Learning from Weak-Label Data: A Deep Forest Expedition[*]

Qian-Wei Wang and **Liang Yang** and **Yu-Feng Li**
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, 210023, China
{wangqw, yangl, liyf}@lamda.nju.edu.cn

Abstract

Weak-label learning deals with the problem where each training example is associated with multiple ground-truth labels simultaneously but only partially provided. This circumstance is frequently encountered when the number of classes is very large or when there exists a large ambiguity between class labels, and significantly influences the performance of multi-label learning. In this paper, we propose LCForest, which is the first tree ensemble based deep learning method for weak-label learning. Rather than formulating the problem as a regularized framework, we employ the recently proposed cascade forest structure, which processes information layer-by-layer, and endow it with the ability of exploiting from weak-label data by a concise and highly efficient label complement structure. Specifically, in each layer, the label vector of each instance from testing-fold is modified with the predictions of random forests trained with the corresponding training-fold. Since the ground-truth label matrix is inaccessible, we can not estimate the performance via cross-validation directly. In order to control the growth of cascade forest, we adopt label frequency estimation and the complement flag mechanism. Experiments show that the proposed LCForest method compares favorably against the existing state-of-the-art multi-label and weak-label learning methods.

Introduction

Weak-label learning (Sun, Zhang, and Zhou 2010), which is a kind of weakly supervised multi-label learning, deals with the problem where each training example is associated with multiple ground-truth labels simultaneously but only partially provided. For example, the image in Fig. 1 is associated with 10 ground-truth labels, but only 5 of them are provided by annotators. Weak-label learning is frequently encountered when the number of classes is very large or when there exists a large ambiguity between classes. The problem of label incompleteness significantly influences the performance of multi-label learning. To alleviate it, many state-of-the-art weak-label learning methods were proposed in recent years (Sun, Zhang, and Zhou 2010; Zhu, Yan, and Ma 2010; Bucak, Jin, and Jain 2011; Xu, Jin, and Zhou 2013; Wu, Jin, and Jain 2013; Lin et al. 2013).

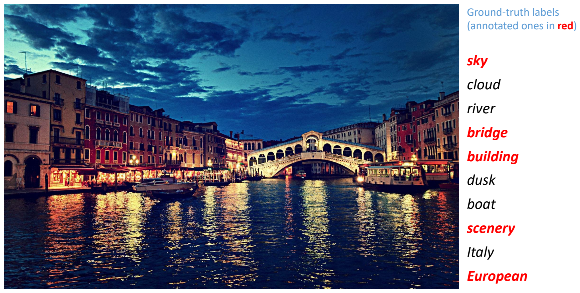

Figure 1: An example of weak-label learning scenario. The image is associated with 10 ground-truth labels, but only 5 of them are provided by annotators.

Formally, let $\mathcal{X} = \mathbb{R}^d$ be the d-dimensional instance space and $\mathcal{Y} = \{y_1, y_2, \ldots, y_n\}$ be the label space with n class labels. Given the weak-label training set $D = \{(\boldsymbol{x}_i, Y_i) | 1 \leq i \leq m\}$, where $\boldsymbol{x}_i \in \mathcal{X}$ is a d-dimensional feature vector $[x_{i1}, x_{i2}, \ldots, x_{id}]^\top$ and $Y_i \subseteq \mathcal{Y}$ is the associated set of labels, the task of weak-label learning is to learn a function $f : \mathcal{X} \mapsto 2^{\mathcal{Y}}$. It is worth noting that, in weak-label learning, since the associated label set of $\boldsymbol{x}_i$ is only partially provided, $Y_{ik} = 1$ means the k-th label is a relevant label for the i-th instance, while $Y_{ik} = 0$ tells nothing.

The difficulties of this problem mainly lie in the following three aspects. Firstly, as a kind of multi-label learning, effective exploitation of the label correlations is crucial for weak-label learning (Zhang and Zhou 2014). Secondly, even if we can ignore the label correlations information. Thus, the weak-label learning can be decomposed into a series of PU (Positive and Unlabeled) learning tasks (Li and Liu 2003; Elkan and Noto 2008), the overwhelming effect caused by false-negative labels is hard to alleviate. Thirdly, class imbalance problem naturally exists in weak-label learning. The class imbalance problem here falls into two categories: for the whole label matrix, the number of relevant labels varies

[*] This research was supported by the National Key R&D Program of China (2017YFB1001903) and the National Natural Science Foundation of China (61772262). Yu-Feng Li is the corresponding author of this work.

from one class to another; for each class, the amount of relevant labels is usually much smaller than irrelevant ones.

