Relative Forest for Visual Attribute Prediction

Shaoxin Li, Shiguang Shan, *Senior Member, IEEE*, Shuicheng Yan, *Senior Member, IEEE*, and Xilin Chen, *Fellow, IEEE*

***Abstract*— Accurate prediction of the visual attributes is significant in various recognition tasks. For many visual attributes, while it is very difficult to describe the exact degrees of their presences, by comparing the pairs of samples, the relative ordering of presences may be easily figured out. Based on this observation, instead of considering such attribute as binary attribute, the relative attribute method learns a ranking function for each attribute to provide more accurate and informative prediction results. In this paper, we also explore pairwise ranking for visual attribute prediction and propose to improve the relative attribute method in two aspects. First, we propose a relative tree method, which can achieve more accurate ranking in case of nonlinearly distributed visual data. Second, by resorting to randomization and ensemble learning, the relative tree method is extended to the relative forest method to further boost the accuracy and simultaneously reduce the computational cost. To validate the effectiveness of the proposed methods, we conduct extensive experiments on four databases: PubFig, OSR, FGNET, and WebFace. The results show that the proposed relative forest method not only outperforms the original relative attribute method, but also achieve the state-of-the-art accuracy for ordinal visual attribute prediction.**

***Index Terms*— Visual attributes, relative attributes, random forest, relative forest.**

I. Introduction

RECENTLY, there have been growing interests in the visual attributes modeling among computer vision community. The visual attributes (e.g. "stripe"), as mid-level representations closer to the low-level features, can be detected from images more easily than the high-level semantics. As a result, the visual attribute has been employed to boost the performance of various computer vision tasks, such as face recognition [2], human recognition [3], object recognition [4], [5] and scene classification [6]. Moreover, the semantically meaningful visual attributes also facilitate a series of new applications in computer vision, such as image describing [7]–[9] and knowledge transferring [10], [11], which as a return further popularize the study of visual attribute modeling.

Among the works exploiting visual attributes, most of them [2]–[5], [7]–[11] model an attribute as a binary property. For many other attributes, however, binary labeling can hardly tell the whole story. For example, given a pair of smiling faces, it can be more informative to tell which face is smiling bigger than the other. Similar examples include the age or height of a person and the naturalness or openness of a scene. For describing the presence of this kind of attributes, their absolute scores are hard to be estimated or even irrational to be used. But it is not difficult for human beings to tell which sample has higher attribute presence given two samples. Actually, this kind of relativity is very common among the visual attributes.

Based on this observation, Parikh and Grauman [1] propose a relative attribute method to learn the ranking function for visual attribute based on numerous labeled sample pairs. In [1], Parikh et al., as an example, describe a scene of country road as "more manmade than mountain but more natural than tall building" rather than simply categorizing it as natural or manmade, which can provide more information to understand the scene. By satisfying as many as possible relative ordering constraints rather than estimating a simple binary presence, the relative attribute method provides more accurate and informative prediction results that impressively facilitates the subsequent recognition or other tasks. However, we doubt that the capability of the relative attribute method might be limited in handling complex nonlinear data, because it learns only a simple linear hyperplane as the ranking function. Moreover, as the exact optimization of the relative attribute method in [1] is very time-consuming, it might meet difficulty in case of large scale applications.

In this paper, to tackle the above two limitations of the relative attribute method, we propose two highly related contributions: 1) we propose a relative tree method to achieve nonlinear ranking instead of linear ranking as in [1]; 2) the relative tree method is extended to the relative forest method to further improve prediction accuracy and efficiency. An overview of the proposed method is shown in Fig. 1.

First, considering that many attributes cannot be ranked linearly, we propose to improve the relative attribute method by using nonlinear ranking. For this purpose, we construct a tree, called by us relative tree, that employs the original relative attribute algorithm as the base ranking function.

Manuscript received October 2, 2013; revised June 10, 2014 and December 21, 2014; accepted April 6, 2016. Date of publication June 14, 2016; date of current version July 1, 2016. This work was supported in part by the 973 Program under Contract 2015CB351802, in part by the Natural Science Foundation of China under Contract 61390511 and Contract 61402443, and in part by the Strategic Priority Research Program through the Chinese Academy of Sciences under Grant XDB02070004. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Peter Tay. *(Corresponding author: Shiguang Shan.)*

S. Li is with the Tencent YouTu Lab, Tencent Shanghai, 200233, China (e-mail: darwinli@tencent.com).

S. Shan and X. Chen are with the Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China (e-mail: sgshan@ict.ac.cn; xlchen@ict.ac.cn).

S. Yan is with the 360 AI Institute, Beijing 100015, China (e-mail: yanshuicheng@360.cn).

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

Digital Object Identifier 10.1109/TIP.2016.2580939

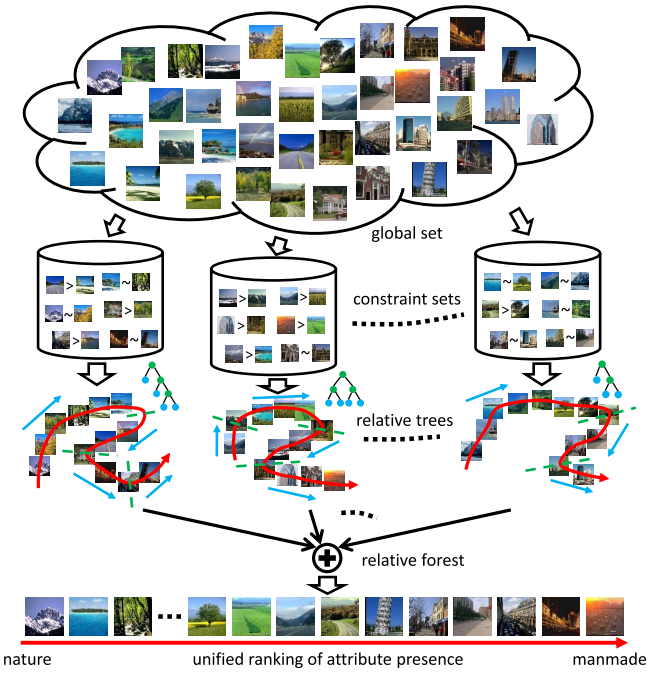


Fig. 1. Overview of proposed relative forest method. In a large scale global dataset, multiple relative trees are learned in an extremely randomized manner to collect a large number of ranking cues for the final prediction of attribute with relative forest (here we take "nature or manmade" attribute as an example).

Concretely, in the recursive procedure of tree construction, for each node of the tree, we first learn a ranking function by the original relative attribute method and then split the training samples into child nodes according to their relative scores. In such a hierarchical structure, the tree actually achieves a globally nonlinear ranking by decomposing gradually finer and finer piecewise linear ranking functions.

Second, inspired by the idea of random forest [12], we propose to learn multiple randomized relative trees to simultaneously reduce the computational cost of model training and improve the prediction accuracy. Specifically, we first quickly learn multiple relative trees by resorting to randomized learning technique to reduce the time cost of building a single tree. Then, these trees are combined and form a forest, named by us relative forest, to further boost the accuracy.

Comprehensive evaluations are conducted on four databases, i.e. OSR [13], PubFig [2], FGNET aging [14] and WebFace [15], [16] databases. The aforementioned two contributions lead to very promising improvements over the relative attribute method [1] and the random forest based methods [12], [17], as well as other state-of-the-art methods [18], [19] on the corresponding databases. These results validate the advantage of using relative modeling for the visual attributes: 1) wider application range , i.e. applicable when only relative labels are available; 2) higher prediction accuracy even in scenario that absolute ranking labels are available. This may be because that the visual attributes usually have "inhomogeneous" changes. Taking facial age as an example, the face of a teenager changes quickly while that of a mid-aged adult stays unchanged for years. As a result, even absolute age labels are available for training, modeling facial age as a relative attribute can still be beneficial (see TABLE IV and V).

Besides the above two contributions, we also carefully study the different roles of two kinds of relative constraints, i.e. similar constraints and ordered constraints, in the training of relative attribute. Taking age estimation as example, we found that similar constraints (e.g. person A and person B have about the same age) can be more crucial for better accuracy, if there are already enough ordered constraints (e.g. person A is older than person B). Based on this observation and also inspired by [20], we design a strategy to collect numerous similar constraints from rich video context and show how they can impressively boost the estimation accuracy of age estimation. It is worth pointing out that, although we only take age estimation as example, similar strategy should be also useful for training relative attribute for other visual attributes.

The rest of the paper is organized as follows. Section II gives a brief review of the related works. Section III validates the relative tree algorithm. Section IV proposes the relative forest algorithm followed by comprehensive experimental evaluation in Section V. Finally, we conclude this work and discuss about further extension in Section VI.

II. Related Work

In this section, we review the related works on attribute prediction and random forest.

