute difference is difficult to be used because its value always changes greatly for different data. Therefore, the condition $|\overline{g} - \widehat{g}| \leq \gamma |\overline{g}|$ to judge, where $\gamma \in (0,1)$. Owning to the fact that $\overline{g}$ cannot be efficiently computed for large-scale data. On other hand, the stability of $\overline{g}$ on the sample S under fixed size is also difficult to be guaranteed due to the presence of randomness. To solve this problem, we have proposed an adaptive sampling algorithm which is one kind of multiphase sampling. In this method, the unlabeled instances are continuously sampled from the set U to form the subset S until the estimation $\overline{g}$ on the obtained subset S satisfies the stopping condition. The stooping condition is related to the number of iterations and the range of the kernel function on the set U. Moreover, fixed-volume unlabeled instances are drawn during each iteration to accelerate the satisfaction of the termination condition. Though the estimation is continuously computed in the process of iteration, it can be efficiently calculated using the by linearly weighting of the results of the previous iteration and but the current results. The pseudocode of adaptive sampling algorithm is listed in algorithm 2.

Algorithm 2 Adaptive Sampling algorithm

```
Input: The unlabel set $U = \{x_{l+1}, x_{l+2}.\cdots, x_{l+u}\}$, the batch size $b$.
Output: The subset $S$.
Initialization: $S = \emptyset$, $r = 0$, $\widehat{g} = 1$, $\beta_r = -\infty$;
while $\widehat{g} > \beta_r(1 + 1/\varepsilon)$ do
    $r = r + 1$, $\beta_r = \lambda\sqrt{\ln(r(r+1)/\delta)/(2r * b)}$;
    Sampling $b$ unlabeled instance without replacement from the set $U$ to be $S_r$ ;
    $S = S \bigcup S_r$, and compute $\widehat{g}$ on the set $S$ ;
end while
return $S$
```

2.4. Robustness

Supposed that the unlabeled instance set U is the population, then S is one sample set of U. Obviously, the expectation of $\widehat{g}$ is that $E(\widehat{g}) = \overline{g}$. According to the Hoeffding inequality, we can get the following lemma.

Lemma 1 *Let $S \subseteq U$ be the set of the fixed size s instances independently sampled from the set U. For any real number $\varepsilon > 0$, it has $P(|\widehat{g} - \overline{g}| \leq \varepsilon) \leq 2exp(-2s\varepsilon^2/\lambda^2)$, where $\lambda = \max\limits_{x_i \in U} K(x_i,) - \min\limits_{x_i \in U} K(x_i)$.*

Proof *Let $z_i = K(x_i)$ be the random variable on the sample space $\{x_{l+1}, x_{l+2}, \cdots, x_{l+u}\}$, $\widehat{g}$ is the mean of z_i value on the sample S, and $E(\widehat{g}) = \overline{g}$. According to the Hoeffding inequality, $P(\widehat{g} - \overline{g} \leq \varepsilon) \leq exp(-2s\varepsilon^2/\lambda^2)$ and $P(\overline{g} - \widehat{g} \leq \varepsilon) \leq exp(-2s\varepsilon^2/\lambda^2)$, then the lemma can be obtained by combining these two inequations.*

The final number of iterations r of algorithm 2 depends on the randomness of the sampled instances, and it is a random variable. Two number r_0 and r_1 are defined, $r_0 = \min\limits_r\{\beta_r \leq \varepsilon|\overline{g}|\}$, and $r_1 = \min\limits_r\{\beta_r \leq \varepsilon|\overline{g}|/(1+2\varepsilon)\}$. Because $\beta(r)$ is a strictly decreasing function, then r_0 and r_1 are uniquely determined. Similar to the result of the paper [11], we can get these two lemmas.

Lemma 2 *[11] For any $\varepsilon > 0$, the probability that adaptive sampling algorithm 2 stops before $t_0 - th$ iteration is smaller than $\delta(1 - 1/r_0)$.*

Lemma 3 *[11] For any $\varepsilon > 0$ and $\delta \in (0,1)$, the probability that adaptive sampling algorithm 2 stops after $t_1 - th$ iteration is smaller than $\delta/(2r_0)$.*

Combining Lemma 2 and Lemma 3, we have the following lemma.

Lemma 4 *For any $\varepsilon > 0$ and $\delta \in (0,1)$, the probability that adaptive sampling algorithm 2 stops between $t_0 - th$ iteration and $t_1 - th$ iteration is larger than $1 - 1 - \delta + \delta/(2r_0)$.*

Proof *$P(r_0 \leq t \leq r_1) = 1 - P(r < r_0) - P(r > r_1) \geq 1 - \delta/(2r_0) - \delta(1 - 1/r_0) = 1 - \delta + \delta/(2r_0)$*

Lemma 5 *[11] For any $\varepsilon > 0$ and $\delta \in (0,1)$, the output of adaptive sampling output $\widehat{g}$ satisfies $P(|\overline{g} - \widehat{g}| \leq \varepsilon\overline{g}|r_0 \leq t \leq r_1) > 1 - 2\delta/r_0$.*

Theorem *For any $\varepsilon > 0$ and $\delta \in (0,1)$, the output of adaptive sampling output $\widehat{g}$ satisfies $P(|\overline{g}-\widehat{g}| \leq \varepsilon\overline{g}) > 1-\delta$.*

Proof *Combining the lemma 5 and lemma 4, we have $P(|\overline{g}-\widehat{g}| \leq \varepsilon\overline{g}) \geq P(|\overline{g}-\widehat{g}| \leq \varepsilon\overline{g}, r_0 \leq t \leq r_1) = P(|\overline{g}-\widehat{g}| \leq \varepsilon\overline{g}|r_0 \leq t \leq r_1) * P(r_0 \leq t \leq r_1) > (1-2\delta/r_0)*(1-2\delta/r_0) = 1-\delta+\delta^2/(2r_0)-\delta^2/(4r_0^2) > 1-\delta$.*

