Learning from label proportions on high-dimensional data

Yong Shi a,b,c,f, Jiabin Liu d, Zhiquan Qi c,a,b,*, Bo Wang e

a Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China
b Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, China
c School of Economics and Management, University of Chinese Academy of Sciences, Beijing, 100190, China
d School of Computer and Control Engineering, University of Chinese Academy Sciences, Beijing 100190, China
e School of Information Technology and Management, University of International Business and Economics, Beijing 100029, China
f College of Information Science and Technology, University of Nebraska at Omaha, NE 68182, USA

ARTICLE INFO

Article history:
Received 4 August 2017
Received in revised form 5 February 2018
Accepted 6 March 2018
Available online 20 March 2018

Keywords:
Optimization
High-dimensional data
Learning from label proportions (LLP)
Random forests

ABSTRACT

Learning from label proportions (LLP), in which the training data is in the form of bags and only the proportion of each class in each bag is available, has attracted wide interest in machine learning. However, how to solve high-dimensional LLP problem is still a challenging task. In this paper, we propose a novel algorithm called learning from label proportions based on random forests (LLP-RF), which has the advantage of dealing with high-dimensional LLP problem. First, by defining the hidden class labels inside target bags as random variables, we formulate a robust loss function based on random forests and take the corresponding proportion information into LLP-RF by penalizing the difference between the ground truth and estimated label proportion. Second, a simple but efficient alternating annealing method is employed to solve the corresponding optimization model. At last, various experiments demonstrate that our algorithm can obtain the best accuracies on high-dimensional data compared with several recently developed methods.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

In the era of big data, the data amount is rapidly increasing, which leads to many classification problems fail to be efficiently solved by the traditional machine learning algorithms (Adankon & Cheriet, 2002; Breiman, 2001; Liu & Yao, 1999; Schmidhuber, 2014). For example, the performance of supervised learning algorithms extremely depends on the amount of labeled training data. However, as the number of training data increases, it is becoming infeasible or quite labor-intensive to obtain labels of instances. In fact, compared to the ground-truth label of each instance, the proportion of each class can be obtained much more easily and cheaply by random sampling or other prior knowledge. As a result, in practice it is very meaningful to learn a classifier only from the information of class proportions and instances without labels.

In recent years, learning from label proportions (LLP) has been proposed as a solution to the above problem (Quadrianto, Smola, Caetano, & Le, 2008; Yu, Liu, Kumar, Jebara, & Chang, 2013). In detail, it is a learning task where the training data is provided in the form of bags and only the proportions of the labels in each bag are available. Fig. 1 provides an illustration of this classification problem. The “o” is the unlabeled data, which is partitioned into four bags and there is no intersection between the data of different bags. In each bag the sizes of red and green rectangles denote the amount of different classes and a proportion information is available according to different sizes. A classifier can be trained only by the proportion information and instances without labels. On the right, the red and green points, respectively, represent different classes and the blue line denotes the hyperplane generated by the classifier.

As increasingly practice applications can be abstracted to this problem, LLP has received amount of attention from the research community. Some real-world applications for LLP are given as follows.

In the case of political election (Rüping, 2010), the voters can be divided into two groups: always-favorable voters and swing voters where the latter will make their decision depending on what the candidates can offer them. If the candidates want to win the election, the proportions of always-favorable voters plus a set of swing voters should exceed 50%. However, due to the limit of finance, campaign time and ability to make election promises, the candidates would like to focus on their attention to the regions where they can achieve the largest gains. As a result, it is very important for them to identify which class each voter belongs to according to proportions information revealed by the previous elections.

https://doi.org/10.1016/j.neunet.2018.03.004
0893-6080/© 2018 Elsevier Ltd. All rights reserved.
where the mean of each bag is treated as an instance. More specifically, \( w, b \) are the model parameters of SVR and \( \xi_i, \xi_i^* \) are two slack variables. The constant \( C \) determines the penalty between the difference of real and predicted values, where the maximum tolerable error is defined by \( \epsilon_i \). Furthermore, \( S_i \) is the number of total instances in ith bag and \( \frac{1}{S_i} \sum_{j \in S_i} (w x_j + b) \) represents the ith bag mean. One limitation of Inverse Calibration is that \( p_i = (1 + \exp(-w^T m_i + b))^{-1} \) is not a good way of measuring the proportion predicted and the performance can be extremely terrible in some uncommon cases argued by Yu et al. (2013).

Recently, a new method based on large-margin framework was proposed by Yu et al. (2013) by jointly optimizing the unknown instance labels and the known label proportions. The hyperplane was obtained by reaching a systemic compromise between large margin principle (LMP) and empirical proportion risk minimization principle (EPRMP). In detail, this LLP algorithm can be expressed as follows:

\[
\min_{y, w, b} \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} L(y_i, w^T \phi(x_i) + b) + C_s \sum_{k=1}^{K} L_p(p_k(y), p_k) \\
\text{s.t. } y_i \in \{-1, 1\}
\]

where \( L() \) is the loss function from SVM and \( L_p(.) \) is a loss function to penalize the difference between the ground-truth and estimated proportion.

This algorithm outperforms the former known methods in most situations, and alleviates the need for making restrictive assumptions on the data. However, there also exist several limitations about this method. On one hand, this algorithm leads to a non-convex integer programming problem, which is always difficult to be solved. On the other hand, the performance of this algorithm is obtained on low-to-medium dimensional data. It is uncertain whether it has a similar performance when dealing with high-dimensional data.

