Unsupervised Learning of Discriminative Relative Visual Attributes

Shugao Ma[1], Stan Sclaroff[1], and Nazli Ikizler-Cinbis[2]

[1] Department of Computer Science, Boston University
[2] Department of Computer Engineering, Hacettepe University

Abstract. Unsupervised learning of relative visual attributes is important because it is often infeasible for a human annotator to predefine and manually label all the relative attributes in large datasets. We propose a method for learning relative visual attributes given a set of images for each training class. The method is unsupervised in the sense that it does not require a set of predefined attributes. We formulate the learning as a mixed-integer programming problem and propose an efficient algorithm to solve it approximately. Experiments show that the learned attributes can provide good generalization and tend to be more discriminative than hand-labeled relative attributes. While in the unsupervised setting the learned attributes do not have explicit names, many are highly correlated with human annotated attributes and this demonstrates that our method is able to discover relative attributes automatically.

1 Introduction

There has been increasing interest in visual attribute models for computer vision [1–4]. The key idea is that visual attributes describe properties of entities and are often shared by many different classes; thus, attribute models learned on a set of classes can be useful for describing other, previously unseen classes.

Visual attributes can be divided into binary and relative attributes. Previous studies have mainly focused on binary attributes. Recently, Parikh and Grauman proposed the use of relative attributes [5], which describe the relative strength of the presence of a visual attribute. In comparison to binary attributes, relative attributes are more natural and informative in describing many visual aspects of objects and scenes.

Attribute learning methods can be divided into supervised and unsupervised based on the availability of a list of annotated attributes on the training data. Jointly learning attributes and class models on datasets with labeled attributes has been widely studied [1–4, 6–9]. All of these methods yield good performance, particularly with respect to the learned attributes' good generalizability to test classes that are not present in learning; however, these methods require a human to predefine the attributes and provide labeled training data.

Supervised learning of attributes has several problems. Firstly, a manually defined set of attributes may be intuitive but not very discriminative for the task at hand. Secondly, some discriminating attributes may be overlooked or difficult

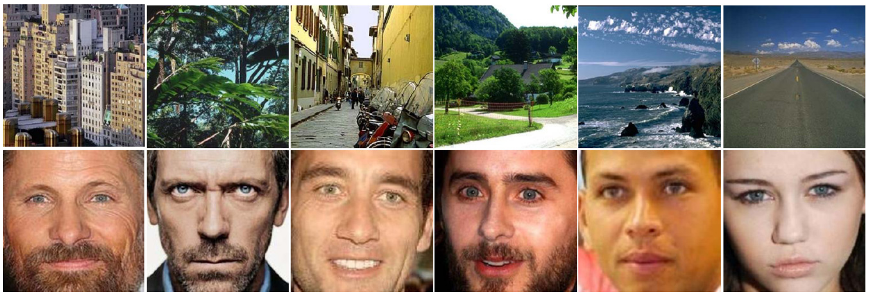

Fig. 1. Example learned relative attributes for the datasets OSR (top row) and PUBFIG (bottom row). Each row shows one sample image from each class, presented in order with respect to the learned attribute. In our experiments, the method discovers attributes that are highly correlated with human-labeled attributes: the illustrated attributes highly correlate with human-labeled attributes *Open* and *Young* respectively.

to express in words. Thirdly, attribute labels are produced by annotators' subjective judgements, which may be erroneous [6]. Fourthly, the required human supervision hinders the method's scalability to a large number of attributes.

Other works employ semi-supervised learning of attributes by leveraging information retrieval techniques, e.g., [6, 10–12]. While these techniques reduce the need for annotation, there are two main problems. Firstly, labels from web search or text retrieval tend to be loose and noisy. Secondly, some visual attributes are rarely described by text and are thus hard to learn with these methods.