In this paper, we propose LCForest (Label Complement cascade Forest), which is, to our best knowledge, the first the first tree ensemble based deep learning method for weak-label learning. Rather than formulating the problem as a regularized framework, we employ the cascade forest structure proposed by Zhou and Feng (2019), which processes information layer-by-layer, and endow it with the ability of exploiting the weak-label data by a concise and highly efficient label complement structure. Specifically, in each layer, we manipulate the training set into 5 folds as 5-fold cross-validation, and in each fold, the label vector of each instance from testing-fold is modified with the predictions of random forests fitted with training-fold. Since the ground-truth label matrix is inaccessible, we can not estimate the performance via cross-validation directly as gcForest. In order to control the growing of cascade forest, we adopt the recently proposed TIcE (Bekker and Davis 2018) method in PU learning to estimate the label frequency of weak-label matrix and the complement flag mechanism.

Our method takes the three difficulties mentioned above into consideration explicitly. Firstly, in each layer of the cascade forest, the pseudo label distribution is concatenated with the original label vector, which takes the label correlations into account. Secondly, to tackle the problem caused by false-negative labels, relevant labels are complemented to the initial label matrix safely in each layer. Thirdly, for the former kinds of class imbalance problems, we introduce the complement flag mechanism to control the label complement for each class, which alleviates the problem to some extents; for the latter one, threshold θ is used to split the predicted probability and stimulate outputting more positive predictions. Experiments show that the proposed LCForest method compares favorably against the existing state-of-the-art multi-label learning, deep neural network, and weak-label learning algorithms.

Furthermore, since LCForest belongs to deep forest algorithms, it can be adapted to learn from the spatial or sequential feature relationships of image or sequence data by multi-grain scanning (Zhou and Feng 2019), which can not achieve directly by other state-of-the-art weak-label learning methods. And compared to deep neural network, LCForest inherits all the merits of deep forest including does not rely on back-propagation and can easily be trained with small datasets and lower computational cost.

The rest of this paper is organized as follows. Section 2 briefly discusses related works. Section 3 introduces LCForest. Section 4 reports the experimental results. Finally, Section 5 concludes.

Related Work

To learn from weak-label examples, many learning methods have been proposed in the past few years. The WELL method (Sun, Zhang, and Zhou 2010) employs the low-density assumption and exploits the label correlation based on the assumption that instance similarities are determined by a group of low-rank similarity matrices. Also, the WELL method explicitly considers the inherent class-imbalance problem in weak-label learning. Zhu et al., (2010) formulated the problem as a decomposition of the user-provided label matrix into a low-rank refined matrix and a sparse error matrix. The MLR-GL method (Bucak, Jin, and Jain 2011) formulates the problem as a ranking based multi-label learning framework and addressed the weak-label problem by exploiting the group lasso technique to combine the ranking errors. And Xu et al., (2013) solved the problem based on low-rank matrix completion and presented a theoretical result on the number of observed entries required for a perfect recovery.

The proposed LCForest method, which belongs to deep forest algorithms, employed the cascade forest structure proposed by Zhou and Feng (2019). After proposition, deep forest has attracted lots of attention and manifests its ability on a broad range of tasks. Pang et al., (2018) presented the gcforest$_{cs}$ method, which improves deep forest by confidence screening, that is, to pass the instances with high confidence directly to the final stage rather than passing through all the layers. The eForest (Feng and Zhou 2018) provides a tree ensemble based method for auto-encoding task, which proves that forests can carry as much information as deep neural networks. Lyu et al., (2019) proposed the casForest method, which formulated the traditional forest representation learning as an additive model, as well as theoretical results from the perspective of margin theory. The MLDF method (Yang et al. 2019) makes a first step on adapting deep forest to multi-label learning tasks by designing a multi-layer structure to learn correlations among labels. Siamese Deep Forest, proposed by Utkin and Ryabinin (2018), adapts deep forest to metric learning tasks, and can also be regarded as an alternative to Siamese neural network. And the BCDForest method (Guo et al. 2018) is an application of deep forest to cancer subtypes classification task.

Furthermore, weak-label learning is related to several other weakly supervised multi-label learning problems. Semi-supervised multi-label learning (Liu, Jin, and Yang 2006; Kong, Ng, and Zhou 2013; Zhao and Guo 2015; Zhan and Zhang 2017) attempts to exploit from a large number of unlabeled training examples in addition to limited multi-label examples. Multi-instance multi-label learning deals with the problem where each training example is associated with not only multiple instances but also multiple class (Zhou and Zhang 2006; Zhou et al. 2012). Partial multi-label learning (Xie and Huang 2018) tackles the problem where each training example is associated with multiple candidate labels which are only partially valid. Semi-supervised Weak-Label Learning (Dong, Li, and Zhou 2018) addresses the problem where only a partial or even empty label set can be observed. It is also related to semi-supervised learning (Li and Liang 2019; Li, Guo, and Zhou 2019).

The LCForest Method

In this section, we introduce LCForest. Rather than formulating the problem as a regularized framework, LCForest employs the cascade forest structure of deep forest algorithms, which processes raw features layer-by-layer. As we all know, tree-based methods have the intrinsic ability to learn from multi-label data, so the cascade forest structure can be adapted to multi-label tasks naturally. To tackle the weak-label problem, a concise and highly efficient label complement structure, which completes the label matrix in each layer, is embedded into the cascade forest. Furthermore, since the ground-truth label matrix is inaccessible, we can not estimate the performance via cross-validation directly as gcForest. In order to control the growing of cascade forest, we adopt the TIcE method to estimate the label frequency of weak-label matrix and the complement flag mechanism. Next, we will first introduce the label complement structure in each cascade layer. Then, we will introduce how we control the growing of LCForest via label frequency estimation and complement flag mechanism, followed by the overall framework of LCForest and training algorithm.

