Iterative Privileged Learning

Xue Li, Bo Du, *Senior Member, IEEE,* Yipeng Zhang, Chang Xu, and Dacheng Tao, *Fellow, IEEE*

***Abstract*—While privileged information may not be as informative as example features in the context of making accurate label predictions, it may be able to provide some comments on the efficacy of the learned model. In a departure from conventional static manipulations of privileged information within the support vector machine (SVM) framework, this paper investigates iterative privileged learning (IPL) within the context of gradient boosted decision trees. As the learned model evolves, the comments learned from privileged information to assess the model should also be actively upgraded instead of remaining static and passive. During the learning phase of the gradient boosted decision tree method, we discover new decision trees to enhance the capability of the model, and iteratively update the comments generated from the privileged information to accurately assess and coach the up-to-date model. The resulting objective function can be efficiently solved within the gradient boosting framework. Experimental results on real-world datasets demonstrate the benefits of studying privileged information in an iterative manner, as well as the effectiveness of the proposed algorithm.**

***Index Terms*—Learning using privileged information, gradient boosted trees.**

I. Introduction

TRADITIONAL supervised classification methods were once developed over a set of data points, each of which consists of an example feature vector and its corresponding label. These methods differ in the techniques used to approximate the underlying mapping from the feature space to the label space. In addition to the feature vectors and labels, it is often practical in some applications to collect some auxiliary information in order to assist the mapping approximation. For example, in an object recognition task, we can manually set bounding boxes to filter out the objects of interest in images, which may decrease the influence of messy backgrounds and improve recognition accuracy. This auxiliary information is often regarded as privileged information, since it only exists in the training stage but cannot easily be applied in the test phase.

An increasing number of studies have demonstrated that privileged information of this kind can help to improve the performance of learning models in various applications. In digit classification tasks, a poetic description of each training image is additionally supplied to further boost the classification performance [1]. In real-world applications, visual data for scene recognition are usually down-sampled as low-resolution images to save on storage, but superior performance can be obtained with the aid of high-resolution images in the training phase [2]. In addition, some facial attributes associated with aging, such as skin smoothness, face shape, face acne, wrinkles and under-eye bags, have been used as privileged information in solving age estimation problems [3].

The concept of learning using privileged information (LUPI) was first introduced by Vapnik and Vashist [1], [4]. As in human learning, where teachers provide additional sources of explanation along with training examples during the learning process, LUPI also investigates training data with additional information (referred to as privileged information) that is only available at the training stage and not available at the test stage. Beginning with the successful SVM+ algorithm which was adapted from the support vector machine (SVM) framework [1], the LUPI paradigm has attracted increasing interest in the community [5]–[11]. SVM+ replaces the original slack variables in the standard SVM by means of a linear auxiliary function of privileged information. In so doing, the mistakes made by the classifier can be assessed with the help of privileged information, which will be beneficial for coaching the learning of an optimal classifier. Since its inception, many variants of SVM+ have been further explored by researchers. For example, the relative attribute support vector machine (raSVM+) algorithm takes relative attributes as privileged information in order to improve the accuracy of age estimation by controlling outliers and guiding the learned predictor over the training data [3]. SVM+ has also been extended to the multi-instance learning scenario, solving object recognition and image retrieval problem by exploiting privileged information derived from web data [6]. In addition, privileged information has been incorporated into structured SVM learning framework to obtain better generalization performance [12]. There have also been some studies on optimization techniques to solve linear SVM+ and kernel SVM+, such as gSMO, CVX-SVM+, MAT-SVM+ and ℓ_2-loss SVM+ [2].

However, while these methods have achieved satisfactory performance by taking advantage of the LUPI paradigm, we must also ask whether it is possible to investigate privileged information beyond the SVM framework. Most importantly, it is instructive to note that practical human learning is not an one-off intervention, but an iterative process. Students enhance their capabilities with the help of comments from teachers, at the same time teachers' comments need to be updated regularly to accurately reflect and address students' current capabilities. Hence, rather than exploiting privileged information in a "static" manner by means of a fixed auxiliary function, the comments generated from privileged information should be updated constantly so that they remain consistent with the most up-to-date model.

X. Li is with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan 430079, China (e-mail: lixue93@whu.edu.cn).

B. Du is with the State Key Laboratory of Software Engineering, School of Computer, Wuhan University, Wuhan 430079, China (e-mail: remoteking@whu.edu.cn).

Y. Zhang is with the Department of Electrical Engineering and Computer Science, Syracuse University, USA (e-mail: yzhan139@syr.edu).

C. Xu and D. Tao are with the UBTech Sydney Artificial Intelligence Centre and the School of Information Technologies, Faculty of Engineering and Information Technologies, The University of Sydney, Sydney, NSW 2008, Australia (e-mail: c.xu@sydney.edu.au; dacheng.tao@sydney.edu.au).

Privileged information has rarely been investigated using this developmental approach. The most relevant work here may be the GB+ method [13], which iteratively encourages the consistent predictions of two decision trees constructed with example features and privileged features, respectively. However, the privileged information is often not informative as the example features to make accurate label predictions in practice, which makes the consistence constraint unfeasible. Therefore, rather than treating privileged information as equivalent to the example features when learning the prediction model, we tend to regard the privileged information as auxiliary data to generate comments which makes some corrections on the learned model.

In this paper, we propose a new method called iterative privileged learning (IPL) within the context of gradient boosted decision trees (GBDT). In each iteration of GBDT, a weak learner (i.e., a new tree) is trained to upgrade the existing model, and a corresponding additional linear auxiliary function is employed to generate comments (e.g., the values of the auxiliary function) from the privileged information on this weak learner. These comments depict the discrepancy (i.e. residuals) between predictions and ground-truth values, and will coach the learning process of a new decision tree. Moreover, we integrate the learning of the new decision tree and the auxiliary function into a unified objective function. In doing so, we closely connect the upgrading of the prediction model with the exploitation of the privileged information. As the comments from the privileged information on each weak learner are updated iteratively, the model can always be accurately assessed and its capability gradually enhanced as more and more single weak learners (i.e., decision trees) are boosted into the model. Considering that in the current iteration, the new learned projection vector of the additional linear auxiliary function may have relations with projection vectors learned in the previous iterations, we further introduce a variant of our IPL method referred as sparse IPL. Experimental results on real-world datasets demonstrate the effectiveness of the proposed iterative privileged learning algorithms by utilizing the privileged information.

The rest of this paper is arranged as the following. The related works is briefly reviewed in Section II. Section III presents the baseline gradient boosted decision tree (GBDT) method. Section IV describes the proposed IPL method and its variant named sparse IPL, and their solution are provided. In section V, the effectiveness of the proposed method is evaluated compared with several approaches on several real-world datasets. Section VI concludes this paper.

II. Related Works

Inspired by the human learning and teaching process, the learning using privileged information (LUPI) framework has attracted increasing interest in the machine learning field since it was proposed by Vapnik and Vashist [1]. Unlike the classical machine learning paradigm, which solely employs example features and their corresponding ground-truth labels in model learning, LUPI also considers comments from teachers. That is, in the LUPI paradigm, models are not only trained using labeled examples, but also can access teachers' comments, explanations, comparisons and so on. Although this help from teachers is only available in the training stage, the superiority of this advanced learning paradigm has been demonstrated in diverse problems [1].

The first approach proposed in the LUPI paradigm, which is based on the support vector machine (SVM) framework, is called SVM+ [1]. The basic assumption of SVM+ is that the misclassification loss of each training example can be measured using a correcting function derived from privileged information. Hence, a classifier is learned contemporaneously with the correcting function. SVM+-based research has attracted extensive attention and many variants of SVM+ have been proposed and implemented for various applications, such as L1-regularized SVM+ [14], multi-class SVM+ [15], multi-task multi-class SVM+ [16], structural SVM+ [12], multi-label SVM+ [17], SVM+ for domain adaptation [6], and the rank transfer method [5], and so on. By introducing L1-norm SVM into the LUPI paradigm, less time will be spent in tuning model parameters and feature selection can be realized in the training stage, unlike the standard L2-norm SVM+ [14]. It is demonstrated that SVM+ is closely associated with the well-known weighted SVM, and that the privileged information from SVM+ can be encoded through instance weights [18].

In addition to the ordinary classification task, the LUPI paradigm has been applied to a number of different problems. One example [19] involved generalizing the classical metric learning methods using a fixed threshold in the generic empirical risk framework by building a locally adaptive decision rule with privileged information. Inspired by knowledge transfer, the standard hash learning method has also been extended in a transfer learning scenario [20]. The quantization error was approximated using a function learned from auxiliary data, and the geometry structure of the auxiliary data was explored to obtain more accurate binary codes in the target domain.

The present work is also related to the decision tree (DT) methods. DT is a non-parametric supervised learning method used for classification and regression. By building simple decision rules learned from the data features, DT attempts to create a model that predicts the value of a target variable. In this paper, we focus on a specific DT method called the classification and regression trees (CART) method [21]. The CART model, which has been one of the most popular decision tree methods since it was first introduced, utilizes tree-building algorithms that employ a set of if-then conditions for split to make prediction or classification. In more detail, CART starts by analyzing all expositive variables and then determines how best to make a binary division of a single expositive variable in order to reduce deviance in the response variable. The split process is continued and repeated for each portion of the data resulting from the previous split, until homogeneous end points or terminal nodes are arrived at in a hierarchical tree.