Another line of work uses active learning to acquire models of attributes. Parikh and Grauman [13] propose a method that automatically identifies binary attributes that are probably *nameable* and asks human annotators to either name the identified attribute or reject it as unnameable. Kovashka, et al. [14] also use active learning to acquire attribute models. Both methods achieve a good balance between the required annotation work and quality of learned attributes; however, they only consider *binary attributes*.

Methods have also been proposed for unsupervised learning of *binary attributes* [2, 4, 12, 15] that provide good generalizability and discriminative properties. However, to the best of our knowledge, there is no previous work on unsupervised learning of *relative attributes*, which is the focus of this paper. Given the demonstrated value of relative attributes and the problems of supervised learning, we would expect our method to be useful. The difficulty in unsupervised learning of relative attributes is that the search space is very large (factorial to the number of classes). Furthemore, certain attributes may only be meaningful to a subset of training classes, depending on whether the strength of the attribute is shared to the same extent among the majority of instances of a class; consequently, the search for relative attributes should also include orderings of subsets of training classes – making the search space even larger.

Contributions: We formulate the unsupervised relative visual attribute learning problem as a mixed integer programming problem and propose an approximation algorithm that performs greedy search in the space of all possible relative attributes. Our method also infers orderings of the relevant training classes with respect to each learned attribute (see examples in Fig. 1). Our experiments on the datasets of [5] show that our automatically learned attributes are discriminative and complementary to hand-labeled relative attributes, and they offer good generalizability to classes that were unseen during training. We also demonstrate that, while in the unsupervised setting the learned attributes do not have explicit semantic names, our method is able to automatically discover all the human annotated attributes for the datasets tested.

2 Automatic Relative Attribute Learning

2.1 Formulation

In training, we are given a set of images $I = \{i\}$ represented by feature vectors $\{\boldsymbol{x_i}\}$ and image class labels $\{c_a\}$. Relative attribute annotations in the form of class orderings are not given; instead, we want to identify those during training along with the set of learned attribute rank functions. To make things simpler, we only consider relative attributes that contain strict pairwise orders. The rank function that we want to learn for each relative attribute a_m is as follows:

$$r_m(\boldsymbol{x}_i^a) = \boldsymbol{w}_m^T \boldsymbol{x}_i^a, \;\; s.t. \;\; \boldsymbol{w}_m^T \boldsymbol{x}_i^a > \boldsymbol{w}_m^T \boldsymbol{x}_j^b \;, \; i \in c_a, j \in c_b, c_a \succ c_b, \tag{1}$$

where $\boldsymbol{x}_i^a$ is the feature vector of training image i from class c_a.

Our formulation is based on the one used in [5], and we introduce decision variables to represent missing attribute annotation information. We define $\mu_a = 1$ if the attribute is relevant to c_a otherwise $\mu_a = 0$, and for $\forall a, b, \; a > b$ we define

$$\delta_{ab} = \begin{cases} 1 & c_a \succ c_b \\ -1 & c_a \prec c_b \\ 0 & \mu_a = 0 \vee \mu_b = 0. \end{cases} \tag{2}$$

Intuitively, if an attribute is relevant to more training classes, then it may represent a general attribute that is likely to be relevant to new classes unseen in training. To embody this intuition, we add the ratio of irrelevant classes into the minimization objective. Finally, our formulation for unsupervised relative attribute learning is:

$$\min_{\boldsymbol{w}_m, \boldsymbol{\xi}, \boldsymbol{\delta}, \boldsymbol{\mu}} \;\; \frac{1}{2}||\boldsymbol{w}_m^T||_2^2 + C_1 \sum \xi_{ij,ab}^2 + C_2 (1 - \frac{1}{N} \sum \mu_a) \tag{3}$$

$$s.t. \;\; \delta_{ab} \boldsymbol{w}_m^T (\boldsymbol{x}_i^a - \boldsymbol{x}_j^b) \geq min(\mu_a, \mu_b) - \xi_{ij,ab}, \quad \forall (i,j), i \in c_a, j \in c_b, a > b \tag{4}$$