Label Complement Structure

In LCForest, the label complement structure is embedded into the cascade forest to complement relevant labels to the annotated weak-label matrix. In the t-th layer of cascade, we manipulate the training data set $D^t = \{X^t, Y^t\}$ into 5 folds as 5-fold cross-validation.

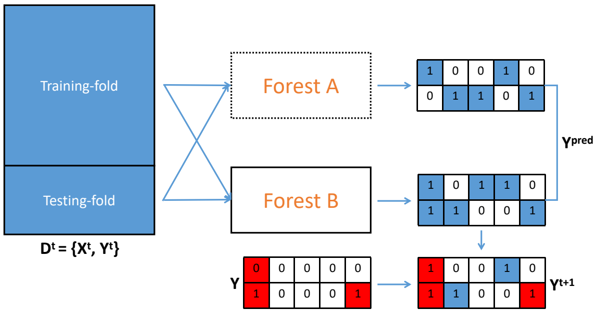

Figure 2: Label complement structure.

For each fold, as shown in Fig. 2, the random forests in the t-th layer are trained with the examples from training-fold and predict the instances from corresponding testing-fold. Then, the label vector of each example from the testing-fold is modified with the predictions from random forests. Here, for the k-th class label of the i-th example from testing-fold, if the predictions from those random forests are positive and consistent, that is, for $1 \leq j \leq numF$, $Y_{ik}^{pred\_j} = 1$, while the original label Y_{ik} is negative, the positive label will be added to the label matrix for the next layer $Y_{ik}^{t+1} = 1$. Here, $numF$ indicates the number of random forests in one layer. It is worth noting that the complementing is not performed on the training label for this layer but the original weak-label training label provided by annotators. We only adopt the simplest way in this work since it is effective and make sense. Complicated ways can be applied if needed. Finally, we can obtain the predictions of the full training set Y^{pred}, and then complete the label matrix with Eq.(1).

$$Y_{ik}^{t+1} = \prod_{j=1}^{numF} Y_{ik}^{pred\_j} \vee Y_{ik} \tag{1}$$

Label Frequency Estimation and Complement Flag Mechanism

Traditional deep forest algorithms, e.g. gcforest, control the cascade layer and avoid over-fitting by estimating the performance of the whole cascade in each layer via cross-validation. For instance, after expanding a new layer, the performance of the current training model will be estimated, and the training procedure will be terminated if there is no significant performance gain. However, in the weak-label learning scenario, the provided training label is deficient, which leads to the inaccurate estimation of cross-validation. So that we can hardly achieve this by previous methods. Actually, the controlling of the training process as well as the hyper-parameter tuning without the ground-truth label matrix is one of the common failings of weakly supervised learning.

In LCForest, we first employ the TIcE method to estimate the label frequency of the provided weak-label matrix. Here, the label frequency defines the probability that a positive example is selected to be labeled. In short, the TIcE method adopts the "selected completely at random" assumption (Elkan and Noto 2008), which always satisfied in the weak-label problem concerned in this paper, and provides us a simple and effective way to estimate the label frequency c. The key insight of TIcE is that subdomains of the data giving a lower bound of c, and finding such subsets can naturally be done via top-down decision tree induction. Technical details of the TIcE method can be found in (Bekker and Davis 2018).

Once the label frequency is estimated, for each class, the number of positive labels in the corresponding supervised label set can be calculated, which can be regarded as the upper bound of the number of positive labels in the complemented label matrix during training. In LCForest, we introduce the complement flag vector $\boldsymbol{f} = [f_1, f_2, \ldots, f_n]^\top$ to control the training process. Here, $f_k = 0$ means that the k-th class is available for complementing labels; while $f_k = 1$ indicates that it is unavailable. At the beginning, all elements of the complement flag vector are initialized to 0, which means that it is permitted to complement labels for all classes. During training, for each class, once the number of positive labels reaches the estimated upper bound after one layer of cascade, its corresponding element in the complement flag vector will be set to 1, and labels of this class will not be changed from now on. The training process will be terminated as long as all elements in the are complement flag vector are 1. Label complement procedure under the supervision of the complement flag mechanism can be represented as Eq.(2).

$$Y_{ik}^{t+1} = \mathbb{I}(f_k = 0) \cdot \prod_{j=1}^{numF} Y_{ik}^{pred\_j} \vee Y_{ik} + \mathbb{I}(f_k = 1) \cdot Y_{ik}^{t} \tag{2}$$

As mentioned above, in weak-label learning, for each class, the amount of relevant labels is usually much smaller than irrelevant ones. For instance, we drop 40% of the relevant labels completely at random on benchmark multi-label learning data set *yeast*, which makes it a weak-label data set,

