Inverse Extreme Learning Machine for Learning with Label Proportions

Limeng Cui
School of Computer and Control Engineering
University of Chinese Academy of Sciences
Beijing, China
lmcui932@163.com

Jiawei Zhang
IFM Lab
Department of Computer Science
Florida State University
FL, USA
jzhang@cs.fsu.edu

Zhensong Chen
School of Economics and Management
University of Chinese Academy of Sciences
Beijing, China
wxzmczs@163.com

Yong Shi
Key Laboratory of Big Data Mining
and Knowledge Management
Chinese Academy of Sciences
Beijing, China
yshi@ucas.ac.cn

Philip S. Yu
Department of Computer Science
University of Illinois at Chicago
IL, USA
psyu@uic.edu

***Abstract*—In large-scale learning problem, the scalability of learning algorithms is usually the key factor affecting the algorithm practical performance, which is determined by both the time complexity of the learning algorithms and the amount of supervision information (i.e., labeled data). Learning with label proportions (LLP) is a new kind of machine learning problem which has drawn much attention in recent years. Different from the well-known supervised learning, LLP can estimate a classifier from groups of weakly labeled data, where only the positive/negative class proportions of each group are known. Due to its weak requirements for the input data, LLP presents a variety of real-world applications in almost all the fields involving anonymous data, like computer vision, fraud detection and spam filtering. However, even through the required labeled data is of a very small amount, LLP still suffers from the long execution time a lot due to the high time complexity of the learning algorithm itself. In this paper, we propose a very fast learning method based on inversing output scaling process and extreme learning machine, namely Inverse Extreme Learning Machine (IELM), to address the above issues. IELM can speed up the training process by order of magnitudes for large datasets, while achieving highly competitive classification accuracy with the existing methods at the same time. Extensive experiments demonstrate the significant speedup of the proposed method. We also demonstrate the feasibility of IELM with a case study in real-world setting: modeling image attributes based on ImageNet Object Attributes dataset.**

Keywords*—*Learning with label proportions, semi-supervised learning, extreme learning machine, attribute modeling, classifier calibration.

I. Introduction

In recent years, a rapid growth of visual data have seen produced by social media, large-scale surveillance cameras, biometrics sensors, and mass media content providers. In such large-scale setting, the visual data requires a large amount of supervision to make machine learning methods effective. To be more specific, in standard supervised learning problem, a set of certainly labeled instances is given to produce a classifier. The goal is to accurately predict its class label given a new unlabeled example [1]. However, in many real-world cases, it is not always feasible to obtain the labels of instances. In the past few decades, researchers have made great efforts to reduce the effort on manually labeling the training dataset. In these learning frameworks, such a certainly labeled dataset for training a classifier is not provided. Specific techniques have been proposed recently in order to deal with such weakly labeled datasets, such as the popular semi-supervised learning (SSL) [2], [3], multi-instance learning (MIL) [4], [5] and learning with label proportions (LLP) [6], [7]. In this paper, we focus on the problem of learning with label proportions, in which the training data is provided in groups and only the proportion of each class in each group is known.

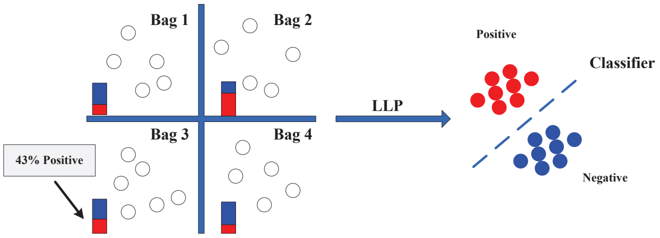
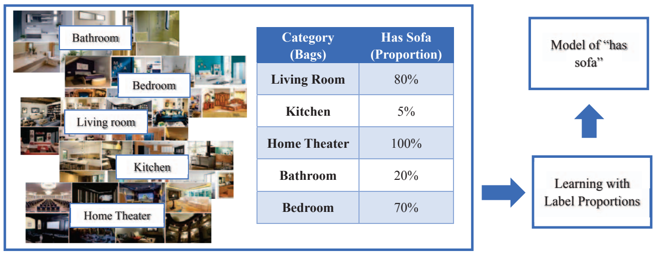


Fig. 1. Illustration of learning with label proportions (LLP). In this example, the training data are provided in 4 bags, each with its label proportion. The learned model is a hyperplane to classify each individual instance.

Recently many algorithms have been proposed to address the problem of LLP [8], [9], [10], [11], [12], [13], [14] and achieve encouraging results. The feasibility of LLP setting has also been verified from a theoretical perspective [15]. Fig. 1 provides an illustration of learning with label proportions. Compared to supervised learning, where the exact labels of all the training instances are known, only the label proportions



Category (Bags)	Has Sofa (Proportion)
Living Room	80%
Kitchen	5%
Home Theater	100%
Bathroom	20%
Bedroom	70%

Fig. 2. Here gives a large set of images and many bags. Each bag represents a scenery, such as living room, bathroom, kitchen, and so on. We can easily obtain the proportions of "has sofa" in these scenarios, which is reasonable in real-world cases. However, we don't know whether each image contains sofa or not. By using the proposed model, we can induce such information based on the input images and the proportion information.

for the bags are given in LLP.

This learning problem has many interesting applications, where only the proportion label is provided [16], [17], [18], [19], [20], [21], [22]. An example comes from visual attribute modeling. Attributes often refer to the visual property of an object that human has the ability to decide whether it is presented or not such as color, texture and shape, which are shared by different categories [23]. Most conventional attribute modeling methods require the concrete label of the attribute on each image. However, it is more straightforward to estimate the proportions based on human common sense rather than to assign a concrete attribute for each instance [23], [24]. For example, it is easy to know things like "25% cats are white", "nearly every living room has window" and "90% Asians are with dark brown eyes". Fig. 2 illustrates the framework by a conceptual example of modeling the attribute "has sofa". In this example, a large set of images and many bags are given. Each bag represents a scenery, such as living room, bathroom, kitchen and so on. We can easily obtain the proportion of "has sofa" in these scenarios, which is reasonable in real-world cases. And we don't know whether each image contains sofa or not. By using the LLP model, we can induce such information based on the input images and the proportion information.

To give another example, in client purchasing behaviors analysis, it is common practice to apply machine learning to client transaction data. Although it seems straight-forward, but in practice, revealing client's transaction data may cause serious legal dispute, especially when this data is provided to a third party for analysis. Overall, storing only the proportion labels over different groups may be a legally advisable way in the above case.

As only the instance feature and proportion label are provided, the key challenge to this research is how to use the existing supervised learning methods to solve the LLP problem. Max-margin based frameworks [7], [25] have been proposed successively. They first assign the labels randomly or by using geometric information, then adjust the labels gradually through the proportion information iteratively. In addition, as the optimization procedure in each iteration is time-consuming, the overall runtime is hard to endure, especially on large datasets. So, in this paper, we will solve the following questions:

- How do we exploit the proportion label in order to fit the LLP problem into existing supervised learning framework?
- How do we deal with the many-to-one relationship where the feature is instance-level and the label is bag-level?
- How do we use an efficient method to accelerate the solving procedure especially on large dataset?

To address the above issues, we first regard proportion label of each bag as conditional class probability. In traditional conditional class probability estimation, a classifier is trained first, and then a scaling function is applied. In stead we apply the inverse scaling function to the given probability and train a regression model. Then, we define *Bag-level Super-instance* to accord the bag-level label. The size of the dataset need to be trained is decreased at the same time. Also, instead of using max-margin based methods, we extend the extreme learning machine concept to solve a novel optimization problem in its dual form efficiently.