$$|\delta_{ab} - \delta_{bc}| \geq |\delta_{ab} - \delta_{ac}|, \quad \forall a > b > c, \mu_a = \mu_b = \mu_c = 1 \tag{5}$$

$$|\delta_{ab}| = \mu_a, \quad \forall a \in \{2, \ldots, N\} \tag{6}$$

$$|\delta_{ab}| = \mu_b, \quad \forall b \in \{1, 2, \ldots, N-1\} \tag{7}$$

$$\xi_{ij,ab} \geq 0, \; \delta_{ab} \in \{-1, 0, 1\}, \; \mu_a \in \{0, 1\}. \tag{8}$$

N is the number of training classes, a, b, c are training class indices, and C_1 and C_2 are tradeoff constants among the margin, loss and ratio of irrelevant classes.

For $a > b > c$, constraint (5) requires that if $c_a \succ c_b$, $c_b \succ c_c$ then $c_a \succ c_c$ and if $c_a \prec c_b$, $c_b \prec c_c$ then $c_a \prec c_c$, which enforces that the pairwise orderings between classes do not contradict each other. Constraints (6) and (7) ensure that the value of δ_{ab} is well defined according to (2). For two classes c_a and c_b, if $c_a \succ c_b$ then $\delta_{ab} = 1, \mu_a = \mu_b = 1$ and thus constraint (4) becomes $\boldsymbol{w}_m^T(\boldsymbol{x}_i^a - \boldsymbol{x}_j^b) \geq 1 - \xi_{ij,ab}$; if $c_a \prec c_b$ then $\delta_{ab} = -1, \mu_a = \mu_b = 1$ and constraint (4) becomes $\boldsymbol{w}_m^T(\boldsymbol{x}_i^b - \boldsymbol{x}_j^a) \geq 1 - \xi_{ij,ab}$; if one (or both) of c_a, c_b is irrelevant to the attribute, then one (or both) of μ_a, μ_b becomes zero and (4) reduces to $\xi_{ij,ab} \geq 0$, which essentially removes constraints of irrelevant classes. We note that if the values of $\boldsymbol{\mu}$ and $\boldsymbol{\delta}$ are fixed (as in the fully supervised setting), then the above reduces to formulation of [5], but considering only strict pairwise orderings.

2.2 Algorithm

The above formulation presents a mixed-integer programming problem and is hard to solve directly. We propose an algorithm to find an effective, approximate solution, which is summarized as follows. At the start of each run of the algorithm we give initial values to $\boldsymbol{\mu}, \boldsymbol{\delta}$. Given $\boldsymbol{\mu}, \boldsymbol{\delta}$ are fixed, we update $\boldsymbol{w}_m$ using an SVM solver. Once $\boldsymbol{w}_m$ is updated, we then fix $\boldsymbol{w}_m$ and update $\boldsymbol{\mu}$, $\boldsymbol{\delta}$. The reduced problem for learning $\boldsymbol{\mu}$, $\boldsymbol{\delta}$ is still a mixed integer program, so we propose a greedy algorithm to solve it. The algorithm is run multiple times initialized with every possible pair of classes and each run yields a candidate relative attribute. After learning the set of candidate relative attributes, we remove redundant ones. In our implementation, if the absolute value of the cosine between $\boldsymbol{w}_m$ and $\boldsymbol{w}_n$ is in the range $[0.9,\ 1]$, then one of the corresponding attributes is removed. Our key ideas in this algorithm are: choosing a broad set of initializations that tend to yield a broad set of useful attributes, and using an efficient algorithm to update $\boldsymbol{\mu}, \boldsymbol{\delta}$. More details of our algorithm are given below.