3. Experiments

To test the effectiveness and efficiency of the proposed algorithm, the comparison in the performance between the classifier trained using the original unlabeled instance set and the classifier using the unlabeled instance subset obtained by adaptive sampling is made. LapSVM algorithm [6] and HGSVM algorithm [12] are selected for their representativeness to be the basic classifier, where the classifiers trained on the sampled subset denotes LapSVM-AS and HGSVM-AS. Meanwhile, a large dataset named phonme size of 5404 is also used, it is divided into the training set and test set in a ratio of 7 to 3, where the labeled instances account for 10% of the training set. The parameters of adaptive sampling algorithm $\varepsilon = 0.5, \delta = 0.2$, and other parameters of LapSVM algorithm and HGSVM algorithm are set as default parameters. All the codes of the adopted algorithms are written by Python3.10, and they are executed on a server of Intel(R) Xeon(R) Silver 4208 CPU @ 2.10GHz and 160GB RAM. Classification accuracy (Acc) and execution time (ET) in seconds of two kinds of algorithms are listed in Table 1.

Table 1. Performance comparison of the two algorithms

LapSVM VS LapSVM-AS				HGSVM VS HGSVM-AS			
LapSVM		LapSVM-AS		HGSVM-AS		HGSVM-AS	
Acc	ET	Acc	ET	Acc	ET	Acc	ET
81.36	2650.32	80.51	1527.95	87.19	2740.33	86.51	1583.58

Table 1 shows that LapSVM-AS algorithm has a slightly lower classification accuracy than LapSVM algorithm, and the difference between them is 0.85% which is tiny compared with the original classification accuracy. Meanwhile, HGSVM-AS algorithm gets a very similar classification accuracy to HGSVM algorithm, and the difference between them is 0.68%. The experience result indicates two representative algorithms trained on the sampled subset obtained by adaptive sampling achieve very similar classification accuracy, and it also validates that the proposed sampling method can get enough distribution information as the original unlabeled instances.

Besides classification performance, execution time is also an important performance evaluation metric. The less execution time, the much higher efficiency. The execution time of LapSVM algorithm and LapSVM-AS algorithm are 2650.32 and 1527.95, where LapSVM-AS algorithm obtains nearly half the time savings compared to algorithm LapSVM algorithm. Meanwhile, HGSVM-AS algorithm also obtains nearly half the time savings compared to algorithm HGSVM algorithm. So the proposed sampling method has high execution efficiency.

4. Conclusions

For the problem that massive unlabeled instances bring a great challenge to efficiently train semi-supervised semi-supervised support vector machine algorithms, this paper has developed an adaptive sampling algorithm. Different from the previous approaches from the view of algorithm optimization, it takes advantage of data reduction to avoid the difficulty of using domain knowledge to improve the efficiency of algorithms. The proposed method continually samples the fixed number of unlabeled instance from the unlabeled set until the estimation on the obtained subset meets the stopping condition, and its robustness has been proved by related lemmas. Moreover, its effectiveness and efficiency have been validated by the experience results on the real dataset. In the future, the proposed algorithm will be popularized and applied to many pattern recognition fields such as question classification, sentiment classification and face recognition in intelligent question answering.

Acknowledgement

This research was supported by Shandong Provincial Natural Science Foundation, China, grant number ZR2020MF146.

References

[1] Zhu X, Goldberg AB. Introduction to semi-supervised learning. Synthesis Lectures on Artificial Intelligence and Machine Learning. 2009;3(1):1-130.

[2] Li Z, Kang Y, Feng D, Wang XM, Lv W, Chang J, et al. Semi-supervised learning for lithology identification using laplacian support vector machine. Journal of Petroleum Science and Engineering. 2020;195:107510.

[3] Bai L, Chen X, Wang Z, Shao YH. Safe intuitionistic fuzzy twin support vector machine for semi-supervised learning. Applied Soft Computing. 2022;123:108906.

[4] Tao J, Zhang N, Chang J, Chen L, Zhang H, Chi Y. Unlabeled sample selection for mineral prospectivity mapping by semi-supervised support vector machine. Natural Resources Research. 2022;31(5):2247-69.

[5] Joachims T, et al. Transductive inference for text classification using support vector machines. In: Proceedings of The Sixteenth International Conference on Machine Learning. vol. 99; 1999. p. 200-9.

[6] Belkin M, Niyogi P, Sindhwani V. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research. 2006;7(11).

[7] Liu Y, Xu Z, Li C. Online semi-supervised support vector machine. Information Sciences. 2018;439:125-41.

[8] Zeng ZQ, Yu HB, Xu HR, Xie YQ, Gao J. Fast training support vector machines using parallel sequential minimal optimization. In: Proceedings of 3rd International Conference on Intelligent System and Knowledge Engineering. vol. 1. IEEE; 2008. p. 997-1001.

[9] Bottou L. Large-scale machine learning with stochastic gradient descent. In: Proceedings of 19th International Conference on Computational Statistics. Springer; 2010. p. 177-86.

[10] Zinkevich M, Weimer M, Li L, Smola A. Parallelized stochastic gradient descent. In: Advances in Neural Information Processing Systems. vol. 23; 2010. p. 2595-603.

[11] Watanabe O. Sequential sampling techniques for algorithmic learning theory. Theoretical Computer Science. 2005;348(1):3-14.

[12] Sun Y, Ding S, Guo L, Zhang Z. Hypergraph regularized semi-supervised support vector machine. Information Sciences. 2022;591:400-21.