Based on the remarks aforementioned, in this paper, we propose a fast LLP method called IELM. IELM can alleviate the need for optimizing each instance iteratively and accelerate the training time. It is approximately hundreds times faster than the existing method and achieves equivalent accuracy at the same time. Extensive experiment evaluations demonstrate the superior performance of the proposed IELM method. To our best knowledge, our work is the first to extend extreme learning machine into the LLP problem. In addition, we apply our method to the attribute modeling problem.

The rest of this paper is structured as follows. In section II, we review the existing methods proposed for LLP in recent years and then present some techniques of classifier calibration that will be used to formulate our model. The model formulation, its solving strategy and discussions are presented in section III while the experimental results are shown in section IV. Finally, section V concludes this paper.

II. Related Work

During the past decades, many approaches have been developed for LLP. In the following, we first give some brief review on the existing methods proposed for LLP including InvCal and $\propto$SVM. Then, we present the existing work on the techniques of classifier calibration, which will be used to formulate our model.

A. Learning with Label Proportions

In recent years, numerous papers have been published on the problem of learning with label proportions. Interested readers can refer to [26], [27] for a comprehensive survey of these various proportion learning methods.

Chen et al. [28] first introduced a new class of data mining problem called learning from aggregate views, which is to learn from multiple aggregate views of the underlying data. Quadrianto et al. [9] proposed a method called MeanMap, which can reconstruct the correct labels with high probability in a uniform convergence sense by using empirical mean of each bag to approximate expectations with respect to the bag distribution. However, to predict the unknown labels in the testing set, the distribution of the labels is required. This assumption does not hold for many real world applications. Rueping et al. [6] presented a popular method called InvCal, which can learn a classifier from group probabilities based on support vector regression. In this learning setting, the mean of each bag is treated as a super-instance that is assumed to have a soft label corresponding to the label proportion. Stolpe et al. [10] contributed a developmental solution based on the clustering with label proportions. This method can adjust its current hypothesis based on the average loss on the training set. Kück et al. [29] introduced a principled probabilistic model to estimate the unknown binary labels of individuals from knowledge of group statistics. Another effective LLP algorithm is $\propto$SVM proposed by Yu et al. [7], which is based on the large margin framework. It recursively optimizes over the unknown instance labels and the known label proportions until the objective converges. Qi et al. [25] proposed a brand new algorithm, called LLPs via nonparallel support vector machine (LLP-NPSVM), to harness satisfactory data adaption, which can be interpreted as an alternative competitive method benefiting from large margin clustering.

However, the main drawback of current methods is the time cost. LLP methods aim to solve the problem in which sufficient labels are not obtained. But in some existing methods, it takes more time to estimate the classifier with group probabilities than to label the instances manually. Although existing methods have shown promising results, the execution time is not taken into consider.

B. Classifier Calibration Techniques

Given unknown probability distribution $P(X, Y)$, where X is the instance space and $Y \in \{-1, +1\}$ is set of labels. For a probabilistic classification task, we aim to find a function $f : X \to [0, 1]$, which can return an estimate of the conditional class probability through

$$f(x) \approx P(y = 1|x) \tag{1}$$

Calibrating classifiers is an effective approach to the probabilistic classification problem. The key of calibration is to find an appropriate scaling function $\sigma : \mathbb{R} \to [0, 1]$ that can transform the decision values to the probabilities:

$$\sigma(f(x)) \approx P(y = 1|x) \tag{2}$$

According to [30], Platt Calibration and Isotonic Regression are two effective probabilistic calibration techniques for a wide range of learning methods. Specifically, Platt Calibration [31] is a method for transforming SVM outputs from $[-\infty, +\infty]$ to posterior probabilities, which has proven its efficiency for many other numerical decision functions as well [30]. Here, suppose $f(x)$ is the output of a learning method, it suggests to get the calibrated probabilities by using the sigmoid function as the scaling function:

$$\sigma(f(x)) = \frac{1}{1 + exp(Af(x) + B)} \tag{3}$$

where A and B are the parameters optimized using maximum likelihood estimation.

Isotonic Regression [32] is a method used to calibrate predictions from boosted naive Bayes, SVM, and decision tree models. The basic assumption is a monotonic dependency existed between the decision function and the conditional class probabilities, which means the only restriction of this method is that the mapping function should be monotonically increasing. In detail, let $f(x)$ be the predication of one model and y be the corresponding ground truth label, the basic assumption in Isotonic Regression is that

$$y = g(f(x)) + \epsilon \tag{4}$$

where $g(\cdot)$ is the monotonically increasing function that used as the scaling function, which is learned from the training set $(f(x), y)$ by minimizing the quadratic loss.

Various calibration techniques including parametric and non-parametric methods have also been studied in the literature, for example quantile binning and ensemble of near isotonic regression. Interested readers are referred to [33], [34] for a comprehensive survey of these techniques.

III. Inverse Extreme Learning for LLP

In this section, we first introduce the extreme learning machine and the LLP problem formulation respectively. Next, we generate the IELM model by inversing the scaling function. Although exploiting the concept of extreme learning machine, the IELM model leads to a very different optimization problem, which fortunately can be solved in its dual form efficiently. The optimization procedure of the proposed IELM is presented. Finally, we will discuss the feasibility and scalability of IELM.

A. Traditional Extreme Learning Machine

In this section, we give a brief review of the traditional extreme learning problem, which was firstly studied by Huang et al. [35].

Given the training set $\{\mathbf{x}_i, y_i\}_{i=1}^N$, where $\mathbf{x}_i = [x_{i_1}, x_{i_2}, ..., x_{i_n}]^T \in \mathbb{R}^n$ represents the input feature vectors and n is the number of features, and $y_i \in \mathbb{R}$ is the corresponding label, a standard single hidden feedforward neural network (SLFN) with $\widetilde{N}$ hidden nodes and activation function $g(\mathbf{x})$ can be mathematically modeled as

$$\sum_{j=1}^{\widetilde{N}} \beta_j g_j(\mathbf{x}_i) = \sum_{j=1}^{\widetilde{N}} \beta_j g(\mathbf{w}_j \cdot \mathbf{x}_i + b_j) = o_i, i = 1, 2, ..., N \tag{5}$$

where $\mathbf{w}_j = [w_{j_1}, w_{j_2}, ..., w_{j_n}]^T$ is the weight vector connecting the j-th hidden node and the input nodes, β_j is the weight connecting the j-th hidden node and the output nodes, and b_j is the threshold of the j-th hidden node, and $\mathbf{w}_j \cdot \mathbf{x}_i$ denotes the inner product of $\mathbf{w}_j$ and $\mathbf{x}_i$.

According to [36], the standard SLFN can approximate these N samples with zero error, which means $\sum_{i=1}^N ||o_i - y_i|| = 0$. Then, we have

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{Y} \tag{6}$$

where

$$\mathbf{H} = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_{\widetilde{N}} \cdot \mathbf{x}_1 + b_{\widetilde{N}}) \\ \vdots & \cdots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_{\widetilde{N}} \cdot \mathbf{x}_N + b_{\widetilde{N}}) \end{bmatrix}_{N \times \widetilde{N}}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{\widetilde{N}} \end{bmatrix}_{\widetilde{N} \times 1} \quad and \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}_{N \times 1}$$

in which $\mathbf{H}$ is called the hidden layer output matrix of the neural network whose j-th column is the j-th hidden node output with respect to inputs $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$.

