Adaboost-LLP: A Boosting Method for Learning With Label Proportions

Zhiquan Qi, Fan Meng, Yingjie Tian, Lingfeng Niu, Yong Shi, and Peng Zhang

***Abstract*— How to solve the classification problem with only label proportions has recently drawn increasing attention in the machine learning field. In this paper, we propose an ensemble learning strategy to deal with the learning problem with label proportions (LLP). In detail, we first give a loss function based on different weights for LLP, and then construct the corresponding weak classifier, at the same time, estimate its conditional probabilities by a standard logistic function. At last, by introducing the maximum likelihood estimation, we propose a new anyboost learning system for LLP (called Adaboost-LLP). Unlike traditional methods, our method does not make any restrictive assumptions on training set; at the same time, compared with alter-$\propto$SVM, Adaboost-LLP exploits more extra weight information and uses multiple weak classifiers that can be solved efficiently to combine a strong classifier. All experiments show that our method outperforms the existing methods in both accuracy and training time.**

***Index Terms*— Classification, ensemble learning, learning with label proportions, machine learning.**

I. Introduction

SUPERVISED learning [1]–[4], semisupervised [5] learning, and unsupervised learning [6] are the main topics in machine learning field. However, these techniques cannot solve all the machine learning problems. Taking a trade investment problem as an example, the following is explained.

A marketing company plans to increase its profit in sales. Sending out discount coupons is a good choice. In fact, we can divide the customers who buy products with coupons into two types: people would not buy without coupons (Type 1 Customers) and that would buy without coupons (Type 2 Customers). Sending coupons to Type 1 Customers will increase profit and it will cut profit if we send coupons to Type 2 Customers. However, according to the historical data, we cannot distinguish the two types of customers from each other. Only the proportions of these people in certain groups can be estimated by the experienced marketing manager.

Similarly, politicians, spam filtering, and democratic election also face the same problem: how to use only the label proportions information to obtain a new classification model. Generally, researchers usually call it learning problem with label proportions (LLP). Unlike other machine learning problems, in the LLP problem, the training instances are provided in "bags," and only the proportion of each class in each bag is known. The goal is to predict a new individual instance's label. The problem can be described by the mathematical language.

Suppose that there is a training set

$$T = \{(\mathcal{X}_1, \mathcal{P}_1), \ldots, (\mathcal{X}_K, \mathcal{P}_K)\} \tag{1}$$

where $\mathcal{X}_k = \{x_{k1}, \ldots, x_{kl_k}\}, x_{kj} \in R^n, \mathcal{P}_k \in [0, 1]$, $k = 1, \ldots, K$, $j = 1, \ldots, l_k$. For each bag $\mathcal{X}_k$, we only know the proportion of labels: $\mathcal{P}_k$, which can be defined as

$$\mathcal{P}_k := \frac{\left|\left\{x_i | x_i \in \mathcal{X}_k, y_i^* = 1\right\}\right|}{|\mathcal{X}_k|} \tag{2}$$

$y_i^* \in \{1, -1\}$ denotes the unknown ground truth label of x_i. Finally, the goal is to find a real function $\mathcal{F}(x)$ in R^n, such that the label y for any instance x can be predicted by the decision function

$$\mathcal{F}(x) = \operatorname{sgn}(g(x)). \tag{3}$$

From the definition of LLP problem, we can associate it with another famous machine learning problem: multiple instance learning (MIL) problem [7]–[14], which has been widely applied in handwriting recognition [15], object detection [16], scene classification [17], and so on. The one thing common between them is that both the problems have the concept of bags, and the label of each instance is unavailable. The essential difference is that in the LLP problem, we know the label proportion information of each bag, but in the MIL problem, there are the definitions of positive bags and negative bags: the positive bag has at least one positive instance, while the negative bag has none. Analyzing from the definition, we are not sure which one contains more label information. Fig. 1 gives the sketches of two kinds of learning problems.

A. Related Work

The related research about learning with label proportions problem can be traced back to 2005 [18], in which Kuck and de Freitas propose an MCMC algorithm to solve

Manuscript received November 21, 2015; revised September 16, 2016 and March 5, 2017; accepted July 9, 2017. Date of publication August 15, 2017; date of current version July 18, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61472390, Grant 61402429, Grant 11271361, and Grant 71401188, and in part by the Key Project of the National Natural Science Foundation of China under Grant 71331005 and Grant 91546201. *(Corresponding author: Yingjie Tian.)*

Z. Qi, Y. Tian, and Y. Shi are with the Key Laboratory of Big Data Mining and Knowledge Management, Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Economics and Management, University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: tyj@ucas.ac.cn).

F. Meng is with the School of Management Science and Engineering, Central University of Finance and Economics, Beijing 100081, China.

L. Niu is with the Key Laboratory of Big Data Mining and Knowledge Management, Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China.

P. Zhang is with the Ant Financial Services Group, Hangzhou 310000, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2727065

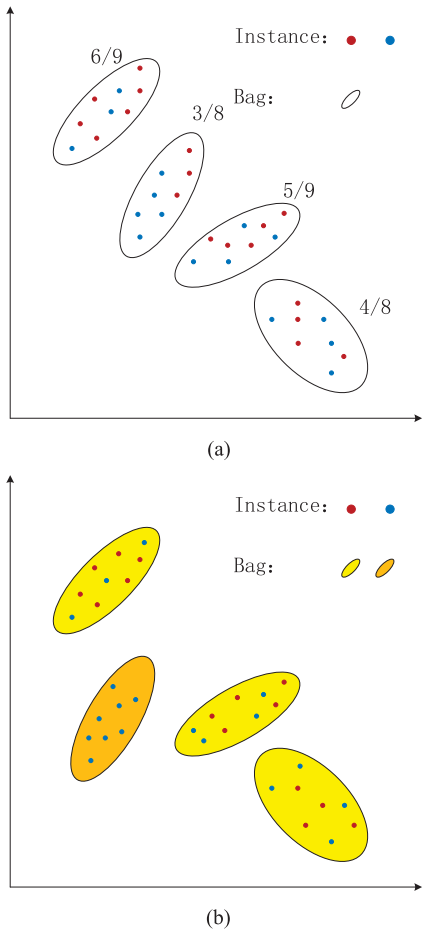


Fig. 1. Comparison between LLP and MIL problem. The red points denote positive points and blue points denote the negative ones. The numbers along with bags in (a) are the proportions of positive points of each bag. The common ground of LLP and MIL is that we do not know all labels of instances in the two problems. Differences are as follows. There are the concept of positive bags and negative bags in MIL, but none in LLP. In MIL, positive bag means that at least one instance belongs to positive class, and negative bag means that all instances belong to negative class. In LLP, each bag only knows the label proportion information between positive classes and negative classes. (a) LLP problem. (b) MIL problem.

this problem that accounts for uncertainty in the parameters and in the unknown individual labels. But its effectiveness is severely limited by the complexity of the problem solved. Next, Chen *et al.* introduce the learning from aggregate views, at the same time developed different learning methods for the special case called learning from projections and counts [19], but the conditional class estimates need to match the observed ones in this approach. Similar ones also include [20]. Quadrianto *et al.* [21] provide consistent estimators, which can reconstruct the correct labels with high probability in a uniform convergence sense. In detail, they assume that the distribution of labels conditioned on the features via a conditional exponential model

$$p(y|x,\theta) = \exp(< \phi(x,y) >, \theta) - g(\theta|x) \tag{4}$$

where $g(\theta|x) = \log \sum_{y\in\mathcal{Y}} \exp < \phi(x,y), \theta >$, is called the log-partition function, and $\phi(x,y)$ denotes a feature map from $\mathcal{X}\times\mathcal{Y}$ to a reproducing kernel Hilbert space $\mathcal{H}$. The parameter θ need be estimated based on labels proportion information. But this algorithm needs a key assumption: $p(x|y,k) = p(x|y)$, where k denotes kth bag. In addition, the estimation process strongly depends on empirical means and expectations scale. Stefan Rueping [22] uses support vector regression and inverse classifier calibration to build a new algorithm for LLP problem, and shows that this new approach outperforms previous approaches. In fact, they take the mean of each bag as a "superinstance," and give a soft label by the corresponding label proportion. But that using "superinstance" to denote the property of bag is not a good choice, and it may be poor in some special cases, especially when then data distribution is dependent on the bags, detailed analysis can be found in [23]. In the same year, Stolpe and Morik [24] give a definition of the problem of learning from label proportions and present a solution based on clustering. But its effectiveness is also limited by the computing complexity. Fan *et al.* [25] present theoretical analysis between the label proportions learning and supervised learning, by giving a sufficient condition of learnable case on binary classification, and then propose a framework to build generative classifiers by density estimation. The experimental results on benchmark data sets show promising performance. In addition, Patrini *et al.* [26] provide a fast learning algorithm that estimates the mean operator via a manifold regularizer with guaranteed approximation bounds. Recently, a new algorithm based on SVM: $\propto$SVM [23], [27] outperforms the other known methods. They make use of a large-margin framework to optimize over the unknown instance labels and the known label proportions where the key distribution is that this algorithm alleviates the need for making restrictive assumptions on the data, either parametric or generative (many assumptions do not hold for many real-world applications). In detail, the LLP algorithm based on the large-margin framework is expressed as

$$\begin{aligned} \min_{y,w,b} \ & \frac{1}{2} w^\top w + C \sum_{i=1}^{N} L(y_i, w^\top \varphi(x_i) + b) \\ & + C_p \sum_{k=1}^{K} L_p(\tilde{p}_k(y), p_k) \\ \text{s.t. } & \forall y_i \in \{-1, 1\} \end{aligned} \tag{5}$$

where $L(\cdot) \geq 0$ is the loss function for classic supervised learning. $L_p(\cdot) \geq 0$ is a function to penalize the error between the ground truth of the proportion label and the predicted one. The goal is to solve y, w, and b. Of course, $\propto$SVM has its own disadvantages too. From (5), we can find that this model is a nonconvex integer programming problem, and is NP-hard. So it is only able to obtain the approximate result in limited time. In addition, other references can be found in [28]–[37]. At present, LLP has also been applied in marketing, election, spam filtering [18], visual attribute modeling [38], [39], video event detection [40], predicting income based on census data [27], and so on.