Algorithm 1 Label complement in each cascade layer

```
Inputs:
$Y$: the original label matrix
$Y^t$: the training label for the $t$-th layer
$Y^{prob}$: the predicted probability of the $t$-th layer
$f$: the complement flag
$\theta$: the threshold for splitting the predicted probability
Outputs:
$Y^{t+1}$: the training labels for the next layer
Process:
Discretize $Y^{prob}$ to $Y^{pred}$ with Eq.(3);
for $i = 1$ to $m$ do
    for $k = 1$ to $n$ do
        Set $Y_{ik}^{t+1}$ with Eq.(2);
    end for
end for
```

and there exists a class only associated with less than 2% relevant labels. In LCForest, threshold θ is used to split the predicted probability Y^{prob} from random forest classifiers and discretize it to Y^{pred} with Eq.(3). The threshold is set to be smaller than 0.5, which stimulates outputting more positive predictions. Alg. 1 summarizes the procedure of label complement with complement flag mechanism and thresholding function.

$$Y_{ik}^{pred\_j} = \begin{cases} 0 \ , & \text{if } Y_{ik}^{prob\_j} < \theta \\ 1 \ , & otherwise \end{cases} \tag{3}$$

Overall Framework of LCForest

Fig. 3 summarizes the overall procedure of the proposed LCForest method. Suppose that the original input is of 100 raw features, and has 5 class labels. The data will be used to train two completely random forests and two random forests in each layer. After processed by the first layer, for each training example, the raw feature vector is concatenated with 4×5=20 dimensional learned representations. The procedure will not be terminated until all elements in the complement flag vector are set to 1 or the number of cascade layer reaches the maximum layer T.

Given a test instance, it will go through the random forests in each layer of cascade, and then the final prediction will be obtained by aggregating the four 5-dimensional class vectors at the last layer, and taking the class with maximum aggregated value. The training procedure is shown in Alg. 2.

Experiments

In this section, we first introduce the experimental setup and then present the evaluation of our proposal compared to several state-of-the-art algorithms on a number of real-world tasks.

Experimental Setup

The performance of LCForest is compared against several state-of-the-art multi-label learning, deep neural network and weak-label learning algorithms, each configured with parameters fine-tuned for weak-label learning tasks:

Algorithm 2 Train LCForest

```
Inputs:
$D$: the weak-label training set $\{(\boldsymbol{x}_i, Y_i) | 1 \leq i \leq m\}$
$T$: the maximum layer
$K$: the number of folds in cross validation
$\theta$: the threshold for splitting the predicted probability
$conF$: the configuration of random forests in one layer
$conT$: the configuration of the TIcE method
Outputs:
$M$: the LCForest model trained with $D$
Process:
Initialize the LCForest model $M = \emptyset$;
Estimate the label frequency of each class via the TIcE
  method with configuration in $conT$;
Calculate the upper bound of positive labels of each
  class $\boldsymbol{u} = [u_1, u_2, \ldots, u_n]^{\mathsf{T}}$;
Initialize the complement flag $\boldsymbol{f}$;
for $t = 1$ to $T$ do
    for $k = 1$ to $n$ do
        if number of positive labels of the $k$-th class is no
          less than $u_k$ then
            Set $f_k$ to unavailable;
        end if
    end for
    if all elements in $\boldsymbol{f}$ are unavailable then
        Return the current LCForest model $M$;
    end if
    Conduct $K$-fold cross-validation and train random
      forests with configuration in $conF$ on training set
      $D^t = \{X^t, Y^t\}$;
    Add layer$_t$ to LCForest: $M = M \cup$layer$_t$;
    Concatenate the learned representation with $X^t$ and
      generate the training data $X^{t+1}$ for the next layer;
    Complete the label matrix and generate the training
      labels $Y^{t+1}$ for the next layer with Alg. 1;
end for
```

- ML-kNN (Zhang and Zhou 2007) first identifies the k-nearest-neighbour for an unseen instance, then classifies it based on statistical information gained from the label sets of these neighbouring instances [configuration: $k = 3$];
- RF-PCT (Kocev et al. 2013) builds ensemble models consisting of predictive clustering trees, which generalize classification trees both locally and globally [configuration: $n\_trees = 100$];
- DBPNN (Hinton and Salakhutdinov 2006) uses the Deep Belief Network (DBN) which consists of several stochastic layers with hidden variables where the upper layers can have symmetric connections [configuration: double-layer network with hidden layer size 20×20];
- WELL (Sun, Zhang, and Zhou 2010) employs the low-density assumption, and exploits the label correlation