In order to train a SLFN, it needs to find the correct $\hat{\mathbf{w}}_j, \hat{b}_j, \hat{\boldsymbol{\beta}}$ by solving the following optimization problem

$$\min_{\mathbf{w}_j, b_j, \boldsymbol{\beta}} ||H(\mathbf{w}_1, ..., \mathbf{w}_{\widetilde{N}}, b_1, ..., b_{\widetilde{N}})\boldsymbol{\beta} - \mathbf{Y}|| \tag{7}$$

To effectively solve the optimization problem above, the extreme learning machine (ELM) was proposed [35]. In the ELM setting, the input weights $\mathbf{w}_j$ and the hidden layer biases b_j are in fact not necessarily tuned and the hidden layer output matrix $\mathbf{H}$ can actually remain unchanged once random values have been assigned to these parameters in the beginning of learning. For the fixed $\mathbf{w}_j$ and b_j, training an SLFN is equivalent to find the smallest norm least squares solution $\hat{\boldsymbol{\beta}}$ of the linear system $\mathbf{H}\boldsymbol{\beta} = \mathbf{Y}$, which is given by

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{Y} \tag{8}$$

where where $\mathbf{H}^{\dagger}$ is the *Moore-Penrose generalized inverse* of matrix $\mathbf{H}$ [37], [38].

B. Inverse Extreme Learning Machine

In learning from label proportions, although the proportion of each bag is given, the label of each instance is unknown.

Suppose we are given training set $\{\mathbf{x}_i, y_i\}_{i=1}^N$ in K bags $\{S_k, P_k\}_{k=1}^K$, where each bag S_k consists of N_k instances $\{\mathbf{x}_i, y_i^*\}_{i=1}^{N_k}$ in which $\mathbf{x}_i$ is the feature vector of the i-th instance in bag S_k and $y_i^* \in \{-1, 1\}$ denotes the unknown ground truth label of $\mathbf{x}_i$. Since the training set is grouped into K bags, the conditional probability of the k-th bag S_k can be defined as

$$P_k = \frac{|\{i|\mathbf{x}_i \in S_k, y_i^* = 1\}|}{|S_k|} \tag{9}$$

As pointed out previously, for some cases, the concrete label of each instance is unknown, while only the label proportion is provided. The goal is to learn a classification model $f : \mathbf{x} \to y$ with minimal error according to the ground truth proportions, such that the label y for any instance $\mathbf{x}$ can be predicted.

Assume the instance labels are explicitly modeled as $\{y_i\}_{i=1}^N$, where $y_i \in \{-1, 1\}$, and $N = \sum_{k=1}^K N_k$ is the total number of training instances. The modeled label proportion of the k-th bag can be obtained by

$$p_k = \frac{|\{i|\mathbf{x}_i \in S_k, y_i = 1\}|}{|S_k|} \tag{10}$$

which can be regarded as the estimate of the conditional class probability P_k.

To simplify notation, we use P, S, $\mathbf{x}$ and y instead of P_k, S_k, $\mathbf{x}_i$ and y_i in the following. In conditional class probability estimation, a classifier f is trained first, and then a scaling function σ is applied to estimate P. Instead, we start with given probability estimates p, fix a scaling function σ, apply this inverse scaling function and train an ELM to predict the values $\sigma^{-1}(p)$, which is shown in Fig. 3.

In our algorithm, we use the scaling function

$$P = \sigma(y) = \frac{1}{1 + e^{-y}} \tag{11}$$

This scaling function was first introduced by Platt et al. [31] for scaling support vector machine, which can formulate the results of the decision values to a fixed range, e.g. $[0, 1]$.

As our approach is to invert the process of estimating probabilities from classifier calibration, here we give the definition of *Inversion of Probability Estimation*.

Definition 3.1. (Inversion of Probability Estimation). We can derive y from p by *inversing the scaling function*

$$y = \sigma^{-1}(P) = -log(\frac{1}{P} - 1) \tag{12}$$

We want to find a mapping function $f(x) = \mathbf{H}(\mathbf{x})\boldsymbol{\beta}$. In order to construct this function, we require $\sigma(y)$ to be a good estimate of P. However, in this problem setting, estimates of P for each instance $\mathbf{x}$ are not given, but only for sets S of instances. Here we give the definition of *Bag-level Super-instance*.

Definition 3.2. (Bag-level Super-instance). Suppose X_k is the bag-level instance of k-th bag, which can reflect the overall instance-level feature $\mathbf{x}_i \in S_k$. The k-th *Bag-level Super-instance* X_k can be defined as

$$X_k = \frac{1}{|S_k|} \sum_{\mathbf{x}_i \in S_k} \mathbf{x}_i \tag{13}$$

Depending on the construction of S, the optimal class probability estimates of the individual instances in S may be very closed to their average P. To address this issue, we only require that f predicts y well over each bag.

$$\mathbf{H}(X_k)\boldsymbol{\beta} \approx y_k, \quad k = 1, \cdots, K \tag{14}$$

in which $\mathbf{H}(X_k) = [g(\mathbf{w}_1 \cdot X_k + b_1), ..., g(\mathbf{w}_{\widetilde{N}} \cdot X_k + b_{\widetilde{N}})]$. Now we manage to formally define the learning task formally in the spirit of Extreme Learning Regression.

Based on the prediction of y_k for each bag S_k, the proportion learning model can be derived as (15), where the output weight vectors in optimization are sample-based and should be normalized in their own relevant subspaces.

$$\begin{aligned} &\min_{\boldsymbol{\beta}} \frac{1}{2}||\boldsymbol{\beta}||^2 + C\sum_{k=1}^{K} ||\xi_k + \xi_k^*||^2 \\ &s.t. \forall_{k=1}^K : \mathbf{H}(X_k)\boldsymbol{\beta} \geq y_k - \xi_k \\ &\quad \forall_{k=1}^K : \mathbf{H}(X_k)\boldsymbol{\beta} \leq y_k + \xi_k^* \\ &\quad \forall_{k=1}^K : \xi_k, \xi_k^* \geq 0 \end{aligned} \tag{15}$$

where $C > 0$ is a penalty parameter, ξ_k and ξ_k^* are the vectors of appropriate dimension. We name this model as Inverse Extreme Learning Machine (IELM).

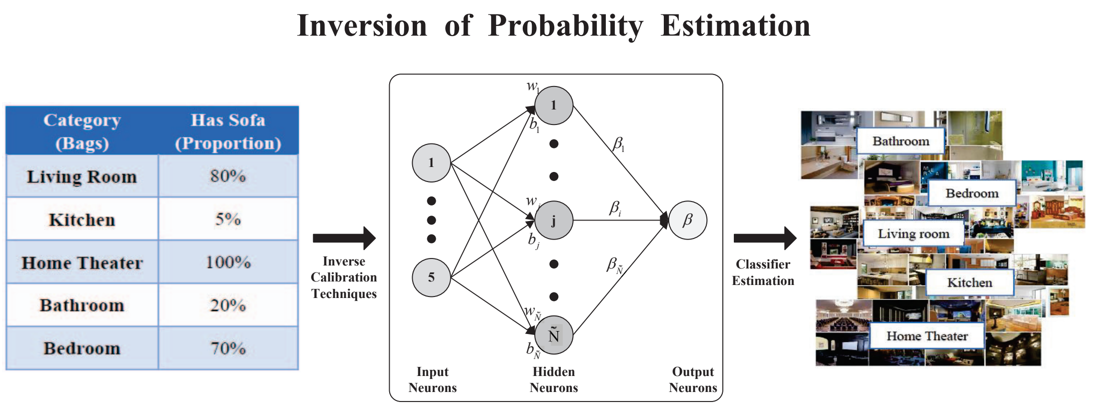


Fig. 3. Estimating a classifier by inverting the calibration process

C. Optimizing the IELM Problem

To estimate the model parameters in IELM, we consider to transfer it into the dual form which can be easily and efficiently solved. Firstly, the optimization problem (15) can be rewritten as

$$\begin{aligned}&\min_{\boldsymbol{\beta}} \frac{1}{2}||\boldsymbol{\beta}||^2 + C\sum_{k=1}^{K}\xi_k^2 \\ &s.t. \forall_{k=1}^{K} : \mathbf{H}(X_k)\boldsymbol{\beta} = y_k - \xi_k\end{aligned} \tag{16}$$