B. Motivation

The essential difference of the above methods is that they use different tactics to describe LLP problem. For example, Quadrianto *et al.* use the mean of each bag and label

proportions to estimate the mean of each class (called MeanMap method) [21]. But the method demands that the class-conditional distribution of data is independent of the bags, which is usually not the case for many real-world applications. Rueping [22] takes the mean of each bag as a "superinstance." But sometimes, the "superinstance" cannot represent the properties of the bags. Yu *et al.* [23] use a large-margin framework to explicitly model the latent unknown instance labels together with the known group label proportions (called $\propto$SVM). Unlike [21] and [22], $\propto$SVM avoids making restrictive assumptions about the data, and outperforms the existing works above. But this method has to solve a nonconvex integer programming problem, which is NP-hard. Although we propose two efficient algorithms to solve the optimization problem, its computational complexity is still very high. As the result, our goal is to find a more effective tactic to describe LLP problem, which requires neither restrictive assumptions, nor solving a complicated optimization problem.

In this paper, we propose a new Learning with Label Proportions based on Boosting (called Adaboost-LLP). Our main idea is that, we build a set of weak classifiers based on the standard logistic function, then use maximum likelihood estimation (MLE) to describe the relationship between prediction function and label proportions of bags, finally solve the LLP problem in the AnyBoost framework, by which the weights of samples and bags can be generated automatically. Similar to [23], Adaboost-LLP does not need to make restrictive assumptions about the data, either. More importantly, our algorithm does not solve a complicated integer programming problem, and the corresponding computational complexity is less than that of $\propto$SVM. All experiments show that Adaboost-LLP outperforms the existing methods in the accuracy and training time.

The remaining parts of this paper are organized as follows. Section II gives our new algorithm: Adaboost-LLP; all experiment results are shown in Section III; concluding remarks are given in Section IV.

II. Boosting Learning With Label Proportions

In this section, we will describe in detail our algorithm, which includes the loss function, weak classifier, the main algorithm, and the discussion. Here, we take the simple linear discriminant function as our weak classifier. The loss function gives the optimization object of the weak classifier. Introducing the logistic function structures the relationship between the proportions label and the output of weak classifier. Adaboost-LLP combines many weak classifiers to build the final strong classifier.

A. Loss Function for Adaboost-LLP

How to define loss function $L_\rho(\cdot)$ is a key step for LLP problem. Simply, we can use the average squared error $L_\rho^{\text{MSE}}(\cdot)$ to describe the matching degree with the given proportions

$$L_\rho^{\text{MSE}}(\rho, \tilde{\rho}) = \sum_{k=1}^{K} (\rho_k - \tilde{\rho}_k)^2 \tag{6}$$

where ρ_k is the true label proportion of kth bag, and $\tilde{\rho}_k$ is the predicted one.

Stolpe and Morik [24] add weights for different bags. So (6) is rewritten as

$$L_\rho^{\text{MSE-weight}}(\rho, \tilde{\rho}) = \sum_{k=1}^{K} \frac{l_k}{N} (\rho_k - \tilde{\rho}_k)^2 \tag{7}$$

where $N = \sum_{k=1}^{K} l_k$. Stolpe and Morik [24] first predict the label of each instance in the bag and the compute the predicted proportion $\tilde{\rho}_k$. In this way, we have $\tilde{\rho}_k = ((|\{x_i | x_i \in \mathcal{X}_k, \tilde{y}_i = 1\}|)/(|\mathcal{X}_k|))$, and $\tilde{y}_i$ denotes the predicted label of $x_i \in \mathcal{X}_k$. Note in Adaboost-LLP, we regard the sum of probabilities of all samples to be positive as the positive proportion, as shown in (14). Due to $L_\rho^{\text{MSE-weight}}(\cdot)$ only using aggregated label information, we cannot avoid the occurrence of mismatch. In other words, minimizing $L_\rho^{\text{MSE-weight}}(\cdot)$ does not mean minimizing the average loss over individual observations. To give a complement from the view of whole data set, literature [24] gives another loss function

$$L_\rho^{\text{prior}}(\rho, \tilde{\rho}) = (\eta - \tilde{\eta})^2 \tag{8}$$

where $\eta = \sum_{k=1}^{K} (l_k/N)\rho_k$, and $\tilde{\eta} = \sum_{k=1}^{K} (l_k/N)\tilde{\rho}_k$ is the predicted value about η. Equation ((8)) catches overall proportions information of the labels, and is a good supplement about ((7)). However, this still cannot completely solve the difficult problem: more mismatches may lead to the lower loss about proportions of the labels. In this paper, we reduce this probability of this case from another point view. Now we consider what kind of bag contains more useful information. Let us discuss in two conditions as follows.

Case 1: Suppose $\rho_k = 0.5$ for some bag, $g(x)$ is a correct classifier. In this case, we can obtain $\tilde{\rho}_k = 0.5$, which is just what we want. But $-g(x)$ can also compute $\tilde{\rho}_k = 0.5$, in this case, $-g(x)$'s predictions for all instances of the bag are completely wrong.

Case 2: Suppose $\rho_k = 0$ or $\rho_k = 1$ for some bag; in this case, we actually have known each instance's label. If $g(x)$ is a right classifier, we cannot obtain another wrong classifier, which can compute $\tilde{\rho}_k = 0$ or $\tilde{\rho}_k = 1$.

Case 3: Based on the two specific cases above, given a proportion label, we make a general calculation of the possible prediction error rate. Suppose $\rho_k = a$, $a \in (0, 1), a \neq 0.5$, $g(x)$ is a correct classifier. In this case, if $g'(x)$ compute $\rho_k = a$ and $b = \min\{a, 1-a\}$, it will predict wrongly $200b\%$ instances at most. Note that the calculation in Case 3 can also be adopted in Cases 1 and 2.

From the three cases, we can find that the value of ρ_k contains the least information in Case 1, and a given $g(x)$ may generate entirely the wrong results. In Case 2, the value of ρ_k contains the most information, and is the equivalent of giving the all labels for the corresponding bag. In Case 3, the information contained in ρ_k is between that of Case 1 and that of Case 2. As the result, we need to give reasonable weights for every bag according to the amount of information they contain.

According to the analysis mentioned previously, the weight formula can be described as

$$w_k^{\text{info}} = \frac{|\rho_k - 0.5| + \tau}{\sum_k |\rho_k - 0.5| + \tau} \tag{9}$$

where τ is a given constant. Adding bag's weight, the total weight is

$$\bar{w}_k = \frac{l_k}{N} \frac{|\rho_k - 0.5| + \tau}{\sum_k |\rho_k - 0.5| + \tau}. \tag{10}$$

So, (7) can be rewritten as

$$L_\rho^{\text{MSE-weight}}(\rho, \tilde{\rho}) = \sum_{k=1}^{K} \bar{w}_k (\rho_k - \tilde{\rho}_k)^2. \tag{11}$$

In practice, we use the combination of (8) and (11) to describe the loss of LLP

$$L_\rho(\rho, \tilde{\rho}) = L_\rho^{\text{MSE-weight}} + L_\rho^{\text{prior}}. \tag{12}$$

B. Weak Classifier

After obtaining (12), the next goal is to minimize the loss function. Now, we first introduce the relationship between the loss function and the decision function. Here we use a standard logistic function to estimate conditional probabilities for some weak classifier $f(x_{kj})$

$$P(\rho_{kj}|x_{kj}) = \frac{1}{1 + e^{-af(x_{kj})+b}} \tag{13}$$

where ρ_{kj} is the conditional probabilitiy for the sample x_{kj}, and a and b are two given constants. So the estimated label proportion $\tilde{\rho}_k$ of the kth bag can be given by

$$\tilde{\rho}_k = P(\rho_k|\mathcal{X}_k) = \frac{1}{l_k} \sum_{j=1}^{l_k} P(\rho_{kj}|x_{kj}). \tag{14}$$

According to (14), we will be able to estimate $\tilde{\rho}_k$, and then to compute (12) for a given $f(x)$. Next, we show how to solve the weak classifier. Suppose

$$f(x) = (\alpha \cdot x_{kj}) + \beta \tag{15}$$

where $\alpha \in R^n$, $\beta \in R$, and $x_{kj} \in R^n$ is an instance. Take (15) into (13), and let $a = 1$ and $b = 0$, we can obtain

$$P(\rho_{kj}|x_{kj}) = \frac{1}{1 + e^{-((\alpha \cdot x_{kj})+\beta)}} \tag{16}$$

and (14) is transformed to

$$\tilde{\rho}_k = \frac{1}{l_k} \sum_{j=1}^{l_k} \frac{1}{1 + e^{-((\alpha \cdot x_{kj})+\beta)}}. \tag{17}$$

Further combining (12) and (17), the corresponding loss function can be expressed as

$$L_\rho(\alpha, \beta) = \sum_{k=1}^{K} \bar{w}_k \left(\rho_k - \frac{1}{l_k} \sum_{j=1}^{l_k} \frac{1}{1 + e^{-((\alpha \cdot x_{kj})+\beta)}} \right)^2 + \frac{1}{N^2} \left(\sum_{k=1}^{K} l_k \rho_k - \sum_{k=1}^{K} \sum_{j=1}^{l_k} \frac{1}{1 + e^{-((\alpha \cdot x_{kj})+\beta)}} \right)^2. \tag{18}$$

Finally, we can use steepest descent or conjugate gradient methods to solve α and β.