Subsequently, a tree-based ensemble method called gradient boosted decision trees (GBDT), which was developed in order to create more powerful prediction models based on decision trees, has achieved widespread use in real-world applications [22]. GBDT [23] is a generalization of gradient boosting [24], [25] that builds additive models of multiple decision trees [26],

[27] and has been successfully applied in the machine learning field [28], [29]. Unlike the random forest method [30], which combines a forest of randomly different trees in parallel, the main purpose of GBDT is to build a series of trees. Typically, GBDT uses many small trees with shallow depth, which are known as 'weak learners' in machine learning. When training each tree, the method tries to correct the mistakes of the previous tree in the series, by sequentially fitting a simple parameterized function to the current target residuals based on the least square. As more and more decision trees are learned and added, the created tree model will make fewer and fewer mistakes.

III. Gradient Boosted Decision Tree

In this section, we briefly introduce the gradient boosted decision tree (GBDT) method. GBDT aims to improve the performance of a single decision tree by fitting a series of decision trees and combining them in order to make predictions. Instead of learning many large trees with a high variance and high depth, GBDT learns and adds small trees with a low depth, combining these weak single decision trees into a strong decision model in an iterative fashion.

Let $G_t(\cdot)$ be the integrated decision function to be maintained, while $G_t(x_i)$ is the prediction of the example x_i in the t-th iteration. Given y_i as the ground-truth label of x_i, the goal of GBDT is to discover the function $G_t(\cdot)$ that can approximately estimate y_i with $\tilde{y}_i = G_t(x_i)$. Mean squared error can be employed to measure the discrepancy between the predicted label $\tilde{y}_i$ and the ground-truth label y_i, and the optimal $G_t(\cdot)$ can be solved by minimizing:

$$Loss = \frac{1}{2}\sum_{i=1}^{n}(G_t(x_i) - y_i)^2. \tag{1}$$

In the t-th iteration, we plan to improve the imperfect model $G_{t-1}(\cdot)$ from the previous iteration by also considering a small decision tree $h_t(\cdot)$, such that $G_t(\cdot)$ can be upgraded with $G_t(x_i) \leftarrow G_{t-1}(x_i) + h_t(x_i)$. The gradient boosting technique considers gradient descent in the example space $X = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{d\times n}$, and the current prediction $G_t(x_i)$ for the example x_i can be adjusted using a gradient step as follows:

$$G_t(x_i) \leftarrow G_{t-1}(x_i) - \beta\frac{\partial Loss}{\partial G_{t-1}(x_i)}, \tag{2}$$

where $\beta > 0$ is referred to as the learning rate, and the negative gradient $-\frac{\partial Loss}{\partial G_{t-1}(x_i)}$ is equal to the residual $r_i = y_i - G_{t-1}(x_i)$ from the previous iteration. We can thus obtain the new decision tree $h_t(\cdot)$ by minimizing the mean squared error, as follows:

$$\min_{h_t}\sum_{i=1}^{n}(h_t(x_i) - r_i)^2. \tag{3}$$

Eq. (3) can be solved using the standard CART algorithm with $\{(x_1, r_1), \ldots, (x_n, r_n)\}$ as the input training data. Two hyper-parameters are employed in the construction of the decision tree, i.e., tree depth k, and the number of features to consider when looking for the best split s. We set tree depth $k \leq 5$ and consider all features for each split according to the suggestions outlined in [22]. There are also two important parameters required to launch gradient boosting, including the maximal number of iterations (i.e., number of boosted trees) N and the learning rate $\beta > 0$. Before the $(t+1)$-th iteration, the residual r_i and the current model $G_t(\cdot)$ are updated using $r_i \leftarrow r_i - \beta h_t(x_i)$ and $G_t(\cdot) = \beta\sum_{j=1}^{t} h_j(\cdot)$, respectively. Finally, when either the maximal number of iterations is reached or the objective value in Eq. (1) becomes stable, the algorithm converges and outputs the final model: $G(\cdot) = \beta\sum_{t=1}^{N} h_t(\cdot)$.

IV. The Proposed Approach

In this section, we first illustrate the theory of our proposed iterative privileged learning (IPL) method. Then, we introduce a variant of our IPL method called sparse IPL method, which learns the projection of the auxiliary function of IPL based on sparse representation.

A. Iterative Privileged Learning

For the GBDT method, in the t-th iteration, the current weak learner $h_t(x_i)$ is learned by satisfying $G_{t-1}(x_i) + h_t(x_i) = y_i$ for each example x_i. That is, the new decision tree (the weak learner) with the training data in order to minimize the difference between the prediction values and the ground-truth labels. However, consider the case where the target value cannot be reached without error, we introduce some auxiliary variables ξ_i to estimate and depict the error as $G_{t-1}(x_i) + h_t(x_i) - y_i = \xi_i$.

As use of the auxiliary data has been demonstrated to assist in constructing a better predictive rule [1], we further investigate GBDT in the LUPI paradigm. Under the LUPI paradigm, we suppose that during training process Ideal Teacher can provide Student with the values of auxiliary variables as privileged information. Thus, triplets are supplied to Student

$$(x_1, \xi_1^0, y_1), (x_2, \xi_2^0, y_2), \ldots, (x_n, \xi_n^0, y_n),$$

where $\xi_i^0, i = 1, \ldots, n$ are the auxiliary variables. In order to learn the decision tree in the t-th iteration, the functional is minimized

$$\sum_{i=1}^{n} L(G_{t-1}(x_i) + h_t(x_i) - y_i - \xi_i). \tag{4}$$

where L is a general loss function which is set to be quadratic in this paper. Define $r_i = y_i - G_{t-1}(x_i)$ and we minimize the functional

$$\sum_{i=1}^{n}(h_t(x_i) - r_i - \xi_i)^2. \tag{5}$$

Noticeably, real Intelligent Teacher actually cannot provide auxiliary values which Teacher does not know. However, they can do something else instead, i.e., defining a space X^* where a set of real-valued auxiliary functions $f(x_i^*)$ to approximate belongs, and generating privileged information for examples in the training set to supply Student with triplets,

$$(x_1, x_1^*, y_1), (x_2, x_2^*, y_2), \ldots, (x_n, x_n^*, y_n),$$

where $x_i \in \mathbb{R}^d$ and $x_i^* \in \mathbb{R}^{d^*}$ are the example feature vector and corresponding privileged feature vector of the i-th training example, and y_i is its ground-truth label. $X =$

$[x_1, x_2, \dots, x_n] \in \mathbb{R}^{d \times n}$ and $X^* = [x_1^*, x_2^*, \dots, x_n^*] \in \mathbb{R}^{d^* \times n}$ are the example feature (EF) training data matrix and privileged feature (PF) training data matrix, respectively. n is the number of training examples in the training set, and d and d^* are the dimensions of the example features and privileged features, respectively. Note that PF data x^* is available only in the training stage, not in the test stage.

More specifically, we employ auxiliary functions which can generate comments from the privileged information to measure the discrepancy (i.e., residuals) between the predictions and ground-truth labels. In this way, the current remaining prediction error of the boosted decision tree model can be more accurately depicted, which will be beneficial for learning the next new decision tree. Consequently, the primary question is now how the auxiliary function might be constructed in order to generate the comments from the privileged information. In particular, we consider a linear auxiliary function defined in the PF space, and some nonlinear function could be straightforwardly applied as well. For the privileged feature x_i^* of the i-th example, the comment generated from x_i^* is supposed to be $f(x_i^*) = wx_i^*$, where $w \in \mathbb{R}^{1 \times d^*}$ is the projection vector. Hence, during the process of learning a new decision tree, the prospective target is not only affected by the difference between the predictions of the current model and the ground-truth labels, but also influenced by the comments from the privileged information.

The next problem to be solved concerns how the most suitable auxiliary function might be selected; in other words, how the best projection w might be found? Our solution is that w is not fixed. In fact, it is iteratively learned and updated each time a new tree is added and the model is updated. That is to say, the comments provided by the privileged information are also iteratively updated as the model grows stronger. This idea, which resembles the teaching process in a human classroom (in that the comments provided by teachers are changed and updated as students become more knowledgeable), is ignored by almost all existing methods in the LUPI paradigm. Therefore, our iterative privileged learning (IPL) aims to iteratively learn, add a new tree and update the comments of the privileged information on the newly derived decision model. Accordingly, in the t-th iteration, rather than generating the new decision tree $h_t(\cdot)$ as GBDT in Eq. (3), we instead consider minimizing the following loss function with the help of the privileged information:

$$\min_{h_t, w_t} \sum_{i=1}^{n} (h_t(x_i) - r_i)^2 + C_1 \sum_{i=1}^{n} (h_t(x_i) - r_i - w_t x_i^*)^2. \quad (6)$$

The second term depicts the difficulty in quantizing the prediction error of the current model. Moreover, it also considers the current auxiliary function $f_t(x_i^*) = w_t x_i^*$ as the tolerance function to allow the little prediction error to a certain extent. In this way, the prediction error can be regularized to avoid over-fitting. And $C_1 > 0$ is a parameter to control the influence of the privileged information on both the residual target and the learning of the decision tree. In this way, the comments provided by the auxiliary function $f_t(x_i^*)$ can be balanced to better correct the residual target.