Initialization: At start of the algorithm, we first pick a pair of classes, say c_a and c_b $(a > b)$, and initialize $\boldsymbol{\mu}$ and $\boldsymbol{\delta}$ as follows:

$$\mu_k = \begin{cases} 1 & k = a \ \vee \ k = b \\ 0 & otherwise \end{cases} \qquad \delta_{kh} = \begin{cases} 1 & k = a \ \wedge \ h = b \\ 0 & otherwise. \end{cases} \tag{9}$$

Thus, all classes except c_a and c_b are set to be irrelevant to the attribute in this initialization. Additional classes can be discovered as relevant in subsequent iterations. An intuitive explanation for this initialization method is that by making as few assumptions about the initial values of $\boldsymbol{\mu}$, $\boldsymbol{\delta}$ as possible, we may consider a broad search space for learning attributes. All possible pairs of classes are used for initialization. As the number of training classes increases, the diversity of the data may also increase, so there may be additional useful relative attributes and the algorithm is run more times to learn these attributes. For a training set of N classes, the total number of class pairs is $\frac{1}{2}N(N-1)$, which is manageable.

Updating $\boldsymbol{w_m}$: When the values for $\boldsymbol{\mu}, \boldsymbol{\delta}$ are fixed, learning $\boldsymbol{w}_m$ reduces to a form that is similar to SVM learning but on pairwise difference vectors. We apply a standard SVM solver [16] to learn $\boldsymbol{w}_m$. To speed up learning, we only generate constraints (4) between classes that are adjacent in the class orderings of that relative attribute. When $\boldsymbol{w_m}$ is learned, we compute the objective value and stop the training if it stops decreasing. If $\mu_a = 1$ for all classes c_a, we also stop training, since no classes remain to be added to the list of relevant classes.

Updating $\boldsymbol{\mu}, \boldsymbol{\delta}$: When $\boldsymbol{w_m}$ is fixed, the formulation in Sec. 2.1 is still hard to solve directly. We use a greedy algorithm to solve this problem. At iteration t, we want to pick a class, say c_a, which is labeled as irrelevant by the previous iteration and which will introduce the least additional loss if labeled as relevant at this iteration. We then want to add c_a to the list of relevant classes and update the values of $\boldsymbol{\mu}, \boldsymbol{\delta}$ accordingly. First, for any class c_b such that $\mu_b^{t-1} = 1$, we compute $m_b^t = \text{median}(\{p_j | p_j = \boldsymbol{w}_m^{tT} \boldsymbol{x}_j^b, j \in c_b\})$. The resulting set of m_b^t will divide the real line into several bins. Then, for any class c_a such that $\mu_a^{t-1} = 0$, we compute the entropy e_a^t of the set $\{p | p = \boldsymbol{w}_m^{tT} \boldsymbol{x}_i^a, i \in c_a\}$ over the histogram of the real line whose bin boundaries are negative infinity, the sorted set of m_b^t, and infinity. Finally, we select the class that has the smallest entropy and add it to the list of relevant classes. This strategy selects the relevant class without explicitly computing the pairwise loss. Although the selected class may not be the class that introduces the least loss under the ranking function, it will introduce low loss because the projections of samples in that class tend to aggregate on the projection line between the medians of projections of relevant classes' samples. After selecting a class c_a to add to the list of relevant classes, we update $\boldsymbol{\mu}, \boldsymbol{\delta}$

$$\mu_k^t = \begin{cases} \mu_k^{t-1}, & k \neq a \\ 1, & k = a \end{cases} \tag{10}$$

$$\delta_{kh}^t = \begin{cases} \delta_{kh}^{t-1}, & k \neq a \wedge h \neq a \\ 1, & (k = a \ \wedge \ m_a^t > m_h^t) \ \vee \ (h = a \ \wedge \ m_k^t > m_a^t) \\ -1, & (k = a \ \wedge \ m_a^t < m_h^t) \ \vee \ (h = a \ \wedge \ m_k^t < m_a^t) \\ 0, & (k = a \ \wedge \ \mu_h^t = 0) \ \vee \ (h = a \ \wedge \ \mu_k^t = 0) \end{cases} \tag{11}$$

where m_a^t is the median value of the set $\{p_i | p_i = \boldsymbol{w}_m^{tT} \boldsymbol{x}_i^a, i \in c_a\}$.