C. Adaboost-LLP

Let $y_{kj} = F(x_{kj})$ be the predicted label of jth instance in the kth bag x_{kj}, and $F(x_{kj}) = \sum_t \lambda_t f^t(x_{kj})$ can be written as the weighted sum of weak classifiers. The probability of a new instance being positive can be expressed as

$$P(\rho_{kj}|x_{kj}) = \zeta(F(x_{kj})) = \frac{1}{1 + \exp(-y_{kj})}. \tag{19}$$

Define the proportion of labels for some bag

$$P(\mathcal{P}_k|\mathcal{X}_k) = \tilde{\rho}_k = \frac{\sum_{j=1}^{l_k} (\rho_{kj})}{l_k}. \tag{20}$$

By introducing the MLE [41], the likelihood probability $\mathcal{L}(F)$ for all training bags can be expressed as

$$L(F) = \prod_k \tilde{\rho}_k^{\mathcal{P}_k} (1 - \tilde{\rho}_k)^{(1-\mathcal{P}_k)}. \tag{21}$$

Next, according to the AnyBoost learning framework [42], [43], $-L(F)$ can be seen a cost function, so the weight of each example can be written by the derivative of $\log L(F)$

$$\nabla \mathcal{L}(F) = \frac{\partial \log(L(F))}{\partial y_{kj}} = w_{kj} = \frac{\mathcal{P}_k - \tilde{\rho}_k}{\tilde{\rho}_k} \rho_{kj} \left(\frac{1}{l_k} \frac{1 - \rho_{kj}}{1 - \tilde{\rho}_k} \right). \tag{22}$$

Theorem 1 gives the process of the derivation for (22).

Theorem 1: Given $\tilde{\rho}_k = ((\sum_{j=1}^{l_k} (\rho_{kj}))/l_k)$, $L(F) = \prod_k \tilde{\rho}_k^{\mathcal{P}_k} (1 - \tilde{\rho}_k)^{(1-\mathcal{P}_k)}$, the derivative of $\log L(F)$ can be expressed as $((\mathcal{P}_k - \tilde{\rho}_k)/(\tilde{\rho}_k)) \rho_{kj} ((1/l_k)((1-\rho_{kj})/(1 - \tilde{\rho}_k)))$.

Proof:

$$\frac{\partial \log L(F)}{\partial y_{kj}} = \frac{\partial \sum_k (\mathcal{P}_k \log \rho_k + (1 - \mathcal{P}_k) \log(1 - \tilde{\rho}_k))}{\partial y_{kj}} \tag{23}$$

$$= \frac{\partial (\mathcal{P}_k \log \tilde{\rho}_k + (1 - \mathcal{P}_k) \log(1 - \tilde{\rho}_k))}{\partial y_{kj}} \tag{24}$$

$$= \frac{\mathcal{P}_k - \tilde{\rho}_k}{\tilde{\rho}_k (1 - \tilde{\rho}_k)} \frac{\partial \tilde{\rho}_k}{\partial y_{kj}} \tag{25}$$

$$= \frac{\mathcal{P}_k - \tilde{\rho}_k}{\tilde{\rho}_k (1 - \tilde{\rho}_k)} \frac{\partial \frac{k}{l_k} \sum_{j=1}^{l_k} (\rho_{kj})}{\partial y_{kj}} \tag{26}$$

$$= \frac{1}{l_k} \frac{\mathcal{P}_k - \tilde{\rho}_k}{\tilde{\rho}_k (1 - \tilde{\rho}_k)} \frac{\partial \rho_{kj}}{\partial y_{kj}} \tag{27}$$

$$= \frac{1}{l_k} \frac{\mathcal{P}_k - \tilde{\rho}_k}{\tilde{\rho}_k (1 - \tilde{\rho}_k)} \rho_{kj} (1 - \rho_{kj}) \tag{28}$$

$$= \frac{\mathcal{P}_k - \tilde{\rho}_k}{\tilde{\rho}_k} \rho_{kj} \left(\frac{1}{l_k} \frac{1 - \rho_{kj}}{1 - \tilde{\rho}_k} \right). \tag{29}$$

□

Combining Theorem 1 and AnyBoost learning framework, Algorithm 1 gives the detailed steps of Adaboost-LLP. As one can see, this is a Adaboost-style two-layer iterative algorithm. At the inner iterations from 1 to s of outer iteration t, we calculate the inner products of differential of current-combined classifiers $\nabla \mathcal{L}(F_{t-1})$ and every weak learners f_s, which are denoted as $\mathcal{D}_{s,kj}$. After the inner iterations of outer

Algorithm 1 Adaboost-LLP

Input: Training set $\{(\mathcal{X}_1, \mathcal{P}_1), \ldots, (\mathcal{X}_K, \mathcal{P}_K)\}$, where $\mathcal{X}_k = \{x_{k1}, \ldots, x_{kl_k}\}$, $x_{kj} \in R^n$, $j = 1, \ldots, l_k$, $k = 1, \ldots, K$, $\mathcal{P}_k \in [0, 1]$, $F_0(x) := 0$.

For $t = 1, \ldots, T$ **do**

  For $s = 1, \ldots, S$ **do**

    for each k, j, we have:

$$\mathcal{D}_{s,kj} = < \nabla\mathcal{L}(F_{t-1}), f_s(x_{kj}) >,$$

$$\mathcal{D}_s = \sum_{k,j} \mathcal{D}_{s,kj} = \sum_{k,j} \nabla\mathcal{L}(F_{t-1}(x_{kj})) f_s(x_{kj}) = \sum_{k,j} \frac{\mathcal{P}_k - \tilde{\rho}_k}{\tilde{\rho}_k} \rho_{kj} \left(\frac{1}{l_k} \frac{1-\rho_{kj}}{1-\tilde{\rho}_k}\right) f_s(x_{kj}) = \sum_{k,j} w_{kj} f_s(x_{kj}).$$

  End for

  $s^* = \text{argmax}_s(\mathcal{D}_s)$,

  $f_t^{\text{sel}}(x) = f_t^{s^*}(x)$.

  choose α_t,

  $F_t := F_{t-1} + \alpha_t f_t^{\text{sel}}(x)$

End for

Output: The final classifier is £º

$F(x) = \sum_{t=1}^{T} (\alpha_t f_t^{\text{sel}}(x))$,

and $p(x) = \zeta(F(x)) = \frac{1}{1+\exp(-F(x))}$.

iteration t are finished, we select the weak learner $f_t^{\text{sel}}(x)$, which maximizes the $\mathcal{D}_s$. Then we choose α_t by a linear search to minimize $F_{t-1} + \alpha f_t^{\text{sel}}(x)$ and set $F_t := F_{t-1} + \alpha_t f_t^{\text{sel}}(x)$. Finally, when T outer iterations finish, we obtain the final classifier $F(x) = \sum_{t=1}^{T} (\alpha_t f_t^{\text{sel}}(x))$ and the predicted possibility of a instance belonging to the positive class $p(x) = \zeta(F(x)) = (1/(1 + \exp(-F(x))))$.

D. Discussion

In Section II-B, we give detailed steps of our method. The biggest difference between Adaboost-LLP and other existing methods is that we attempt to use the idea of ensemble learning strategy to solve LLP problem. Due to alter-∝SVM outperforming the state-of-the-art, in the following, we take it as the main comparison for computational complexity analysis. For the original model of alter-∝SVM, it is a nonconvex integer programming problem, which is NP-hard. However, Adaboost-LLP is not a NP-hard problem, and is more likely to be solved efficiently. So our algorithm is superior to alter-∝SVM from this perspective. In order to overcome the complexity of integer programming in alter-∝SVM, Yu *et al.* [23] give a simple alternating optimization method to solve it approximately. Of course, it suffers sacrifice on precision. According to [23], alter-∝SVM is solved by a standard SVM's quadratic programming problem and an integer programming problem in each iteration. We suppose that the first step's time complexity is $\mathcal{O}(n^3)$ (n is the number of variables),[1] and the integer programming problem can be solved in $\mathcal{O}(n\log(J))$ time, $J = \max_{k=1,\ldots,K}|\mathcal{X}_k|$, and n denotes the number of all samples. So its running time is proportion to ♯iteration × $(\mathcal{O}(n^3)+\mathcal{O}(n\log(J)))$ (Note: ♯iteration also contains iterations of choosing the optimal parameters). For Adaboost-LLP, we suppose that the complexity of weak classifier is $\mathcal{O}(n^3)$. So Adaboost-LLP's running time is proportion to ♯iteration × $\mathcal{O}(n^3)$. Considering from the two formulas themselves, alter-∝SVM has one more item $\mathcal{O}(n\log(J))$. However, since we do not know its specific numbers of iterations, it is hard to judge, which is faster in the training time. But, we can find that the two algorithms' training time is in the same amounts level from the theoretical analysis result. In the experiments, we will give the final answer which has more advantages in the training time.

For the precision analysis, which we have already mentioned above, alter-∝SVM sacrifices some precision in order to obtain a fast solving algorithm, resulting in a disadvantage. In the model construction aspect, alter-∝SVM uses the maximal margin method to solve LLP problem. Adaboost-LLP takes a standard logistic function to estimate the weak classifier's conditional probabilities, and then uses the MLE to build an LLP model. In addition, we give different weights to

[1]LIBSVM's solving method is able to obtain a smaller time complexity. However, to be fair, here we only compare them in the common case of not using special optimization method

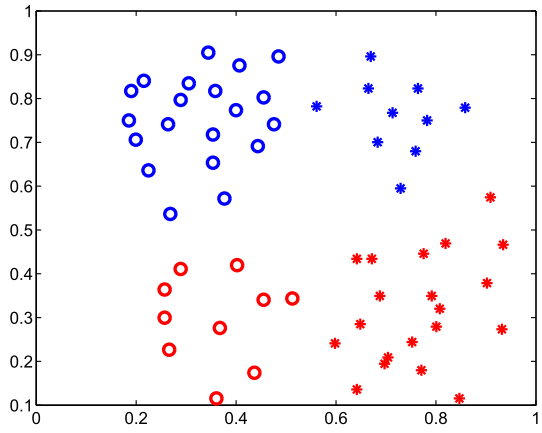


Fig. 2. First toy data. "Blue" and "red" denote the first and second bags, respectively. The "o" and "*" denote the positive and negative classes, respectively.