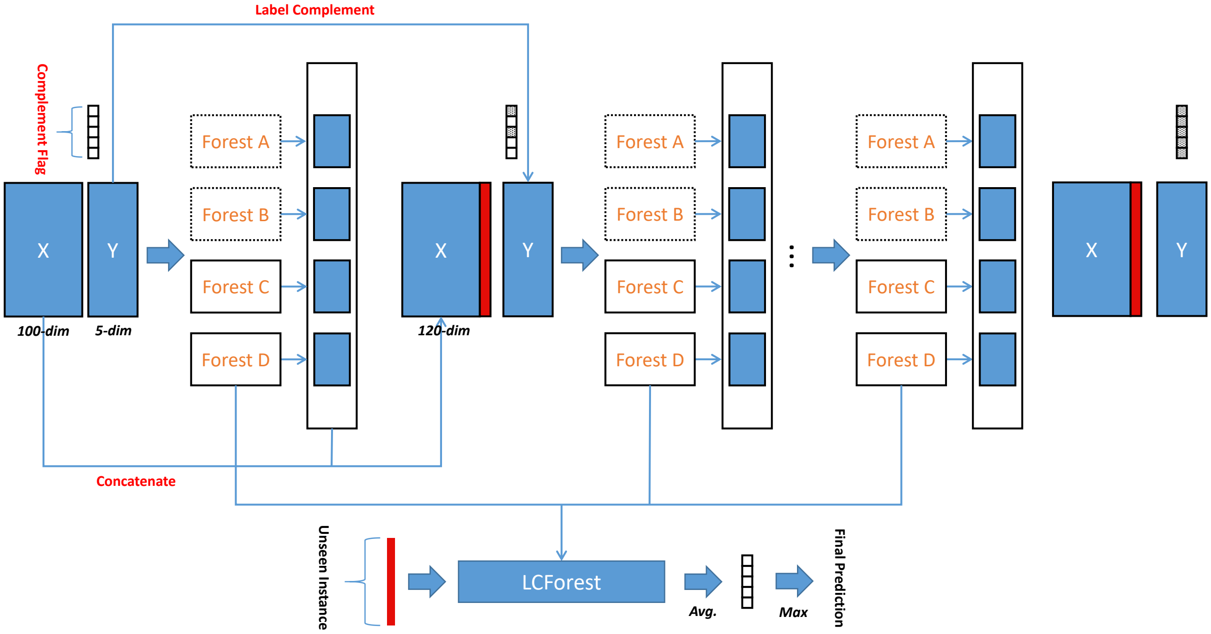

Figure 3: The overall Framework of LCForest.

based on the assumption that instance similarities are determined by a group of low-rank similarity matrices [configuration: $\alpha = 100, \beta = 10$].

We measure the classification results in terms of three multi-label evaluation criteria that are both instance-wise and label-wise effective (Wu and Zhou 2017), i.e., Micro-F1, Macro-F1 and Hamming Loss (H.L.). Hamming Loss evaluates the fraction of misclassified instance-label pairs; Macro-F1 and Micro-F1 which take both precision and recall into account. The larger the value of Micro-F1 and Macro-F1, the better the performance. For hamming loss, the smaller the value, the better the performance. More details about the evaluation metric please refer to (Zhang and Zhou 2014).

As shown in Alg. 2, hyper-parameters employed by LCForest are set as follows: $T = 10$, $K = 5$, $\theta = 0.4$. For configuration of random forests, we used one random forest and one completely random forest to encourage the diversity, and each random forest contains 200 decision trees. For configuration of the TIcE method, the max-bepp parameter $k = 5$, the maximum number of split is $M = 500$, and the minimum number of total examples in subset is $minT = 5$. We fixed all the hyper-parameters of our method in the experiments, since there exists no supervised validation set available for fine-tuning. As the comparing method WELL for weak-label learning is a transductive method, in our experiments, we first obtained the predicted complete labels for training examples by performing WELL, then classified the unseen instances by the ML-kNN method.

For each data set, we consider the incomplete label ratio (I.L. Ratio) by dropping $\{20\%, 30\%, 40\%, 50\%\}$ of the relevant labels on training data completely at random. We compared all methods using the same setting. In the rest of this section, we evaluated the performance by performing 5-fold cross-validation. The LCForest method as well as the comparing methods were trained with the training-fold from weak-label data set and then evaluated on the corresponding testing-fold from their completely supervised version.

Gene Function Analysis Task

The first task is to predict the gene function classes of the Yeast Saccharomyces cerevisiae, which is one of the best studied organisms. The *yeast* data set (Elisseeff and Weston 2001) is a gene function classification data set with 2417 examples and 14 class labels. Each gene is expressed with 103 microarray expression features. The average number of labels for each instance is 4.24.

Results are summarized in Table 1. It can be seen that LCForest obtains quite promising performance against the compared methods. It achieves the best performance on all subtasks except on Macro-F1 when the I.L. ratio is 20%. Although WELL outperforms LCForest on this subtask, it is too vulnerable to the change of I.L. ratio. When the I.L. ratio is larger than 40%, the Macro-F1 as well as Micro-F1 of WELL decrease rapidly. Also, the Hamming Loss of WELL stays high in the experiments of this task, while our method is quite robust to the change of I.L. ratio and different evaluation criteria.