Besides, Stolpe & Morik (2011) proposed a method based on clustering, which suffered from the extremely high computing complexity. Fan, Zhang, Yan, Wang, Zhang, and Feng (2014) formulated the learning from label proportion problem in a density estimation framework and proved sample complexity upper bound of this setting. Wang, Chen, and Qi (2015) proposed a new classification model based on twin SVM, which is in a large-margin framework and only needs to solve two smaller problems. Chen, Qi, Wang, Cui, Meng, and Shi (2017) and Qi, Wang, Meng, and Niu (2017) tried to address the LLP problem via NPSVM, where the former solved it by a generalized classifier instead of a transductive learning framework, and the latter incorporated the label proportion information with the unknown labels into one optimization model under a large-margin framework. Ding, Li, and Yu (2017) presented an efficient SVM-based classification framework for high-resolution SAR images. More specifically, it extended the method of Yu et al. (2013) by implementing a reweighting strategy for the training data. Shi, Cui, Chen, and Qi (2017) proposed a new SVM-based method by replacing hinge loss with pinball loss, and the model is effective to eliminate the impact of noise. Fish and Reyzin (2017) solved foundational questions which consider the computational complexity of LLP. In detail, they compared the computational complexity of learning in this model to classical PAC learning, and also gave an algorithm that demonstrated the feasibility of learning under well-behaved distributions.

### 1.2. Motivation

In the era of big data, as the dimension of data has increased substantially, high-dimensional data has become a trend for many machine learning problems. Meanwhile, how to solve
high-dimensional LLP problem is becoming more and more essential and indispensable. For example, in the case of filtering images (Quadrianto et al., 2008), images are classified into images with proper content or image with improper content according to their contained objects. Traditional methods are adapting supervised learning to train a classifier, which is quite labor-intensive to obtain adequate labeled training instances. In practice, it is considerably cheaper and easier to obtain a estimate of the proportions of proper and improper content. That is said, it is a typical high-dimensional LLP problem and an efficient solution to it can extremely reduce labor cost.

Although most of the proposed methods give effective solutions to LLP (Quadrianto et al., 2008; Rüping, 2010; Yu et al., 2013), their performances are evaluated only on low-to-medium dimensional data. As we all know, learning methods can behave very differently when the dimension of learning data is changed (Caruana, Karampatziakis, & Yessenalina, 2008). That is to say, an algorithm may perform worse on high-dimensional data while it has a good result on low-dimensional data. As a result, a practical problem is extremely when the dimension of learning data is changed (Caruana, 2001). That is to say, an algorithm may perform worse on high-dimensional data while it has a good performance on low-to-medium dimensional data (Banfield, Hall, Bowyer, & Kegelmeyer, 2007). Meanwhile, Caruana et al. (2008) presented an empirical evaluation on high dimensional data of different methods, and found that random forests perform consistently well across all dimensions compared with other methods. Additionally, it is easy for random forests to be parallelized, which makes them very easy for multicore and GPU implementations. Sharp (2008) has shown that GPU can accelerate the random forests and have great advantage compared to CPU in processing speed, which is very useful for practical applications. Recently, random forests have been applied in video segmentation (Perbet, Stenger, & Maki, 2009), object detection (Gall & Lempitsky, 2009), image classification (Bosch, Zisserman, & Munoz, 2007) and remote sensing (Pal, 2005) due to its advantages.

In this section, the random forests which are used for our classification are presented. Random forests are an ensemble learning method together with a bagging procedure for classification and other tasks, where each basic classifier is a decision tree and each tree depends on a collection of random variables. More specifically, during splitting of a randomized tree, each decision node randomly selects a set of features and then picks the best among them according to some quality measurement (e.g., information gain or Gini index) (Saffari, Leistner, Santner, & Godec, 2009). Furthermore, as each tree in the forest is built and tested independently from other trees, the overall training and testing procedures can be performed in parallel (Leistner, Saffari, Santner, & Bischof, 2009).

We denote the $m$th tree of random forests as $f(x, \theta_m)$, where $\theta_m$ is a random vector representing the various stochastic elements of the tree. Meanwhile, let $\rho_m(k|x)$ represent the estimated density of class labels for the $m$th tree and $M$ be the total number of the trees in the forests. In practice, the final prediction results of random forests are given by probability towards different classes. As a result, the estimated probability for predicting class $k$ in random forests can be defined as follows:

$$F_k(x) = \frac{1}{M} \sum_{m=1}^{M} \rho_m(k|x), \quad k \in \gamma = \{1, 2, \ldots, K\},$$

where $K$ is the total number of classes. In particular, a decision can be made by simply taking the maximum over all individual probabilities of the trees for a class $k$ with

$$C(x) = \arg \max_{k \in \gamma} F_k(x), \quad \gamma = \{1, 2, \ldots, K\}$$

where the final result of $C(x)$ is the index of the corresponding class.