Algorithm 1 Iterative Privileged Learning (IPL)

```
Input: Dataset $\{(x_1, x_1^*, y_1), \dots, (x_n, x_n^*, y_n)\}$, parameters including tree depth $k$, the number of trees $N$, learning rate $\beta$, tradeoff parameters $C_1$ and $C_2$
Initialization: $t = 1$, $w_t \in \mathbb{R}^{1 \times d^*}$ is randomly initialized in range (0,1), $r_i = y_i$, where $i$ ranges from 1 to $n$
for $t = 1$ to $N$ do
    repeat
        Update $h_t(\cdot)$ based on Eq. (10):
        $r_i^* = r_i + \frac{C_1}{C_1+1} w_t x_i^*$, where $i$ ranges from 1 to $n$
        $h_t(\cdot) \leftarrow CART((x_1, r_1^*), \dots, (x_n, r_n^*), k)$
        Update $w_t$ based on Eq. (12):
        $w_t = \frac{C_1}{C_1+C_2} A X^{*\mathrm{T}} (X^* X^{*\mathrm{T}})^{-1}$
    until convergence or maximum iteration is reached
    Update $w_{t+1} \leftarrow w_t$, $r_i \leftarrow r_i - \beta h_t(x_i)$, where $i$ ranges from 1 to $n$
end for
Output: Final IPL model $G(\cdot) = \beta \Sigma_{t=1}^{N} h_t(\cdot)$
```

It should be noted that, in the LUPI paradigm, privileged information effectively serves as auxiliary information used to improve the decision tree model's predictions on the example feature data; the values of the auxiliary function can be either positive or negative. In order to prevent its magnitude from growing too large, the value of the auxiliary function $f_t(x_i^*)$ has to be constrained. Therefore, we further introduce an L2 regularization to assist in this matter. The final objective of the proposed IPL method is as follows:

$$\begin{aligned} \min_{h_t, w_t} & \sum_{i=1}^{n} (h_t(x_i) - r_i)^2 + C_1 \sum_{i=1}^{n} (h_t(x_i) - r_i - w_t x_i^*)^2 \\ & + C_2 \sum_{i=1}^{n} \|w_t x_i^*\|_2^2, \end{aligned} \quad (7)$$

where $C_2 > 0$ is a tradeoff parameter used to balance the influence of the third term on the whole minimization problem.

At the t-th iteration, the best h_t and w_t are learned by solving the optimization problem (7), while the new tree $h_t(\cdot)$ is added to the current model $G_t(\cdot) = \beta \sum_{j=1}^{t} h_j(\cdot)$. Moreover, the learned w_t in the current iteration serves to initialize the projection vector w_{t+1} of the auxiliary function in the next iteration. The residual target $r_i \leftarrow r_i - \beta h_t(x_i)$ for each example is then updated for the next iteration.

B. Sparse IPL

In the proposed IPL method, the projection w is updated at each iteration. Noticeably, since the projection w_t in the t-th iteration may have some relations with the previous learned projections $\{w_i\}, i = 1, \dots, t-1$, we consider to represent w with a linear combination of $\{w_i\}, i = 1, \dots, t-1$. We assume after η iterations, there are efficient learned projections $\{w_j\}, j = 1, 2, \dots \eta$ that can construct a dictionary $D = [w_1^T, w_2^T, \dots, w_\eta^T] \in \mathbb{R}^{d^* \times \eta}$, where η represents the number of atoms. Thus, in the m iteration ($m > \eta$), we can represent the w_m as $w_m = Dv_m$. Moreover, due to the similarity of the projections $\{w_j\}$, v_m can be sparse. In order to reduce the cost

Algorithm 2 Sparse IPL

```
Input: Dataset $\{(x_1, x_1^*, y_1), \dots, (x_n, x_n^*, y_n)\}$, parameters including tree depth $k$, the number of trees $N$, learning rate $\beta$, the number of atoms $\eta$, tradeoff parameters $C_1$ and $C_2$
Initialization: $t = 1$, $w_t \in \mathbb{R}^{1\times d^*}$ is randomly initialized in range (0,1), $r_i = y_i$, where $i$ ranges from 1 to $n$
for $t = 1$ to $N$ do
    repeat
        if $t \leq \eta$ then
            Update $h_t(\cdot)$ based on Eq. (10):
            $r_i^* = r_i + \frac{C_1}{C_1+1} w_t x_i^*$, where $i$ ranges from 1 to $n$
            $h_t(\cdot) \leftarrow CART((x_1, r_1^*), \dots, (x_n, r_n^*), k)$
            Update $w_t$ based on Eq. (12):
            $w_t = \frac{C_1}{C_1+C_2} A X^{*\mathrm{T}} (X^* X^{*\mathrm{T}})^{-1}$
        else
            Update $h_t(\cdot)$ based on Eq. (14):
            $r_i^* = r_i + \frac{C_1}{C_1+1} x_i^{*\mathrm{T}} D v_t$, where $D = [w_1^T, w_2^T, \dots, w_\eta^T] \in \mathbb{R}^{d^* \times \eta}$ and $i$ ranges from 1 to $n$
            $h_t(\cdot) \leftarrow CART((x_1, r_1^*), \dots, (x_n, r_n^*), k)$
            Update $v_t$ based on Eq. (17) using the coordinate descent in Scikit-learn [31]
        end if
    until convergence or maximum iteration is reached
    Update $w_{t+1} \leftarrow w_t$, $r_i \leftarrow r_i - \beta h_t(x_i)$, where $i$ ranges from 1 to $n$
end for
Output: Final IPL model $G(\cdot) = \beta \Sigma_{t=1}^N h_t(\cdot)$
```

of time and memory of the IPL method, we further introduce sparse representation to learn v_m. And the optimal v_m can be obtained by solving the new sparse IPL problem which is formulated as:

$$\min_{h_m, v_m} \sum_{i=1}^{n} (h_m(x_i) - r_i)^2 + C_1 \sum_{i=1}^{n} (h_m(x_i) - r_i - x_i^{*\mathrm{T}} D v_m)^2 + C_2 \sum_{i=1}^{n} \|v_m\|_1, \tag{8}$$

where v_m is a η-dimensional sparse vector (i.e., v_m has only a few nonzero entries), whose entries are the weights of the corresponding atoms in D; $C_1 > 0$ are tradeoff parameters used to balance the influence of the second term on the whole minimization problem. $C_2 > 0$ is a parameter to balance the effect of the $L1$ norm sparse term on the objective function.

C. Solution for IPL

From the viewpoint of optimization, both learning steps in the IPL method with respect to the new tree $h_t(\cdot)$ and the projection vector w_t are convex optimization problems. Therefore, we adopt an alternating optimization strategy to efficiently solve the objective problem in Eq. (7), i.e., the new tree $h_t(\cdot)$ and the projection vector w_t are alternatively updated until the convergence is reached. The convergence is shown in Fig. 8 in the experimental part. After almost 5 iterations, an optimum can be achieved in the experiments. More specifically, we first fix the variable w_t and solve for the new tree $h_t(\cdot)$, and then the current learned new tree $h_t(\cdot)$ is fixed and the variable w_t is updated.

1) Optimization of $h_t(\cdot)$: If $t = 1$, we initialize w_t with random values between 0 and 1. Otherwise, w_t is set as the same value of w_{t-1} obtained for the last model. The next decision tree $h_t(\cdot)$ can be obtained by solving the following optimization problem:

$$\min_{h_t} \sum_{i=1}^{n} (h_t(x_i) - r_i)^2 + C_1 \sum_{i=1}^{n} (h_t(x_i) - r_i - w_t x_i^*)^2. \tag{9}$$

By means of some simple mathematical operations, Eq. (9) can be further reformulated as follows:

$$\min_{h_t} \sum_{i=1}^{n} [h_t(x_i) - (r_i + \frac{C_1}{C_1+1} w_t x_i^*)]^2. \tag{10}$$

Eq. (10) can now be solved efficiently by the standard classification and regression trees (CART) algorithm [21]. Here, $r_i^* = r_i + \frac{C_1}{C_1+1} w_t x_i^*$ plays the same role as r_i in Eq. (3). However, it is instructive to note that the privileged feature x_i^* has been included in order to adjust the target value r_i^*. Given the input features $\{x_1, \dots, x_n\}$ and target labels $\{r_1^*, \dots, r_n^*\}$, classical CART solvers can be straightforwardly applied. There are two hyper-parameters to be determined before launching CART: the maximum depth of the single tree k, and the number of features to consider when looking for the best split at each split s. We let s be equal to the number of all features by default [22]. The tree and its predictor value of input data $\{x_1, \dots, x_n\}$ can then be solved based on CART algorithm, as $h_t(\cdot) \leftarrow Cart(\{(x_1, r_1^*), \dots, (x_n, r_n^*)\}, k)$.

2) Optimization of w_t: After obtaining the new tree $h_t(\cdot)$, we fix it and update w_t by solving the following optimization problem:

$$\min_{w_t} C_1 \sum_{i=1}^{n} (h_t(x_i) - r_i - w_t x_i^*)^2 + C_2 \sum_{i=1}^{n} \|w_t x_i^*\|_2^2. \tag{11}$$

In order to simplify the expression of the problem (11), we transform it into the matrix optimization problem. We let $H = [h_t(x_1), \dots, h_t(x_n)] \in \mathbb{R}^{1\times n}$, $R = [r_1, \dots, r_n] \in \mathbb{R}^{1\times n}$, and $A = H - R \in \mathbb{R}^{1\times n}$, such that Eq. (11) can be rewritten as:

$$\min_{w_t} C_1 \|A - w_t X^*\|_2^2 + C_2 \|w_t X^*\|_2^2. \tag{12}$$

We solve the optimal w_t by setting the deviation with respect to w_t to zero. In this way we can obtain the optimal $w_t = \frac{C_1}{C_1+C_2} A X^{*\mathrm{T}} (X^* X^{*\mathrm{T}})^{-1}$.

D. Solution for Sparse IPL

Although the proposed Sparse IPL method is non-convex with respect to the new tree $h_m(\cdot)$ and the sparse vector v_m; however, the local solution of each variable can be solved when the other is fixed. Then the local optimum of the proposed sparse IPL method can be obtained.