3 Experiments

To evaluate our formulation, we used two datasets: the Outdoor Scene Recognition (OSR) Dataset [17] containing 2688 images of 8 categories, and a subset of the Public Figure Face Database (PUBFIG) dataset [4] containing 772 images from 8 random identities (almost 100 images each). This is the same setting provided by the authors of [5] and we used the same features: a 512-D gist [17] descriptor for OSR and a combination of 512-D gist descriptor and 45-D *lab* color histogram for PUBFIG. The train/test split was provided with the datasets.

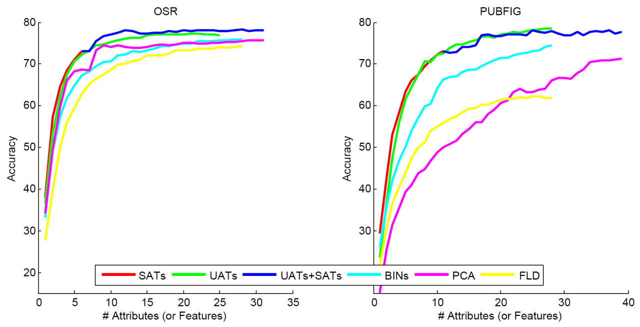

Fig. 2. Multi-class classification results

We conducted experiments that evaluate our learned relative attributes in multi-class classification and K-shot classification. We also conducted an experiment that examines the correlation between our automatically-learned relative attributes and human-labeled relative attributes. Our Matlab implementation will be made available via ftp. The CPU time for attribute learning in Matlab is 127 seconds for OSR and 102 seconds for PUBFIG (28 attributes on each dataset) on a laptop (2.4 GHz Intel Core 2 Duo, 2G memory).

3.1 Multi-class Classification

In this experiment, we trained multi-class SVM classifiers with an RBF kernel by libsvm [18] using: (a) relative attributes learned by our unsupervised algorithm (UATs); (b) relative attributes learned by the supervised algorithm [5] (SATs); (c) combination (UATs+SATs); (d) linear SVM learned between each pair of classes (BINs); (e) PCA; (f) Fisher's Linear Discriminant between every pair of classes (FLD). The latter three are included as baselines of mid-level features. For SATs, we used the attribute values provided by the authors of [5]. 30 images per class were used for training and the others for testing. The classification accuracy was computed as the mean per-class accuracy, i.e., the number of correctly classified test samples *vs.* total number of test samples. Fig. 2 reports classification accuracy as a function of the number of attributes used. For UATs, SATs, BINs and FLD, we ran the experiments 30 times using different attribute (or feature) orders and report the mean. For UATs+SATs, we used all the SATs while varying the number of UATs, according to the reverse order of their correlation to the SATs, with least-correlated UAT added first, as will be explained in Sec. 3.3. For PCA, the principal components are used in order of their eigenvalue (maximum first).

In Fig. 2, it appears that the performance using SATs is limited by the number of labeled attributes: only 6 annotated relative attributes for OSR and 11 for PUBFIG. However, our algorithm learns more attributes: we learned 25 relative attributes for OSR (3 redundant attributes were removed) and 28 for PUBFIG. We also observe that combining SATs+UATs increases the classification

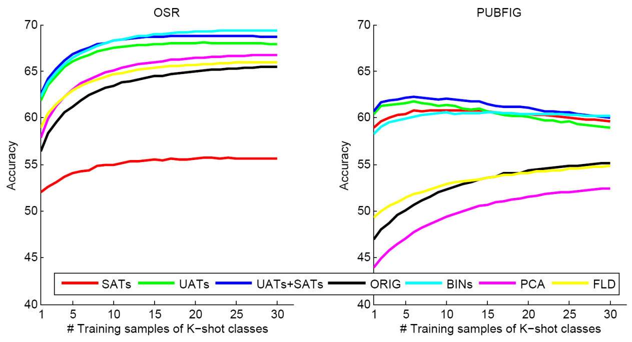

Fig. 3. K-shot classification performance. Total number of classes remains 8.