The classification margin measures the extent to which the average number of votes for the right class exceeds the average for any other class, which is introduced by Breiman (2001), and is expressed as follows:

$$m_g(x, y) = F_y(x) - \max_{k \neq y} F_k(x).$$

Obviously, if the classification is correct, there should be $m_g(x, y) > 0$. In other words, the larger the margin is, the more confidence in the classification. The generalization error of random forests is in the form of

$$GE = E_{x,y}(m_g(x, y) < 0),$$

where the expectation is measured over the entire distribution of $(X, Y)$.

Random forests have shown its advantages in both classification (Breiman, 2001) and clustering (Schlkopf, Platt, & Hofmann, 2006). In particular, experiments have shown that high accuracy can be achieved by random forests when classifying high dimensional data (Banfield, Hall, Bowyer, & Kegelmeyer, 2007). Meanwhile, Caruana et al. (2008) presented an empirical evaluation on high dimensional data of different methods, and found that random forests perform consistently well across all dimensions compared with other methods. Additionally, it is easy for random forests to be parallelized, which makes them very easy for multicore and GPU implementations. Sharp (2008) has shown that GPU can accelerate the random forests and have great advantage compared to CPU in processing speed, which is very useful for practical applications. Recently, random forests have been applied in video segmentation (Perbet, Stenger, & Maki, 2009), object detection (Gall & Lempitsky, 2009), image classification (Bosch, Zisserman, & Munoz, 2007) and remote sensing (Pal, 2005) due to its advantages.

### 3. The LLP-RF algorithm

In this section, we present a novel learning from label proportions algorithm called LLP-RF, which use random forests to solve high-dimensional LLP problem. In order to leverage random forests to LLP, the hidden class labels inside bags are defined as the optimization variables. Meanwhile, we formulate a robust loss function to LLP, the hidden class labels inside bags are defined as the optimization variables. In this paper, we assume that the bags are disjoint. Let $B_i, i = 1, \ldots, n$ denote the $i$th bag in the training set. As a result, the total training data can be expressed as follows:

$$D = B_1 \cup B_2 \cup \cdots \cup B_n$$

$$B_i \cap B_j = \emptyset, \forall i \neq j,$$

where the total number of training data is $N$. The $i$th bag consists of $m_i$ instances and is in the form of

$$B_i = [x_{i1}^1, \ldots, x_{im_i}^i] \{p_i\}, \quad i \in \{1, 2, \ldots, n\},$$

where the associated $p_i$ indicates the label proportion of the $i$th bag. As a result, the $j$th instance in the $i$th bag can be expressed as $\chi_{ij}$.

The ground-truth labels of instances are modeled as $Y$, and the ground-truth labels of instances are modeled as $Y$.

$$\tilde{y}_i = \{y_1, \ldots, y_n\}^T,$$

where $y_1$ is the unknown label of $x_1$. Furthermore, we can define the proportion of $i$th bag as follows:

$$p_i = \frac{\sum_{k \in \gamma} \frac{|\{k \in \gamma | y_k = 1\}|}{|B_i|}}{|B_i|}, \quad \forall k \in \{1, 2, \ldots, N\},$$

where $|B_i|$ is the number of instances in the $i$th bag. The ground-truth labels of instances are modeled as $Y$.
in which \( y^*_k \in \{1, -1\} \) is the unknown ground-truth label of \( x_k \) and \(|B_i|\) denotes the bag size of the \( i \)th bag. In practice, the above formulation is equivalent to the following:

\[
p_i = \frac{\sum_{k \in B_i} y^*_k}{2|B_i|} + \frac{1}{2}, \quad \forall k \in \{1, 2, \ldots, N\}.
\] (10)

3.2. The LLP-RF framework

The above LLP-RF learning setting is very intuitive and the final objective is to train a classifier in the instance level. To this end, inspired by Leistner, Saffari, and Bischof (2010) and Yu et al. (2013), we formulate a robust loss function based on random forests and take the corresponding proportion information into LLP-RF by penalizing the difference between the ground-truth label and estimated label proportion. Therefore, the final objective function of LLP-RF is formulated as follows:

\[
\begin{align*}
\argmin_{F(.)} & \sum_{j \in B_i} \sum_{l=1}^{m_l} \ell(F_j(x^*_l)) + C_L \sum_{l=1}^{m_l} L_p(p_l, p_l) \\
\text{s.t.} & \forall j, y_j \in \{1, -1\},
\end{align*}
\] (11)

where the hidden class labels \( y \) are defined as the optimization variables and the task is to simultaneously optimize the labels \( y \) and the model \( F(.) \).

Specifically, \( L(.) \) is a loss function which is defined over the entire set of instances and \( L_j(.) \) is a loss function used to penalize the difference between the ground-truth label proportion and the estimated label proportion based on \( y \). Different weights can be added for the loss of bag proportions by changing the value of \( C_L \).

Note that \( F_j(x) \) is the confidence of classifier for the \( k \)th class, which is got from random forests.

Furthermore, our proposed framework permits choosing different loss functions for \( L(.) \). In our paper, different loss functions including hinge loss, logistic loss and entropy are tuned to obtain better classification results. In this paper, we consider \( L_j(.) \) as the absolute loss:

\[
L_p(p_l, p_l) = |p_l(p_l) - p_l|.
\] (12)

where \( p_l \) is the true label proportion of \( i \)th bag and \( p_l(y) \) is the estimated label proportion of \( i \)th bag.