A. Attribute Prediction

Before the relative attribute method was proposed, most attribute modeling approaches [2], [3], [5], [7]–[11] treat the visual attribute as a binary property and learn a binary classifier for each attribute. The most commonly used learning method is binary SVM. Although binary prediction of visual attributes also provides valuable mid-level cue for various vision tasks, the binary description of attribute fails to depict the degree of presence of some attribute. In other words, many attributes are unsuitable to describe in binary mode. So, the relative attribute method [1] was proposed. Its basic idea is to learn a linear ranking function to maximize rank margin between given pairs of samples with ordered constraints. Similar idea was first introduced in web page ranking and information retrieval as a pairwise learning-to-rank approach [21], [22], called RankSVM. Formally , to learn RankSVM, one needs a training set $\mathscr{X} = \{x_i \in \Re^d | i = 1, 2, \ldots, N\}$, where x_i is the feature of the i-th training sample. Accompanying the set $\mathscr{X}$, one also needs an ordered (or preference) constraint set $\mathscr{O} \subseteq \mathscr{X} \times \mathscr{X} = \{(x_i, x_j) | x_i \succ x_j\}$ where $x_i \succ x_j$ denotes the relative score of the attribute presence of sample x_i is larger than that of sample x_j. Then, RankSVM aims at learning a ranking function $h(x) = W^T x$, so that for any ordered constraint $(x_i \succ x_j) \in \mathscr{O}$, $h(x_i) > h(x_j)$. For this purpose, the objective of RankSVM is formulated as:

$$\min_W \frac{1}{2} W^T W + C_{\mathscr{O}} \sum_{(x_i, x_j) \in \mathscr{O}} max(0, 1 - W^T (x_i - x_j))^p. \quad (1)$$

where $C_{\mathscr{O}}$ is the weight for trade-off between maximizing rank margin and satisfying ordered constraints. Intuitively, besides the ordered constraint pairs, there should be a lot of sample pairs whose degrees of attribute presence are indistinguishable. They form the similar constraint pairs, which also provide useful information for attribute modeling. Formally, a similar constraint set is denoted as $\mathscr{O} \subseteq \mathscr{X} \times \mathscr{X} = \{(x_i, x_j) | x_i \sim x_j\}$, where $(x_i \sim x_j)$ indicates the relative score of the attribute presence of sample x_i is very similar to that of sample x_j. Obviously, a favorable ranking function $h(x)$ should generate similar relative scores for the sample pairs in $\mathscr{S}$. Therefore, the relative attribute method [1] combines ordered and similar constraints to learn the ranking function by optimizing:

$$\min_{W} \frac{1}{2} W^T W + C_{\mathscr{S}} \sum_{(x_i, x_j) \in \mathscr{S}} |W^T (x_i - x_j)|^p + C_{\mathscr{O}} \sum_{(x_i, x_j) \in \mathscr{O}} max(0, 1 - W^T (x_i - x_j))^p. \quad (2)$$

where $C_{\mathscr{S}}$ and $C_{\mathscr{O}}$ are the weights for similar and ordered constraints respectively. Although not emphasized in [1], similar constraints are actually crucial for relative attribute learning, as will be seen in our experiments. By considering two kinds of relative information, i.e. similar and ordered constraints, relative attribute Eqn. (2) can result in much more informative description of attribute presence.

The optimization of the relative attribute in Eqn. (2) is very similar to that of the RankSVM, i.e., Eqn. (1). As illustrated in [22] and [23], a linear SVM solver can be directly used to train a linear RankSVM by simply constructing a new training set consisting of difference vectors $x_i - x_j$. This is also true for the training of relative attributes. However, as the new training set can be very large, the direct adaptation of an SVM classifier can be very slow. Therefore, several techniques have been proposed to accelerate the training process of RankSVM, such as the structure SVM learning framework proposed by Joachims [24] and the primal RankSVM proposed by Chapelle and Keerthi [23]. In [1], the primal RankSVM method is exploited to optimize the relative attribute. Its complexity is $O(Nd + C)$, where $C = |\mathscr{O}| + |\mathscr{S}|$ is the total constraint number. In the worst case, $C = N(N-1)/2$. In practice, when the training set is very large, its training process might still be very time-consuming.

B. Random Forest

The relative forest proposed later in this paper can be also derived from the seminal work [12] named random forest. Therefore, here we briefly introduce this technology.

Generally, the random forest can be viewed as a framework for assembling multiple tree-structured classifiers. In order to benefit from multiple classifiers' ensemble, the independence and complementarity of different classifiers is crucial. As a result, the key ingredient of random forest [12] is to diversify different tree-structured classifiers by injecting a variety of randomization in the tree construction process. In Breiman's original algorithm, mainly two kinds of randomization are injected into the tree construction process: 1) each tree is trained on a bootstrapped subset of the training samples; 2) for each non-leaf node, a decision function is made based on a subset of feature dimensions. Other randomization is also introduced in the literature, for example in [25], authors proposed to use random rather than optimal split in the non-leaf nodes.

Due to superior generalization ability and efficiency, random forest and its extensions are widely used in various computer vision tasks, such as image segmentation [26], human pose estimation [27] and object classification [28]. Random forest is also one of the most promising method for web-page ranking task. In Yahoo Learning to Rank competition [29], almost all of the winning teams used random forest in one form or another.

Our method is also an extension of random forest, the main difference is that the relative attribute method Eqn. (2) is employed as the decision function. As additional randomization over relative constraints can be injected into the random forest framework, we find relative attribute and random forest can benefit from each other and the combination of them generates a very powerful visual attribute prediction method.

III. Relative Tree

In this section, we first present how to construct a relative tree to achieve nonlinear ranking by employing the original relative attribute method [1] to learn the splitting function for each tree node. Then we describe how to predict the attribute presence of a test sample with the constructed relative tree.

A. Tree Construction

Formally, the training data consists of three parts, i.e. the set of training samples $\mathscr{X}$ and the corresponding sets of the ordered constraints $\mathscr{O} \subseteq \mathscr{X} \times \mathscr{X} = \{(x_i, x_j) | x_i \succ x_j\}$ and the similar constraints $\mathscr{S} \subseteq \mathscr{X} \times \mathscr{X} = \{(x_i, x_j) | x_i \sim x_j\}$. Given such triple training data $(\mathscr{X}, \mathscr{O}, \mathscr{S})$, the relative attribute method learns single linear transformation function W with Eqn. (2) as prediction function,

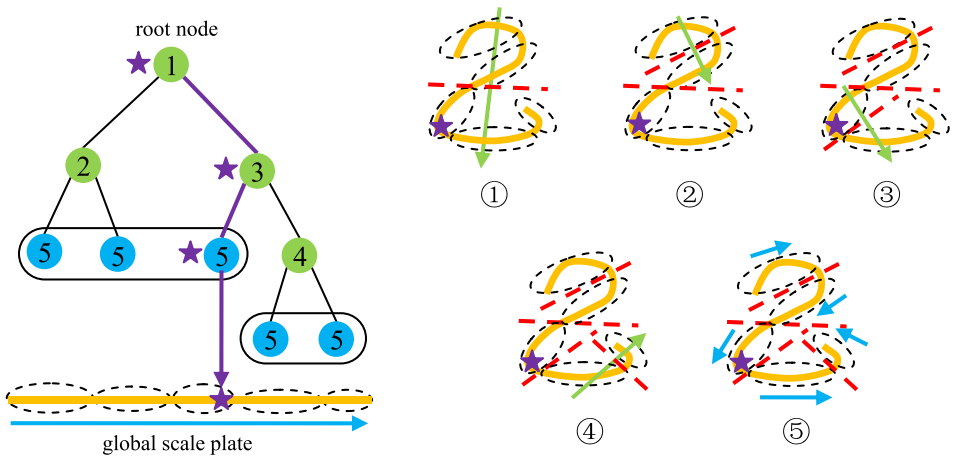
$$W = relative\_attribute(\mathscr{X}, \mathscr{O}, \mathscr{S}). \quad (3)$$

This visual attribute prediction function W is a simple linear function. Thus, the corresponding relative scores of training data can be derived from simple linear projection of training features,

$$R = W^T X. \quad (4)$$

Note, X is the features of training samples $\mathscr{X}$.