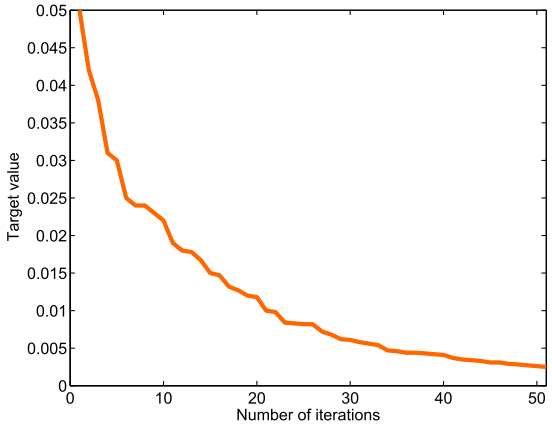
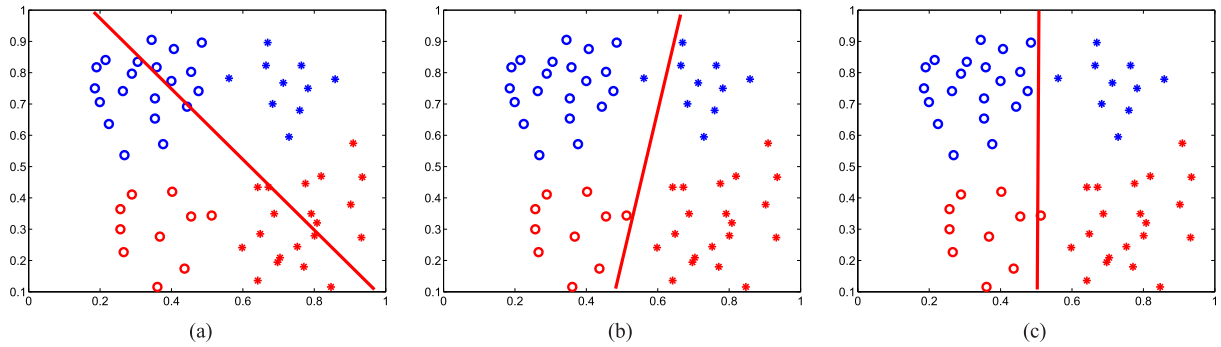


Fig. 3. Change of object value with the increase of iterations.



(a) (b) (c)

Fig. 4. Results under different iterations. With the increase of the number of iterations, the result is getting better and better. After 51st iteration, we get the best boundary. (a) Result at 14th iteration. (b) Result at 27th iteration. (c) Result at 51st iteration.

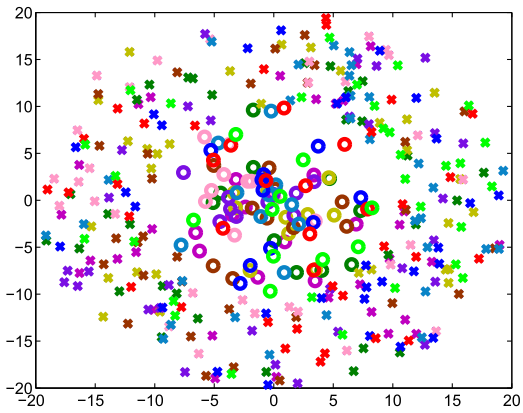


Fig. 5. Second toy data. "o" and "*" denote positive class and negative class, and the different colors denote the different classes, respectively.

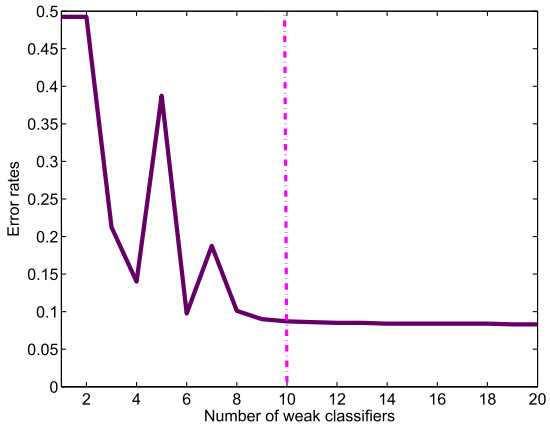


Fig. 6. Error rate changes as the number of weak classifiers grows on the second toy data. When the number of weak classifiers is larger than 10, the error rate is almost no change.

different bags according to three factors: the size of bag, the whole proportions information of the whole training set, and the amount of useful information each bag contains, which is better to use the proportions labels information, and our algorithm has an obvious advantage at this point. The extensive experiments demonstrate the state-of-the-art of Adaboost-LLP.

TABLE I

BASIC INFORMATION OF UCI DATA SETS USED IN EXPERIMENTS. $\mathcal{P}_k$ DENOTES THE PROPORTION OF THE SELECTED POSITIVE CLASS

Dataset	Size	#Attributes	#Classes	$\mathcal{P}_k$
vowel	180	10	2	31.47%
heart	270	13	2	44.44%
colic	368	22	2	24.18%
vote	435	16	2	46.55%
australian	690	14	2	67.83%
letter	1555	16	2	49.26%
dna	2000	180	3	48.89%
satimage	4435	36	6	23.20%
acoustic	78823	50	3	24.17%
combined	78823	100	3	23.22%
connect-4	67557	126	3	65.83%
covtype	581012	54	7	36.46%

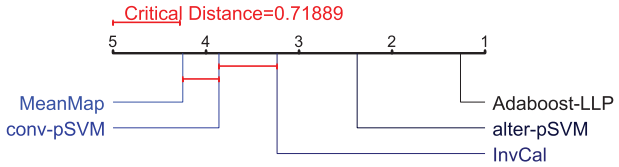


Fig. 7. Statistical significance diagram. CD diagram for Nemenyi tests performed on all the 12 data sets. Average ranks of examined methods are presented. Bold lines indicate groups of classifiers which are not significantly different from each other (their average ranks differ by less than CD value).

III. EXPERIMENT

Running environment: MATLAB 2010 on a PC with an Intel Core I5 processor and 4 GB RAM.

A. Toy Data Set

In this section, at first, we generate the first toy data to evaluate the performance of the weak classifier (18). The toy data includes two bags, which are shown by the blue and red points. "o" and "*" denote different classes, respectively (see Fig. 2). ρ_1 of the first bag is 0.67, and ρ_2 is 0.33. Our goal is to classify the two classes as accurately as possible according to the given ρ_1 and ρ_2. As the result, we get the minimum value of (18) over 51 iterations. Fig. 3 shows the iterative process and Fig. 4 shows the classification's result under different iterations. From Fig. 3, it is easy to find that the toy data are nearly correctly classified in the case of only using the proportions information, which indicates that our weak

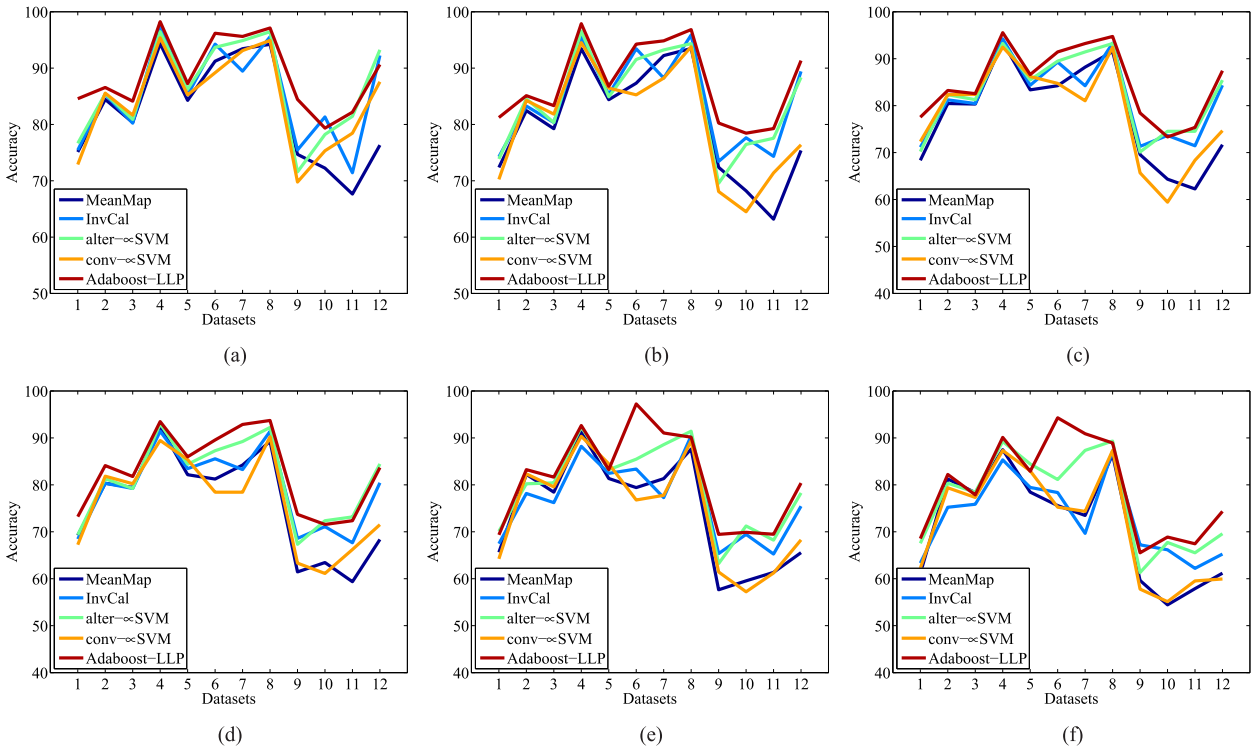


Fig. 8. Accuracy comparison on different data sets. Each subfigure denotes the result under different bags. x-axis—1: vowel. 2: heart. 3: colic. 4: vote. 5: australian. 6: letter. 7: dna. 8: satimage. 9: acoustic. 10: combined. 11: connect-4. 12: covtype. (a) Sizes of bag: 2. (b) Sizes of bag: 4. (c) Sizes of bag: 8. (d) Sizes of bag: 16. (e) Sizes of bag: 32. (f) Sizes of bag: 64.

classifier is reasonable. Of course, this is only an intuitive example; in the following, we will conduct more experiments to verify the effectiveness of Adaboost-LLP.