Then, based on the Karush-Kuhn-Tucker (KKT) theorem [39], the Lagrangian function of problem (16) is given by

$$\begin{aligned}L(\boldsymbol{\beta}, \xi, \boldsymbol{\alpha}) = \; & \frac{1}{2}||\boldsymbol{\beta}||^2 + C\sum_{k=1}^{K}\xi_k^2 \\ & - \sum_{k=1}^{K}\alpha_k(\mathbf{H}(X_k)\boldsymbol{\beta} - y_k + \xi_k)\end{aligned} \tag{17}$$

where α_k is the Lagrange multiplier corresponding to the k-th bag. We can then obtain the KKT sufficient and necessary optimality conditions of the problem (16) as follows

$$\boldsymbol{\beta} - \sum_{k=1}^{K}\alpha_k\mathbf{H}(X_k)^T = 0 \tag{18}$$

$$\alpha_k - C\xi_k = 0 \tag{19}$$

$$\mathbf{H}(X_k)\boldsymbol{\beta} - y_k + \xi_k = 0 \tag{20}$$

Therefore, the dual form of the primal problem (16) can be achieved as follows:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{H}\mathbf{H}^T\boldsymbol{\alpha} + \frac{1}{C}\boldsymbol{\alpha}^T\boldsymbol{\alpha} \tag{21}$$

in which $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_K]^T$, $\mathbf{H} = [\mathbf{H}_1; \mathbf{H}_2; ...; \mathbf{H}_K]$ is the concatenation of $\mathbf{H}_k$ by column, and $\mathbf{H}_k = [g(\mathbf{w}_1 \cdot X_k + b_1), ..., g(\mathbf{w}_{\widetilde{N}} \cdot X_k + b_{\widetilde{N}})]$. By solving this dual optimization problem above, the classification model $f(\mathbf{x}, \boldsymbol{\beta})$ can be obtained.

Algorithm 1 IELM

```
**Input:** Training datasets in bags $\{S_k, P_k\}_{k=1}^K$, activation function $g(\mathbf{x})$ and number of hidden nodes $\widetilde{N}$.
**Output:** Classification model $f(\mathbf{x}, \boldsymbol{\beta})$.
**Begin**
• Randomly assign input weight $\mathbf{w}_j$ and bias $b_j$ for the $j$-th node, $j = 1, 2, ..., \widetilde{N}$.
• For each bag $S_k$, compute $\mathbf{H}_k = [g(\frac{1}{|S_k|}\sum_{\mathbf{x}_i \in S_k}(\mathbf{w}_1 \cdot \mathbf{x}_i + b_1)), ..., g(\frac{1}{|S_k|}\sum_{\mathbf{x}_i \in S_k}(\mathbf{w}_{\widetilde{N}} \cdot \mathbf{x}_i + b_{\widetilde{N}}))]$, $k = 1, ..., K$.
• Compute $\mathbf{Y} = [y_1, ..., y_K]^T$ by inversing the scaling function.
• Solve the dual problem to obtain $\boldsymbol{\alpha} = (\frac{1}{C} + \mathbf{H}\mathbf{H}^T)^{-1}\mathbf{Y}$, where $\mathbf{H} = [\mathbf{H}_1; \mathbf{H}_2; ...; \mathbf{H}_K]$.
• Achieve the weight vector $\hat{\boldsymbol{\beta}} = \mathbf{H}^T\boldsymbol{\alpha}$.
• Construct instance-level classification model $f(\mathbf{x}, \hat{\boldsymbol{\beta}}) = \mathbf{H}(\mathbf{x})\hat{\boldsymbol{\beta}}$ based on the randomly assigned input weight $\mathbf{w}_j$ and bias $b_j$ to predict the label $y$ for each instance $\mathbf{x}$.
**End**
```

According to [36], the optimization problem can be effectively solved based on the ELM regression algorithm, which has been proved that the norm least squares solution is unique and the smallest training error can be reached. Therefore, we proposed the following algorithm called IELM, which is presented in **Algorithm** 1.

Following the above steps, (15) can be solved in a lower computational complexity since the learning time of IELM is mainly spent on calculating the Moore-Penrose generalized inverse of the hidden layer output matrix. In fact, the proposed method can achieve fast speed because 1) it randomly assigns the hidden-layer parameters, thus saves a great amount of computation time and 2) training an IELM is a linear least squares problem whose solution can be directly generated by the generalized inverse of the hidden layer output matrix. The SVD is used to calculate Moore-Penrose generalized inverse of H in our case and the computation complexity of IELM is $O(K^2\widetilde{N} + K\widetilde{N}^2)$. Here K is the number of bags and $\widetilde{N}$ is the number of hidden nodes.

In addition, the goal of most machine learning algorithms is to minimize the loss. But many may get stuck in a local minima, even with infinite iterations. In contrast, it is easier to get the smallest norm of weights by using extreme learning machine, in which the solution is unique.

D. Discussion

1) Feasibility of IELM: IELM provides a powerful complexity-reduction learning paradigm through adjusting the output layer connections only while randomly fixing the hidden parameters. In fact, even IELM adjusts partial connections in FNN, it does not degrade the generalization capability by selecting the appropriate active functions. Since [40] has proved that ELM can achieve the almost optimal generalization error bound for the polynomial, Nadaraya-Watson and sigmoid activation functions, the selection of sigmoid active function for IELM is feasible.

In addition, there is a close connection between the number of hidden layer nodes and the number of training samples is necessary to realize the almost optimal generalization error bound. Moreover, the induced hidden layer output matrix in IELM is full column rank when the active function is algebraic polynomial, which means the well-known generalized inverse technique can be applied effectively.

2) Scalability of IELM: For the case where the number of training bags is not huge, we compute $\boldsymbol{\alpha}$ by substituting (18) and (19) into (20):

$$(\frac{1}{C} + \mathbf{H}\mathbf{H}^T)\boldsymbol{\alpha} = \mathbf{Y} \tag{22}$$

where $\mathbf{Y} = [y_1, ..., y_K]^T$ and $\mathbf{H}\mathbf{H}^T \in \mathbb{R}^{K \times K}$. Then, the solution of IELM can be obtained by

$$\boldsymbol{\beta} = \mathbf{H}^T(\frac{1}{C} + \mathbf{H}\mathbf{H}^T)^{-1}\mathbf{Y} \tag{23}$$

Furthermore, for the large-scale applications, from Eq. (18) and (19), we can get

$$\begin{aligned} \boldsymbol{\beta} &= C\mathbf{H}^T\boldsymbol{\xi} \\ \boldsymbol{\xi} &= \frac{1}{C}(\mathbf{H}^T)^{\dagger}\boldsymbol{\beta} \end{aligned} \tag{24}$$

By substituting (24) into (20), we have

$$\mathbf{H}^T(\mathbf{H} + \frac{1}{C}(\mathbf{H}^T)^{\dagger})\boldsymbol{\beta} = \mathbf{H}^T\mathbf{Y} \tag{25}$$

Thus, IELM can get the solution based on

$$\boldsymbol{\beta} = (\frac{1}{C} + \mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{Y} \tag{26}$$

where $\mathbf{H}^T\mathbf{H} \in \mathbb{R}^{\widetilde{N} \times \widetilde{N}}$.