1) Optimization of $h_m(\cdot)$: To optimize $h_m(\cdot)$, we first fix sparse vector v_m, and solve the following optimization problem regarding $h_m(\cdot)$:

$$\min_{h_m} \sum_{i=1}^{n} (h_m(x_i) - r_i)^2 + C_1 \sum_{i=1}^{n} \left(h_m(x_i) - r_i - x_i^{*\mathrm{T}} D v_m\right)^2. \tag{13}$$

By means of some simple mathematical operations, Eq. (13) can be further reformulated as follows:

$$\min_{h_m} \sum_{i=1}^{n} [h_m(x_i) - (r_i + \frac{C_1}{C_1+1} x_i^{*\mathrm{T}} D v_m)]^2.. \tag{14}$$

Similarly as in IPL, we adopt the standard classification and regression trees (CART) algorithm to efficiently solve the optimization problem in Eq. (14), and $r_i^* = r_i + \frac{C_1}{C_1+1} x_i^{*\mathrm{T}} D v_m$ plays the same role as r_i in Eq. (3). By using CART, the current tree $h_m(\cdot)$ and its predictor value of input data $\{x_1, \ldots, x_n\}$ can then be solved as $h_m(\cdot) \leftarrow Cart(\{(x_1, r_1^*), \ldots, (x_n, r_n^*)\}, k)$.

2) Optimization of v_m: Now we consider the optimization of v_m. Considering $h_m(\cdot)$ is fixed, the optimization problem is rewritten as:

$$\min_{v_m} C_1 \sum_{i=1}^{n} [h_m(x_i) - r_i - x_i^{*\mathrm{T}} D v_m]^2 + C_2 \sum_{i=1}^{n} \|v_m\|_1 \tag{15}$$

By means of some simple mathematical operations, Eq. (15) can be further reformulated as follows:

$$\min_{v_m} C_1 \sum_{i=1}^{n} (b_i - v_m^{\mathrm{T}} \tilde{x})^2 + C_2 \sum_{i=1}^{n} \|v_m\|_1 \tag{16}$$

where $b_i = h_m(x_i) - r_i \in \mathbb{R}$ and $\tilde{x_i} = D^{\mathrm{T}} x_i^* \in \mathbb{R}^{\eta \times 1}$. Note that the minimization problem in Eq. (16) is a typical least absolute shrinkage and selection operator (LASSO) optimization problem [32]. And Eq. (16) can be further reformulated as Eq. (17), which can be efficiently solved by the coordinate descent in Scikit-learn [31].

$$\min_{v_m} \frac{1}{2n} \|B - \tilde{X}^{\mathrm{T}} v_m\|_2^2 + \|v_m\|_1 \tag{17}$$

where $B = [b_1, \ldots, b_n]^{\mathrm{T}} \in \mathbb{R}^n$, $\tilde{X} = [\tilde{x_1}, \ldots, \tilde{x_n}] \in \mathbb{R}^{\eta \times n}$, and $\lambda = \frac{C_2}{2C_1}$.

V. Experiments

In this section, we evaluate the efficiency of our proposed IPL method and compare it with several representative existing algorithms: 1) two algorithms use only example feature data, including a gradient boosted decision tree (GBDT) algorithm [22] and a standard support vector machine (SVM) method using the Gaussian kernel; 2) four algorithms that use privileged information, including linear kernel SVM+ referred as SVM+(linear), gaussian kernel SVM+ referred as SVM+(rbf) [1], L2-SVM+ [2], and gradient boosted with privileged information referred as GB+ [13]. Specifically, we first conduct two simulated experiments on a UCI dataset named Pima Indians Diabetes and a Galaxy dataset [33]. Then, we study different types of privileged information in real applications, including holistic description for digit classification task, attribute annotations for object recognition task, and depth information for face pose classification task. In order to further evaluate the performance of sparse IPL that is a variant of our IPL method, we evaluate it on the RGB-D dataset, as an example.

TABLE I
Description of Datasets.

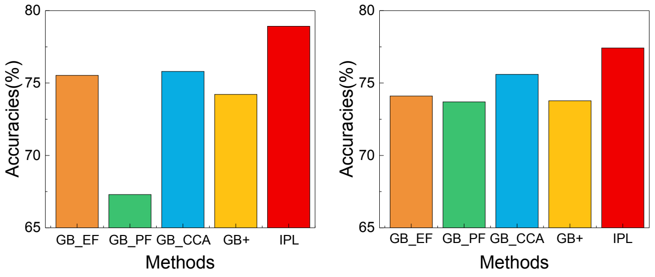
Dataset	Examples	PF	EF	Classes
Pima Indians Diabetes	768	4	4	2
Galaxy	505	21	127	2

A. Parameter setting

There are five parameters in our proposed IPL model, including the depth of a tree k, the number of trees to be boosted N, the learning rate β, and two tradeoff parameters C_1 and C_2. We tune the parameters β, d and N from the range of {0.05, 0.1, 0.2, 0.3}, {2, 5, 10} and {100, 300, 500}, respectively. We also tune the parameters C_1 and C_2 of IPL from the range of $10^{[-2,-1,\ldots,1,2]}$. For the GBDT and GB+ method, we also vary the tree depth k and the learning rate β and the number of trees N in the range of {2, 5, 10}, {0.05, 0.1, 0.2, 0.3}, and {100, 300, 500}, respectively. For the standard SVM, SVM+(rbf), L2-SVM+ and SVM+(linear) methods, the tradeoff parameter C is set in the range of $10^{[-2,-1,\ldots,1,2]}$. The Gaussian kernel is used for the standard SVM, SVM+(rbf), and L2-SVM+methods, and the linear kernel is used for the SVM+(linear) method.

B. Simulated experiments

1) Datasets: In this section, in order to show the potential of methods using the privileged information, we control the features that constitute the example and privileged feature spaces, respectively. The experiments are conducted on two publicly available datasets, including a standard classification dataset in the UCI repository (Pima Indians Diabetes), and a novel dataset that predicts the galaxy types (Galaxy) [33]. The Pima Indians Diabetes dataset contains 768 data items described by 8 attributes and classified into two classes. Following the same setting as [13], we select a certain number of features (i.e., the 1st, 4th, 5th and 7th attributes) as the example features, and the rest (i.e., the 2nd, 3rd, 6th and 8th attributes) as the privileged features. Another dataset, Galaxy contains galaxy images derived from the Sky Survey



(a) Pima Indians Diabetes (b) Galaxy

Fig. 1. Classification performances of GB_EF, GB_PF, GB_CCA, GB+ and IPL methods on two datasets: (a) Pima Indians Diabetes, and (b) Galaxy.

TABLE II
MEANS AND STANDARD DEVIATIONS OF THE CLASSIFICATION ACCURACIES (%) OBTAINED BY DIFFERENT METHODS ON THE PIMA INDIANS DIABETES AND GALAXY DATASETS.

Dataset	GBDT	SVM	Linear-SVM+	SVM+(rbf)	L2-SVM+	GB+	IPL
Diabetes	75.53±2.62	74.94±2.35	67.16±5.18	75.55±2.69	75.86±2.56	74.21±2.48	**78.92±2.49**
Galaxy	73.68±3.29	73.08±3.00	61.78±4.46	73.28±2.80	73.86±1.50	73.77±3.13	**77.42±3.16**

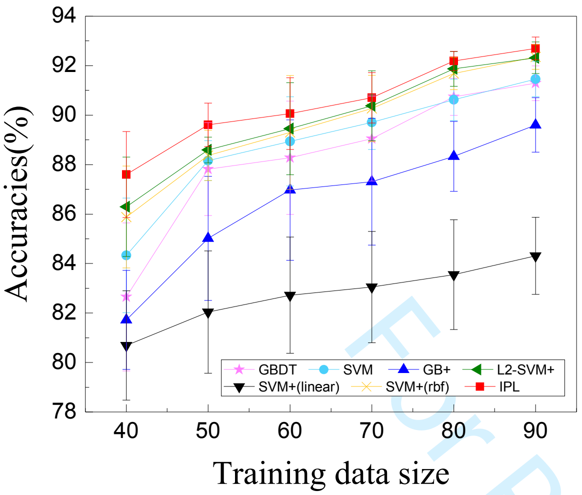


Fig. 2. Means and standard deviations of the classification accuracies obtained by different methods on the MNIST+ dataset.

Database [34]. Similar to the settings used in [33], galaxy images are classified into the spiral and non-spiral classes. And the shape information is used to generate privileged features into a 21-dimensional vector, while example features into a 100-dimensional vector are extracted from color information. Details of the information used in the experiments are summarized in Table I, which lists the total number of examples in the dataset, the size/dimensions of both privileged feature data and example feature data, and the number of classes. For both datasets, 80% of examples are randomly split into the training set for five times and the remaining examples are used as the test set. We use 5-fold cross-validation scheme in the training set to find the best parameters.

2) Results: The average classification accuracies over five random trials are summarized in Table II for different methods on the Pima Indians Diabetes and Galaxy datasets. It can be seen from the classification results that our IPL method achieves the best performance on both datasets. Compared with the baseline GBDT, our IPL achieves obviously better results on two datasets, which demonstrates the efficiency of the proposed IPL method that uses privileged information to learn a better classification model. In addition, our IPL method performs better than GB+. This supports our decision to regard privileged information as auxiliary data in order to generate comments, as well as underscores the fact that privileged information is insufficient on its own for making label predictions. The GB+ method can be observed to be slightly more accurate as GBDT on the Galaxy dataset. But its accuracy is obviously lower than that of GBDT on the Pima Indians Diabetes dataset. One possible explanation is that the privileged information in this dataset is not informative enough to be equivalent to the example features for learning the prediction model, or the probability distribution of EF data and PF data is different a lot. Therefore the consistency constraint of GB+ is unfeasible and the use of privileged information in GB+ does not help to improve the classification performance. The possible explanation is discussed in detail in the following subsection.

SVM+(rbf) and L2-SVM+ can obtain better classification accuracies than the baseline SVM, which demonstrates their effectiveness to use privileged information. And SVM+(rbf)and l2-SVM+ obviously outperform linear-SVM+, which suggests the efficiency of the gaussian kernel strategy. Generally, L2-SVM+ gets better results than other methods except our IPL.