The above LLP-RF framework is fairly straightforward and intuitive. However, it leads to a non-convex integer programming problem because it needs to simultaneously optimize the labels \( y^*_k \) and trains a random forest. In practice, the problem is often NP-hard. Therefore, one key issue is how to solve the optimization problem efficiently. In this paper, a simple but efficient alternating optimization strategy based on annealing is employed to minimize the overall learning objective.

3.3. How to solve the LLP-RF

The strategy to solve (11) is similar to the rule from Yu et al. (2013). There are two variables \( F \) and \( y \) in the optimization formula, where the unknown instance labels \( y \) can be seen as a bridge between supervised learning loss and label proportion loss. Therefore, we solve the problem by alternating optimizing the two variables \( F \) and \( y \).

- We fix the \( y \). The optimization problem becomes a native random forests problem, which can be expressed as follows:

\[
\argmin_{F(.)} \sum_{j = 1}^{m} \sum_{l=1}^{m_l} \ell(F_j(x^*_l)).
\] (13)

- Then, \( F \) is fixed. The problem can be transformed to the following:

\[
\begin{align*}
\argmin_{y_j} & \sum_{l=1}^{m_l} \ell(F_j(x^*_l)) + C_L \sum_{l=1}^{m_l} L_p(p_l, p_l) \\
\text{s.t.} & \forall j, y^*_j \in \{1, -1\}.
\end{align*}
\] (14)

The first term of the objective is defined over the entire instances. However, the proportion information \( p_l \) of the second term is provided in the bag level. In order to use the proportion information efficiently, the above formula can be written to the following:

\[
\begin{align*}
\argmin_{y_j} & \sum_{l=1}^{m_l} \left\{ \ell(\hat{F}_j(x^*_l)) + C_L L_p(p_l, p_l) \right\} \\
\text{s.t.} & \forall j, y^*_j \in \{1, -1\}.
\end{align*}
\] (15)

As the bags are disjoint to each other, the contribution of each bag to the objective is independent. As a result, the objective can be optimized on each bag separately and the final result is equivalent to the summation of every bag. In particular, solving \( \{y_j^*_j \in B_i\} \) yields the following optimization problem:

\[
\begin{align*}
\argmin_{\{y_j \in B_i\}} & \sum_{j \in B_i} \ell(F_j(x^*_j)) + C_L L_p(p_l, p_l) \\
\text{s.t.} & \forall j \in B_i, \quad y^*_j \in \{1, -1\}.
\end{align*}
\] (16)

Obviously, the original optimization problem has changed to solve the formula (16), whose solution can be found by the following optimization strategy.

Remark 1. The steps for solving formula (16).

- Compute all the possible values of the second term in formula (16), where there are totally \(|B_i| + 1\) values. In practice, the \( k \)th value can be expressed as follows:

\[
F_2(k) = \frac{k - 1}{|B_i|} - p_l, \quad k \in \{1, 2, \ldots, |B_i|, |B_i| + 1\}.
\] (17)

- Obtain all the values of the first term \( F_1(k) \) corresponding to the second term \( F_2(k) \).

- Pick the smallest objective value from

\[
C * F_1(k) + C_L * F_2(k), \quad k \in \{1, 2, \ldots, |B_i|, |B_i| + 1\},
\] (18)

yielding the optimal solution of (16).

The above strategy is fairly intuitive and straightforward. The main focus is how to obtain the value of the first term corresponding to the second term. In practice, there are totally \(|B_i| + 1\) values about the second term. For a fixed value of the second term, steps can be taken as Proposition 1.

Proposition 1. For a fixed \( p_l(y) = \theta \), we can find the solution of (16) by the iterative steps as follows.

- Initialize \( y_j = -1, \quad \forall j \in \{1, 2, \ldots, |B_i|\} \), where \(|B_i|\) is the number of instances in \( i \)th bag.

- Compute the value of \( \ell(F_j(x^*_j)) \), \( j \in \{1, 2, \ldots, |B_i|\} \).

- Flip the sign of \( y_j = 1, \quad \forall j \in \{1, 2, \ldots, |B_i|\} \).

- Compute the value of \( \ell(F_j(x^*_j)) \), \( j \in \{1, 2, \ldots, |B_i|\} \).

- Let \( \bar{y}^*_j \equiv C(\ell(F_j(x^*_j)) - \ell(F_j(x^*_j))) \), \( j \in \{1, 2, \ldots, |B_i|\} \) denote the reduction of the first term in (16) through flipping the sign of \( y_j^*_j \).

- Sort \( \bar{y}^*_j, \forall j \in \{1, 2, \ldots, |B_i|\} \) in descending way. Then flip the signs of \( y_j^*_j \) of the top-R \((R = \theta |B_i|)\) which have the highest
obtain the smallest subjective value as the final result.

**Algorithm 1 LLP-RF**

Require: Bags\(B_i\); The corresponding proportion \(p_i\) of \(B_i\);
Randomly initialize \(y_j^i \in \{1, -1\}, \forall j \in \{1, \ldots, |B_i|\}\);\(C^* = 10^{-3}C\).

while \(C^* < C\) do

\(C^* = \min((1 + \Delta)kC^*, C)\).

repeat

Fix \(y\) to solve \(F(\text{Train the Random Forests: } trainRF(y^i))\)
Fix \(F\) to solve \(y\) using the strategy discussed in Remark 1).
Update \(y_j^i, \forall j, \forall^m_i, \forall^m_j\).

until the decrease of the objective is smaller than a threshold or reach the setting iteration.
end while

4. Experiment

The purpose of this section is to evaluate the proposed algorithms on different datasets. In our experiment, we compare our method to InvCal (Rüping, 2010), conv-\(\alpha\)-SVM (Yu et al., 2013) and alter-\(\alpha\)-SVM (Yu et al., 2013), which have been proven to outperform the previous algorithms.