Text Categorization Task

The second task is a text classification task collected from SIAM Text Mining Competition (TMC) 2007. Each document is an aviation safety report documenting one or more

Table 1: Experimental results (mean±std) on *yeast*. ↑ (↓) indicates the larger (smaller) the better. The best performance and its comparable performances are bolded.

	I.L. Ratio	LCForest	WELL	ML-kNN	RF-PCT	DBPNN
Hamming Loss(↓)	20%	**0.186±0.003**	0.623±0.004	0.216±0.006	0.214±0.003	0.211±0.008
	30%	**0.188±0.002**	0.627±0.002	0.229±0.004	0.241±0.004	0.230±0.004
	40%	**0.190±0.004**	0.622±0.006	0.251±0.003	0.276±0.003	0.265±0.008
	50%	**0.217±0.007**	0.628±0.003	0.274±0.006	0.295±0.003	0.288±0.005
Macro-F1(↑)	20%	0.407±0.015	**0.476±0.006**	0.353±0.008	0.241±0.011	0.290±0.010
	30%	**0.394±0.012**	0.384±0.003	0.304±0.008	0.169±0.011	0.236±0.005
	40%	**0.365±0.016**	0.076±0.007	0.220±0.006	0.083±0.006	0.139±0.016
	50%	**0.264±0.036**	0.012±0.003	0.147±0.015	0.029±0.005	0.059±0.009
Micro-F1(↑)	20%	**0.674±0.011**	0.591±0.005	0.561±0.019	0.530±0.006	0.563±0.017
	30%	**0.662±0.007**	0.435±0.003	0.495±0.014	0.398±0.012	0.465±0.015
	40%	**0.645±0.008**	0.094±0.007	0.372±0.011	0.176±0.008	0.260±0.027
	50%	**0.512±0.032**	0.015±0.004	0.234±0.032	0.052±0.006	0.101±0.019

Table 2: Experimental results (mean±std) on *TMC*. ↑ (↓) indicates the larger (smaller) the better. The best performance and its comparable performances are bolded.

	I.L. Ratio	LCForest	WELL	ML-kNN	RF-PCT	DBPNN
Hamming Loss(↓)	20%	**0.064±0.003**	0.088±0.003	0.089±0.002	0.087±0.003	0.081±0.003
	30%	**0.065±0.004**	0.090±0.001	0.091±0.001	0.092±0.002	0.084±0.003
	40%	**0.067±0.004**	0.096±0.001	0.092±0.002	0.097±0.002	0.088±0.002
	50%	**0.077±0.003**	0.100±0.001	0.095±0.001	0.099±0.002	0.089±0.003
Macro-F1(↑)	20%	**0.247±0.028**	0.166±0.040	0.153±0.015	0.076±0.010	0.240±0.015
	30%	**0.221±0.016**	0.076±0.014	0.127±0.014	0.041±0.005	0.194±0.014
	40%	**0.193±0.010**	0.039±0.009	0.092±0.007	0.018±0.008	0.153±0.014
	50%	**0.152±0.020**	0.017±0.007	0.064±0.007	0.003±0.003	0.123±0.018
Micro-F1(↑)	20%	**0.630±0.026**	0.225±0.054	0.335±0.020	0.270±0.014	0.482±0.019
	30%	**0.610±0.032**	0.106±0.020	0.255±0.023	0.147±0.012	0.417±0.022
	40%	**0.572±0.030**	0.054±0.013	0.201±0.010	0.042±0.006	0.344±0.014
	50%	**0.417±0.041**	0.024±0.010	0.102±0.015	0.003±0.002	0.298±0.020

Table 3: Experimental results (mean±std) on *scene*. ↑ (↓) indicates the larger (smaller) the better. The best performance and its comparable performances are bolded.

	I.L. Ratio	LCForest	WELL	ML-kNN	RF-PCT	DBPNN
Hamming Loss(↓)	20%	**0.101±0.030**	0.115±0.004	0.119±0.020	0.133±0.021	0.115±0.005
	30%	**0.100±0.025**	0.110±0.008	0.120±0.011	0.146±0.018	0.117±0.011
	40%	**0.107±0.033**	0.137±0.006	0.138±0.021	0.159±0.012	0.134±0.010
	50%	**0.117±0.037**	0.159±0.002	0.152±0.008	0.166±0.010	0.148±0.004
Macro-F1(↑)	20%	0.431±0.064	**0.581±0.026**	0.363±0.036	0.254±0.073	0.518±0.030
	30%	0.431±0.059	**0.460±0.047**	0.355±0.066	0.187±0.074	0.438±0.054
	40%	**0.386±0.081**	0.263±0.036	0.253±0.074	0.147±0.046	0.313±0.075
	50%	**0.318±0.120**	0.113±0.015	0.181±0.050	0.092±0.048	0.214±0.033
Micro-F1(↑)	20%	**0.690±0.101**	0.681±0.022	0.604±0.079	0.414±0.146	0.621±0.025
	30%	**0.688±0.079**	0.588±0.040	0.568±0.057	0.305±0.146	0.544±0.051
	40%	**0.634±0.037**	0.397±0.046	0.426±0.140	0.198±0.113	0.424±0.070
	50%	**0.529±0.209**	0.195±0.022	0.305±0.071	0.135±0.095	0.316±0.030

problems that occurred on certain flights. The goal is to label the documents with respect to what types of problems

Table 4: Experimental results (mean±std) on *medical*. ↑ (↓) indicates the larger (smaller) the better. The best performance and its comparable performances are bolded.