accuracy over using SATs alone, which shows that the UATs can capture some discriminative information that may be overlooked by humans when labeling the relative attributes. However, the overall performance of UATs+SATs is similar to using only UATs. This is consistent with results reported for binary attributes in [2], and we think it may be due to the high correlation between the labeled relative attributes and some of the relative attributes learned by our algorithm. On both datasets, UATs perform better than mid-level feature baselines. The results show our method performs better than dimensionality reduction techniques PCA and FLD. Note that, our method should not be seen as a dimensionality reduction technique because when the number of classes increases, additional useful attributes may be learned and the resulting attribute space may not be smaller than the raw feature space.

We also compared the ability of BINs and UATs to order the classes, using an entropy criterion similar to the one in Sec. 2.2. The results show that UATs produce class orderings that have lower entropy than for BINs and thus better separate the classes (detailed results are omitted due to space limitations).

3.2 K-Shot Classification

To evaluate the generalizability of learned attributes, we performed K-shot classification. In our setting, 2 classes (we call them *K-shot classes*) were left out and attributes were trained on the other 6 classes. The learned attributes were then used in a 1NN classifier for multi-class classification, with K training images for each K-shot class and all training images for the other classes (30 per class). We plotted the classification accuracy against varying K value for the 1NN classifier. For each possible choice of K-shot classes and choice of K, we repeated the experiment 10 times (each time randomly selecting K training images from each K-shot class) and computed the mean accuracy. We compared 6 image representations: (a) original features (ORIG); (b) SATs; (c) UATs; (d) UATs+SATs; (e) BINs; (f) PCA; (g) FLD. For SATs, we used the code and parameter settings provided by the authors of [5] to learn the attributes.

For PCA, the number of used principal components is set to be the same as number of attributes in UATs+SATs. Fig. 3 reports the results.

The results indicate that the attributes learned by the unsupervised algorithm and the supervised algorithm can complement each other, and also show that the UATs can yield good discrimination even if there are few training examples for a test class. For the OSR dataset, SATs perform worse than the original features (ORIG), while for the PUBFIG dataset SATs preform better than ORIG. We suspect that this may due to the attribute annotations: it is hard to control the quality (in terms of discriminative power and generalizability) of manually labeled attributes. Besides, there are only 6 labeled relative attributes for the OSR dataset, which may also limit performance. Our method discovers a larger set of relative attributes for this dataset, and sidesteps this limitation. When K is small (the more interesting case), UATs perform better than mid-level feature baselines. When K is large, the performances of UATs and BINs are similar, but we emphasize here that comparing to mid-level features, our method is explicitly designed to identifying useful class orderings (Sec. 3.3).

3.3 Correlation Analysis

In this section we analyze the correlation between automatically learned relative attributes and human-labeled relative attributes. For each pair of automatically learned class ordering and manually labeled class ordering, we compute the Kendall Tau correlation:

$$\tau = \frac{n_c - n_d}{\frac{1}{2}n(n-1)} \tag{12}$$

Table 1. Correlation between manually labeled class orderings and automatically learned class orderings. In OSR, classes are coast (C), forest (F), highway (H), inside-city (I), moutain (M), open-country (O), street (S) and tall-building (T). In PUBFIG, classes are Alex Rodriguez (A), Clive Owen (C), Hugh Laurie (H), Jared Leto (J), Miley Cyrus (M), Scarlett Johansson (S), Viggo Mortensen (V) and Zac Efron (Z).