4.1. How does LLP-RF works

In this section, we give a simple example on the 2-D toy data shown in Fig. 2 to demonstrate how our method works. Specially, there are totally two classes and 100 points in each class about the data, where "□" and "□" represent different classes, respectively. The data is partitioned into 2 bags and different bags are marked with different colors.

In order to demonstrate how LLP-RF works, a relatively worse result is chosen to present the iterative process of our algorithm. Fig. 3 provides the iterative process of our algorithm on the toy data. In detail, the accuracy on the top indicates the predicted result in each step and it takes LLP-RF eight steps to achieve a stable result with the final accuracy reaching to 100%. The blue lines are the output of a trained classifier on a 2D grid obtained by the LLP-RF.

Obviously, from the change of accuracy on the top, the convergence process of the algorithm can be easily revealed, where the accuracy is monotonously increasing with the increase of iteration number. At the last figure, we can clearly see that there is only one blue line and all samples are correctly classified, where all positive points and negative points are in the opposite side of blue line. In fact, the number of iterations is related to the initiation \(y\) and our algorithm can always be stabilized in ten iterations.

4.2. Experimental setup

In bag setting, the data is randomly split into different bag sizes (number of instances in the bag) in our experiment, with varied bag sizes ranged in 2, 4, 8, 16, 32, and 64. Furthermore, the data is partitioned into training and test sets, where 80% are used for training, and 20% for test. We repeat the experiment 5 times with re-split training and testing data and the final performance is given by the average classification accuracy.

Initialization is necessary for alter-\(\alpha\)-SVM and LLP-RF because both of them adapt an alternating optimization method to get the final results. In practice, a stochastic method is used to generate initial labels for alter-\(\alpha\)-SVM and LLP-RF according to the label proportions of different bags. Furthermore, in order to reduce the influence of random initialization, we randomly initialize alter-\(\alpha\)-SVM and LLP-RF 20 times, and pick the result from the one with lowest objective function in multiple tests.

In particular, for LLP-RF, 15 trees are used in the experiment and the information gain is chosen for the node splitting criterion. Meanwhile, we set the maximum tree depth 32. For alternating optimization process, we allow a maximum of 10 iterations.

All the parameters are tuned in the criterion of fivefold cross validation. In detail, the parameters of different algorithms are tuned as follows:

- **InvCal**: \(C_p \in [0.1, 1, 10], \, \varepsilon \in [0, 0.1, 0.01]\)
- alter-\(\alpha\)-SVM: \(C \in [0.1, 1, 10], \, C_p \in [1, 10, 100]\)
- conv-\(\alpha\)-SVM: \(C \in [0.1, 1, 10], \, \varepsilon \in [0, 0.1, 0.01]\)
- LLP-RF: \(C \in [0.1, 1, 10], \, C_p \in [1, 10, 100]\)

In our experiment, we mainly focus on binary classification situation. For datasets with multiple classes, we pick two classes from the datasets and perform the 1-versus-1 binary classification.

4.3. Machine learning datasets

We compare the four methods on 7 binary classification problems whose dimensions range from 166 to 10,000. A summary...
The iterative process of LLP-RF in the toy data. In detail, the accuracy on the top indicates the predicted result in each step, and it takes LLP-RF eight steps to achieve a stable result with the final accuracy reaching to 100%. The blue lines are the output of a trained classifier on a 2D grid obtained by the LLP-RF.

Table 1
Datasets in the experiment.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size</th>
<th>Attributes</th>
<th>Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Musk1</td>
<td>476</td>
<td>166</td>
<td>2</td>
</tr>
<tr>
<td>Milt-Feature</td>
<td>2000</td>
<td>649</td>
<td>10(1, 2)</td>
</tr>
<tr>
<td>Mnist</td>
<td>600</td>
<td>780</td>
<td>10(1, 2)</td>
</tr>
<tr>
<td>Hand</td>
<td>200</td>
<td>2500</td>
<td>5(1, 2)</td>
</tr>
<tr>
<td>SVHN</td>
<td>26032</td>
<td>3072</td>
<td>10(1, 2)</td>
</tr>
<tr>
<td>GISETTE</td>
<td>1000</td>
<td>5000</td>
<td>2</td>
</tr>
<tr>
<td>ARCENE</td>
<td>900</td>
<td>10000</td>
<td>2</td>
</tr>
</tbody>
</table>

Of these datasets is presented in Table 1. Specifically, Musk1 and Mnist are two common datasets, where Musk1 is to predict whether new molecules will be musks or non-musks, and Mnist is a handwritten digit images dataset. Furthermore, GISETTE is to separate the highly confusable digits ‘4’ and ‘9’, and Milt-Feature consists of features of handwritten numerals extracted from a collection of Dutch utility maps. ARCENE is to distinguish cancer versus from mass-spectrometric data, and HAND is collected by conducting six daily life grasps using the National Instrument Labview. With respect to SVHN, it is a real-world image dataset for developing machine learning and object recognition algorithms. In particular, for the 4 multiple-class problem, we choose 2 classes from them and the indexes in the table represent the classes we selected.