Typically, this simple linear function cannot handle complex non-linear data. For clarity, we present a toy sample of learning relative prediction function on an "S-shape" manifold as shown in Fig. 2. Ideally, the top left end of the "S-shape" manifold should have the highest rank and the bottom right end of the "S-shape" manifold should have the lowest rank. As can be seen from Fig. 2(b): ①, the ranking function learned in the root node (see Fig. 2(a)) is actually the result of the relative attribute method [1]. Unfortunately, this prediction function gives an almost lowest relative ranking score to the exemplified test sample indicated by the purple star, which is



(a) relative tree

(b) hierarchical ranking functions

Fig. 2. Illustration of the learning procedure of relative tree (a) on an "S-shape" nonlinear manifold and its corresponding set of hierarchical ranking functions (b). Circled numbers indicate the index of the training step. Nodes in green are non-leaf nodes while those in blue are leaf nodes. Local ranking functions of all leaf nodes are aligned to a global piece-wise linear ranking function in order that unified relative scores can be obtained at different leaf nodes. (better viewed in color)

however unacceptable (if not completely wrong). To address this problem, our relative tree automatically learns finer and finer local ranking functions according to the intrinsic structure of the data manifold.

As shown in Fig. 2(a), the relative tree method achieves nonlinear ranking via a hierarchical tree structure. And the learning process is presented in Fig. 2(b). Briefly speaking, the linear splitting function W is iteratively learned via the relative attribute method in each tree node and the training data of each tree node is split to its child nodes according to the relative scores R. As shown Fig. 2(b), the "S-shape" nonlinear manifold is finally divided into 5 near-linear segments, each of which is circled in a dotted ellipse. As a result, the purple star can be ranked, i.e. test sample, more accurately.

For easy understanding, we summarize the construction procedure of the relative tree with pseudo code in Algorithm 1. It is worth pointing out that in this algorithm, we assume the training data is organized in the type of ordered and similar pairs. In some scenarios, rather than this kind of pairwise constraints, the ordinal attribute label (e.g. the age) of each single sample in the training set might be available. In this case, $\mathscr{O}$ and $\mathscr{S}$ can be easily generated by exploiting the ordinal labels, which implies Algorithm 1 can be applied without difficulty.

Overall speaking, the algorithm is easy to follow except two points needing further explanation: 1) how to determine whether the current node is divisible or not (Algorithm 1, Line 4); 2) how to find the optimal splitting threshold (Algorithm 1, Line 10). We detail them in the following.

1) Divisibility Determination: Clearly, the divisibility of a node depends on the diversity of the training samples for the node. According to the label type of the training samples, we have two different strategies determining the divisibility. The first one corresponds to the case that only pairwise constraints (ordered and similar) are available in the training set. Here, we simply exploit the number of ordered constraints

Algorithm 1 Tree Construction

Input:
Training sample set: $\mathscr{X}$
Ordered constraint set: $\mathscr{O}$
Similar constraint set: $\mathscr{S}$
Output:
Relative tree: T

1: $T \Leftarrow (\mathscr{X}, \mathscr{O}, \mathscr{S})$
2: $priority\_queue.push(T)$
3: $N = priority\_queue.top()$
4: **while** $isDivisible(N)$ **do**
5: $priority\_queue.pop()$
6: $(\mathscr{X},\ \mathscr{O},\ \mathscr{S}) = (N.\mathscr{X},\ N.\mathscr{O},\ N.\mathscr{S})$
7: $W = relative\_attribute(\mathscr{X},\ \mathscr{O},\ \mathscr{S})$
8: $R = W^T X$
9: $b = optimal\_split(R,\ \mathscr{O},\ \mathscr{S})$
10: $N \Leftarrow (W,\ R,\ b)$
11: Split training data, for each $x_i \in \mathscr{X}$:

$$x_i \in \begin{cases} \mathscr{X}^{(L)} & if\ R_i \le b \\ \mathscr{X}^{(R)} & if\ R_i > b \end{cases}$$

12: Split relative constraints, for each $(x_i, x_j) \in \mathscr{O}\ or\ \mathscr{S}$:

$$(x_i, x_j) \in \begin{cases} \mathscr{O}^{(L)}\ or\ \mathscr{S}^{(L)} & if\ x_i, x_j \in \mathscr{X}^{(L)} \\ \mathscr{O}^{(R)}\ or\ \mathscr{S}^{(R)} & if\ x_i, x_j \in \mathscr{X}^{(R)} \end{cases}$$

13: $N.left\_node \Leftarrow (\mathscr{X}^{(L)}, \mathscr{O}^{(L)}, \mathscr{S}^{(L)})$
14: $N.right\_node \Leftarrow (\mathscr{X}^{(R)}, \mathscr{O}^{(R)}, \mathscr{S}^{(R)})$
15: $priority\_queue.push(N.left\_node)$
16: $priority\_queue.push(N.right\_node)$
17: $N = priority\_queue.top()$
18: **end while**
19: **return** T

$|\mathscr{O}|$ as the diversity measurement, and an empirical threshold is used to make the division decision. The second strategy is only applicable to the case that the ordinal attribute label of

each sample is available. In this scenario, we make use of the entropy of ordinal labels as the divisibility measurement. Then, a threshold is empirically set to determine the division of the node. Note, during the construction of relative tree, we maintain a $priority\_queue$ according to the divisibility and the $priority\_queue.top()$ operation would return the most divisible node in the queue.

2) Splitting Objective: Once a node is determined as divisible, we then need to split the triple training data of it appropriately. To achieve this goal, we first apply the original relative attribute method, i.e. Eqn. (2), to learn a linear ranking function. Then, similar to previous discussions, we again design different strategies for two distinct types of training set, as described in the following.

In the case that the ordinal attribute labels are available for each training sample, similar to [26], we look for the optimal splitting threshold b based on maximal information gain criterion:

$$\max_b \ \Delta H(b) = H(Y) - \frac{|\mathscr{X}^{(L)}|}{|\mathscr{X}|} H(Y^{(L)}) - \frac{|\mathscr{X}^{(R)}|}{|\mathscr{X}|} H(Y^{(R)}), \tag{5}$$

where $\mathscr{X}/\mathscr{Y}$, $\mathscr{X}^{(L)}/\mathscr{Y}^{(L)}$ and $\mathscr{X}^{(R)}/\mathscr{Y}^{(R)}$ indicate the set of samples/labels that are partitioned to the parent, the left child and the right child nodes respectively. Function $H(\cdot)$ indicates the Shannon entropy of the set and the operator $|\cdot|$ indicates the size of the set.

As for the case that only relative pairwise constraints, i.e. $\mathscr{O}$ and $\mathscr{S}$, are available, we determine the optimal splitting threshold b which separates the maximal number of the ordered pair and maintains the maximal number of the similar pair:

$$\max_b \ \Delta I(b) = I(\mathscr{O}, \mathscr{S}) - \frac{|\mathscr{X}^{(L)}|}{|\mathscr{X}|} I(\mathscr{O}^{(L)}, \mathscr{S}^{(L)}) - \frac{|\mathscr{X}^{(R)}|}{|\mathscr{X}|} I(\mathscr{O}^{(R)}, \mathscr{S}^{(R)}), \tag{6}$$

where $I(\mathscr{O}, \mathscr{S}) = |\mathscr{O}| - |\mathscr{S}|$. Although more complex and accurate approximation of entropy $H(\cdot)$ may be obtained based on $\mathscr{O}$ and $\mathscr{S}$. In this paper, we simply use this heuristic splitting objective given by Eqn. (6).

3) Complexity Analysis: Although the relative attribute method needs to be learned in each tree node, the time complexity of the proposed relative tree is acceptable. As mentioned in the last paragraph of Section II-A, the time complexity of the relative attribute is $O(N^2)$. For the proposed relative tree, if we get a complete binary tree, the complexity of the relative tree is still $\sum_{i=0}^{\log_2^N} 2^i O((\frac{N}{2^i})^2) = O(N^2)$. As will be presented in the experiments, our method indeed tends to choose balanced split, and the training time of relative tree is approximately 3 or 4 times of relative attribute.

4) Discussion: As mentioned previously, our relative tree method aims at nonlinear ranking. One may doubt why not directly apply the kernel trick to Eqn. (2) for nonlinear ranking. Indeed, kernel trick might be a possible way to achieve nonlinearity. However, kernel trick has two drawbacks in comparison with our solution. Firstly, each type of kernel function actually implies some form of data distribution. For example, RBF kernel assumes Gaussian distribution. The data distribution however is hard to estimate. In contrast, a set of tree structured projections learned in a data-driven manner, as in our method, can automatically adapt to nonlinear manifold structure without involving any hypothesis about data distribution, as proven in [30]. Intuitively, the nonlinear ranking is achieved by a manner of piecewise linear ranking along the data manifold. Secondly, the scalability of kernel method is relatively poor when facing a large number of training samples, because all the training samples have to be loaded to compute the kernel matrix in the testing stage. On the contrary, the proposed relative tree has the same asymptotic computational complexity as the linear relative attribute method, as analyzed previously.

B. Tree Prediction

After introduce the construction process of relative tree, in this subsection, we elaborate how to predict the ordinal label of a test sample (see Fig. 2(a)), based on the constructed relative tree. This problem is not trivial, since the output of the leaf-node ranking function, as a local score rather than expected global one, cannot be used directly. As before, we again need to differentiate two kinds of training set, i.e. ordinal labels are available or only relative pairwise constraints are provided.