In most applications, since the number of hidden nodes $\widetilde{N}$ can be much smaller than the number of training bags K: $\widetilde{N} \ll K$, the computational cost can be reduced dramatically. Therefore, IELM has much better computational scalability with regard to the number of training bags K.

TABLE I. THE UCI DATASETS USED IN OUR EXPERIMENTS

Dataset	#Size	#Feature
sonar	208	60
heart	270	13
vote	435	16
credit-a	690	15
diabetes	768	8
pima	768	8
splice	1,000	60
Musk	6,598	166
Magic	19,020	10
cod-rna	59,535	8

IV. EXPERIMENTAL RESULTS

To validate the performance of our proposed method, in this section, experiment comparisons with InvCal [6], $\propto$SVM[1] [7] and pNPSVM [41] are designed. In the following, we first evaluate the performance on the UCI repository in section 4.1, and then present the experimental results on the ImageNet in section 4.2.

A. Results on the UCI Repository

In this section, performance of the proposed IELM algorithm is compared with the popular algorithms InvCal [6], $\propto$SVM [7] and pNPSVM [41] on the UCI datasets[2] shown in Table I. In order to avoid scaling issues in the learning process, the features of each dataset are scaled to $[-1, +1]$. All the experiments for these four algorithms including IELM, InvCal, $\propto$SVM and pNPSVM are carried out in MATLAB 8.4 environment running in a 2.5 GHz Intel Core i7 CPU.

To formulate the LLP classification problems, the training data is randomly partitioned into a particular fold of bags with fixed size σ. We test various bag size σ: 2, 4, 8, 16, 32, 64 for the first seven datasets in Table I and 2^{10}, 2^{11}, 2^{12} for the others (with the last bag smaller than σ, if necessary). In each single experiment, the accuracy has been assessed by five-fold cross-validation. Furthermore, we repeat the above process five times and report the mean accuracy.

The parameters of each algorithm are tuned through the following rules: For InvCal, the parameters are tuned from $C_p \in \{0.1, 1, 10\}$ and $\varepsilon \in \{0, 0.01, 0.1\}$. For $\propto$SVM, the parameters are tuned from $C \in \{0.1, 1, 10\}$ and $C_p \in \{1, 10, 100\}$. For pNPSVM , the parameters $c_i (i = 1, 2, 3, 4)$ are tuned for the best classification accuracy in the range 0.1 to 10 and $c_p \in \{0.1, 1, 10\}$. Linear kernel is taken for InvCal, $\propto$SVM and pNPSVM in our experiments. For IELM, the number of hidden nodes is gradually increased by an increment of 5 and the nearly optimal number of nodes for IELM is then selected based on the cross-validation method. In our experiments, the outputs (targets) have been normalized into $[-1, 1]$. The results of numerical experiments are summarized in the following tables, where the best accuracy is shown by bold figures.

As observed from Table II and Table III, where the classification accuracy and its standard deviations are presented, IELM can achieve highly competitive or even better accuracy than InvCal, $\propto$SVM and pNPSVM on all the datasets. In detail,

[1] https://github.com/felixyu/pSVM.

[2] http://archive.ics.uci.edu/ml/datasets.html.

TABLE II. ACCURACY OF DIFFERENT METHODS ON SMALL DATASETS WITH DIFFERENT BAG SIZE σ. IN THIS TABLE, THE BEST ACCURACY IS SHOWN BY BOLD FIGURES.

Dataset	Method	2	4	8	16	32	64	Average Rank
sonar	IELM	73.55±0.02 (2)	**71.21±0.05** (1)	**72.17±0.08** (1)	**70.16±0.12** (1)	64.89±0.07 (2)	**63.04±0.12** (1)	**1.33**
	InvCal	73.09±0.08 (3)	70.69±0.06 (2)	65.37±0.11 (2)	64.94±0.10 (2)	**67.29±0.13** (1)	50.02±0.07 (4)	2.33
	$\propto$SVM	72.65±0.08 (4)	68.72±0.07 (3)	54.77±0.06 (4)	55.76±0.09 (4)	51.99±0.12 (3)	59.16±0.11 (2)	3.34
	pNPSVM	**78.85±0.08** (1)	63.94±0.08 (4)	63.54±0.10 (3)	57.12±0.10 (3)	48.04±13.76 (4)	51.92±0.12 (3)	3.00
heart	IELM	**82.59±0.04** (1)	80.74±0.06 (2)	**78.89±0.04** (1)	73.70±0.09 (3)	**77.78±0.06** (1)	**78.52±0.04** (1)	**1.50**
	InvCal	81.85±0.06 (2)	79.26±0.06 (3)	77.78±0.06 (2)	76.67±0.12 (2)	60.00±0.15 (4)	70.37±0.14 (2)	2.50
	$\propto$SVM	78.52±0.04 (3)	**81.48±0.05** (1)	77.41±0.04 (3)	**80.00±0.04** (1)	65.93±0.20 (2)	44.07±0.24 (4)	2.33
	pNPSVM	70.74±0.25 (4)	61.11±0.23 (4)	57.78±0.19 (4)	56.67±0.26 (4)	63.33±0.17 (3)	60.74±0.11 (3)	3.67
vote	IELM	**95.63±0.02** (1.5)	93.56±0.03 (3)	91.72±0.01 (3.5)	91.72±0.04 (3)	**94.02±0.02** (1)	**91.49±0.02** (1)	2.17
	InvCal	95.62±0.01 (3)	**95.86±0.02** (1)	**95.63±0.03** (1)	**95.17±0.01** (1)	92.41±0.02 (3)	90.11±0.03 (2)	**1.83**
	$\propto$SVM	94.48±0.03 (4)	87.59±0.17 (4)	93.56±0.04 (2)	94.25±0.03 (2)	92.87±0.04 (2)	87.82±0.04 (3)	2.83
	pNPSVM	**95.63±0.01** (1.5)	95.63±0.02 (2)	91.72±0.07 (3.5)	76.55±0.13 (4)	87.59±0.06 (4)	59.54±0.22 (4)	3.17
credit-a	IELM	**86.09±0.02** (1)	84.78±0.04 (2)	80.29±0.08 (3)	79.71±0.06 (3)	78.55±0.10 (2)	**78.84±0.08** (1)	2.00
	InvCal	85.51±0.03 (3)	84.64±0.03 (3)	**85.80±0.03** (1)	**84.35±0.03** (1)	**84.49±0.03** (1)	75.07±0.04 (2)	**1.83**
	$\propto$SVM	85.94±0.03 (2)	84.49±0.03 (4)	84.20±0.04 (2)	81.59±0.05 (2)	73.48±0.14 (3)	67.97±0.19 (4)	2.84
	pNPSVM	78.41±0.17 (4)	**85.51±0.04** (1)	79.86±0.06 (4)	57.83±0.22 (4)	65.36±0.15 (4)	71.74±0.13 (3)	3.33
diabetes	IELM	**78.56±0.04** (1)	**74.61±0.03** (1)	**73.18±0.03** (1)	**70.05±0.05** (1)	**68.61±0.05** (1)	**65.62±0.05** (1)	**1.00**
	InvCal	76.56±0.02 (2)	72.53±0.03 (4)	72.54±0.05 (2)	69.15±0.06 (3)	66.28±0.04 (3)	65.10±0.05 (2)	2.67
	$\propto$SVM	74.73±0.04 (3)	73.44±0.03 (2)	67.32±0.03 (3)	69.28±0.03 (2)	66.92±0.04 (2)	64.88±0.02 (3)	2.50
	pNPSVM	63.69±0.15 (4)	73.17±0.05 (3)	53.55±0.15 (4)	57.52±0.15 (4)	54.17±0.16 (4)	56.27±0.11 (4)	3.83
pima	IELM	75.92±0.03 (2)	**75.39±0.02** (1)	**74.34±0.11** (1)	**71.49±0.06** (1)	**68.62±0.11** (1)	**66.80±0.04** (1)	**1.17**
	InvCal	**76.43±0.04** (1)	70.69±0.05 (3)	71.62±0.05 (2)	71.09±0.02 (2)	65.76±0.03 (3)	65.10±0.02 (3)	2.33
	$\propto$SVM	74.86±0.04 (3)	73.19±0.06 (2)	71.36±0.04 (3)	67.31±0.07 (3)	67.31±0.06 (2)	65.89±0.03 (2)	2.50
	pNPSVM	57.16±0.19 (4)	58.30±0.21 (4)	57.40±0.14 (4)	59.11±0.08 (4)	54.44±0.12 (4)	54.30±0.15 (4)	4.00
splice	IELM	75.80±0.04 (4)	71.50±0.02 (3)	66.30±0.04 (3)	64.90±0.02 (2)	**65.10±0.04** (1)	**62.40±0.06** (1)	2.33
	InvCal	78.30±0.03 (2)	**74.60±0.03** (1)	67.40±0.05 (2)	64.00±0.02 (3)	64.60±0.07 (2)	62.10±0.08 (2)	**2.00**
	$\propto$SVM	76.70±0.02 (3)	73.60±0.03 (2)	**68.30±0.05** (1)	**66.20±0.05** (1)	56.80±0.14 (3)	59.50±0.09 (3)	2.17
	pNPSVM	**81.20±0.01** (1)	70.10±0.06 (4)	53.60±0.04 (4)	53.70±0.03 (4)	56.00±0.09 (4)	51.90±0.05 (4)	3.50