Table 2 shows fivefold cross validation performance of InvCal, conv-αSVM, alter-αSVM, and LLP-RF on each of the seven datasets. In detail, the datasets are arranged in the order of increasing dimension and the bag size is becoming bigger from left to right. Meanwhile, the best accuracies are denoted by bold numbers.

As can be seen from this table, our method is always superior to InvCal, alter-SVM, and conv-SVM on most datasets, where LLP-RF acquires 29 best from the totally 42 results. Especially on the mnist dataset, LLP-RF outperforms the other methods in all bag sizes. Furthermore, it is obvious that the accuracies of all the methods decrease in different degrees with the increase of the bag size, which indicates bigger bag size is more challenge.

In particular, on the 2 datasets GISETTE and ARCENE, LLP-RF achieves highly competitive accuracies with InvCal, and is superior to the alter-αSVM and conv-αSVM. In practice, LLP-RF has considerable advantages over InvCal with big bag size, while InvCal performs better in small bag size. On the other 5 datasets, LLP-RF achieves the best performance for most of the bag size and InvCal is prone to acquire more competitive results than alter-αSVM and conv-αSVM.

Additionally, in order to clearly present the different classification performance of the selected methods, we introduce the average classification accuracies on the 7 datasets. The results are shown in Fig. 4, where the y-axis represents different bag sizes and different algorithms are denoted by different colors. From the results, we can clearly see that the LLP-RF obtains the smallest average error rates on different bag sizes and conv-α SVM performs relatively worse compared with the other three methods.

4.4. Caltech-101

For better performing the categorization experiments, a popular image classification benchmark dataset Caltech-101 is selected. Specially, the Caltech-101 dataset (Li, Fergus, & Perona, 2007) consists of images from 101 object classes and 1 additional background class, where each class contains from 31 to 800 images and there are 9144 images in total. Furthermore, most of images are medium resolution and about 300 × 300 pixels. Some images in Caltech-101 are shown in Fig. 5. They are part of butterflies, sunflowers, leopards, laptops and pianos ranging from top to bottom.