In the former case, as the ordinal attribute labels Y are available for all training samples, the ordinal label of any test sample can be set as the average of the labels of all the samples belonging to the leaf node that the test sample falls into. This is a normal method exploited in typical decision tree method. Formally, if we suppose the test sample x_t falls into the j-th leaf node of the relative tree T after traversing, its ordinal label $\overline{y}_t$ can be estimated as follows:

$$\overline{y}_t = \frac{1}{|N_j|} \sum_{k \in N_j} y_k, \tag{7}$$

where N_j indicates the index set of training samples that fall into the j-th node, the $|N_j|$ is the number of training samples in the j-th node, and y_k is the ground truth ordinal label of the k-th training sample.

Eqn.(5) makes use of all the training samples falling to the leaf node, which is not necessarily good enough for the test sample especially for a large leaf node. Therefore, in this paper, we slightly modify the prediction method by averaging only the labels of the training samples that are close to the test sample in terms of the output score of the ranking function of that leaf node. Formally, assuming that a test sample x_t fall into the j-th leaf node, we denote its relative scores as r_t, and the relative scores of the training samples belonging to N_j as $r_k,\ k \in N_j$. We predict the ordinal label of the test sample as follows:

$$\overline{y}_t = \frac{1}{|\Omega_{jt}|} \sum_{k \in \Omega_{jt}} y_k, \tag{8}$$

where $\Omega_{jt} \subseteq N_j$ contains the nearest neighbor training samples of the test sample in terms of the ranking scores generated by the ranking function of this leaf node.

As for the second case, i.e., only ordered or similar constraints are available, the technique for the first case will fail, as no ordinal labels can be used to average. In this case, what we have are only the relative scores R given by different leaf-node ranking functions. The problem is the relative scores R output by different leaf-node ranking functions are NOT compatible, as they are learned to satisfy only constraints ranked locally. In other words, these local ranking functions are not globally aligned. To address this problem, we need to align or normalize the outputs of these local ranking functions to achieve globally compatible ranking. Formally, given a arbitrary set $\mathscr{R}$ of relative scores, we need to map the values of them to a target range $[b_{inf}, b_{sup}]$. For this purpose, we adopt a simple yet effective linear mapping strategy,

$$\widehat{r} = b_{inf} + (\frac{r - r_{inf}}{r_{sup} - r_{inf}} - b_{inf})/(b_{sup} - b_{inf}). \quad (9)$$

Note, r is the original value of the relative score and $\widehat{r}$ is the corresponding mapped value in the target range. For the stability consideration, the α and $1-\alpha$ percentile of the relative scores r_{inf} and r_{sup} of set $\mathscr{R}$ are mapped to the boundary values of the target range, i.e. b_{inf} and b_{sup},

$$r_{inf} : \ |\{r_k | r_k \in \mathscr{R}, \ r_k < r_{inf}\}| = \alpha|\mathscr{R}|, \quad (10)$$

$$r_{sup} : \ |\{r_k | r_k \in \mathscr{R}, \ r_k < r_{sup}\}| = (1-\alpha)|\mathscr{R}|. \quad (11)$$

Note, we empirically set α as 5% in this work.

More specifically, given a relative tree, we first manually set the global range of root node as [0, 1]. Then, for each non-root node, a sub-range is automatically assigned according to the range and splitting threshold of its parent node. We summarize the procedure about how to globalize the relative scores of non-root nodes in Algorithm 2.

With the score globally aligned, the proposed relative tree can generate unified relative scores even with only pairwise ordered and similar constraints, thus achieving adaptive piecewise linear ranking for nonlinear data.

Algorithm 2 Tree Globalization Without Ordinal Label

Input:
Relative tree: T
Output:
Globalized tree: T, with globalized parameters, i.e. b_{inf}, b_{sup}, r_{inf}, r_{sup}, for every tree node N

```
Initialize the global range T.b_inf = 0, T.b_sup = 1
queue.push(T)
N = queue.top()
while N != NULL do
    queue.pop()
    (b_inf, b_sup, R, b) = (N.b_inf, N.b_sup, N.R, N.b)
    Calculate percentile r_inf and r_sup with Eqn. (10)&(11)
    N <= (r_inf, r_sup)
    Globalize threshold b -> b_hat with Eqn. (9)
    if N.left_node != NULL then
        Set target range of left node:
            N.left_node <= (b_inf, b_hat).
        queue.push(N.left_node)
    end if
    if N.right_node != NULL then
        Set target range of right node:
            N.right_node <= (b_hat, b_sup).
        queue.push(N.right_node)
    end if
    N = queue.top()
end while
return T
```

IV. Relative Forest

Although the relative tree method achieves more accurate ranking on nonlinear manifold compared to the relative attribute method, it may suffer from over-fitting and is more time-consuming. In this section, we propose the relative forest method based on the relative tree to address these problems.

A. Randomized Learning

Similar to random forest, we also resort to randomization to build multiple relative trees, which are then combined together to achieve accuracy and efficiency. In the original random forest algorithm [12], randomization is injected in two ways: 1) bootstrap a subset of training data to grow a random tree and 2) choose a subset of features to learn a splitting function f. In the proposed relative forest, we also employ the two randomization strategies.

Moreover, we further implement randomization on selecting subset of pairwise constraints. Especially, in the case that the ordinal labels are given for all training samples, instead of using all possible ordered pairwise constraints, we randomly select a subset of constraints. Specifically, given N samples with ordinal labels, we randomly select only $2N$ or $4N$ pair constraints from all the possible $O(N^2)$ pairs, to train the node splitting functions of relative trees with the relative attribute method based on Eqn. (2). As the optimization of relative attribute method is of $O(Nd + C)$ complexity, where C is the total number of relative constraints, this randomization directly ensures the linear time complexity of optimizing the relative attribute in each tree node. Empirically, $2N$ or $4N$ constraints are also sufficient for training. Therefore, to balance speed and accuracy, $2N$ or $4N$ constraints can be selected safely. This property makes the relative attribute method especially suitable for learning the splitting function in the randomized relative tree.

B. Tree Ensemble

As the randomization diversifies the learned relative trees, the ensemble of multiple randomized relative trees is expected to significantly improve the ranking accuracy. As shown in Fig. 3, given the relative scores of M relative trees, the prediction of relative forest can be calculated as:

$$\overline{y}_t = \frac{1}{M} \sum_{i=1}^{M} \overline{y}_{ti}, \quad (12)$$

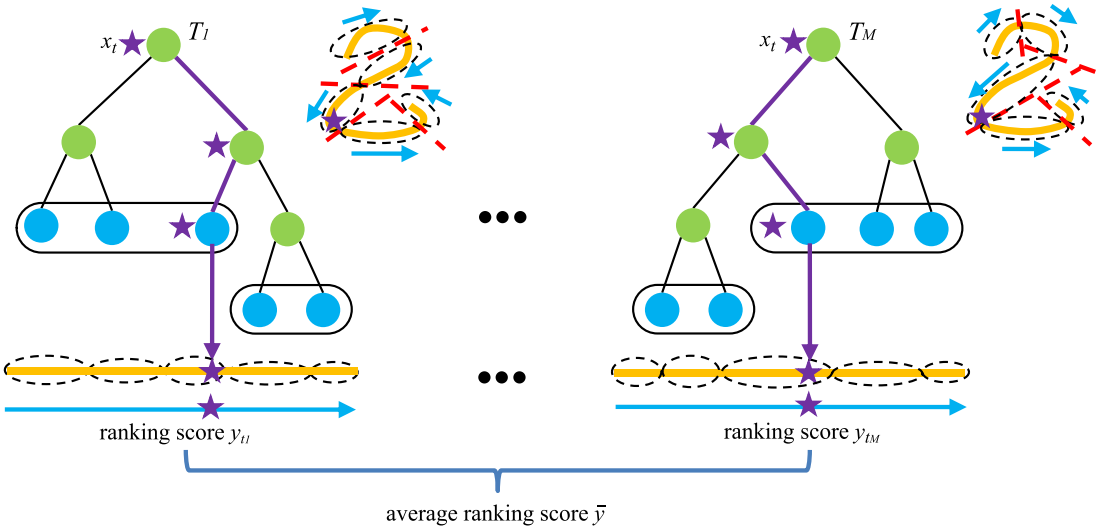


Fig. 3. Relative trees' Ensemble. Due to randomized learning, different trees may learn different hierarchical ranking functions. We average the ranking score of n random relative trees to produce final ranking results of relative forest.