3) Quality of the privileged information: We also evaluate the quality of the privileged information and indicate its effects on the performance of classification of example feature data in some aspects. Since the example feature (EF) data and the privileged feature (PF) data are in different feature spaces, we perform CCA [35] between EF and PF data to map them into a common feature space (referred as a CCA space), and the transformed EF data and PF data are referred as EF_CCA and PF_CCA, respectively. We compare the performance of several algorithms under different combinations of EF_CCA data and PF_CCA data all in the CCA space, including GB_EF, GB_PF, and GB_CCA. Also, GB+ and the proposed IPL method are also compared without using CCA.

Specifically, GB_EF only uses the EF_CCA data to train and test the GBDT model. GB_PF indicates that GBDT is trained with PF_CCA data and tested on the EF_CCA data. While GB_CCA uses both EF_CCA data and PF_CCA data to train a GBDT and then tests on the testing set of EF_CCA data. The GB+ and the proposed IPL are trained using both EF and PF data and are then tested on EF data. We can observe that in Fig. 1 (a), the classification results of GB_PF and GB+ are obviously worse than those of GB_EF on the Pima Indians Diabetes dataset. Moreover, GB_CCA also gets almost the same results though twice as many training examples are used. While from the results shown in Fig. 1 (b), GB_PF obtains slightly lower performance than GB_EF, and GB_CCA exhibits better performance than GB_EF. And GB+ also gets a slightly better result than GBDT as shown in Table II. Noticeably, for methods such as the GB+ method, which depend heavily on auxiliary data, the better performance is obtained when the predictions of example feature data and auxiliary data are as similar as possible. When the probability distributions of the example feature data and auxiliary data are quite different, or the quality of PF is not good enough, the performance of GB+ may suffer. This is evident from the results on the Pima Indians Diabetes shown in Table II,

TABLE III
CLASSIFICATION ACCURACIES FOR ONE-VS-ALL BINARY CLASSIFICATIONS. THE FEATURE EXTRACTED FROM THE IMAGE IS THE EXAMPLE VIEW, AND THE ATTRIBUTE ANNOTATION OF THE IMAGE IS THE AUXILIARY VIEW. BEST ACCURACIES ARE HIGHLIGHTED IN BOLDFACE.

	GBDT	SVM	SVM+(linear)	SVM+(rbf)	L2-SVM+	GB+	IPL
Bag	75.48±1.24	76.66±2.60	74.64±1.92	76.43±2.08	76.79±2.17	71.43±1.96	**77.40±1.39**
Building	81.34±1.88	84.22±2.59	80.41±2.31	83.61±3.42	84.22±2.95	78.66±1.71	**84.23±1.74**
Carriage	74.78±1.87	76.52±3.19	73.91±5.30	77.61±2.88	76.74±3.49	69.87±2.13	**78.04±2.10**
Centaur	80.67±2.49	85.33±3.33	85.33±2.79	87.33±2.98	87.33±2.78	89.23±0.82	**89.33±2.78**
Donkey	86.19±4.29	89.28±4.68	89.76±3.73	**90.48±3.91**	90.24±4.26	89.15±3.73	89.76±3.82
Goat	75.71±3.61	81.02±4.11	77.35±3.56	80.41±3.58	81.63±3.22	76.43±2.96	**82.45±3.18**
Jetski	77.17±3.21	79.67±3.00	76.42±1.83	80.42±1.74	80.33±1.41	72.08±3.42	**81.58±1.94**
Monkey	67.86±2.93	72.14±1.94	71.25±2.78	**72.67±1.74**	72.14±1.32	65.98±2.80	71.96±1.92
Mug	77.94±1.61	79.41±1.69	77.64±2.58	79.56±1.83	79.12±1.85	72.97±3.58	**79.56±1.27**
Statue of people	76.77±2.57	78.71±3.14	77.74±2.46	79.35±2.70	**80.00±2.98**	72.87±2.42	79.52±2.20
Wolf	75.24±1.75	77.05±2.46	74.92±3.09	76.89±1.97	77.05±1.42	69.90±2.10	**77.70±1.86**
Zebra	75.07±2.19	76.99±2.13	76.02±3.06	77.53±1.48	77.94±2.44	70.77±2.88	**78.49±2.93**

where the accuracies obtained by GB+ are even lower than those obtained by GB_EF. However, our IPL method uses the privileged information as auxiliary data to provide comments on the learned model and help to improve its learning, which can be successful even if the PF data is not as informative as the example features for making accurate label predictions.

C. Holistic Description as Privileged Information

1) Dataset: We consider the digit classification task of classifying two digits “5” and “8” in the MNIST+ dataset with the help of privileged information. MNIST+ dataset has been widely used to evaluate the effectiveness of the auxiliary textual descriptions as the privileged information [1]. The MNIST+ dataset contains 2943 images of the digit “5” and 3025 images of digit “8” from the MNIST database. And each image is additionally supplied with a holistic (poetic) description (see [1] for examples) that serves as the privileged feature data. In order to make it more difficult to distinguish between these two digits, all images of two digits are resized into 10×10 pixels in the MNIST+ dataset. We treat the 100-dimensional vector of raw pixels as the example feature data for each image. The holistic (poetic) description for each image is translated into a 21-dimensional feature vector.

The 100 examples of 10×10 images are used as a training set, and remaining examples are randomly split into a validation set of 4002 images and a test set of 1866 images. In the experiment, we use training sets of increasing size of 40, 50, 60, ..., 90. For each method, we perform five rounds of experiments using randomly selected samples from the training set, validation set and test set, and the average of test classification accuracies and the standard deviations are reported in Fig. 2.

2) Results: As we can see in Fig. 2, the classification accuracies of all methods show an upwards trend when the training data size increases. As the number of training examples varies, SVM+(rbf) and L2-SVM+ obviously outperform the baseline SVM, and IPL clearly improves classification accuracies compared with its baseline GBDT. This shows the utilizing holistic description as privileged information for the digit classification is generally useful. Moreover, the proposed IPL method outperforms all compared methods, which demonstrates its robustness and efficiency in the utilizing of the privileged information. However, GB+ gets poorer results than GBDT, which shows the consistence constraint of GBDT is unfeasible on this dataset. According to the results, L2-SVM+ generally performs better than SVM+(rbf), due to the effective new formulation of L2-SVM+ and its dual coordinate descent algorithm. And SVM+(linear) generally gets the worst results and performs significantly worse than SVM+(rbf) that uses the Gaussian kernel, which demonstrates that the linear kernel is less suitable than the gaussian kernel on this dataset.

D. Attributes as Privileged Information

1) Dataset: We use the a-Yahoo dataset [36], which contains twelve object categories from the Yahoo image search. The objects in the a-Yahoo dataset are: bag, building, carriage, centaur, donkey, goat, jet ski, monkey, mug, statue of people, wolf, and zebra. All images in the dataset are used as the example data, and they are also labeled with attribute annotations which are used as the privileged information. According to [36], the example feature data for each image is given by a 9751-dimensional feature by describing local texture, HOG, edge, and color descriptors inside the bounding box, shapes and locations. Then PCA is performed for the example

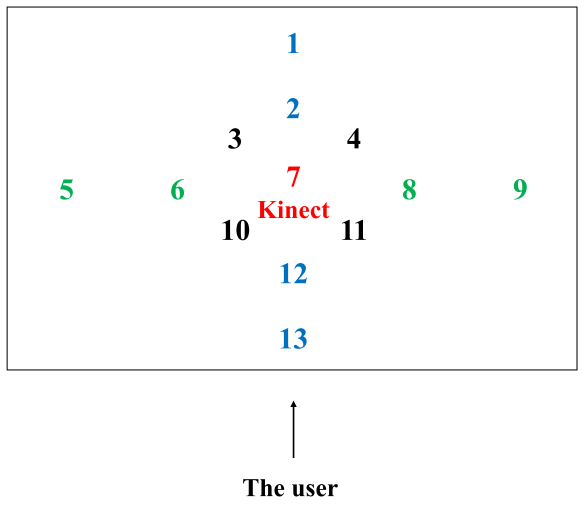


Fig. 3. Each person looks at fixed points on a wall behind the Kinect sensor. The Kinect is placed on point number 7.

TABLE IV
MEANS AND STANDARD ERRORS OF THE AP PERFORMANCE OVER 5 RUNS. THE FEATURE EXTRACTED FROM THE RGB IMAGE IS THE EXAMPLE VIEW, AND THE FEATURE OBTAINED FROM THE DEPTH IMAGE IS THE AUXILIARY VIEW. BEST ACCURACIES ARE HIGHLIGHTED IN BOLDFACE.

	GBDT	SVM	SVM+(linear)	SVM+(rbf)	L2-SVM+	GB+	IPL	Sparse IPL
Class 1 vs Class 2	93.52±0.97	94.53±1.11	93.86±1.56	95.76±0.93	**95.87±0.85**	90.71±1.16	95.42±0.81	95.76±0.49
Class 2 vs Class 3	76.87±1.10	78.50±2.11	69.91±3.28	78.66±1.77	79.28±1.85	74.12±1.99	79.30±1.20	**80.89±1.42**
Class 1 vs Class 3	88.03±2.54	88.52±2.74	82.41±3.36	88.66±2.78	89.01±2.96	82.43±2.80	90.00±1.40	**90.63±1.67**

feature to reduce computational expense and we fix the PCA dimension to 200 in the experiment. The attribute annotations capture 64 binary properties that characterize shape, material, and the presence of important parts of the visible object. And we use the 64 dimensional attributes as the privileged information. We conduct 12 binary classification experiments for each class versus the rest classes. We use 40%, 30% and 30% examples from the desired class and the same number of examples randomly drawn from the remaining classes for training, validation and testing, respectively. We repeat the experiment for five rounds by using different randomly sampled pairs. And the average of classification accuracies and the standard deviations are reported in Table III, where the maximum values of the average accuracies in the corresponding rows are marked in bold.