In particular, we resize the images into 240 × 320 pixels and HOG is used to describe the image appearance. HOG is to compute a histogram of gradients, with each gradient quantized by its angle and weighed by its magnitude. For color images, the gradient is computed separately for each color channel and then choose the
Table 2
The 5-fold cross validation’s results under the optimal parameters with bag size 2, 4, 8, 16, 32, and 64. Bold numbers denote the best accuracies.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>musk1</td>
<td>InvCal</td>
<td>63.86</td>
<td>65.68</td>
<td><strong>64.95</strong></td>
<td>59.16</td>
<td>59.79</td>
<td>56.00</td>
</tr>
<tr>
<td></td>
<td>alter-αSVM</td>
<td>73.71</td>
<td>63.59</td>
<td>60.55</td>
<td>54.75</td>
<td>55.37</td>
<td>54.70</td>
</tr>
<tr>
<td></td>
<td>conv-αSVM</td>
<td>76.67</td>
<td>66.32</td>
<td>52.00</td>
<td>56.21</td>
<td>47.37</td>
<td>49.26</td>
</tr>
<tr>
<td></td>
<td>LLP-RF</td>
<td><strong>78.20</strong></td>
<td><strong>67.16</strong></td>
<td>62.84</td>
<td><strong>66.95</strong></td>
<td><strong>60.42</strong></td>
<td><strong>56.83</strong></td>
</tr>
<tr>
<td>mfeat</td>
<td>InvCal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>alter-αSVM</td>
<td>97.25</td>
<td>83.25</td>
<td>70.00</td>
<td>66.25</td>
<td>61.00</td>
<td>58.75</td>
</tr>
<tr>
<td></td>
<td>conv-αSVM</td>
<td>20.75</td>
<td>41.00</td>
<td>25.25</td>
<td>63.25</td>
<td>80.74</td>
<td>63.00</td>
</tr>
<tr>
<td></td>
<td>LLP-RF</td>
<td>99.75</td>
<td><strong>99.25</strong></td>
<td><strong>99.50</strong></td>
<td><strong>99.00</strong></td>
<td><strong>98.62</strong></td>
<td><strong>98.62</strong></td>
</tr>
<tr>
<td>mnist</td>
<td>InvCal</td>
<td>93.83</td>
<td>89.83</td>
<td>91.67</td>
<td>94.00</td>
<td>91.33</td>
<td>90.33</td>
</tr>
<tr>
<td></td>
<td>alter-αSVM</td>
<td>75.00</td>
<td>64.00</td>
<td>58.16</td>
<td>56.67</td>
<td>56.67</td>
<td>55.33</td>
</tr>
<tr>
<td></td>
<td>conv-αSVM</td>
<td>60.60</td>
<td>6.06</td>
<td>24.00</td>
<td>26.50</td>
<td>66.50</td>
<td>32.00</td>
</tr>
<tr>
<td></td>
<td>LLP-RF</td>
<td><strong>97.10</strong></td>
<td><strong>96.66</strong></td>
<td><strong>96.00</strong></td>
<td><strong>94.02</strong></td>
<td><strong>96.33</strong></td>
<td><strong>93.83</strong></td>
</tr>
<tr>
<td>hand</td>
<td>InvCal</td>
<td>57.00</td>
<td>46.00</td>
<td>57.00</td>
<td>48.50</td>
<td>53.50</td>
<td>50.50</td>
</tr>
<tr>
<td></td>
<td>alter-αSVM</td>
<td>59.50</td>
<td>51.00</td>
<td>59.00</td>
<td>55.00</td>
<td>55.50</td>
<td>55.00</td>
</tr>
<tr>
<td></td>
<td>conv-αSVM</td>
<td>47.00</td>
<td>51.50</td>
<td>51.50</td>
<td>47.52</td>
<td>49.50</td>
<td>52.00</td>
</tr>
<tr>
<td></td>
<td>LLP-RF</td>
<td><strong>63.50</strong></td>
<td><strong>59.00</strong></td>
<td><strong>60.50</strong></td>
<td><strong>59.00</strong></td>
<td><strong>58.00</strong></td>
<td><strong>54.00</strong></td>
</tr>
<tr>
<td>SVNH</td>
<td>InvCal</td>
<td>47.00</td>
<td>50.25</td>
<td>47.75</td>
<td>51.25</td>
<td>52.00</td>
<td>50.75</td>
</tr>
<tr>
<td></td>
<td>alter-αSVM</td>
<td>50.75</td>
<td>54.75</td>
<td>53.50</td>
<td>56.75</td>
<td><strong>56.25</strong></td>
<td><strong>55.25</strong></td>
</tr>
<tr>
<td></td>
<td>conv-αSVM</td>
<td>63.50</td>
<td>62.25</td>
<td>56.25</td>
<td>55.25</td>
<td>55.75</td>
<td>53.50</td>
</tr>
<tr>
<td>GISTEYE</td>
<td>InvCal</td>
<td><strong>94.40</strong></td>
<td><strong>91.10</strong></td>
<td><strong>88.90</strong></td>
<td><strong>88.30</strong></td>
<td><strong>79.50</strong></td>
<td><strong>79.30</strong></td>
</tr>
<tr>
<td></td>
<td>alter-αSVM</td>
<td>79.00</td>
<td>63.80</td>
<td>59.50</td>
<td>57.80</td>
<td>55.60</td>
<td>54.40</td>
</tr>
<tr>
<td></td>
<td>conv-αSVM</td>
<td>76.67</td>
<td>66.32</td>
<td>52.00</td>
<td>56.21</td>
<td>47.37</td>
<td>49.26</td>
</tr>
<tr>
<td></td>
<td>LLP-RF</td>
<td>90.40</td>
<td>90.00</td>
<td><strong>88.50</strong></td>
<td><strong>86.00</strong></td>
<td><strong>84.10</strong></td>
<td><strong>79.50</strong></td>
</tr>
<tr>
<td>ARCENE</td>
<td>InvCal</td>
<td><strong>84.50</strong></td>
<td><strong>71.50</strong></td>
<td><strong>66.00</strong></td>
<td><strong>64.50</strong></td>
<td><strong>57.50</strong></td>
<td><strong>61.00</strong></td>
</tr>
<tr>
<td></td>
<td>alter-αSVM</td>
<td>80.50</td>
<td>72.00</td>
<td>63.00</td>
<td>62.00</td>
<td>62.00</td>
<td>62.00</td>
</tr>
<tr>
<td></td>
<td>conv-αSVM</td>
<td>72.00</td>
<td>67.00</td>
<td>55.00</td>
<td>51.50</td>
<td>46.00</td>
<td>58.50</td>
</tr>
<tr>
<td></td>
<td>LLP-RF</td>
<td>77.00</td>
<td><strong>72.50</strong></td>
<td><strong>64.50</strong></td>
<td><strong>64.00</strong></td>
<td><strong>62.50</strong></td>
<td><strong>63.50</strong></td>
</tr>
</tbody>
</table>

Fig. 4. The mean accuracies for the classification of different methods on the machine learning datasets. Specially, the x-axis represents different bag sizes, and different algorithms are denoted by different colors. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

In the next experiment, 4 binary classifiers are trained from the Caltech-101 dataset. In detail, the first data consists of leopards, grand–pianos and laptops, where the leopards are regarded as positive instances and the others negative. The second data is to distinguish butterflies from sunflowers and the third data is to classify pianos and desktops. Two kinds of animals dolphins and ibises are selected to form the fourth data. Furthermore, two algorithms InvCal and alter-αSVM which perform relatively better in machine learning dataset are chosen to compare with our method and conv-αSVM is excluded from the experiment because it falls far below the other methods. Fig. 6 shows the results of different data in a varied bag size. In detail, (a), (b), (c) and (d) represent different data, respectively, and the x-axis represents different bag sizes. Furthermore, different algorithms are denoted by different colors.

From the results, we can clearly see that LLP-RF outperforms the other two algorithms in most cases, with a 79% chance to rank first (acquiring 19 best from the total of 24 results), a 12.5% chance to rank second (acquiring 3 s from the total of 24 results), and less than an 8% chance to rank last (acquiring 2 last from the total of 24 results). Especially on the first dataset, LLP-RF outperforms all the other methods. Furthermore, as the increasing of the bag size, the accuracies of all methods decrease in different degrees, which is similar to the results in machine learning datasets. In addition, it is obvious to us that the average accuracies of different data vary a lot, where the first data achieves a better result than other data. In practice, the reason why the first datasets here are much better compared to the other data is that the latter problems are much more difficult.