TABLE III. ACCURACY OF DIFFERENT METHODS ON LARGE DATASETS WITH DIFFERENT BAG SIZE σ. IN THIS TABLE, THE BEST ACCURACY IS SHOWN BY BOLD FIGURES. $\propto$SVM DOESN'T RUN ON THE COD-RNA DUE TO ITS HIGH COMPUTATIONAL COST AND PNPSVM TAKES TOO LONG TO TEST ON MUSK, MAGIC AND COD-RNA.

Dataset	Method	2^{10}	2^{11}	2^{12}
Musk	IELM	**83.80±0.03**	**84.69±0.01**	**84.65±0.02**
	InvCal	80.05±0.04	84.59±0.01	84.59±0.01
	$\propto$SVM	73.73±0.02	74.92±0.05	79.87±0.06
	pNPSVM	NA	NA	NA
Magic	IELM	**74.46±0.07**	70.03±0.03	72.26±0.15
	InvCal	64.84±0.00	64.84±0.00	64.84±0.01
	$\propto$SVM	73.92±0.01	**74.05±0.00**	**73.55±0.01**
	pNPSVM	NA	NA	NA
cod-rna	IELM	**74.72±0.05**	**72.45±0.08**	**70.30±0.12**
	InvCal	66.90±0.00	68.90±0.01	68.90±0.02
	$\propto$SVM	NA	NA	NA
	pNPSVM	NA	NA	NA

TABLE IV. TRAINING TIME COMPARISON (IN SECONDS) ON SMALL DATASETS. IELM CAN ACHIEVE THE LEAST TRAINING TIME, WHILE MAINTAINING HIGHLY COMPETITIVE OR EVEN BETTER ACCURACY.

Dataset	Method	2	4	8	16	32	64
sonar	IELM	**0.02**	**0.03**	**0.02**	**0.02**	**0.02**	**0.02**
	InvCal	1.06	0.93	0.92	0.90	0.89	0.86
	$\propto$SVM	5.31	4.00	3.07	2.48	2.01	2.80
	pNPSVM	5.90	5.59	11.78	8.84	16.56	5.42
heart	IELM	**0.03**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**
	InvCal	1.43	1.06	1.01	1.01	0.97	0.95
	$\propto$SVM	7.50	5.79	4.56	3.84	3.20	2.94
	pNPSVM	8.90	11.29	4.73	6.07	13.74	9.94
vote	IELM	**0.04**	**0.03**	**0.03**	**0.04**	**0.04**	**0.03**
	InvCal	3.53	3.41	3.31	3.28	3.26	3.24
	$\propto$SVM	14.15	11.06	9.67	8.54	7.25	6.62
	pNPSVM	12.38	8.89	8.22	7.42	10.18	7.61
credit-a	IELM	**0.03**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**
	InvCal	1.21	0.53	0.40	0.35	0.32	0.29
	$\propto$SVM	18.43	14.58	12.36	9.60	8.58	8.16
	pNPSVM	39.11	17.54	19.04	27.24	19.28	27.39
diabetes	IELM	**0.03**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**
	InvCal	1.31	0.58	0.41	0.34	0.32	0.27
	$\propto$SVM	26.58	22.81	16.79	14.88	13.26	9.92
	pNPSVM	16.35	52.04	58.30	15.41	13.00	14.92
pima	IELM	**0.03**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**
	InvCal	1.36	0.56	0.41	0.35	0.32	0.27
	$\propto$SVM	29.25	22.01	18.99	12.99	11.20	9.86
	pNPSVM	14.32	76.55	27.60	16.83	15.49	14.50
splice	IELM	**0.04**	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**
	InvCal	3.70	1.17	0.58	0.35	0.30	0.28
	$\propto$SVM	50.04	44.19	36.61	28.47	23.47	18.57
	pNPSVM	283.43	212.69	181.38	177.32	175.20	165.46

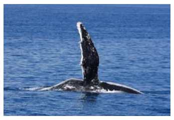
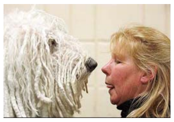
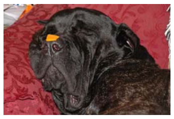
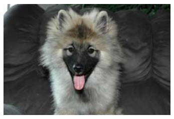
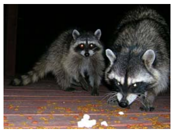
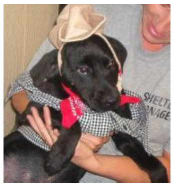
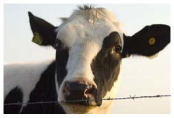
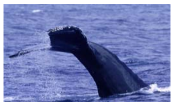
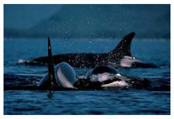
the average classification accuracy of IELM is higher than InvCal, $\propto$SVM and pNPSVM on the datasets sonar, heart, diabetes and pima since it achieves the best accuracy with some bag size and obtains highly competitive results on the others. On the datasets vote, credit-a and splice, the average classification accuracy of IELM is not worse than 6% compared with the best result. Due to the high computational cost, $\propto$SVM doesn't run on the dataset cod-rna, and pNPSVM doesn't run on the datasets Musk, Magic and cod-rna. Moreover, as seen from IELM algorithm, the learning time of IELM is mainly spent on calculating the Moore-Penrose generalized inverse $\mathbf{H}^{\dagger}$ of the hidden layer output matrix $\mathbf{H}$. Specifically, IELM is much faster than InvCal, $\propto$SVM and pNPSVM on all the datasets, which can be seen from Table IV and Table V. As presented in these two tables, the training time for $\propto$SVM and pNPSVM is increasing rapidly with the growth of the amount of training datasets, which is several to hundreds times slower than InvCal and IELM. The training time for IELM is several times, even hundreds times on some datasets, faster compared with InvCal. For example, for the Magic dataset, with σ equals to 2^{10}, IELM takes 0.02 sec, while InvCal takes 300 times more and $\propto$SVM takes 135,000 times more. For the cod-rna dataset, with σ equals to 2^{10}, IELM takes 0.19 sec, while InvCal costs around 10,0000 times more, and $\propto$SVM and pNPSVM do not even run.