where $\overline{y}_{ti}$ indicates the ranking score prediction results of the i-th relative tree. Besides averaging tree predictions with the same weight, i.e. $\frac{1}{M}$ in Eqn. (12), a more effective weighted average of tree predictions can be conducted when the ordinal label Y is available for training samples. The basic idea is giving higher weights to more reliable tree predictions. As only partial training samples are bootstrapped to train each relative tree, the rest of the samples (so called out-of-bag samples) can be used to evaluate the reliability of relative trees. Since the predictions of relative tree are actually made in the leaf node, the reliability of the predictions actually depend on the leaf node's "quality". Suppose the j-th leaf node of the i-th relative tree T_i is represented as N_{ij} and the corresponding index set of out-of-bag samples that fall in N_{ij} is represented as $\mathbf{\Omega}_{ij}$. Then we formulate the reliability weight ω_{ij} of the leaf node N_{ij} as:

$$\omega_{ij} = \Sigma_{k \in \mathbf{\Omega} ij} P(\overline{y}_k | y_k), \tag{13}$$

where $\overline{y}_k$ is the predicted ranking score of the k-th out-of-bag sample, which receives the prediction at the leaf node N_{ij}, and y_k is the ground truth ordinal label of the k-th out-of-bag sample. $P(\overline{y}_k | y_k)$ measures the "quality" of the prediction. For example, in facial age estimation scenario, we assume the predicted relative scores are distributed as a Gaussian with respect to the ground truth ordinal label y_k, i.e. $\overline{y}_k \sim \mathcal{N}(y_k, \sigma^2)$. In this paper, the variance of the Gaussian distribution σ is empirically set as 5 for age estimation. Intuitively, if a large number of out-of-bag samples receive accurate prediction in a leaf node, this leaf node should be more reliable. Once the weights of all leaf nodes are calculated, given a test sample x_t, for the i-th relative tree T_i, suppose x_t will fall in (j_{ti})-th leaf node $N_{i(j_{ti})}$, then the weighted average of relative trees' predictions can be formulate as:

$$\overline{y}_t = \frac{\Sigma_{i=1}^{M} \omega_{i(j_{ti})} \cdot \overline{y}_{ti}}{\Sigma_{i=1}^{M} \omega_{i(j_{ti})}}. \tag{14}$$

TABLE I
ATTRIBUTE ASSIGNMENTS OF OSR AND PUBFIG DATABASE

	Binary	Relative
OSR	T I S HC OMF	
natural	0 0 0 0 1 1 1 1	T≺I∼S≺H≺C∼O∼M∼F
open	0 0 0 1 1 1 1 0	T∼F≺I∼S≺M≺H∼C∼O
perspective	1 1 1 1 0 0 0 0	O≺C≺M∼F≺H≺I≺S≺T
large-objects	1 1 1 0 0 0 0 0	F≺O∼M≺I∼S≺H∼C≺T
diagonal-plane	1 1 1 1 0 0 0 0	F≺O∼M≺C≺I∼S≺H≺T
close-depth	1 1 1 1 0 0 0 1	C≺M≺O≺T∼I∼S∼H∼F
PubFig	ACHJ MS V Z	
Masculine-looking	1 1 1 1 0 0 1 1	S≺M≺Z≺V≺J≺A≺H≺C
White	0 1 1 1 1 1 1 1	A≺C≺H≺Z≺J≺S≺M≺V
Young	0 0 0 0 1 1 0 1	V≺H≺C≺J≺A≺S≺Z≺M
Smiling	1 1 1 0 1 1 0 1	J≺V≺H≺A∼C≺S∼Z≺M
Chubby	1 0 0 0 0 0 0 0	V≺J≺H≺C≺Z≺M≺S≺A
Visible-forehead	1 1 1 0 1 1 1 0	J≺Z≺M≺S≺A∼C∼H∼V
Bushy-eyebrows	0 1 0 1 0 0 0 0	M≺S≺Z≺V≺H≺A≺C≺J
Narrow-eyes	0 1 1 0 0 0 1 1	M≺J≺S≺A≺H≺C≺V≺Z
Pointy-nose	0 0 1 0 0 0 0 1	A≺C≺J∼M∼V≺S≺Z≺H
Big-lips	1 0 0 0 1 1 0 0	H≺J≺V≺Z≺C≺M≺A≺S
Round-face	1 0 0 0 1 1 0 0	H≺V≺J≺C≺Z≺A≺S≺M

* this table is provide by the authors of relative attribute [1].

where $\omega_{i(j_{ti})}$ is the weight of the leaf node $N_{i(j_{ti})}$, and $\overline{y}_{ti}$ is the ranking score of test sample x_t generated by the i-th relative tree. Note $\overline{y}_{ti}$ is predicted by the (j_{ti})-th leaf node $N_{i(j_{ti})}$ of relative tree T_i.

Note the only difference between Eqn. (12) and Eqn. (14) is the combination weights. Hereafter, we term the proposed method as simple or weighted relative forest if Eqn. (12) or Eqn. (14) is used to integrate multiple trees' predictions. As will be shown, weighted relative forest can achieve better performance compared with simple relative forest.

V. EXPERIMENTS

In this section, we conduct extensive evaluations on four databases: 1) Outdoor Scene Recognition (OSR) database [13]

TABLE II
EVALUATION ON OSR DATASET (%)

	Natural	Open	Perspective	Large	Diagonal	Close
Binary SVM [31]	91.10	86.06	78.98	65.01	80.77	87.20
SVR [31]	94.01	90.42	85.51	86.17	86.87	87.34
Random forest [12]	94.12	90.57	86.64	86.05	89.11	87.84
iGBRT [17]	94.93	91.18	87.40	87.47	90.16	88.79
Relative attribute [1]	94.40	91.01	85.08	86.39	87.52	88.71
Relative tree	95.24	92.39	87.58	88.34	89.43	89.54
Relative forest	96.25	93.14	88.04	88.64	90.64	90.67

TABLE III
EVALUATION ON PUBFIG DATASET (%)

	Masculine	White	Young	Smiling	Chubby	Forhead
Binary SVM [31]	70.12	64.64	75.49	66.97	59.37	76.50
SVR [31]	75.36	69.98	76.25	76.58	72.34	85.79
Random forest [12]	81.30	78.01	80.18	78.63	73.84	87.16
iGBRT [17]	83.63	79.33	81.30	80.64	75.59	87.47
Relative attribute [1]	81.00	77.31	81.05	79.66	76.14	87.91
Relative tree	85.33	82.59	84.41	83.36	78.97	88.83
Relative forest	85.50	83.62	84.19	83.92	81.04	91.16
	Eyebrow	Eye	Nose	Lips	Face	
Binary SVM [31]	69.05	74.90	66.29	74.52	74.25	
SVR [31]	75.22	77.08	69.14	71.86	74.23	
Random forest [12]	80.96	80.97	72.43	76.31	78.83	
iGBRT [17]	82.10	81.93	74.20	78.53	80.50	
Relative attribute [1]	78.89	80.72	74.84	78.07	80.46	
Relative tree	81.84	83.15	80.43	81.87	86.31	
Relative forest	82.94	84.97	82.08	83.65	86.16	

which contains $2,688$ images of 8 categories; 2) a subset of Public Figure Face (PubFig) database [2], which contains 800 images of 8 random identities (100 images each); 3) FGNET face aging database [14] which contains $1,002$ images of 82 individuals; and 4) WebFace database [16] which contains $55,930$ face images.

In Section V-A, we first conduct evaluations on OSR and PubFig databases. Then age estimation as a typical ordinal attribute prediction problem is further evaluated on FGNET and WebFace databases in Section V-B. After that, we comprehensively analyze the effects of key parameters of the relative forest method in Section V-C. Finally, some preliminary results are presented to provide some insight on how to cheaply collect relative constraints for visual attribute prediction.

A. Evaluation on OSR and PubFig Database

For OSR and PubFig databases, we conduct experiments with the same feature and train\test partitions used in [1]. The attributes assignments used in [1] and this work are shown in Table I, where the characters in the middle column of the table denote different scenes (for OSR) or individuals (for PubFig). We use the data provided by authors of [1], i.e. 512-dimensional gist feature [13] for OSR and gist plus 45-dimensional Lab color histogram feature for PubFig, to conduct experiments. For each image pair (i, j) , if the predicted relative scores $\overline{y}_i$ and $\overline{y}_j$ have the same order as the ground-truth relative score, this pair is considered to be ranked correctly. The overall prediction accuracy of all the ordered image pairs in the test set ($2,448$ images for OSR, 560 images for PubFig) is shown in Table II (OSR) and Table III (PubFig). Note, in the OSR and PubFig databases, the ground truth ranking labels of the attributes "Natural" scene, "Smiling" and "Chubby" face et al. are not available and also not applicable. A trivial solution is assigning hard ranking levels for each attribute according to the relative relationships. For example, for the "Natural" attribute, the relative relationships are $T < I \sim S < H < C \sim O \sim M \sim F$. We assign the hard ranking labels of them as $T = 1$, $I = 2$, $S = 2$, $H = 3$, $C = 4$, $O = 4$, $M = 4$, $F = 4$ to obtain corresponding results of SVR [31], random forest [12] and iGBRT [17]. As shown, the proposed relative tree and relative forest methods not only outperforms simple baseline methods, namely binary SVM, SVR, relative attribute and random forest, but also achieves better performance than the state-of-the-art random forest based ranking method [17] in almost all given attribute prediction tasks. For random forest method [12] and initialized Gradient Boosted Regression Trees (iGBRT) method [17], we use the code provided by the author of [17] to generate the results presented in Table II and III.