	I.L. Ratio	LCForest	WELL	ML-kNN	RF-PCT	DBPNN
Hamming Loss(↓)	20%	**0.013±0.002**	0.022±0.003	0.019±0.001	0.018±0.001	0.026±0.001
	30%	**0.013±0.002**	0.024±0.002	0.021±0.001	0.022±0.001	0.027±0.001
	40%	**0.019±0.005**	0.025±0.001	0.023±0.001	0.023±0.001	0.027±0.000
	50%	0.025±0.000	0.026±0.001	**0.024±0.001**	0.025±0.000	0.028±0.000
Macro-F1(↑)	20%	**0.194±0.013**	0.177±0.033	0.153±0.019	0.151±0.034	0.013±0.006
	30%	**0.189±0.007**	**0.189±0.043**	0.138±0.015	0.096±0.016	0.010±0.007
	40%	0.112±0.044	**0.147±0.015**	0.104±0.017	0.069±0.011	0.002±0.002
	50%	0.047±0.014	**0.093±0.024**	0.072±0.007	0.047±0.007	0.001±0.002
Micro-F1(↑)	20%	**0.742±0.053**	0.486±0.050	0.555±0.041	0.522±0.041	0.133±0.051
	30%	**0.735±0.027**	0.403±0.046	0.461±0.084	0.364±0.027	0.075±0.056
	40%	**0.460±0.218**	0.333±0.012	0.352±0.047	0.288±0.018	0.015±0.013
	50%	0.186±0.026	0.146±0.037	**0.262±0.043**	0.191±0.026	0.014±0.019

they describe. Each document may belong to more than one class. The *TMC* data set (Srivastava and Zane-Ulman 2005) is a large text data set with 28,596 instances and 22 class labels in total. Each document is expressed with 49060 features and has on average 3.57 labels. Here, we used its short version. We randomly sampled 2000 examples for evaluation and only the former 500 features were selected in the experiments.

Results are summarized in Table 2. We can observe that the proposed LCForest method significantly outperforms all the compared methods on all subtasks. It is also interesting to find that the Hamming Loss of LCForest when the I.L. ratio is 50% is better than the Hamming Loss of all the compared methods when the I.L. ratio is 20%, which demonstrates the effectiveness of our method.

Scene Classification Task

The third task is a multi-label semantic scene classification task. The *Scene* data set (Boutell et al. 2004) is a labeled image data set with 2407 images in 6 object classes. Each image is represented with spatial color moments in Luv space as features, which is commonly used in scene classification literature. After conversion to Luv space, the image is divided into 49 blocks using a 7×7 grid. Then, the first and second moments of each band are computed, corresponding to a low-resolution image and to computationally inexpensive texture features respectively and finally represented as a 294-dimension feature vector. The average number of labels for each instance is 1.07.

Results are summarized in Table 3. We can observe that although WELL performs better than LCForest on Macro-F1 when the I.L. ratio is less than 30%, LCForest remains promising performance when the I.L. ratio is larger. And on Hamming Loss and Micro-F1, LCForest always achieves the best performance.

Medical Natural Language Processing Task

The last task is a medical natural language processing task. The *Medical* data set (Read et al. 2011) contains 978 instances and 1449 features. It is used in the Medical Natural Language Processing Challenge3 in 2007, whose instance is a document that contains a brief free-text summary of a patient symptom history. It has 45 labels in total. The goal is to annotate each document with the probable diseases from the International Classification of Diseases (ICD-9-CM). Results are summarized in Table 4.

Conclusion

In this paper, a tree-based method LCForest is proposed to solve weak-label learning problems. Rather than formulating the problem as a regularized framework, we employ the recently proposed cascade forest structure, which processes information layer-by-layer, and endow it with the ability of exploiting the weak-label data by a concise and highly efficient label complement structure. We demonstrate that our method solved the three main difficulties of weak-label learning well and can be adapted to learn from image or sequence data, which can not achieve directly by other methods. Extensive comparative studies clearly validate the effectiveness of LCForest. In the future, we consider adapting our method to learn from image data and sequential data by multi-grained scanning, and consider exploiting from other kinds of weakly supervised data with deep forest methods.

References

Bekker, J., and Davis, J. 2018. Estimating the class prior in positive and unlabeled data through decision tree induction. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

Boutell, M. R.; Luo, J.; Shen, X.; and Brown, C. M. 2004. Learning multi-label scene classification. *Pattern recognition* 37(9):1757–1771.

Bucak, S. S.; Jin, R.; and Jain, A. K. 2011. Multi-label learning with incomplete class assignments. In *Proceedings of 24th IEEE Conference on Computer Vision and Pattern Recognition*, 2801–2808.