Next, we generate the second toy data to evaluate the performance of Adaboost-LLP. Similarly, it is also binary classification of including ten bags (seeing Fig. 5, different colors represent different bags. "o" is positive class composed of 100 points, and "*" is negative class composed of 100 points). Fig. 6 gives the result based on ten weak classifiers. We noticed that the error ratio of the strong classifier reduces to 8.7%, and as the growing of number of weak classifiers, the corresponding error ratio is steadily dropping, which shows convergence our algorithm in practice.

B. UCI Data Sets

In this section, we use 12 UCI repository data sets[2] (see Table I).

Data sets: "heart," "colic," "vote," "australian," "letter," and "satimage" are used in [23], so we first choose them. The size of "vowel" is 180, but it has 11 classes, so we pick up the data set. And for the same reason, we select "dna" data set. In addition, we also select some big size's data sets: "acoustic," "combined," "connect-4," and "covtype." Among them, "combined" and "connect-4" have 100 and 126 features, respectively. The "connect-4" and "covtype" have sparse features, and all data of "connect-4" is only composed of 0 and 1. Therefore, our data sets have a good diversity.

[2]http://archive.ics.uci.edu/ml/,http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/

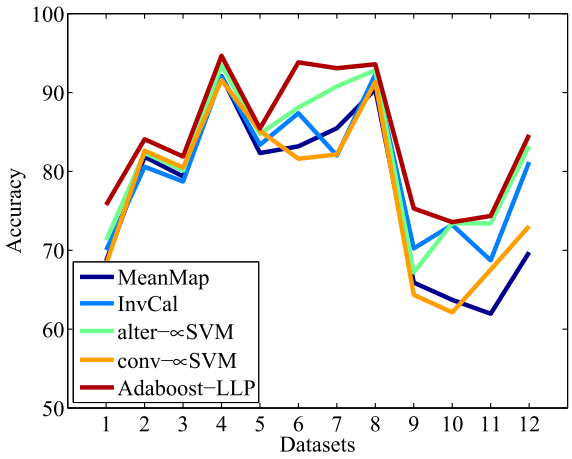


Fig. 9. Average accuracy comparison on different data sets. The average accuracy denotes the mean of all bags' accuracies for some data set. x-axis–1: vowel. 2: heart. 3: colic. 4: vote. 5: australian. 6: letter. 7: dna. 8: satimage. 9: acoustic. 10: combined. 11: connect-4. 12: covtype.

The detailed setup is as follows: the experiment only solves the binary classification. For multiclassification, we use one-vs-rest method to convert it into the binary classification. Attributes of each data set are scaled to $[-1, 1]$. Each bag is randomly selected from the original data set, whose size is separately set to $2, 4, 8, 16, 32, 64$. All experiments are conducted with fivefold cross validation. We compute the average classification accuracy with standard deviations by repeating

TABLE II

FIVEFOLD CROSS VALIDATION'S RESULTS UNDER THE OPTIMAL PARAMETERS. BLACK-FACE LETTERS DENOTE THE BEST ACCURACIES

Dataset	Method	2	4	8	16	32	64
vowel	MeanMap	75.11 ± 1.08	72.36 ± 1.28	68.35 ± 1.32	69.24 ± 1.55	65.63 ± 1.74	61.12 ± 1.26
	InvCal	75.44 ± 0.26	74.26 ± 1.09	71.15 ± 1.15	68.45 ± 1.98	67.46 ± 1.35	63.32 ± 1.29
	alter-∝SVM	76.64 ± 1.23	73.86 ± 1.21	70.21 ± 1.22	69.35 ± 1.54	70.12 ± 1.32	67.54 ± 1.89
	conv-∝SVM	72.88 ± 1.66	70.24 ± 0.23	72.34 ± 2.28	67.24 ± 1.34	64.24 ± 1.33	62.22 ± 1.80
	Adaboost-LLP	**84.56 ± 1.43**	**81.23 ± 0.72**	**77.54 ± 0.61**	**73.23 ± 0.66**	**69.34 ± 0.24**	**68.55 ± 0.09**
heart	MeanMap	84.43 ± 1.71	82.46 ± 1.32	80.45 ± 1.22	80.34 ± 1.22	82.23 ± 2.44	81.25 ± 0.86
	InvCal	85.34 ± 2.26	83.26 ± 2.21	81.23 ± 2.12	80.32 ± 2.84	78.16 ± 1.32	75.22 ± 2.20
	alter-∝SVM	85.44 ± 1.84	84.56 ± 1.43	82.31 ± 1.46	81.31 ± 2.24	80.22 ± 2.34	80.34 ± 1.34
	conv-∝SVM	85.56 ± 1.26	84.25 ± 1.22	82.34 ± 0.88	81.84 ± 1.22	82.34 ± 1.89	79.37 ± 1.33
	Adaboost-LLP	**86.56 ± 0.81**	**85.11 ± 0.40**	**83.24 ± 0.61**	**84.10 ± 0.87**	**83.24 ± 0.44**	**82.22 ± 2.11**
colic	MeanMap	80.35 ± 1.44	79.24 ± 1.14	80.34 ± 1.21	79.44 ± 1.24	78.44 ± 1.46	78.34 ± 1.34
	InvCal	80.22 ± 1.21	80.34 ± 1.56	80.45 ± 1.54	79.23 ± 2.67	76.21 ± 2.46	75.86 ± 4.91
	alter-∝SVM	80.88 ± 2.11	80.34 ± 0.88	81.28 ± 0.33	79.22 ± 2.36	80.43 ± 1.21	**78.44 ± 1.36**
	conv-∝SVM	81.64 ± 1.23	81.83 ± 2.09	82.38 ± 0.46	80.24 ± 2.17	79.56 ± 2.23	77.31 ± 1.76
	Adaboost-LLP	**84.12 ± 0.44**	**83.33 ± 1.24**	**82.56 ± 0.89**	**81.77 ± 2.12**	**81.64 ± 0.45**	77.87 ± 1.30
vote	MeanMap	94.34 ± 1.34	93.44 ± 1.25	93.54 ± 1.21	92.43 ± 1.53	91.28 ± 1.25	87.48 ± 2.31
	InvCal	97.48 ± 1.21	95.47 ± 1.48	94.23 ± 2.45	91.43 ± 2.32	88.21 ± 1.33	85.33 ± 3.79
	alter-∝SVM	96.45 ± 2.24	96.58 ± 2.21	93.26 ± 2.88	93.01 ± 1.64	92.32 ± 1.31	89.23 ± 1.24
	conv-∝SVM	95.39 ± 3.21	94.51 ± 2.44	92.47 ± 1.63	89.45 ± 1.37	90.34 ± 0.46	87.33 ± 1.64
	Adaboost-LLP	**98.24 ± 0.31**	**97.89 ± 0.66**	**95.55 ± 0.75**	**93.48 ± 0.72**	**92.64 ± 0.48**	**90.11 ± 2.21**
australian	MeanMap	84.27 ± 1.34	84.38 ± 2.37	83.36 ± 2.29	82.16 ± 1.88	81.34 ± 1.78	78.45 ± 3.41
	InvCal	85.25 ± 1.35	85.31 ± 0.46	84.35 ± 1.48	83.45 ± 1.67	82.45 ± 1.22	79.44 ± 1.78
	alter-∝SVM	86.24 ± 2.24	84.89 ± 0.56	85.46 ± 2.67	84.37 ± 0.66	83.22 ± 1.78	**84.44 ± 1.28**
	conv-∝SVM	85.24 ± 1.53	86.46 ± 0.34	86.20 ± 1.71	85.38 ± 1.78	**84.49 ± 2.44**	82.98 ± 1.32
	Adaboost-LLP	**87.24 ± 0.54**	**86.78 ± 0.77**	**86.61 ± 0.27**	**85.98 ± 0.33**	83.24 ± 0.87	82.88 ± 1.24
letter	MeanMap	91.23 ± 0.57	87.32 ± 1.54	84.25 ± 1.33	81.23 ± 1.34	79.42 ± 2.03	75.58 ± 1.42
	InvCal	94.25 ± 1.25	93.43 ± 1.88	89.29 ± 1.95	85.55 ± 1.66	83.37 ± 1.21	78.36 ± 1.21
	alter-∝SVM	93.69 ± 0.39	91.53 ± 2.15	89.48 ± 0.88	87.27 ± 1.73	85.46 ± 1.03	81.14 ± 1.11
	conv-∝SVM	89.18 ± 1.87	85.25 ± 0.87	84.74 ± 1.28	78.44 ± 2.51	76.78 ± 1.44	75.24 ± 1.28
	Adaboost-LLP	**96.18 ± 1.56**	**94.25 ± 1.54**	**91.44 ± 1.28**	**89.55 ± 3.75**	**97.24 ± 0.27**	**94.29 ± 2.21**
dna	MeanMap	93.44 ± 1.24	92.22 ± 1.24	88.20 ± 1.24	84.21 ± 1.64	81.32 ± 1.90	73.48 ± 2.40
	InvCal	89.45 ± 2.21	88.23 ± 1.23	84.23 ± 1.45	83.25 ± 1.16	77.27 ± 1.90	69.66 ± 1.46
	alter-∝SVM	94.88 ± 1.32	93.23 ± 1.45	91.44 ± 0.89	89.23 ± 1.53	88.56 ± 1.23	87.34 ± 1.68
	conv-∝SVM	93.11 ± 1.57	88.22 ± 1.81	81.03 ± 3.58	78.44 ± 2.51	77.77 ± 4.93	74.34 ± 1.38
	Adaboost-LLP	**95.61 ± 0.52**	**94.84 ± 0.55**	**93.24 ± 0.88**	**92.87 ± 3.25**	**91.04 ± 0.47**	**90.89 ± 2.14**
satimage	MeanMap	94.22 ± 2.38	93.47 ± 1.28	91.50 ± 1.25	89.25 ± 2.26	87.62 ± 1.22	86.44 ± 0.62
	InvCal	95.56 ± 2.23	95.87 ± 1.73	93.32 ± 2.23	91.39 ± 1.34	90.45 ± 1.32	87.23 ± 0.24
	alter-∝SVM	96.46 ± 1.89	94.34 ± 1.08	93.23 ± 4.21	92.22 ± 1.38	**91.43 ± 1.34**	**89.33 ± 2.90**
	conv-∝SVM	94.90 ± 2.56	93.78 ± 1.98	92.43 ± 1.34	90.24 ± 1.66	89.36 ± 3.25	87.33 ± 0.33
	Adaboost-LLP	**97.12 ± 0.45**	**96.83 ± 0.42**	**94.74 ± 0.34**	**93.71 ± 0.23**	90.15 ± 0.89	88.92 ± 0.78
acoustic	MeanMap	74.68 ± 2.55	72.39 ± 3.12	69.57 ± 2.45	61.45 ± 1.34	57.65 ± 2.52	59.64 ± 2.44
	InvCal	75.46 ± 3.13	73.43 ± 2.03	71.32 ± 1.28	68.59 ± 2.23	65.35 ± 2.32	**67.21 ± 2.24**
	alter-∝SVM	71.57 ± 2.23	69.54 ± 1.65	70.13 ± 2.27	67.33 ± 1.54	63.22 ± 1.56	61.33 ± 2.30
	conv-∝SVM	69.77 ± 2.68	68.09 ± 2.49	65.68 ± 2.22	63.33 ± 3.07	61.43 ± 2.24	57.78 ± 1.23
	Adaboost-LLP	**84.44 ± 0.67**	**80.24 ± 1.22**	**78.44 ± 1.34**	**73.71 ± 1.43**	**69.45 ± 0.33**	65.52 ± 2.12
combined	MeanMap	72.24 ± 1.25	68.21 ± 2.45	64.33 ± 1.57	63.46 ± 1.84	59.55 ± 1.90	54.44 ± 1.48
	InvCal	**81.33 ± 3.13**	77.63 ± 1.33	73.65 ± 2.24	71.11 ± 1.49	69.45 ± 1.11	66.17 ± 1.55
	alter-∝SVM	78.22 ± 2.23	76.45 ± 2.12	**74.53 ± 1.27**	72.36 ± 1.09	**71.24 ± 1.46**	67.73 ± 2.37
	conv-∝SVM	75.32 ± 1.38	64.49 ± 1.46	59.44 ± 2.52	61.13 ± 1.55	57.22 ± 1.89	55.12 ± 1.25
	Adaboost-LLP	79.34 ± 2.27	**78.44 ± 1.56**	73.34 ± 1.89	**71.56 ± 1.73**	69.88 ± 2.34	**68.88 ± 2.42**
connect-4	MeanMap	67.64 ± 2.11	63.18 ± 1.57	62.26 ± 1.33	59.37 ± 0.33	61.35 ± 1.33	57.88 ± 1.98
	InvCal	71.42 ± 2.22	74.33 ± 2.13	71.45 ± 2.12	67.66 ± 1.29	65.25 ± 1.56	62.21 ± 1.21
	alter-∝SVM	81.44 ± 1.32	77.55 ± 1.32	74.53 ± 1.27	**73.16 ± 1.27**	68.22 ± 1.73	65.53 ± 2.11
	conv-∝SVM	78.44 ± 2.18	71.39 ± 2.26	68.37 ± 1.48	66.22 ± 1.81	61.22 ± 2.29	59.54 ± 1.09
	Adaboost-LLP	**82.17 ± 1.88**	**79.26 ± 1.04**	**75.35 ± 1.06**	72.36 ± 2.11	**69.48 ± 2.24**	**67.44 ± 1.23**
covtype	MeanMap	76.33 ± 2.42	75.38 ± 1.25	71.67 ± 1.78	68.37 ± 0.23	65.55 ± 1.50	61.18 ± 1.11
	InvCal	92.24 ± 1.05	89.41 ± 1.23	84.34 ± 1.99	80.44 ± 1.88	75.45 ± 1.54	65.25 ± 1.26
	alter-∝SVM	**93.24 ± 1.06**	88.35 ± 1.89	85.43 ± 1.32	**84.46 ± 1.48**	78.28 ± 1.83	69.59 ± 2.91
	conv-∝SVM	87.57 ± 0.45	76.39 ± 2.09	74.67 ± 1.66	71.52 ± 1.51	68.28 ± 2.28	59.94 ± 1.07
	Adaboost-LLP	90.63 ± 0.48	**91.33 ± 1.35**	**87.44 ± 1.44**	83.67 ± 1.16	**80.38 ± 1.14**	**74.34 ± 2.23**