Using the relative scores generated by the attribute prediction models as features, we also conduct recognition experiments on OSR and PubFig databases with the nearest neighbor classifier. The results are shown in Fig. 4. As shown, the relative scores generated by the models with relative modeling (i.e. Relative attribute, Relative tree and Relative forest) are better for conducting recognition tasks compared to the models trained without relative modeling (i.e. Binary SVM, SVR, Random forest and iGBRT). Even though iGBRT achieves promising results in the attribute prediction tasks, its prediction scores are less effective for the recognition

TABLE IV
AGE ESTIMATION ON FGNET FACE AGING DATABASE

Methods	OHRanker [18]	MTWGP [19]	MHR [32]	LARR [33]	BIF[34]	Relative Attribute [1]	Random forest [12]	iGBRT [17]	Relative Tree	Relative Forest	Weighted Relative Forest
MAEs	4.48	4.83	4.87	5.07	4.77	6.01	6.09	5.78	5.26	4.45	4.21

TABLE V
AGE ESTIMATION ON WEBFACE DATABASE

Methods	SVR [35]	SVC [35]	Ridge Regression [16]	Relative Attribute [1]	Random forest [12]	iGBRT [17]	Relative Tree	Relative Forest	Weighted Relative Forest
MAEs	11.13	12.75	11.42	10.97	11.21	10.68	9.94	9.67	9.16

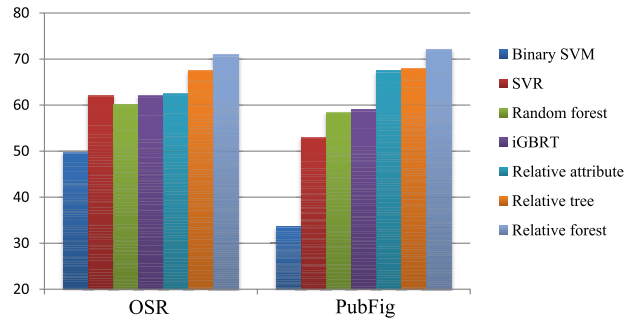


Fig. 4. Rank 1 recognition results on OSR and PubFig databases with attribute prediction scores.

task than the relative attribute method. This may indicate that the hard assignment of ranking labels according to the relative constraints can be systematically wrong and the real margins between samples in different ordinal constraints are different. Thus, relative modeling of visual attribute is more effective. Overall, our method which uses the relative attribute as splitting function, performs considerably better in both attribute prediction tasks in TABLE II&III and subsequent recognition tasks in Fig. 4.

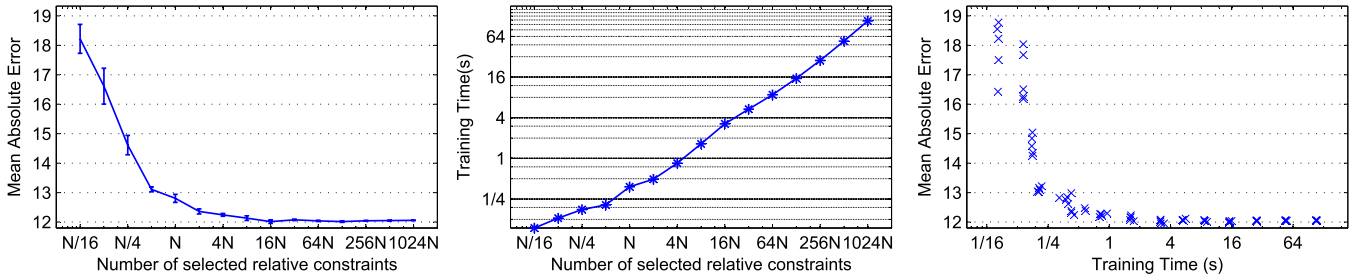
B. Evaluation on FGNET and WebFace Database

To verify the effectiveness of the relative forest for visual attribute prediction in scenario when ordinal labels are available, we conduct a facial age estimation experiments on FGNET and WebFace databases. For the FGNET database, we conduct leave-one-out experiments according to the evaluation protocol. Similar to [33], we first use OLPP [36] to extract features from gray scale face images for obtaining the results of the relative attribute, random forest, iGBRT and our proposed relative tree and forest methods. The results of other methods are directly cited from the corresponding papers. For the WebFace database, the images are randomly split into 4 equally sized subsets to conduct 4-fold cross validation. Similar to [16], we first extract HOG features from raw images and then reduce the feature dimension to 200 with PCA. After feature extraction, the relative forest is used to predict the age. As mentioned in the last part of Section IV, for each database, we report two results of relative forest with different averaging strategies, i.e. the simple average Eqn. (12) and the weighted average Eqn. (14). On FGNET database, the Mean Absolute Error (abbr. MAE) results of our methods and the state-of-the-art age estimation methods [18], [19], [32]–[34] are shown in TABLE IV. With the similar features used in [33], our method achieves the best MAE results on FGNET database. On WebFace database ,shown in TABLE V, since the training set is very large, we only use $32N$ random similar and ordered constraints to train the relative attribute and relative tree models. On the FGNET and WebFace database, all the key parameters of relative forest method are set as default values, which will be clarify in the next subsection, except that only 2% samples are bootstrapped to train each random relative tree and 100 random relative trees are assembled to generate final results on the WebFace database. Note, with the same features, our method also outperforms the random forest method [12] and its state-of-the-art variant iGBRT [17], this validates the advantage of relative modeling for visual attribute prediction in scenario that the absolute ranking labels are available. This may be because visual attributes usually have "inhomogeneous" changes. Taking facial age as an example, the face of a teenager may changes quickly while that of a mid-aged adult may stays unchanged for years. As a result, using relative modeling, which tends to preserve the original structure of data manifold, is still beneficial even the absolute age labels are available (see TABLE IV&V).

C. Empirical Study of the Key Parameters

In this section, we comprehensively study the key parameters of the relative forest method. Since both relative attribute and relative tree methods serve as important components of relative forest method, for clarity, we discuss these key parameters in three levels, i.e. relative attribute level, relative tree level and relative forest level respectively.

1) Key Parameters in Relative Attribute: As the relative attribute method serves as base ranking function of relative forest, in this subsection, we first present the results to clarify the effects of using random subset of ordered and similar constraints for training relative attribute. As aforementioned, exact optimization of the relative attribute algorithm with primal RankSVM method [23] yields an $O(Nd+C)$ complexity. In the worst case $C = O(N^2)$. As a result, exact optimization of the relative attribute method is time-consuming even with only thousands of training samples. Instead of tangling in how to find a condensed representation of all the constraints, we try



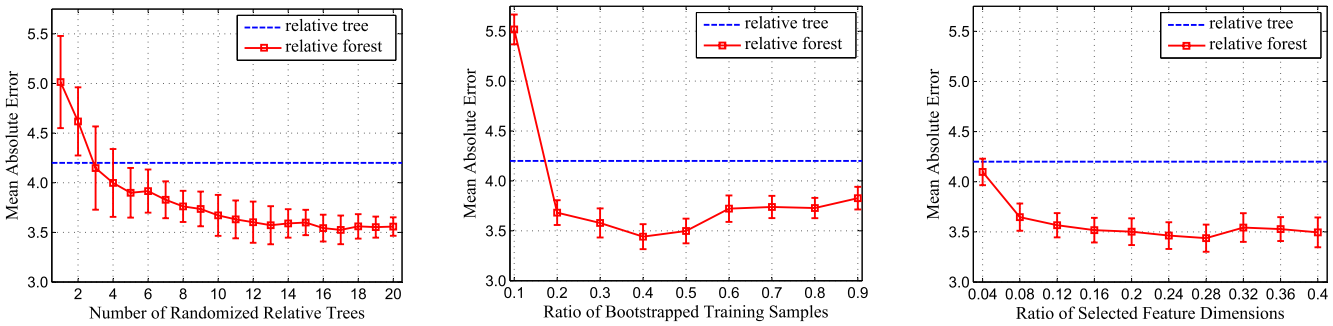
(a) (b) (c)

Fig. 5. Effect of the number of relative constraints randomly generated from a subset of $N = 2200$ samples from WebFace database. (a) How MAE changes with more and more relative constraints; (b) How time cost increases with increasing number of relative constraints and (c) The relationship between MAE and training time.