2) Results: As we can see from Table III, in general utilizing attribute annotations as privileged information for the object classification is useful. The proposed IPL method performs better than the baseline GBDT in 12 cases out of 12. And SVM+(rbf) outperforms the baseline SVM in 8 cases out of 12, while L2-SVM+ also outperforms SVM in 10 cases out of 12. The GB+ method only exhibits better performance than the baseline GBDT method in 3 cases out of 12, which shows the consistence constraint of GBDT is unfeasible in most cases on this dataset. SVM+(linear) obtains worse results than both SVM+(rbf) and L2-SVM+ in 12 cases out of 12. Moreover, SVM+(linear) performs better than SVM only in 5 cases out of 12, which demonstrates that the Gaussian kernel is more suitable on the dataset than the linear kernel, and the standard SVM using the Gaussian kernel is a competitive baseline. Generally, the proposed IPL method outperforms all the other methods in 9 cases out of 12, followed by SVM+(rbf) obtains the best performance in 2 cases out of 12. This demonstrates the effectiveness and robustness of our IPL method.

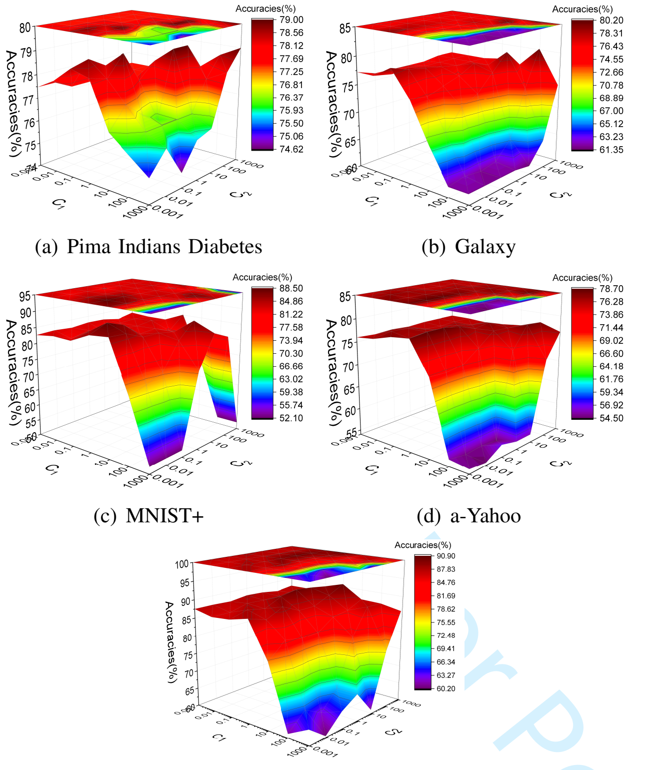
E. Depth Information as Privileged Information

1) Dataset: In this subsection, we evaluate the performance of the invariant of IPL method named sparse IPL on the RGB-D Face dataset [37]. Specifically, we perform face pose recognition on the dataset, which contains color and depth images and are taken by a Kinect sensor exactly at the same time. The RGB-D Face dataset contains color images and their corresponding depth images from 31 persons in different face poses. And each pose for each person is repeated 3 times, which results in 3×31 image pairs for each pose. The different face positions are produced by making each person sequentially look at thirteen fixed points (seen in Fig. 3) on a wall behind the Kinect sensor. Since the number of images per face pose is relatively small, we merge the poses into three groups: poses in the vertical direction (point number 1, 2, 12, 13 in blue) referred as class 1, the horizontal direction (point number 5, 6, 8, 9 in green) referred as class 2, and the non-vertical and non-horizontal direction (point number 3, 4, 10, 11 in black) referred as class 3. Then we perform binary classification on each pair of groups.

We use 40% color-depth image pairs per class for training, 30% image pairs per class for validation and the rest 30% for testing. And the train/validation/test split is repeated for five times. For all images in the data set, we first crop each image into a fixed size of 150×150. For each RGB color image, it is converted into the gray image. Then, we divide each image into 15×15 non-overlapping subregions with the size of 10×10, and extract the LBP feature from each subregion. A single 5900-dimensional feature vector is formed by concatenating the LBP features from all the 100 subregions. The same strategy is also performed to extract a 5900-dimensional feature vector for each depth image. In the experiment, we use feature vectors extracted from the color images as the example data representation and those extracted from the depth images as the auxiliary data. Finally, PCA is applied for both example features and privileged features for dimension reduction to obtain 150-dimensional compact representations. For sparse IPL, we vary the parameter η in the range of $\{50, 70, 100, 150\}$. The average classification accuracies and the standard deviations of sparse IPL, IPL and all compared methods over five random trials are reported in Table IV. The best results are highlighted in boldface.

2) Classification Accuracies: From the results shown in Table IV, we observe that the utilizing of the depth images as privileged information for face pose classification is useful. The SVM+(rbf) and L2-SVM+ methods achieve better results than the baseline SVM method in all 3 cases. Similar results can also be observed in the table that the proposed IPL and sparse IPL methods clearly outperform the baseline GBDT method. Noticeably, we can find that sparse IPL further improves the performance of IPL (in all 3 cases) and outperforms L2-SVM+ (in 2 out of 3 cases), which reveals the effectiveness of sparse IPL that uses sparse representation to learn a better auxiliary function base on the previous learned projections.

Moreover, SVM+(linear) which uses the linear kernel clearly gets worse results than both SVM+(rbf) and L2-SVM+ which use the Gaussian kernel. Also, the nonlinear GBDT, GB+, IPL and sparse IPL methods generally outperform SVM+(linear). However, the results obtained by GB+ are worse than those of the baseline GBDT method. This shows that the strategy of GB+ that iteratively encourages the consistent predictions of two decision trees constructed with example features and privileged features is not effective for the face pose classification problem with depth information



(a) Pima Indians Diabetes

(b) Galaxy

(c) MNIST+

(d) a-Yahoo

(e) RGB-D Face

Fig. 4. Effects of the parameters C_1 and C_2 on different datasets: (a) Pima Indians Diabetes, (b) Galaxy, (c) MNIST+, (d) a-Yahoo, and (e) RGB-D Face.

as privileged information on this dataset. Rather, the good performance of the IPL and sparse IPL shows their robustness and effectiveness compared with GB+.

3) Comparison of Training Time Between IPL and Sparse IPL: We further use the RGB-D Face dataset as an example to show the training time of our proposed IPL and Sparse IPL methods. The experiments are conducted on a workstation with Intel Xeon CPU@3.50GHz. We compare the average training times from five rounds of the binary classification of class 1 and class 3, as an example. The number of trees N, the learning rate β, the tree depth k and parameter η are set to 500, 0.1, 2 and 70, respectively. As a result, IPL costs 28.85 seconds, while sparse IPL takes 20.03 seconds. And sparse IPL reduces training time by 30% compared with IPL.

F. Experimental Analysis

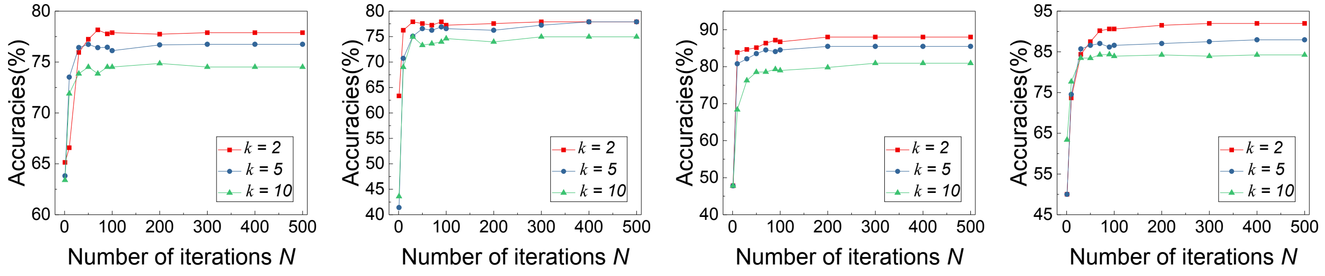
1) On the parameters C_1 and C_2: We introduce a parameter C_1 that is expected to balance the influence of real-valued auxiliary functions in the privileged space on both the residual target and the learning of the decision tree in Eq. (7). The reason is that the privileged information is sometimes not informative as the example features to make accurate label predictions in practice. If C_1 is too small, the privileged information has small effect on the new learned model. While a too large C_1 may degrade the performance of our IPL method because of not informative privileged information. Moreover, the parameter C_2 modulates the influence of the L2 regularization term $\sum_{i=1}^{n} \|w_t x_i^*\|_2^2$ in Eq. (7). If C_2 is too small, the values of the auxiliary function may be too large, which will introduce much error for the residual target. In the following, we discuss the effects of C_1 and C_2, as well as the interplay between the two corresponding loss terms.

The performance comparisons of varying C_1 and C_2 from 10^{-3} to 10^3 are shown in Fig. 4, where the Pima Indians Diabetes, Galaxy, MNIST+, a-Yahoo and RGB-D Face datasets are used as examples. Specifically, for the MNIST+ dataset we use 40 training examples. For the RGB-D Face dataset, the accuracies of the binary classification of class 1 and class 3 are reported. And for the a-Yahoo dataset, the result of the binary classification of bag is illustrated as an example.