the above processes five times. Algorithm's parameters are tuned by the fivefold cross validation on the training subset. Parameter settings of all algorithms are shown as follows.

MeanMap: $\lambda \in [0.1, 1, 10]$.

InvCal: $C_p \in [0.1, 1, 10]$, $\varepsilon \in [0, 0.01, 0.1]$.

alter-∝SVM: $C \in [0.1, 1, 10]$, $C_p \in [1, 10, 100]$.

conv-∝SVM: $C \in [0.1, 1, 10]$, $\varepsilon \in [0, 0.01, 0.1]$.

LLP-NPSVM: $C \in [0.01, 0.1, 1]$, $\varepsilon \in [0.001, 0.01, 0.1]$.

For InvCal, alter-∝SVM conv-∝SVM, only RBF kernel is considered: $\gamma = [0.01, 0.1, 1]$.

Table II and Figs. 8 and 9 give the final comparison results with other methods. Adaboost-LLP outperforms Meanmap, InvCal, alter-∝SVM, and conv-∝SVM in the most cases. For heart, vote, and dna data sets, our method completely outperforms other methods. As seen from Fig. 9, especially for dna data set, the accuracy of our method is average higher 1% than that of second best method: alter-∝SVM. Except for Adaboost-LLP, alter-∝SVM also obtain better results than MeanMap, InvCal and alter-∝SVM. For more challenging case of 64 bag sizes, the results of both the two

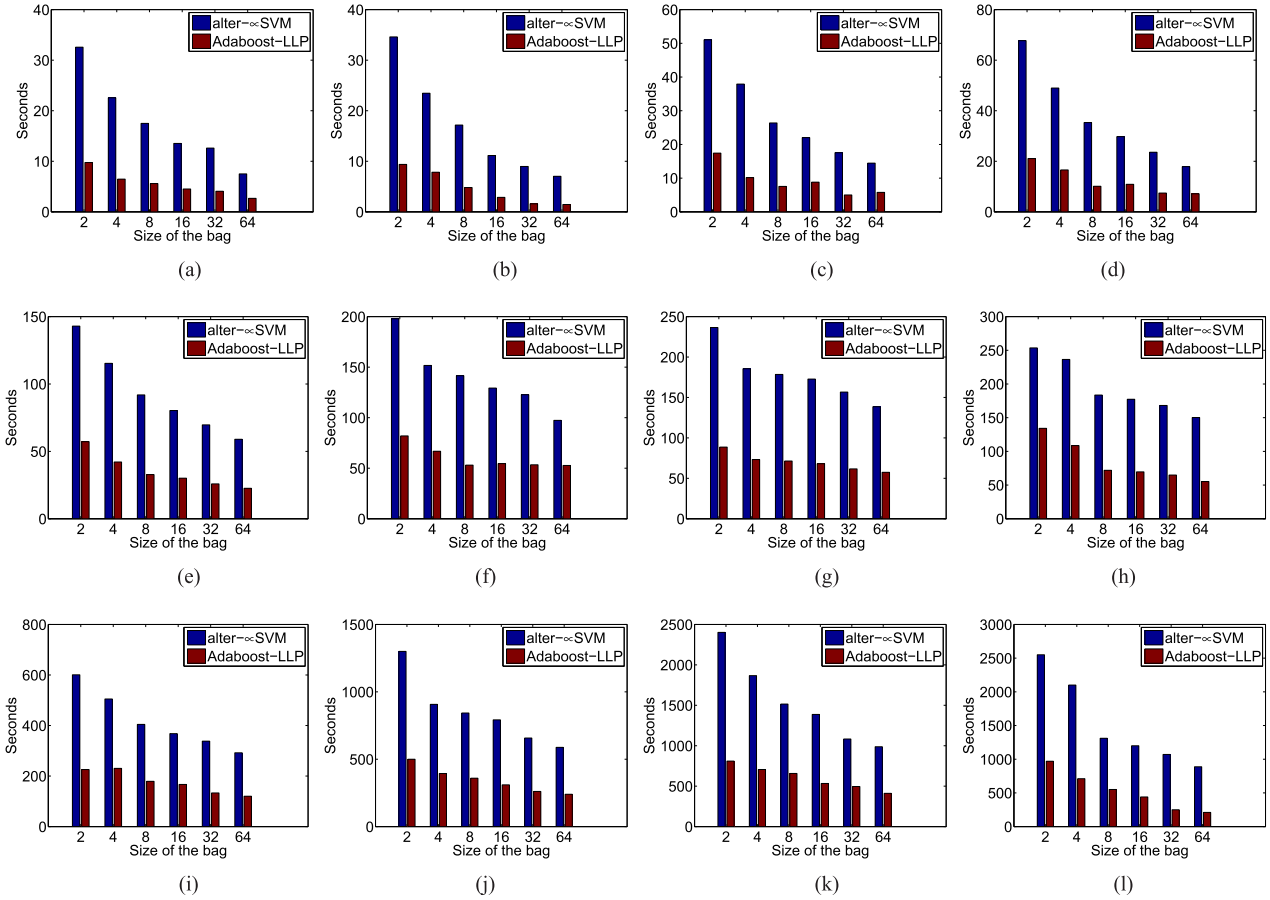


Fig. 10. Time's comparison between Adaboost-LLP and alter-$\propto$ SVM. (a) heart. (b) colic. (c) vote. (d) australian. (e) dna. (f) satimage. (g) dna. (h) satimage. (i) dna. (j) satimage. (k) dna. (l) satimage.

maximum margin methods and Adaboost-LLP are superior to that MeanMap and InvCal, at the same time, Adaboost-LLP is slightly better than alter-$\propto$SVM and conv-$\propto$SVM.