TABLE VI

EVALUATION ON SPLITTING OBJECTIVE AND PREDICTION STRATEGY OF RELATIVE TREE METHOD IN TWO DISTINCTIVE SCENARIOS, i.e. ORDINAL LABELS ARE GIVEN AND ONLY RELATIVE PAIRWISE CONSTRAINTS ARE AVAILABLE

Splitting objective	Prediction rule	Natural	Open	Perspective	Large	Diagonal	Close
Ordinal Label, Eqn. (5)	Ordinal Label, Eqn. (8)	96.25	93.14	88.04	88.64	90.64	90.67
Ordinal Label, Eqn. (5)	Relative Pair, Eqn. (9)	94.89	91.83	87.66	87.80	90.09	89.96
Relative Pair, Eqn. (6)	Ordinal Label, Eqn. (8)	96.91	93.50	88.43	89.98	91.20	92.39
Relative Pair, Eqn. (6)	Relative Pair, Eqn. (9)	95.59	91.53	88.08	89.04	89.73	89.72



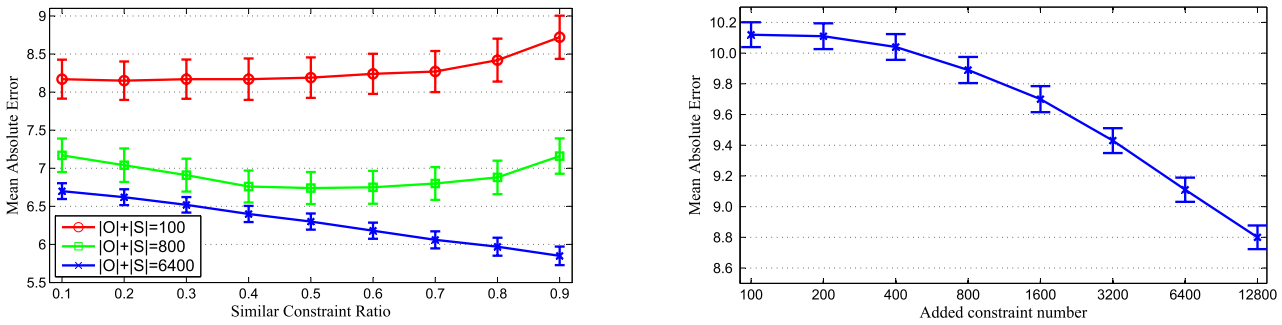
(a) (b) (c)

Fig. 6. Comparison between relative tree and relative forest methods. (a) Evaluation of relative forest with growing number of trees; (b) Evaluation of relative forest with different bootstrap ratio for training each tree; (c) Evaluation of relative forest with different dimension of selected features that are used to train splitting function at each tree node.

to use only an $O(N)$ sized subset of constraints to train the relative attribute for relative forest.

In practice, we find the above strategy usually ensures acceptable performance of the learned ranking function and significantly reduces the training time. As shown in Fig. 5, we present age estimation results on a subset of WebFace database. More specifically, only $N = 2200$ samples are randomly selected for evaluation, which allows us to completely examine whether the error would asymptotically go down for a sufficiently large number of used constraints. As shown in Fig. 5(a), such asymptotical effects is observed in our evaluation. However, the accuracy improves extremely slow when more than $16N$ ordered and similar constraints are used. In the meanwhile, as shown in Fig. 5(b), the consumed time increases linearly according to the constraint number. We also directly present the trade-off between the training time and achieved MAE in Fig. 5(c).

2) Key Parameters in Relative Tree: As illustrated in Section III, the proposed relative tree method aims at attribute prediction with either ordinal labels or relative pairs. In this subsection, we conduct experiments on OSR database to see whether ordinal labels are more informative than relative pairwise constraints for searching optimal splitting threshold (Eqn. (5) with ordinal labels or Eqn. (6) with relative pairs) and normalizing ranking score (Eqn. (8) with ordinal labels or Eqn. (9) with relative pairs). More specifically, we evaluate four different kinds of configurations for relative tree method. As shown in Table VI, consist of relative trees with different configurations, all variants of the relative forest methods achieve acceptable performance on OSR database. While ordinal labels provide more information for robust ranking score normalization compared to relative pairs, the relative pair based splitting rule Eqn. (6) seems to be better than information gain based splitting rule Eqn. (5).



(a) (b)

Fig. 7. Evaluation on the different effects of similar and ordered constraints for training relative attribute. (a) Evaluation on different similar constraint ratio with different number of total constraint; (b) Evaluation on mining similar constraints from video context.

3) Key Parameters in Relative Forest: For relative forest, we analyze the effects of three key parameters, which are: 1) the number of randomized relative trees in the relative forest (see Fig. 6(a)); 2) the ratio of samples bootstrapped from the whole training set for constructing single relative tree (see Fig. 6(b)) and 3) the ratio of selected feature dimensions for training the relative attribute in each node splitting step (see Fig. 6(c)). Note that the default values of parameters, namely tree number, ratio of bootstrap samples and ratio of selected feature dimensions are 20, 40% and 30% respectively. When analyzing one parameter, we fix the other two. For each setting, we repeat the relative forest method 10 times and present the average results.

The evaluation is conducted on the FGNET dataset. Images of the first person in this database (15 images) are treated as the test set, and images of all other persons (987 images) are used for training. MAE is used to evaluate the estimation accuracy. Note, in this section, only simple relative forest is used for comparison. Since the effects of constraint level randomization is already examined, in the experiments of this section, we empirically set the number of selected constraints as $2N$.

As shown in Fig. 6(a), when the size of the forest grows larger than 5, the relative forest method performs consistently better than the relative tree method. As shown in Fig. 6(b), when more than 20% samples are bootstrapped to train a single tree, the relative forest can successfully capture the intrinsic structure of the data manifold and achieve better ranking results. Similarly, as shown in Fig. 6(c), even when only 4% feature dimensions (8/200) are selected to train the splitting function, the relative forest can perform better than the relative tree. If too many samples are bootstrapped or too many features are selected, MAE increases slightly. In extreme situations, if all parameters are set as the maximal value, the performance of the relative forest method will degrade to that of the relative tree method since all the trees will be exactly the same and an ensemble will not bring any performance gain. For clarity, in TABLE VII, we present the training time we used to obtain the results in TABLE II, III, IV and V. On the OSR and PubFig databases, due to the lack of training samples, only approximately 240 are used to train each attribute, we bootstrap 95% training samples for training each randomized relative tree. On the WebFace database, $32N$ random subset of ordered and similar constraints are used for training relative attribute and relative tree method. If all constraints are used, the training would be $N/64 \approx 650$ times longer. As aforementioned, The relative forest method only bootstrap 2% samples to train each random relative tree and assembles 100 random relative trees to obtain the results on the WebFace database. Except for the parameters mentioned above, all other parameters are set as default values, namely random selected relative constraints number, tree number, ratio of bootstrapped samples and ratio of selected feature dimensions are set as 2N, 20, 40% and 30% respectively.

TABLE VII

TRAINING TIME COMPARISONS (S). THE TRAINING TIME OF RELATIVE ATTRIBUTE, RELATIVE TREE AND RELATIVE FOREST METHODS FOR OBTAINING RESULTS IN TABLE II, III, IV AND V ARE PRESENTED

Database	relative attribute	relative tree	relative forest
OSR	1.2	3.4	33.7
PubFig	1.5	4.0	34.3
FGNET	121.8	417.7	60.3
WebFace	2603.1	9075.6	6882.5

D. Relative Attribute Benefits From Video Context

Since pairwise relative constraints are difficult to label, one of the remaining problems of our method is how to collect enough relative constraints to reliably train the relative attribute model as base ranking function. This problem is already alleviated by the fact that only $O(N)$ constraints may already be enough to train a relative attribute. However, to further address this problem, more insight into the ordinal attribute prediction is indispensable. Take age estimation as an example, which kind of constraints (similar or ordered) is more crucial for accurate prediction? To study this problem, we fix the total number of randomly selected constraints and change the ratio of similar constraints. The experiments are conducted on FGNET database. For each setting, the experiments are repeated 1000 times and mean MAE with corresponding standard deviation is shown in Fig. 7(a). As shown, if only a small number of constraints are available, more ordered constraints are essential. However, if there are enough

ordered constraints, an additional large number of similar constraints may help further improve ranking accuracy.