Dong, H.-C.; Li, Y.-F.; and Zhou, Z.-H. 2018. Learning from semi-supervised weak-label data. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2926–2933.

Elisseeff, A., and Weston, J. 2001. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems 14*, 681–687.

Elkan, C., and Noto, K. 2008. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 213–220.

Feng, J., and Zhou, Z.-H. 2018. Autoencoder by forest. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 2967–2973.

Guo, Y.; Liu, S.; Li, Z.; and Shang, X. 2018. Bcdforest: a boosting cascade deep forest mdoel towards the classification of cancer subtypes based on gene expression data. *BMC bioinformatics* 19(5):118.

Hinton, G. E., and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *science* 313(5786):504–507.

Kocev, D.; Vens, C.; Struyf, J.; and Džeroski, S. 2013. Tree ensembles for predicting structured outputs. *Pattern Recognition* 46(3):817–833.

Kong, X.; Ng, M. K.; and Zhou, Z.-H. 2013. Transductive multilabel learning via label set propagation. *IEEE Transactions on Knowledge and Data Engineering* 25(3):704–719.

Li, Y.-F., and Liang, D.-M. 2019. Safe semi-supervised learning: A brief introduction. *Frontiers of Computer Science* 13(4):669–676.

Li, X., and Liu, B. 2003. Learning to classify texts using positive and unlabeled data. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, volume 3, 587–592.

Li, Y.-F.; Guo, L.-Z.; and Zhou, Z.-H. 2019. Towards safe weakly supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Lin, Z.; Ding, G.; Hu, M.; Wang, J.; and Ye, X. 2013. Image tag completion via image-specific and tag-specific linear sparse reconstructions. In *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition*, 1618–1625.

Liu, Y.; Jin, R.; and Yang, L. 2006. Semi-supervised multilabel learning by constrained non-negative matrix factorization. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 421–426.

Lyu, S.-H.; Yang, L.; and Zhou, Z.-H. 2019. A refined margin distribution analysis for forest representation learning. In *Advances in Neural Information Processing Systems 32*, 5531–5541.

Pang, M.; Ting, K.-M.; Zhao, P.; and Zhou, Z.-H. 2018. Improving deep forest by confidence screening. In *The IEEE International Conference on Data Mining*, 1194–1199.

Read, J.; Pfahringer, B.; Holmes, G.; and Frank, E. 2011. Classifier chains for multi-label classification. *Machine learning* 85(3):333.

Srivastava, A. N., and Zane-Ulman, B. 2005. Discovering recurring anomalies in text reports regarding complex space systems. In *IEEE Aerospace Conference*, 3853–3862.

Sun, Y.-Y.; Zhang, Y.; and Zhou, Z.-H. 2010. Multi-label learning with weak label. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 593–598.

Utkin, L. V., and Ryabinin, M. A. 2018. A siamese deep forest. *Knowledge-Based Systems* 139:13–22.

Wu, X.-Z., and Zhou, Z.-H. 2017. A unified view of multilabel performance measures. In *Proceedings of the 34th International Conference on Machine Learning*, 3780–3788.

Wu, L.; Jin, R.; and Jain, A. K. 2013. Tag completion for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(3):716–727.

Xie, M.-K., and Huang, S.-J. 2018. Partial multi-label learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Xu, M.; Jin, R.; and Zhou, Z.-H. 2013. Speedup matrix completion with side information: Application to multi-label learning. In *Advances in Neural Information Processing Systems 26*, 2301–2309.

Yang, L.; Wu, X.-Z.; Jiang, Y.; and Zhou, Z.-H. 2019. Multilabel learning with deep forest. *CoRR* abs/1911.06557.

Zhan, W., and Zhang, M.-L. 2017. Inductive semisupervised multi-label learning with co-training. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1305–1314.

Zhang, M.-L., and Zhou, Z.-H. 2007. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition* 40(7):2038–2048.

Zhang, M.-L., and Zhou, Z.-H. 2014. A review on multilabel learning algorithms. *IEEE Transactions on Knowledge and Data Engineering* 26(8):1819–1837.

Zhao, F., and Guo, Y. 2015. Semi-supervised multi-label learning with incomplete labels. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 4062–4068.

Zhou, Z.-H., and Feng, J. 2019. Deep forest. *National Science Review* 6(1):7486.

Zhou, Z.-H., and Zhang, M.-L. 2006. Inductive semisupervised multi-label learning with co-training. In *Proceedings of the 19th Internation Conference on Neural Information Processing Systems*, 1609–1616.

Zhou, Z.-H.; Zhang, M.-L.; Huang, S.-J.; and Li, Y.-F. 2012. Multi-instance multi-label learning. *Artificial Intelligence* 176(1):2291–2320.

Zhu, G.; Yan, S.; and Ma, Y. 2010. Image tag refinement towards low-rank, content-tag prior and error sparsity. In *Proceedings of the 18th ACM International Conference on Multimedia*, 461–470.