OSR			
Attr. Name	Sem. Attr.	Auto. Learned Attr.	$\hat{\tau}$
natural	$T\prec I\sim S\prec H\prec C\sim O\sim M\sim F$	$S\prec I\prec H\prec F\prec O$	0.89
open	$T\sim F\prec I\sim S\prec M\prec H\sim C\sim O$	$T\prec F\prec S\prec O\prec C\prec H$	0.86
perspective	$O\prec C\prec M\sim F\prec H\prec I\prec S\prec T$	$O\prec F\prec H\prec I\prec S$	1
large-objects	$F\prec O\sim M\prec I\sim S\prec H\sim C\prec T$	$F\prec M\prec S\prec H\prec C\prec T$	0.97
diagonal-plane	$F\prec O\sim M\prec C\prec I\sim S\prec H\prec T$	$F\prec O\prec M\prec I\prec H\prec S$	0.79
close-depth	$C\prec M\prec O\prec T\sim I\sim S\sim H\sim F$	$M\prec O\prec F\prec I\prec S$	0.84
PUBFIG			
Attr. Name	Sem. Attr.	Auto. Learned Attr.	$\hat{\tau}$
Masculine-looking	$S\prec M\prec Z\prec V\prec J\prec A\prec H\prec C$	$S\prec M\prec Z\prec A\prec H\prec C$	1
White	$A\prec C\prec H\prec Z\prec J\prec S\prec M\prec V$	$A\prec Z\prec H\prec J\prec S$	0.80
Young	$V\prec H\prec C\prec J\prec A\prec S\prec Z\prec M$	$V\prec H\prec C\prec J\prec A\prec M$	1
Smiling	$J\prec V\prec H\prec A\sim C\prec S\sim Z\prec M$	$J\prec H\prec C\prec A\prec Z$	0.95
Chubby	$V\prec J\prec H\prec C\prec Z\prec M\prec S\prec A$	$J\prec H\prec C\prec Z\prec A\prec M$	0.87
Visible-forehead	$J\prec Z\prec M\prec S\prec A\sim C\sim H\sim V$	$J\prec Z\prec M\prec C\prec A\prec H$	0.89
Bushy-eyebrows	$M\prec S\prec Z\prec V\prec H\prec A\prec C\prec J$	$S\prec M\prec Z\prec A\prec H\prec C$	0.73
Narrow-eyes	$M\prec J\prec S\prec A\prec H\prec C\prec V\prec Z$	$M\prec A\prec J\prec H\prec C$	0.80
Pointy-nose	$A\prec C\prec J\sim M\sim V\prec S\prec Z\prec H$	$A\prec M\prec V\prec J\prec H$	0.84
Big-lips	$H\prec J\prec V\prec Z\prec C\prec M\prec A\prec S$	$H\prec J\prec V\prec M\prec A$	1
Round-face	$H\prec V\prec J\prec C\prec Z\prec A\prec S\prec M$	$V\prec J\prec Z\prec A\prec S$	1

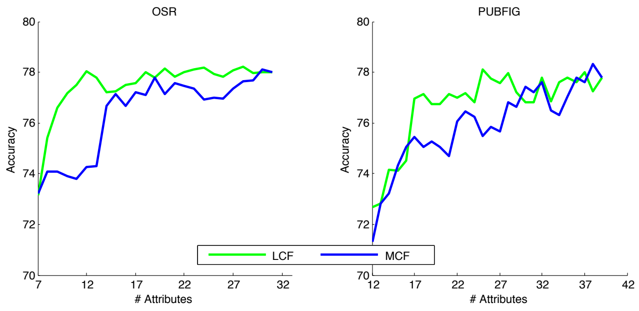

Fig. 4. Multi-class classification by adding unsupervised relative attributes according to the order of their correlation to the labeled relative attributes: LCF - least correlated is added first; MCF - most correlated is added first

where n_c and n_d are the number of concordant and discordant pairs between the two orderings respectively and n is the total number of pairs. The range of τ is in $[-1, 1]$ and it is 1 if the two orderings are the same (-1 if reversely the same). Considering anti-correlation, we used its absolute value: $\hat{\tau} = |\tau|$. Learned class orderings may contain only a subset of classes and in these cases we compute the correlation between the learned class ordering and the corresponding sub-list of the labeled class ordering.