Since similar constraints are more important when there are enough ordered constraints, inspired by [20], we propose to cheaply collect a large number of similar constraints from video context. Intuitively, all faces in a tracking sequence of the same video should have similar age prediction results. Using the same video sequences introduced in [20], we find that for age estimation, if the number of ordered labels is big enough (fixed as $6,400$ in this test), similar constraints collected from video context indeed help improve ranking accuracy of the relative attribute. As shown in Fig. 7(b), with the increasing number of the added similar constraints, the MAE steadily decreases. Note that the MAE results of Fig. 7(b) are higher than those of Fig. 7(a), because automatic face alignment is used to extract features and OMRON face detector [37] is utilized in this experiments.

VI. Conclusion

The human-namable visual attributes bear intuitive appeal in strengthening various vision tasks, which makes the accurate estimation of them especially important. Aiming at accurate and efficient prediction of the visual attributes, we first propose the relative tree method to cope with more complex nonlinear data, and then propose the relative forest method to further boost the performance and reduce the training time. Modeling the visual attributes as relative properties rather than binary or multi-class properties, our proposed relative forest method not only has a wider application range, i.e. applicable when only pairwise constraints are available, but also achieves higher accuracy in almost all the attribute prediction tasks and the subsequent recognition tasks. The possibility of using video context to cheaply collect a large number of relative constraints for training the relative attribute is also explored. In future, training the relative forest with auxiliary video context data shall be further studied for accurate visual attribute predictions.

References

[1] D. Parikh and K. Grauman, "Relative attributes," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 503–510.

[2] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *Proc. IEEE 12th Int. Conf. Comput. Vis. (ICCV)*, Sep./Oct. 2009, pp. 365–372.

[3] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 1331–1338.

[4] J. Wang, K. Markert, and M. Everingham, "Learning models for object recognition from natural language descriptions," in *Proc. British Mach. Vis. Conf. (BMVC)*, Sep. 2009, p. 2.

[5] G. Wang and D. Forsyth, "Joint learning of visual attributes, object classes and visual saliency," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sep./Oct. 2009, pp. 537–544.

[6] G. Patterson and J. Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 2751–2758.

[7] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 1778–1785.

[8] L. Bourdev, S. Maji, and J. Malik, "Describing people: A poselet-based approach to attribute classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 1543–1550.

[9] B. Siddiquie, R. S. Feris, and L. S. Davis, "Image ranking and retrieval based on multi-attribute queries," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 801–808.

[10] R. Fergus, H. Bernal, Y. Weiss, and A. Torralba, "Semantic label sharing for learning with many categories," in *Proc. 11th Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 762–775.

[11] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 951–958.

[12] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[13] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[14] FGNET. (2002). *The Fg-Net Aging Database*. [Online]. Available: http://sting.cycollege.ac.cy/~alanitis/fgnetaging/index.htm

[15] B. Ni, Z. Song, and S. Yan, "Web image mining towards universal age estimator," in *Proc. 17th ACM Int. Conf. Multimedia (MM)*, 2009, pp. 85–94.

[16] S. Zheng, "Visual image recognition system with object-level image representation," Ph.D. dissertation, Dept. Elect. Comput. Eng., Nat. Univ. Singapore, Singapore, 2012.

[17] A. Mohan, Z. Chen, and K. Q. Weinberger, "Web-search ranking with initialized gradient boosted regression trees," *Proc. JMLR, Workshop Conf.*, vol. 14. Jan. 2011, pp. 77–89.

[18] K.-Y. Chang and Y.-P. Hung, "Ordinal hyperplanes ranker with cost sensitivities for age estimation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 585–592.

[19] Y. Zhang and D.-Y. Yeung, "Multi-task warped Gaussian process for personalized age estimation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2622–2629.

[20] Z. Song, B. Ni, D. Guo, T. Sim, and S. Yan, "Learning universal multi-view age estimator using video context," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 241–248.

[21] R. Herbrich, T. Graepel, and K. Obermayer, "Large margin rank boundaries for ordinal regression," in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 115–132.

[22] T. Joachims, "Optimizing search engines using clickthrough data," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining (SIGKDD)*, 2002, pp. 133–142.

[23] O. Chapelle and S. S. Keerthi, "Efficient algorithms for ranking with SVMs," *Inf. Retr.*, vol. 13, no. 3, pp. 201–215, 2010.

[24] T. Joachims, "Training linear SVMs in linear time," in *Proc. 12th ACM Int. Conf. Knowl. Discovery Data Mining (SIGKDD)*, 2006, pp. 217–226.

[25] P. Geurts and G. Louppe, "Learning to rank with extremely randomized trees," *J. Mach. Learn. Res.*, vol. 14, pp. 49–61, Jan. 2011.

[26] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[27] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," *Commun. ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[28] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *Proc. IEEE 11th Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–8.

[29] O. Chapelle and Y. Chang, "Yahoo! Learning to rank challenge overview," *J. Mach. Learn. Res.*, vol. 14, pp. 1–24, Jan. 2011.

[30] Y. Freund, S. Dasgupta, M. Kabra, and N. Verma, "Learning the structure of manifolds using random projections," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 473–480.

[31] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 27.

[32] T. Qin, X.-D. Zhang, D.-S. Wang, T.-Y. Liu, W. Lai, and H. Li, "Ranking with multiple hyperplanes," in *Proc. ACM Int. Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2007, pp. 279–286.

[33] G. Guo, Y. Fu, C. R. Dyer, and T. S. Huang, "Image-based human age estimation by manifold learning and locally adjusted robust regression," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1178–1188, Jul. 2008.

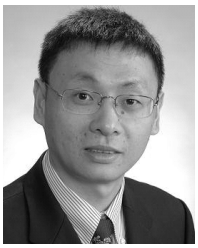
[34] G. Guo, G. Mu, Y. Fu, and T. S. Huang, "Human age estimation using bio-inspired features," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 112–119.

[35] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.

[36] D. Cai, X. He, J. Han, and H.-J. Zhang, "Orthogonal Laplacianfaces for face recognition," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3608–3614, Nov. 2006.
[37] H. Ai, C. Huang, B. Wu, and S. Lao, "Specified object detection apparatus," U.S. Patent 7457432, Nov. 25, 2008.

Shaoxin Li received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2015. In 2013, he served as a Research Assistant with the Learning and Vision Research Group, Department of Electrical and Computer Engineering, National University of Singapore. He is currently a Research Engineer with the Tencent YouTu Lab., Social-Network-Group, Tencent. His research interests include machine learning, computer vision, deep learning, and especially focus on large-scale video face recognition in the wild environments.

Shiguang Shan (SM'14) received the M.S. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1999, and the Ph.D. degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 2004. He joined ICT, CAS, in 2002, where he has been a Professor since 2010. He is currently the Deputy Director of the Key Laboratory of Intelligent Information Processing with CAS. He has published more than 200 papers in refereed journals and proceedings. His research interests cover computer vision, pattern recognition, and machine learning. He especially focuses on face recognition related research topics. He received the China's State Scientific and Technological Progress Award in 2005 and the China's Natural Science Award in 2015 for his work on visual information processing and face recognition technologies. He also received the Young Scientist Award of China Computer Federation in 2015. He has served as an Area Chair for many international conferences, including ICCV'11, ICPR'12, ACCV'12, FG'13, ICPR'14, ICASSP'14, and ACCV'16. He is an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, *Neurocomputing*, *Pattern Recognition Letters*, and *EURASIP Journal of Image and Video Processing*.



Shuicheng Yan (SM'10) is a Chief Scientist of 360, Director of 360 AI Institute, and also the Dean's Chair Associate Professor with the National University of Singapore. He has authored/co-authored 100 technical papers over a wide range of research topics, with Google Scholar citation over 26 000 times and has an H-index of 65. His research areas include machine learning, computer vision, and multimedia. He was an ISI Highly Cited Researcher 2014 and 2015, and an IAPR Fellow 2014. He received the Best Paper Awards from ACM MM'13 (both Best Paper and Best Student Paper), ACM MM'12 (Best Demo), PCM'11, ACM MM'10, ICME'10, and ICIMCS'09, the runner-up prize of ILSVRC'13, the winner prize of ILSVRC'14 detection task without extra data, the winner prizes of the classification task in PASCAL VOC 2010–2012, the winner prize of the segmentation task in PASCAL VOC 2012, the honorable mention prize of the detection task in PASCAL VOC'10, the 2010 TCSVT Best Associate Editor Award, the 2010 Young Faculty Research Award, the 2011 Singapore Young Scientist Award, and the 2012 NUS Young Researcher Award.

Xilin Chen (F'15) received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1988, 1991, and 1994, respectively, all in computer science. He was a Professor with the Harbin Institute of Technology from 1999 to 2005. He has been a Professor with the Institute of Computing Technology, Chinese Academy of Sciences, since 2004. He is also a FiDiPro Professor from 2012 to 2015 with Oulu University. He has authored one book and over 200 papers in refereed journals and proceedings in the areas of computer vision, pattern recognition, image processing, and multimodal interfaces. He is a fellow of the China Computer Federation.