In order to better show the advantage of our method, we also use confusion matrix to compare the performance of different algorithms. The final results are shown in Table 3. In detail, TP, FN, FP and TN, respectively, represent true positive, false positive, false positive and true negative. In particular, the results consist of

Table 3
Confusion matrix.

<table>
<thead>
<tr>
<th>Method</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LPL-RF</td>
<td>2194</td>
</tr>
<tr>
<td>alter-αSVM</td>
<td>2045</td>
</tr>
<tr>
<td>invCal</td>
<td>2001</td>
</tr>
<tr>
<td>LPL-RF</td>
<td>320</td>
</tr>
<tr>
<td>alter-αSVM</td>
<td>558</td>
</tr>
<tr>
<td>invCal</td>
<td>472</td>
</tr>
</tbody>
</table>

Note that the results show that LPL-RF outperforms the other two algorithms in most cases.
Fig. 5. Some images from the Caltech-101 dataset. They are part of butterflies, sunflowers, leopards, laptops and pianos ranging from top to bottom.

Fig. 6. The test accuracies of LLP-RF, InvCal and alter–c SVM on Caltech-101 dataset in the case of 5-fold cross validation. In detail, (a), (b), (c) and (d) represent different data, respectively, and the x-axis is the bag size. Furthermore, different algorithms are denoted by different colors. (a) Leopard–piano, laptop. (b) Butterfly–sunflower. (c) Piano–desktop. (d) dolphin and ibis. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
the Caltech-101 datasets with different bag sizes. Intuitively, the performance of different algorithms depends on the value of TP and TN, and a better algorithm will result in a bigger value of TP and TN. From the results, we can clearly see that LLP-RF outperforms the other two algorithms, with a relatively big value of TP and TN.

4.5. Wilcoxon signed-rank test

In order to better demonstrate the advantage of our algorithm, we employ Wilcoxon Signed-rank Test to compare the effect of different algorithms. More specifically, it is a nonparametric test when the observations are paired, and it can be used to determine whether two dependent samples have the same distribution. In practice, the total results of two different methods are as the input, and LLP-RF is selected to compare with the other three algorithms. Furthermore, the final output results are p-value and corresponding R+ and R−, and we can know whether the test reject the null hypothesis of zero median by the p-value.

Our results are shown in Table 4. From the table, we can see that our algorithm is significantly superior to the other three algorithms in dealing with high-dimensional problems. In detail, LLP-RF and the other one algorithm have different distributions as p-value is close to zero. Furthermore, we can conclude that the median grade of LLP-RF is greater than the median grade of the other three methods.

4.6. Effect of dimension

In this section we show the trends in performance with the change of dimension, which follows the method from Caruana et al. (2008). Specifically, the method evaluated performance on three metrics: accuracy, AUC, and squared loss and studied the effect of increasing dimensionality on the performance of different learning algorithms. The results are shown in Fig. 7, where the x-axis represents dimension on problems of increasing dimension on a log scale and the y-axis is the mean accuracy of each learning method in all bag size. Meanwhile, different methods are denoted by different colors. In particular, conv-αSVM is also excluded from the graph because it incorporates the results from the Caltech-101 datasets.

From Fig. 7, it is clear that LLP-RF outperforms the other two methods in most dimensions and InvCal is prone to perform better than alter-αSVM. In addition, it is obvious to us that the average accuracies of different dimensions vary a lot, which mainly results from the varied difficulty of different dimensional problems.

5. Conclusion

In this work, we proposed a new algorithm for high-dimensional LLP problem based on random forests. In detail, the hidden class labels inside bags were defined as random variables, and a robust loss function based on the posterior probability of random forests was formulated to leverage random forests to LLP. Meanwhile, the corresponding proportion information was taken into LLP-RF by penalizing the difference between the ground truth and estimated label proportion. In particular, a non-convex integer programming problem needs to be faced in the optimization model. To this end, a simple but efficient alternating optimization based on annealing was employed. Furthermore, two aspects were considered in the experiment. On one hand, we demonstrated how LLP-RF works with a 2D toy data. On the other hand, we evaluated the performances of several proposed LLP algorithms on high dimensional data, and abundant experiments proved our method outperformed the existing methods in dealing with high-dimensional LLP problem.

In practice, solving the problem investigated by this work could be a benefit to many practical applications. For example, our method can efficiently solve the problem of filtering images according to proportion information. However, there are also some limitations about this work. On one hand, the stability of our method is limited by the random initialization to the y. On the other hand, it is always time consuming as our method needs to do an alternating optimization strategy based on annealing. In the future work, both the efficiency and the stability of the current provided method are planned to be further improved. We also would like to present an algorithm which can solve the overlapping data and an algorithm with the bag proportion not a value but a range.

Acknowledgments

This work was partially supported by the grants from National Natural Science Foundation of China (Nos. 91546201, 61402429, and 61702099), key project of National Natural Science Foundation of China (No. 71331005), Major International (Regional) Joint Research Project (No. 71110107026).

References