Table 1 shows the results and from it we can see that for all of the human-labeled relative attributes, there are automatically learned relative attributes that are highly correlated with them. One potential application is to learn relative attributes using our unsupervised algorithm before labeling semantic relative attributes, and use these learned attributes as initial guidance for annotation. Furthermore, we observe that some learned relative attributes are not highly correlated with any of the human-labeled relative attributes. These may correspond to attributes that were overlooked by the annotator or hard to be concisely described. We now test the hypothesis that these attributes are indeed useful and can complement the human-labeled relative attributes. In the multi-class classification setting (Sec. 3.1), we start from using all SATs and add UATs one by one, according to their correlation to labeled semantic attributes: (a) most correlated first (MCF); (b) least correlated first (LCF). Fig. 4 shows the results, which validate our hypothesis.

4 Conclusion

We propose a formulation that can efficiently learn relative attributes and infer the training class orderings with respect to each of the learned attributes. The formulation also considers an attribute's relevance to each training class. In our experiments, the learned relative attributes are discriminative and provide good generalizability and classification performance. Compared to supervised learning of relative attributes [5], our algorithm does not need human labeling of attributes and can also learn useful attributes that may be difficult to describe with concise labels. Our method also finds attributes that are highly correlated with all the semantic attributes identified by humans for the datasets tested. The present formulation learns relative attributes at the class level. An interesting direction for future work is learning relative attributes at the instance level.

Acknowledgments. This work was supported in part through US NSF grants 0855065, 0910908, and 1029430.

References

1. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: CVPR (2009)
2. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.A.: Describing objects by their attributes. In: CVPR (2009)
3. Farhadi, A., Endres, I., Hoiem, D.: Attribute-centric recognition for cross-category generalization. In: CVPR (2010)
4. Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K.: Attribute and simile classifiers for face verification. In: ICCV (2009)
5. Parikh, D., Grauman, K.: Relative attributes. In: ICCV (2011)
6. Mahajan, D.K., Sellamanickam, S., Nair, V.: A joint learning framework for attribute models and object descriptions. In: ICCV (2011)
7. Wang, Y., Mori, G.: A Discriminative Latent Model of Object Classes and Attributes. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 155–168. Springer, Heidelberg (2010)
8. Yu, X., Aloimonos, Y.: Attribute-Based Transfer Learning for Object Categorization with Zero/One Training Example. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 127–140. Springer, Heidelberg (2010)
9. Wang, G., Forsyth, D.A.: Joint learning of visual attributes, object classes and visual saliency. In: ICCV (2009)
10. Ferrari, V., Zisserman, A.: Learning visual attributes. In: NIPS (2007)
11. Berg, T.L., Berg, A.C., Shih, J.: Automatic Attribute Discovery and Characterization from Noisy Web Data. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 663–676. Springer, Heidelberg (2010)
12. Rohrbach, M., Stark, M., Szarvas, G., Gurevych, I., Schiele, B.: What helps where - and why? semantic relatedness for knowledge transfer. In: CVPR (2010)
13. Parikh, D., Grauman, K.: Interactively building a discriminative vocabulary of nameable attributes. In: CVPR (2011)
14. Kovashka, A., Vijayanarasimhan, S., Grauman, K.: Actively selecting annotations among objects and attributes. In: ICCV (2011)
15. Torresani, L., Szummer, M., Fitzgibbon, A.: Efficient Object Category Recognition Using Classemes. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 776–789. Springer, Heidelberg (2010)
16. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. JMLR 9 (2008)
17. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV 42(3) (2001)
18. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM TIST 2, 27:1–27:27 (2011), Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`