According to the analysis above, it can be concluded that IELM achieves great performance on the standard weakly la-

TABLE V. Training time comparison (in seconds) on large datasets. IELM can achieve the highly competitive or even better accuracy with less training time. $\propto$SVM doesn't run on the cod-rna due to its high computational cost and pNPSVM takes too long to test on Musk, Magic and cod-rna.

Dataset	Method	2^{10}	2^{11}	2^{12}
Musk	IELM	**0.04**	**0.03**	**0.03**
	InvCal	0.95	0.95	1.03
	$\propto$SVM	1027.81	940.71	881.11
	pNPSVM	NA	NA	NA
Magic	IELM	**0.02**	**0.04**	**0.03**
	InvCal	6.59	6.92	6.92
	$\propto$SVM	2786.30	2766.42	2597.34
	pNPSVM	NA	NA	NA
cod-rna	IELM	**0.19**	**0.18**	**0.21**
	InvCal	1944.00	1932.31	1875.21
	$\propto$SVM	NA	NA	NA
	pNPSVM	NA	NA	NA

TABLE VI. The results on ImageNet Object Attributes dataset. For each image, the attributes are derived by different methods, where the correct attributes are marked in red while others are marked in blue.

Images	Ground Truth	IELM	InvCal	$\propto$SVM
	black long shiny wet	black green red rough shiny wet	black green red rough	black furry white
	furry rough white	furry rough white	rough white	gray vegetarian strip
	black furry smooth	black furry green smooth	black furry green smooth	black furry rough shiny
	black furry gray white	black furry gray white	black gray white	black furry white
	black furry gray	black furry gray	black furry white	black furry white
	black furry smooth red	black furry gray brown red	rough furry strip smooth	furry wet smooth
	black rough smooth white	black smooth	green shiny smooth	gray white
	black long shiny wet	black long metallic shiny	black long metallic shiny	black long metallic shiny
	black long smooth shiny wet	black long metallic smooth wet	black long metallic shiny	black long metallic shiny

beled datasets. Specifically, it can obtain the highly competitive or even better classification accuracy with much less training time. IELM shows great application prospects, especially in the big data era.

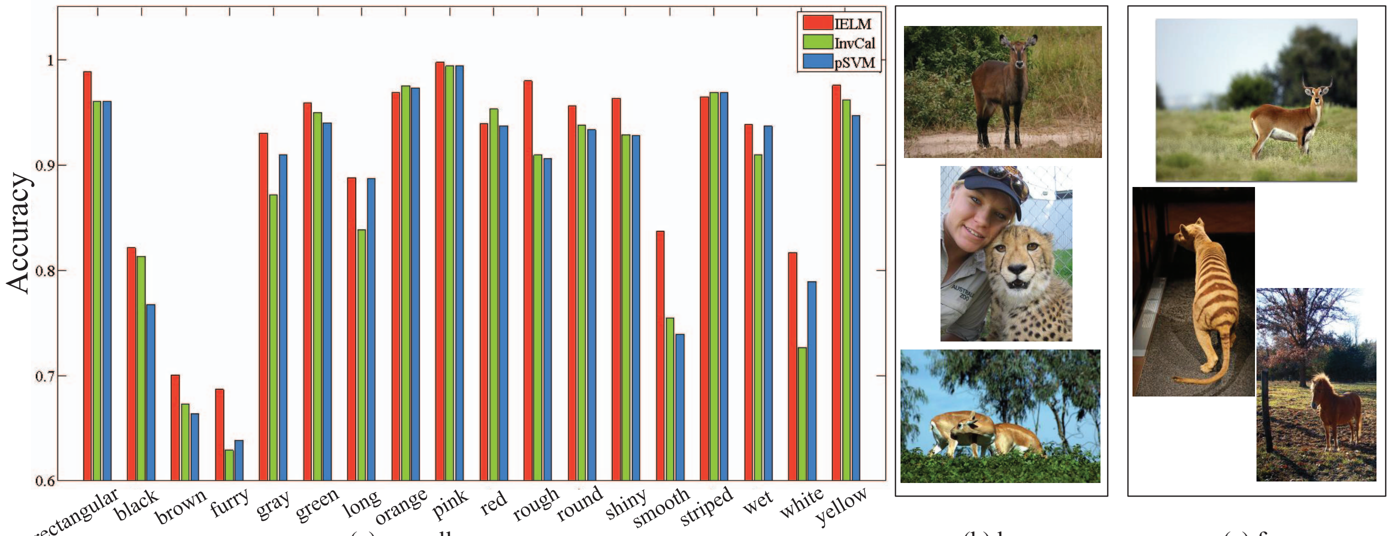
B. Case Study on ImageNet

In this section, to further evaluate the efficiency of IELM, we are going to present the experimental results on the ImageNet [42]. As it is known, ImageNet is an image dataset organized according to the WordNet hierarchy. In our experiment, we use the Object Attributes dataset. The dataset consists of 9600 images from 384 synsets and 25 attributes including color, pattern, shape and texture. In order to obtain the ground truth data, Russakovsky et al. [42] use workers on Amazon Mechanical Turk (AMT) to label 25 images randomly chosen from each synset. An image is considered to be a positive (negative) example of an attribute if all subjects agree that this is a positive (negative) example. Each image is presented by three types of normalized histogram features: 1) RGB color histogram, 2) texture histogram of quantized SIFT descriptors and 3) shape histogram of quantized shape-context features with edges computed using the Pb edge detector [43].

The proposed method IELM can be applied in the image tagging and image retrieval tasks. With the input image, the attributes can be easily generated. Also, each image can be represented through a vector where each dimension represents an attribute. Images with similar distribution of the attributes can be matched together in the retrieval task.

Similarly, to formulate the LLP problem, images belonging to the same synsets can be regarded as a bag. The proportion of positive attributes such as "is brown" is the bag-level label. We take 80% images to be the training datasets and the other 20% as the testing ones. Individual classifiers can be trained for each attribute on all the testing images and the generalization performance can also be evaluated. Similarly, pNPSVM doesn't run on ImageNet due to its high computational cost. Part of the results is shown in Table VI. As presented in this table, for each image, the obtained attributes by different methods are shown in the last three columns, where the correct attribute is marked in red and the inaccurate one is marked in blue. For example, for the fifth image, which shows two raccoons, the IELM predicts its features are black, furry and gray, while InvCal and $\propto$SVM predict black, furry and white. Compared with the ground truth, the IELM obtains all the correct features but InvCal and $\propto$SVM predict one incorrect feature. In summary, the IELM achieves the best results in all cases, while InvCal achieves a draw on two cases.

In addition, the overall accuracy on each attribute on ImageNet Object Attributes dataset is presented in Fig. 4(a). As it is shown, IELM obtains the best performance at most attributes and gets highly competitive results on the others compared with InvCal and $\propto$SVM. Moreover, all these three methods get unsatisfying performance on the attributes "brown" and "furry", as a result of the great variety of the exemplars (see Fig. 4(b) and Fig. 4(c) for examples of "brown" and "furry" images), which makes the features of these attributes hard to capture effectively.