It can be seen from Fig. 4 (a), (b), (c), (d) and (e), the accuracies with varying C_1 and C_2 generally exhibit similar tendencies. The classification accuracy is poor when C_1 is too large and C_2 is small, which demonstrates the necessity of the L2 regularization term $\sum_{i=1}^{n} \|w_t x_i^*\|_2^2$ and its tradeoff parameter. And since the target residual learning a new tree is almost totally decided by the term $\sum_{i=1}^{n} (h_t(x_i) - r_i - w_t x_i^*)^2$, C_2 should be set at a large value to penalize the regularization term much, to control its magnitude from growing too large. While when C_1 is large and C_2 becomes larger, the performance gets better. Generally, when C_1 is set at a middle value, our IPL method performances better than that when C_1 is larger or smaller. This demonstrates its efficiency by using privileged information to provide comments on the residual targets when learning a new tree. If C_1 is small, the privileged information makes small contribution to the learning of the model and it performs similar as GBDT. The effectiveness of the parameter C_1 and C_2 is clearly demonstrated according to the results shown in Fig. 4.

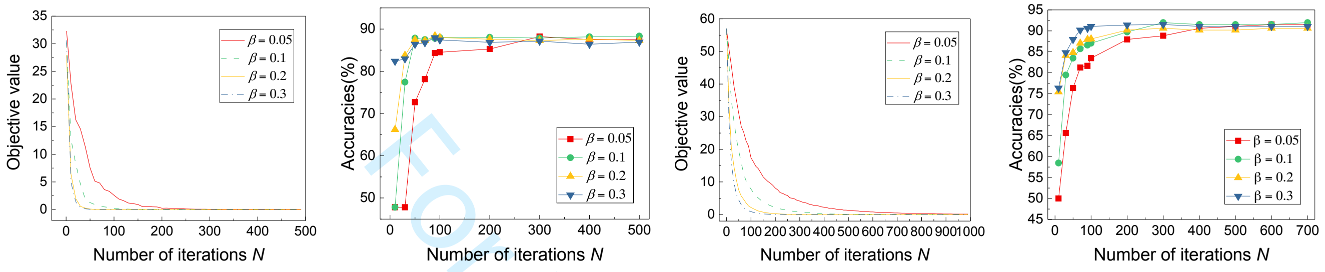
2) On the parameters k and N: The depth of a single decision tree k and the number of iterations (i.e., number of trees) N are two important parameters in the proposed IPL method. We further evaluate the performance of our IPL method with different k as N is increased from 1 to 500. According to the previous researches on the baseline GBDT [23], [38], they suggest that the depth should be small and is approximately equal to 4. Therefore, we compare the performance when k is set at 2, 5 and 10 as N grows, and other parameters are fixed. The results are illustrated in Fig. 5, where the Pima Indians Diabetes, Galaxy, MNIST+, and RGB-D Face datasets are used as examples. Specifically, for the MNIST+ dataset we used 40 training examples as an example, while for the RGB-D Face dataset the result of the binary classification of class 1 and class 3 is reported.

In general, for depths of 2, 5, and 10, as N increases, the new trees are iteratively added and the comments of the privileged information are iteratively updated. The results show that the classification accuracies also generally show an upwards trend across these datasets. This demonstrates the positive impact of iteratively learning new trees from example features and corresponding new comments from privileged information on classification performance. More specifically, in Figures 5 (a), (b), (c) and (d), the accuracies at different tree depths generally exhibit similar tendencies. When N is less than a certain value (depending on the datasets and tree depths), the accuracies improve obviously and rapidly. As the number



(a) Pima Indians Diabetes (b) Galaxy (c) MNIST+ (d) RGB-D Face

Fig. 5. Classification accuracies of the proposed IPL method with different tree depths k and increasing numbers of boosted trees N on different datasets: (a) Pima Indians Diabetes, (b) Galaxy, (c) MNIST+, and (d) RGB-D Face.



(a) (b) (c) (d)

Fig. 6. Effects of the parameter β on the convergence and performance of IPL on the MNIST+ and RGB-D Face datasets: (a) Convergence curves and (b) performance on the MNIST+ dataset; (c) Convergence curves and (d) performance on the RGB-D Face dataset.

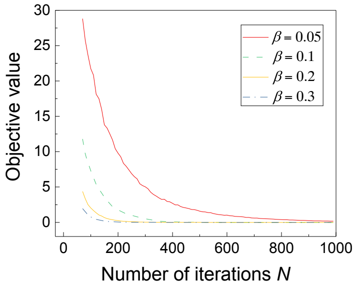
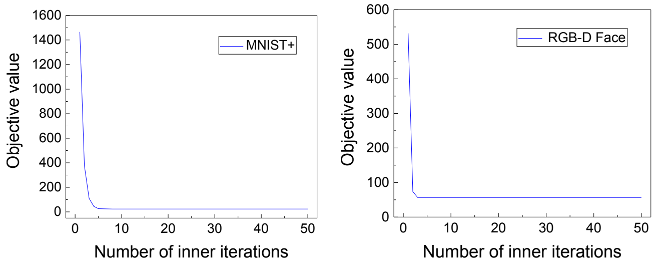


Fig. 7. Effects of the parameter β on convergence of sparse IPL on the RGB-D Face dataset.



(a) (b)

Fig. 8. Inner convergence on the (a) MNIST+ dataset; and (b) RGB-D Face dataset.

of iterations continues to increase, the accuracies gradually change slowly before the rate of improvement stabilizes.

Moreover, it is observed that the classification results are obviously influenced by different tree depths k. Generally, better accuracies can be obtained when the tree depth k is smaller. The proposed IPL method performs the best when k is set at 2. When k is equal to 10, the performance of the proposed IPL method becomes significantly worse than those obtained by IPL with k equaling 2 and 5. Therefore, the tree depth should be small for better performance of the proposed IPL method. From the results shown in Fig. 5, the effectiveness of the tree depth k and the number of boosted trees N is clearly demonstrated.

3) On the parameter β: The learning rate $\beta > 0$ is a shrinkage parameter which shrinks the contribution of each tree to the IPL model and should be substantially less than 1. Usually more trees (i.e., larger N) are required if β is decreased, and vice versa. Generally, a smaller β and relatively larger N are preferable, which is conditional on specific situations [39]. In the following we investigate the effects of β on the convergence and performance of our IPL method. The MNIST+ dataset with a training set of size 40 and the RGB-D Face dataset where the binary classification of class 1 and class 3 are used as examples. The objective values and classification accuracies of our IPL method are illustrated in Fig. 6 by varing β in the range of $\{0.05, 0.1, 0.2, 0.3\}$.

We can observe from Fig. 6 (a) and (c) that the rate of convergence is significantly influenced by the learning rate β on both datasets. The smaller the learning rate β is, the slower the IPL method converges. While the larger β is, the faster the IPL method converges. Fig. 6 (b) and (d) show that the bigger value for β approaches obviously better predictive performance when N is small. While the smaller values for β gradually get better performance as N grows, and generally require hundreds of trees to reach minimum error. Generally, on the MNIST+ dataset, when $\beta \geq 0.1$, the method converges in less than 100 iterations. And when $\beta = 0.05$, the algorithm converges in almost 300 iterations. While on the RGB-D Face dataset, it takes larger N to lead to a convergence. It takes almost 200, 300, 400 and 500 iterations to converge when β is equal to 0.05, 0.1, 0.2 and 0.3, respectively. The results shown in Fig. 6 clearly demonstrate the effectiveness of the learning rate β and the number of boosted trees N.

4) On parameter β and convergence of sparse IPL: In this subsection, we analyze the effect of the influence of the learning rate β on the convergence of sparse IPL. We use the binary classification between class 1 and 3 on the RGB-D Face dataset as an example. The results are shown in Fig. 7. We can observe the rate of convergence is significantly influenced by the learning rate β. The β is larger, sparse IPL converges faster.

5) Inner convergence: We discuss the inner convergence of the optimization of the proposed IPL algorithm, by using the MNIST+ and RGB-D Face datasets as examples. Fig. 8 shows the objective function values as the number of inner iterations grows. According to the convergence curves, it is observed that the convergence to optimal solution is guaranteed after almost 5 iterations.

VI. Conclusion

In this paper, we propose a novel iterative privileged learning (IPL) method within the context of gradient boosted decision trees from the LUPI paradigm aspect. Rather than letting the comments used to assess the model remain static and passive, the proposed model ensures that comments from privileged information are iteratively updated to keep them compatible with the latest classification model. More specifically, in each iteration of GBDT, a new decision tree is trained to accurately assess and coach the up-to-date model, while an additional linear auxiliary function is also employed to generate comments from the privileged information. The IPL method integrates the learning of the new decision tree and the auxiliary function into a unified objective function, which can be efficiently optimized. And a variant of IPL named sparse IPL is proposed. Experimental results on real-world datasets have demonstrated the advantages of exploiting privileged information in an iterative manner, as well as the the effectiveness of the proposed IPL and sparse IPL algorithms.

References

[1] V. Vapnik and A. Vashist, “A new learning paradigm: Learning using privileged information,” *Neural Networks*, vol. 22, no. 5, pp. 544–557, 2009.

[2] W. Li, D. Dai, M. Tan, D. Xu, and L. Van Gool, “Fast algorithms for linear and kernel SVM+,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2258–2266.

[3] S. Wang, D. Tao, and J. Yang, “Relative attribute SVM+ learning for age estimation,” *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 827–839, 2016.

[4] D. Pechyony and V. Vapnik, “On the theory of learnining with privileged information,” in *Advances in neural information processing systems*, 2010, pp. 1894–1902.

[5] V. Sharmanska, N. Quadrianto, and C. H. Lampert, “Learning to rank using privileged information,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 825–832.

[6] W. Li, L. Niu, and D. Xu, “Exploiting privileged information from web data for image categorization,” in *European Conference on Computer Vision*. Springer, 2014, pp. 437–452.

[7] V. Vapnik and R. Izmailov, “Learning using privileged information: similarity control and knowledge transfer.” *Journal of Machine Learning Research*, vol. 16, no. 55, pp. 2023–2049, 2015.