In order to statistically evaluate differences between methods, Friedman rank test with post-hoc Nemenyi tests is carried out for results in Table II. Friedman rank test [44] at significance level $\alpha = 0.05$ is carried out with accuracy being chosen as a performance metric to evaluate the systematic difference among all the methods. We tested Adaboost-LLP and baselines on all the 12 data sets and the results are shown in Fig. 7. From the critical difference (CD) diagram for post-hoc Nemenyi tests [44] in Fig. 7, we can see that Adaboost-LLP outperforms all the other methods, and alter-$\propto$SVM is the second best one. These two methods are significantly better than the other three. MeanMap is confirmed as the least accurate classifier. The difference between MeanMap and InvCal is significant; however, that between MeanMap and conv-$\propto$SVM is not.

Fig. 10 gives the final training time comparison between Adaboost-LLP and alter-$\propto$SVM. The training speed of Adaboost-LLP is about three times of alter-$\propto$SVM. Especially, the advantage of Adaboost-LLP is more obvious when the size of the bag is relatively small. The main reason is that the iterations of our algorithm are much fewer than that of alter-$\propto$SVM. As the alter-$\propto$SVM, conv-$\propto$SVM, and Adaboost-LLP outperform others methods, we can see that it is helpful to improve the accuracy of classification by making restrictive assumptions for training set. The success of Adaboost-LLP shows that solving LLP problem in the ensemble learning framework has more advantages, and is worthy of further study and exploitation.

C. Weighting Mechanisms Comparison

As mentioned previously, our method benefits from incorporating extra weights information into loss function. In this section, we test the effect of weights in Adaboost-LLP. Particpating data sets include: "heart," "colic," "vote," "australian," "letter," and "satimage." Adaboost-LLP-1 denotes the method with the weight of (7); Adaboost-LLP-2 denotes the one with the weight of (7) and (8); Adaboost-LLP-3 denotes the one with weight of (12). Fig. 11 give the Adaboost-LLP's results under different weights of bags. With the help of weight formula (8), the accuracy of Adaboost-LLP-2 increased about 0.24%. With the help of weight formula (12), the accuracy of Adaboost-LLP-3 again increased about 0.48% on the basis of Adaboost-LLP-2, which indicate that our algorithm benefits from adding these extra weights information.

D. Animals With Attributes Data Set

In this section, we will apply our algorithm into the attribute-based representation. The Animals with Attributes (AwAs) data set contains 30475 images of

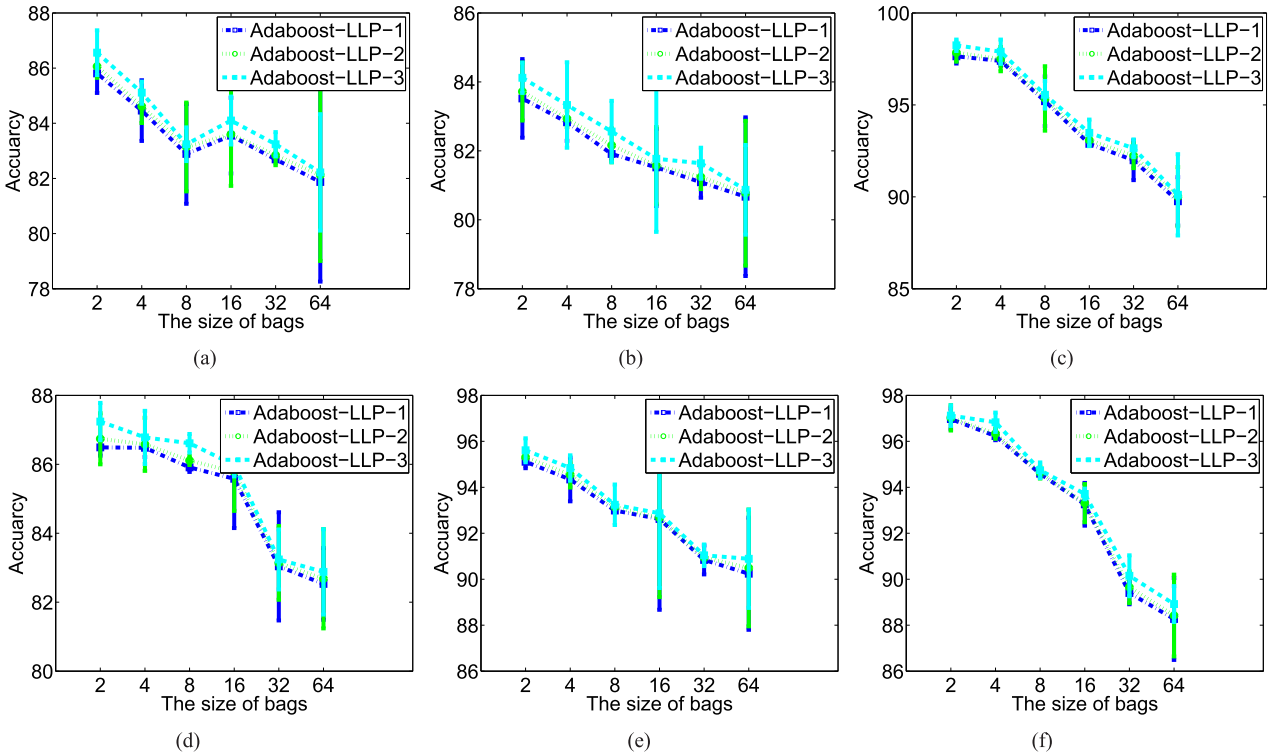


Fig. 11. Accuracy comparison on the different weights of bags. Adaboost-LLP-1 denotes the one with the weight of (7); Adaboost-LLP-2 denotes the one with the weight of (7) and (8); Adaboost-LLP-3 denotes the one with the weight of (12). With the help of weight formula (8), the accuracy of Adaboost-LLP-2 increases about 0.24%. With the help of weight formula (12), the accuracy of Adaboost-LLP-3 again increases about 0.48% on the basis of Adaboost-LLP-2, which indicate that our algorithm benefits from adding these extra weights' information. (a) Heart. (b) Colic. (c) Vote. (d) Australian. (e) Dna. (f) Satimage.

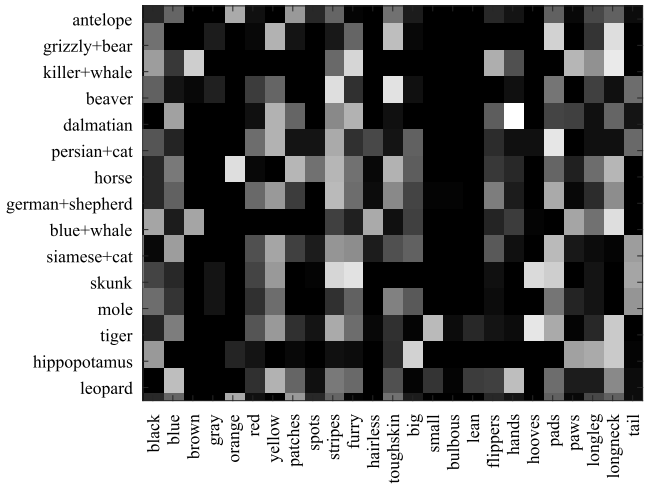


Fig. 12. Class-attribute matrices of the AwAs data set.

50 animal categories [45]. In addition, in the data set, there is a 50×85 class-attribute matrix (see Fig. 12). The goal is to classify unknown classes by learning known classes (more detailed description can be found in [45]). In [45], they first thresholds the matrix to 0-1 matrix, and then they train 85 attribute classifiers for each attribute. Yu *et al.* [39] consider that the binarization step leads to a sever information loss. So they treat the similarity matrix as a category-attribute proportion, and use $\propto$SVM to train the attribute models. Experiments show that their models are better than that of [45]. Here, we adopt the method of [45] to construct our attribute models by Adaboost-LLP. The low-level features are provided in http://attributes.kyb.tuebingen.mpg.de/. The experiment is performed with random class split using fivefold cross validation, and 40 classes are taken as the training set and remaining ten classes as test set. The quality of the prediction is measured by the normalized multiclass accuracy (MC acc.). Table III gives the final result. Our algorithm improves about 6.4% than the method of [45] and improves about 2.1% than alter-$\propto$SVM.

TABLE III
RESULTS ON THE AWAS DATA SET

Method	[45][3]	alter-$\propto$SVM	Adaboost-LLP
MC acc.	41.4	45.7	47.8

IV. CONCLUSION

The main contribution of this paper is that we give another alternative learning system to solve LLP problem. In the course of building the loss function, we consider three factors to set the corresponding weights: the size of bag, the whole proportions information of the whole training set, and the amount of useful information that each bag contains, which is very helpful to improve the performance. Introducing the

standard logistic function enables the output of weak classifiers to describe each bag's proportion label. According to the principle of MLE, we construct the ensemble learning method for LLP in the AnyBoost framework. Unlike existing methods, our method does not make any restrictive assumptions on training set and exploits extra weight' information based on the proportion of bags. All experiments show that our method outperforms the existing methods in both accuracy and training time. In the future, one possible work is to extend Adaboost-LLP to an online learning problem. LLP problem with data noises is also interesting under our consideration.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers and handling editors for their careful work and thoughtful suggestions that have helped improve this paper substantially.

REFERENCES

[1] Y.-H. Shao, C.-H. Zhang, X.-B. Wang, and N.-Y. Deng, "Improvements on twin support vector machines," *IEEE Trans. Neural Netw.*, vol. 22, no. 6, pp. 962–968, Jun. 2011.