(a) overall accuracy (b) brown (c) furry

Fig. 4. The overall accuracy on ImageNet Object Attributes dataset is shown in (a). For each attribute, we can see the accuracy achieved by different methods. As it is shown, IELM obtains the best performance at most attribute and gets the highly competitive results on the other compared with InvCal and ∝SVM. Moreover, all these three methods get poorly performance on the attributes "brown" and "furry". (b) and (c) shows some example images labeled by the human subjects as "brown" and "furry" respectively. As it is shown, huge difference exist between different images for the same attribute (i.e. "brown"), which makes it is hard to capture the effective features to describe the attribute.

V. Conclusion

We present a fast attribute modeling method (IELM) with label proportions that improves performance compared to previously published methods. It can achieve the highly competitive or even better classification accuracy compared with some state-of-the-art LLP algorithms on the standard weakly labeled UCI datasets. In addition, the proposed IELM is several to hundreds times faster, which can effectively alleviate the issue that the labeled data is quite labor-intensive and time-consuming to be acquired. The IELM is also applied on an image database to evaluate its efficiency. In conclusion, the proposed method can be a feasible method for learning with label proportions, which is conceivable in many practical applications.

Source codes will be made publicly available.

Acknowledgment

This work is supported in part by NSF through grants IIS-1526499 and CNS-1626432, and NSFC through grants 61672313, 91546201, 71331005.

References

[1] J. Hernández-González, I. Inza, and J. A. Lozano, "A novel weakly supervised problem: Learning from positive-unlabeled proportions," in *Advances in Artificial Intelligence*. Springer, 2015, pp. 3–13.

[2] O. Chapelle, B. Schölkopf, A. Zien *et al.*, "Semi-supervised learning," 2006.

[3] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1–130, 2009.

[4] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Advances in neural information processing systems*, 2002, pp. 561–568.

[5] R. C. Bunescu and R. J. Mooney, "Multiple instance learning for sparse positive bags," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 105–112.

[6] S. Rueping, "Svm classifier estimation from group probabilities," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 911–918.

[7] F. Yu, D. Liu, S. Kumar, J. Tony, and S.-F. Chang, "∝ svm for learning with label proportions," in *Proceedings of the 30th International Conference on Machine Learning*, 2013, pp. 504–512.

[8] D. R. Musicant, J. M. Christensen, and J. F. Olson, "Supervised learning by training on aggregate outputs," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 252–261.

[9] N. Quadrianto, A. J. Smola, T. S. Caetano, and Q. V. Le, "Estimating labels from label proportions," *The Journal of Machine Learning Research*, vol. 10, pp. 2349–2374, 2009.

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

[11] G. Patrini, R. Nock, T. Caetano, and P. Rivera, "(almost) no label no cry," in *Advances in Neural Information Processing Systems*, 2014, pp. 190–198.

[12] G. Patrini, F. Nielsen, R. Nock, and M. Carioni, "Loss factorization, weakly supervised learning and label noise robustness," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, pp. 708–717.

[13] E. M. Ardehaly and A. Culotta, "Domain adaptation for learning from label proportions using self-training," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. AAAI Press, 2016, pp. 3670–3676.

[14] J. Hernández-González, I. Inza, and J. A. Lozano, "Learning from proportions of positive and unlabeled examples," *International Journal of Intelligent Systems*, vol. 32, no. 2, pp. 109–133, 2017.

[15] F. X. Yu, K. Choromanski, S. Kumar, T. Jebara, and S.-F. Chang, "On learning from label proportions," *arXiv preprint arXiv:1402.5902*, 2014.

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

[17] 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

Proceedings of the ACM International Conference on Multimedia. ACM, 2014, pp. 367–376.

[18] K.-T. Lai, F. X. Yu, M.-S. Chen, and S.-F. Chang, "Video event detection by inferring temporal instance labels," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 2251–2258.

[19] T. Liebig, M. Stolpe, and K. Morik, "Distributed traffic flow prediction with label proportions: from in-network towards high performance computation with mpi," in *Proceedings of the 2nd International Conference on Mining Urban Data-Volume 1392*. CEUR-WS. org, 2015, pp. 36–43.

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

[21] J. Hernández-González, I. Inza, L. Crisol-Ortíz, M. Guembe, M. Iñarra, and J. Lozano, "Fitting the data from embryo implantation prediction: Learning from label proportions." *Statistical methods in medical research*, 2016.

[22] D. Hübner, T. Verhoeven, K. Schmid, K.-R. Müller, M. Tangermann, and P.-J. Kindermans, "Learning from label proportions in brain-computer interfaces: online unsupervised learning with guarantees," *PloS one*, vol. 12, no. 4, p. e0175856, 2017.

[23] F. X. Yu, L. Cao, M. Merler, N. Codella, T. Chen, J. R. Smith, and S.-F. Chang, "Modeling attributes from category-attribute proportions," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 977–980.

[24] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 453–465, 2014.

[25] Z. Qi, B. Wang, F. Meng, and L. Niu, "Learning with label proportions via npsvm," *IEEE Transactions on Cybernetics*, 2016.

[26] V. Cheplygina, D. M. Tax, and M. Loog, "On classification with bags, groups and sets," *Pattern Recognition Letters*, vol. 59, pp. 11–17, 2015.

[27] J. Hernández-González, I. Inza, and J. A. Lozano, "Weak supervision and other non-standard classification problems: a taxonomy," *Pattern Recognition Letters*, vol. 69, pp. 49–55, 2016.

[28] B.-C. Chen, L. Chen, R. Ramakrishnan, and D. R. Musicant, "Learning from aggregate views," in *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*. IEEE, 2006, pp. 3–3.

[29] H. Kuck and N. de Freitas, "Learning about individuals from group statistics," *arXiv preprint arXiv:1207.1393*, 2012.

[30] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 625–632.

[31] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[32] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 694–699.

[33] I. Cohen and M. Goldszmidt, "Properties and benefits of calibrated classifiers," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2004, pp. 125–136.

[34] M. P. Naeini and G. F. Cooper, "Binary classifier calibration using an ensemble of near isotonic regression models," in *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 2016, pp. 360–369.

[35] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 2. IEEE, 2004, pp. 985–990.

[36] G.-B. Huang, Q.-Y. Zhu, and C. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[37] J. A. Fill and D. E. Fishkind, "The moore–penrose generalized inverse for sums of matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 2, pp. 629–635, 2000.

[38] F. Toutounian and A. Ataei, "A new method for computing moore–penrose inverse matrices," *Journal of Computational and applied Mathematics*, vol. 228, no. 1, pp. 412–417, 2009.

[39] R. Tapia, "Practical methods of optimization, volume 2: Constrained optimization (r. fletcher)," 1984.

[40] X. Liu, S. Lin, J. Fang, and Z. Xu, "Is extreme learning machine feasible? a theoretical assessment (part i)," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 1, pp. 7–20, 2015.

[41] Z. Chen, Z. Qi, B. Wang, L. Cui, F. Meng, and Y. Shi, "Learning with label proportions based on nonparallel support vector machines," *Knowledge-Based Systems*, 2016.

[42] O. Russakovsky and L. Fei-Fei, "Attribute learning in large-scale datasets," in *European Conference on Computer Vision*. Springer, 2010, pp. 1–14.

[43] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 5, pp. 530–549, 2004.