[8] L. Niu, W. Li, and D. Xu, “Exploiting privileged information from web data for action and event recognition,” *International Journal of Computer Vision*, vol. 118, no. 2, pp. 130–150, 2016.

[9] X. Xu, W. Li, and D. Xu, “Distance metric learning using privileged information for face verification and person re-identification,” *IEEE transactions on neural networks and learning systems*, vol. 26, no. 12, pp. 3150–3162, 2015.

[10] S. Fouad, P. Tino, S. Raychaudhury, and P. Schneider, “Incorporating privileged information through metric learning,” *IEEE transactions on neural networks and learning systems*, vol. 24, no. 7, pp. 1086–1098, 2013.

[11] J. Tang, Y. Tian, P. Zhang, and X. Liu, “Multiview privileged support vector machines,” *IEEE transactions on neural networks and learning systems*, no. 99, pp. 1–15, 2017.

[12] J. Feyereisl, S. Kwak, J. Son, and B. Han, “Object localization based on structural SVM using privileged information,” in *Advances in Neural Information Processing Systems*, 2014, pp. 208–216.

[13] R. Pasunuri, P. Odom, T. Khot, K. Kersting, and S. Natarajan, “Learning with privileged information: Decision-trees and boosting,” in *International Joint Conference on Artificial Intelligence Workshop*, 2016.

[14] L. Niu, Y. Shi, and J. Wu, “Learning using privileged information with L-1 support vector machine,” in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on*, vol. 3. IEEE, 2012, pp. 10–14.

[15] J. Liu, W. Zhu, and P. Zhong, “A new multi-class support vector algorithm based on privileged information,” *Journal of Informational and Computational Science*, vol. 10, no. 2, pp. 443–450, 2013.

[16] Y. Ji, S. Sun, and Y. Lu, “Multitask multiclass privileged information support vector machines,” in *2012 21st International Conference on Pattern Recognition (ICPR)*. IEEE, 2012, pp. 2323–2326.

[17] Y. W. C. X. D. T. Shan You, Chang Xu, “Privileged multi-label learning,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 3336–3342.

[18] M. Lapin, M. Hein, and B. Schiele, “Learning using privileged information: SVM+ and weighted SVM,” *Neural Networks*, vol. 53, pp. 95–108, 2014.

[19] X. Yang, M. Wang, L. Zhang, and D. Tao, “Empirical risk minimization for metric learning using privileged information.” in *IJCAI*, 2016, pp. 2266–2272.

[20] J. T. Zhou, X. Xu, S. J. Pan, I. W. Tsang, Z. Qin, and R. S. M. Goh, “Transfer hashing with privileged information,” in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. AAAI Press, 2016, pp. 2414–2420.

[21] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC Press, 1984.

[22] A. Mohan, Z. Chen, and K. Weinberger, “Web-search ranking with initialized gradient boosted regression trees,” in *Proceedings of the Learning to Rank Challenge*, 2011, pp. 77–89.

[23] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” *Annals of Statistics*, pp. 1189–1232, 2001.

[24] P. Geurts, L. Wehenkel, and F. d’Alché Buc, “Gradient boosting for kernelized output spaces,” in *Proceedings of the 24th International Conference on Machine Learning*. ACM, 2007, pp. 289–296.

[25] R. A. Hutchinson, L.-P. Liu, and T. G. Dietterich, “Incorporating boosted regression trees into ecological latent variable models.” in *AAAI*, vol. 11, 2011, pp. 1343–1348.

[26] V. Kazemi and J. Sullivan, “One millisecond face alignment with an ensemble of regression trees,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1867–1874.

[27] W. Liu and I. W. Tsang, “Sparse perceptron decision tree for millions of dimensions.” in *AAAI*, 2016, pp. 1881–1887.

[28] F. Diego and F. A. Hamprecht, “Structured regression gradient boosting,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1459–1467.

[29] B. Manning, “Extreme gradient boosting and behavioral biometrics.” in *AAAI*, 2017, pp. 4969–4970.

[30] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[31] F. Pedregosa, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, “Scikit-learn: Machine learning in python,” *Journal of Machine Learning Research*, vol. 12, no. 10, pp. 2825–2830, 2011.

[32] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[33] D. S. Dhami, “Morphological classification of galaxies into spirals and non-spirals,” Ph.D. dissertation, Indiana University, 2015.

[34] K. W. Willett, C. J. Lintott, S. P. Bamford, K. L. Masters, B. D. Simmons, K. R. Casteels, E. M. Edmondson, L. F. Fortson, S. Kaviraj, W. C. Keel *et al.*, “Galaxy zoo 2: detailed morphological classifications for 304 122 galaxies from the Sloan Digital Sky Survey,” *Monthly Notices of the Royal Astronomical Society*, vol. 435, no. 4, pp. 2835–2860, 2013.

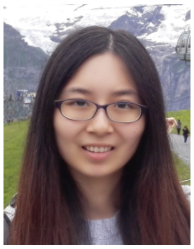
[35] D. R. Hardoon, S. Szedmak, and J. Shawetaylor, “Canonical correlation analysis: an overview with application to learning methods.” *Neural Computation*, vol. 16, no. 12, p. 2639, 2004.

[36] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, “Describing objects by their attributes,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 1778–1785.

[37] R. I. Hg, P. Jasek, C. Rofidal, K. Nasrollahi, T. B. Moeslund, and G. Tranchet, “An rgb-d database using microsoft’s kinect for windows for face detection,” in *Eighth International Conference on Signal Image Technology and Internet Based Systems*, 2012, pp. 42–46.

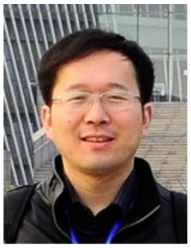
[38] Z. Zheng, H. Zha, T. Zhang, O. Chapelle, K. Chen, and G. Sun, “A general boosting method and its application to learning ranking functions for web search,” in *Advances in Neural Information Processing Systems 20: Proceedings of the 2007 Conference*, 2008, pp. 1697–1704.

[39] J. Elith, J. R. Leathwick, and T. Hastie, “A working guide to boosted regression trees.” *Journal of Animal Ecology*, vol. 77, no. 4, pp. 802–813, 2008.



Xue Li received the B.S. degree in sciences and techniques of remote sensing in 2015 from Wuhan University, Wuhan, China, where she is currently working toward the Ph.D. degree in the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing.

Her research interests include transfer learning in remote sensing images, hyperspectral-image processing, and machine learning.



Bo Du (MâĂŹ10âĂŞSMâĂŹ15) received the B.S. degree and the Ph.D. degree in Photogrammetry and Remote Sensing from State Key Lab of Information Engineering in Surveying, Mapping and Remote sensing, Wuhan University, Wuhan, China, in 2005, and in 2010, respectively.

He is currently a professor with the School of Computer, Wuhan University, Wuhan, China. He was with the Centre for Quantum Computation & Intelligent Systems (QCIS), University of Technology Sydney. He has more than 40 research papers published in the IEEE Transactions on Geoscience and Remote Sensing (TGRS), IEEE Transactions on image processing (TIP), IEEE Journal of Selected Topics in Earth Observations and Applied Remote Sensing (JSTARS), and IEEE Geoscience and Remote Sensing Letters (GRSL), etc. His major research interests include pattern recognition, hyperspectral image processing, and signal processing.

He is currently a senior member of IEEE. He received the best reviewer awards from IEEE GRSS for his service to IEEE Journal of Selected Topics in Earth Observations and Applied Remote Sensing (JSTARS) in 2011 and ACM rising star awards for his academic progress in 2015. He was the Session Chair for the IEEE International Geoscience and Remote Sensing Symposium 2016 and 4th IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS). He also serves as a reviewer of 20 Science Citation Index (SCI) magazines including IEEE TGRS, TIP, JSTARS, and GRSL.

Yipeng Zhang received the B.S. degree in the Electrical Engineering from Wuhan University, Wuhan, China, in 2014. He completed the M.S. degree in the Electrical Engineering in the Syracuse University (SU) in 2016, where he is currently pursuing his Ph.D in computer engineering. His research interest involves the artificial neural network, acceleration algorithm, embedded design, FPGA, VLSI.

Chang Xu is a Lecturer in Machine Learning and Computer Vision at the School of Information Technologies, The University of Sydney. He obtained a Bachelor of Engineering from Tianjin University, China, and a Ph.D. degree from Peking University, China. While pursing his PhD degree, Chang received fellowships from IBM and Baidu. His research interests lie in machine learning, data mining algorithms and related applications in artificial intelligence and computer vision, including multi-view learning, multi-label learning, visual search and face recognition. His research outcomes have been widely published in prestigious journals and top tier conferences.

Dacheng Tao (FâĂŹ15) is Professor of Computer Science and ARC Future Fellow in the School of Information Technologies and the Faculty of Engineering and Information Technologies, and the Inaugural Director of the UBTech Sydney Artificial Intelligence Centre, at The University of Sydney. He mainly applies statistics and mathematics to Artificial Intelligence and Data Science. His research interests spread across computer vision, data science, image processing, machine learning, and video surveillance. His research results have expounded in one monograph and 500+ publications at prestigious journals and prominent conferences, such as IEEE T-PAMI, T-NNLS, T-IP, JMLR, IJCV, NIPS, CIKM, ICML, CVPR, ICCV, ECCV, AISTATS, ICDM; and ACM SIGKDD, with several best paper awards, such as the best theory/algorithm paper runner up award in IEEE ICDMâĂŹ07, the best student paper award in IEEE ICDMâĂŹ13, and the 2014 ICDM 10-year highest-impact paper award. He received the 2015 Australian Scopus-Eureka Prize, the 2015 ACS Gold Disruptor Award and the 2015 UTS Vice-ChancellorâĂŹs Medal for Exceptional Research. He is a Fellow of the IEEE, OSA, IAPR and SPIE.