[2] W. Bian and X. Chen, "Neural network for nonsmooth, nonconvex constrained minimization via smooth approximation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 545–556, Mar. 2014.

[3] Z. B. Xu, R. Zhang, and W. F. Jing, "When does online BP training converge?" *IEEE Trans. Neural Netw.*, vol. 20, no. 10, pp. 1529–1539, Oct. 2009.

[4] D. Wang, J. Zhai, H. Zhu, and X. Wang, "An improved approach to ordinal classification," in *Machine Learning and Cybernetics*. Springer, 2014, pp. 33–42.

[5] G. Huang, S. Song, J. N. D. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2405–2417, Dec. 2014.

[6] J. Yu and P. Hao, "Comments on 'the multisynapse neural network and its application to fuzzy clustering,"' *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 777–778, May 2005.

[7] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 1998, pp. 570–576.

[8] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, no. 1, pp. 31–71, 1997.

[9] A. Blum and A. Kalai, "A note on learning from multiple-instance examples," *Mach. Learn.*, vol. 30, no. 1, pp. 23–29, 1998.

[10] S. Sabato and N. Tishby, "Multi-instance learning with any hypothesis class," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 2999–3039, 2012.

[11] B. Babenko, N. Verma, P. Dollar, and S. J. Belongie, "Multiple instance learning with manifold bags," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 81–88.

[12] M.-L. Zhang and Z.-H. Zhou, "Improve multi-instance neural networks through feature selection," *Neural Process. Lett.*, vol. 19, no. 1, pp. 1–10, 2004.

[13] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li, "Multi-instance multi-label learning," *Artif. Intell.*, vol. 176, no. 1, pp. 2291–2320, 2012.

[14] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 561–568.

[15] J. D. Keeler, D. E. Rumelhart, and W. K. Leow, "Integrated segmentation and recognition of hand-printed numerals," in *Proc. Adv. Neural Inf. Process. Syst.*, 1991, pp. 557–563.

[16] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1. Sep. 2002, pp. I-900–I-903.

[17] Z.-H. Zhou and M.-L. Zhang, "Multi-instance multi-label learning with application to scene classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1609–1616.

[18] H. Kuck and N. de Freitas, "Learning about individuals from group statistics," in *Proc. 21st UAI*, 2005, pp. 332–339.

[19] B.-C. Chen, L. Chen, R. Ramakrishnan, and D. R. Musicant, "Learning from aggregate views," in *Proc. IEEE 22nd Int. Conf. Data Eng. (ICDE)*, 2006, pp. 1–3.

[20] D. R. Musicant, J. M. Christensen, and J. F. Olson, "Supervised learning by training on aggregate outputs," in *Proc. 7th IEEE Int. Conf. Data Mining (ICDM)*, Sep. 2007, pp. 252–261.

[21] N. Quadrianto, A. J. Smola, T. S. Caetano, and Q. V. Le, "Estimating labels from label proportions," *J. Mach. Learn. Res.*, vol. 10, pp. 2349–2374, Apr. 2009.

[22] S. Rueping, "SVM classifier estimation from group probabilities," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 911–918.

[23] F. X. Yu, D. Liu, S. Kumar, T. Jebara, and S.-F. Chang. (Jun. 2013). "$\propto$SVM for learning with label proportions." [Online]. Available: https://arxiv.org/abs/1306.0886

[24] M. Stolpe and K. Morik, "Learning from label proportions by optimizing cluster model selection," in *Machine Learning and Knowledge Discovery in Databases*. Berlin, Germany: Springer-Verlag, 2011, pp. 349–364.

[25] K. Fan, H. Zhang, S. Yan, L. Wang, W. Zhang, and J. Feng, "Learning a generative classifier from label proportions," *Neurocomputing*, vol. 139, pp. 47–55, Apr. 2014.

[26] G. Patrini, R. Nock, T. Caetano, and P. Rivera, "(Almost) no label no cry," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 190–198.

[27] F. X. Yu, K. Choromanski, S. Kumar, T. Jebara, and S.-F. Chang. (Feb. 2014). "On learning from label proportions." [Online]. Available: https://arxiv.org/abs/1402.5902

[28] J. Hernández-González, I. Inza, and J. A. Lozano, "Learning Bayesian network classifiers from label proportions," *Pattern Recognit.*, vol. 46, no. 12, pp. 3425–3440, 2013.

[29] S. Chen, B. Liu, M. Qian, and C. Zhang, "Kernel k-means based framework for aggregate outputs classification," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Sep. 2009, pp. 356–361.

[30] T. Cour, B. Sapp, and B. Taskar, "Learning from partial labels," *J. Mach. Learn. Res.*, vol. 12, pp. 1501–1536, Apr. 2011.

[31] Z. Wang and J. Feng, "Multi-class learning from class proportions," *Neurocomputing*, vol. 119, pp. 273–280, Oct. 2013.

[32] T. Ni, F.-L. Chung, and S. Wang, "Support vector machine with manifold regularization and partially labeling privacy protection," *Inf. Sciences*, vol. 294, pp. 390–407, Feb. 2015.

[33] J. Hernández and I. Inza, "Learning naive Bayes models for multiple-instance learning with label proportions," in *Advances in Artificial Intelligence*. Berlin, Germany: Springer-Verlag, 2011, pp. 134–144.

[34] K. Balasubramanian, P. Donmez, and G. Lebanon, "Unsupervised supervised learning II: Margin-based classification without labels," *J. Mach. Learn. Res.*, vol. 12, pp. 3119–3145, Nov. 2011.

[35] D. Kotzias, M. Denil, N. De Freitas, and P. Smyth, "From group to individual labels using deep features," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 597–606.

[36] R. Nock, G. Patrini, and A. Friedman, "Rademacher observations, private data, and boosting," in *Proc. 32nd ICML*, 2015, pp. 948–956.

[37] Y. Jiang, Z. Deng, S. Wang, and T. Ni, "Training probability transductive classifiers for group probability datasets based on regression," in *Proc. 10th Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD)*, 2013, pp. 444–449.

[38] T. Chen, F. X. Yu, J. Chen, Y. Cui, Y.-Y. Chen, and S.-F. Chang, "Object-based visual sentiment concept analysis and application," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 367–376.

[39] F. X. Yu *et al.*, "Modeling attributes from category-attribute proportions," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 977–980.

[40] K.-T. Lai, F. X. Yu, M.-S. Chen, and S.-F. Chang, "Video event detection by inferring temporal instance labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Sep. 2014, pp. 2251–2258.

[41] F. Scholz, "Maximum likelihood estimation," in *Encyclopedia of Statistical Sciences*. Hoboken, NJ, USA: Wiley, 2006.

[42] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting algorithms as gradient descent," in *Proc. NIPS*, 1999, pp. 1–7.

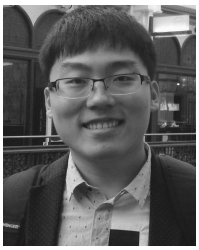
[43] C. Zhang, J. C. Platt, and P. A. Viola, "Multiple instance boosting for object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1417–1424.

[44] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

[45] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 453–465, Mar. 2014.

Zhiquan Qi received the master's and Ph.D. degrees from the College of Science, China Agricultural University, Beijing, China, in 2006 and 2011, respectively.

He is currently a Research Assistant with the Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing. His current research interests include data mining and the application in weak label learning.



Fan Meng received the bachelor's degree from Peking University, Beijing, China, in 2012 and the Ph.D. degree in management science and engineering from the University of Chinese Academy of Sciences, Beijing, in 2017.

He is currently a Teaching Assistant with the School of Management Science and Engineering, Central University of Finance and Economics, Beijing. His current research interests include data mining, and weak label learning and its applications in computer vision.

Yingjie Tian received the bachelor's degree in mathematics from Shandong Normal University, Shandong, China, in 1994, the master's degree in applied mathematics from the Beijing Institute of Technology, Beijing, China, in 1997, and the Ph.D. degree in management science and engineering from China Agricultural University, Beijing, China.

He is currently a Professor with the Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing, China. He has authored four books on SVMs, one of which has been cited over 1 500 times. His current research interests include support vector machines, optimization theory and applications, data mining, intelligent knowledge management, and risk management.

Lingfeng Niu received the B.S. degree in mathematics from Xian Jiaotong University, Xi'an, China, in 2004 and the Ph.D. degree in mathematics from the Chinese Academy of Sciences, Beijing, China, in 2009.

She is currently an Associate Professor with the Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing. Her current research interests include optimization, machine learning, and data mining.

Yong Shi received the B.S. degree in mathematics from the Southwest Petroleum Institute, Chengdu, China, in 1982 and the Ph.D. degree in management science from the University of Kansas, Lawrence, KS, USA, in 1991.

He is currently the Director of the Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing, China. Since 1999, he has been the Charles W. and Margre H. Durham Distinguished Professor of Information Technology with the College of Information Science and Technology, Peter Kiewit Institute, University of Nebraska, Lincoln, NE, USA. His current research interests include business intelligence, data mining, and multiple criteria decision making.

Peng Zhang received the Ph.D. degree from the University of the Chinese Academy of Sciences, Beijing, China, in 2009.

He is currently a Senior Data Analyst with the Ant Financial Services Group, Hangzhou, China. Since 2009, he has been with one national engineering laboratory in China, two universities in the USA, and University of Technology Sydney, NSW, Australia. He has published over 100 research papers in the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), IEEE International Conference on Data Mining (ICDM), International Joint Conference on Artificial Intelligence (IJCAI), Association for the Advancement of Artificial Intelligence (AAAI), and The International Conference of World Wide Web. His current research interests include machine learning and data mining.

Dr. Zhang also serves as a PC Member (Reviewer) in TKDE, KDD, ICDM, IJCAI, AAAI, and NIPS. He is an Associate Editor of two Springer journals, *Journal of Big Data* and *Annals of Data Science*.