# Predicting Multiple Attributes via Relative Multi-task Learning

Lin Chen, Qiang Zhang, Baoxin Li
Arizona State University, Tempe, AZ
lchen109, qzhang53, baoxin.li@asu.edu

## Abstract

*Relative attributes learning aims to learn ranking functions describing the relative strength of attributes. Most of current learning approaches learn ranking functions for each attribute independently without considering possible intrinsic relatedness among the attributes. For a problem involving multiple attributes, it is reasonable to assume that utilizing such relatedness among the attributes would benefit learning, especially when the number of labeled training pairs are very limited. In this paper, we proposed a relative multi-attribute learning framework that integrates relative attributes into a multi-task learning scheme. The formulation allows us to exploit the advantages of the state-of-the-art regularization-based multi-task learning for improved attribute learning. In particular, using joint feature learning as the case studies, we evaluated our framework with both synthetic data and two real datasets. Experimental results suggest that the proposed framework has clear performance gain in ranking accuracy and zero-shot learning accuracy over existing methods of independent relative attributes learning and multi-task learning.*

## 1. Introduction

Recent literature has witnessed fast development of the new methodology of relative attribute learning, whose goal is to overcome the limitation of traditional learning approaches based only on binary labels. In general, a traditional learning approach using binary labels can only map low-level features to one of the two labels, without capturing the "relativeness" of the concepts that the labels are supposed to represent. For example, in Figure 1, we may see that 1(a) is "natural" and 1(c) is "man-made", but we may be less certain on assigning either of the labels to 1(b). Unlike learning with binary labels, relative attributes learning is to capture the strength of the attributes under consideration. For example, this would allow us to say 1(b) is less "natural" but more "man-made" than 1(a) while being more "natural" but less "man-made" than 1(c).

Many practical applications involve multiple attributes

(a) (b) (c)

Figure 1. An example of relative attributes. Considering binary label learning, (a) is labeled as "natural" and (c) is labeled as "man-made", however, it is hard to lable (b) as "natural" or "man-made." In relative attributes, (a) is more "natural" and "open" than (b) and (b) is more "natural" and "open" than (c).

(like the two concepts, "natural" and "man-made", in the above image labeling example). Current relative attributes learning approaches train separate ranking functions independently for each of the attributes under consideration. For a given problem, if multiple attributes are involved, they usually exhibit correlation among them. For example, the more "natural" a scene image is, the more "open" it may be, where two attributes "being natural" and "being open" often have positive correlation. In addition, even if two attributes are disjointed in the high-level semantic space, in a practical algorithm they may be dependent of some common low-level features, and thus are made to be related to each other in some sense. Both factors suggest that the correlation among different attributes of the same problem should be dealt with in a principled way for effective relative attributes learning.

To exploit potential correlation among multiple attributes for learning better ranking functions, in this paper, we employ multi-task learning (MTL) in relative attributes learning and propose a new multi-attribute relative learning framework. MTL is a machine learning approach that learns several tasks simultaneously for potential performance gain through utilizing "relatedness" among different tasks, which provides a principled way for us to model correlation among attributes if we view the attributes as tasks. In the proposed framework, a new cost function is defined to capture the joint effect of the individual objective func-

tions in original relative attribute learning. Further, a regularization term is introduced to model the potential correlation among the attributes. As a result, the proposed framework could learn the relative strength of the attributes simultaneously while utilizing the correlation among the attributes/tasks. Under this framework, we developed an algorithm employing Block Coordinate Descent principles. Our algorithm solves the learning problem through alternating optimization steps dealing with capturing the relative ranking information and the attribute correlation information iteratively. The proposed approach has been tested on both synthetic data and two real datasets, with comparison with results from the state-of-the-art approaches of relative attributes learning and MTL.

The key contribution of this work lies in a novel formulation of relative attributes learning that handles multiple attributes jointly to capture the potential correlation among them for improved learning performance. Additionally, an algorithm is developed to find a solution under the formulation. As demonstrated by our experiments, the proposed method is able to deliver good performance even with a small number of training pairs, owing to its ability to exploit correlation among the attributes.

In the remaining of the paper, we first discuss related work in Section 2. The proposed approach is presented in Section 3. Experiments and results are demonstrated in Section 4. We concluded the paper in Section 5.

**Notations:** In this paper, we represent scalars, vectors, matrices and sets as lower case letters $x$, bold face lower case letters $\boldsymbol{x}$, capital letters $X$ and calligraphic capital letters $\mathcal{O}$ respectively. $\boldsymbol{x}_i$ denotes the $i$-th column of the matrix $X$. $\|\cdot\|$ and $\|\cdot\|_F$ represent Euclidean and Frobenius norms respectively. $\|X\|_{p,q}$ is defined as the $\ell_{p,q}$ norm $(\sum_i((\sum_j x_{ij}^q)^{\frac{1}{q}})^p)^{\frac{1}{p}}$. $\|X\|_* = \sum_{i=1}^r \sigma_i(X)$ is the trace norm, with $r = rank(X)$ and $\sigma_i(X)$ the $i$-th non-zero singular value in non-increasing order.

## 2. Related Work

As the work is mostly related to multi-task learning and relative attributes learning, we briefly review the literature on these two approaches in the following.

### 2.1. Multi-task Learning

Multi-task learning aims to improve generalization performance by training several tasks together to capture their intrinsic correlation. Various types of MTL approaches and applications have been proposed. Neural network approaches [4][6][20] utilized a hidden layer with a few nodes and a set of network weights shared by all tasks. Hierarchical Bayes approach [3][21][22][23] enforced task relatedness through a common prior probability distribution on the tasks' parameters.

In recent years, more attention has been paid to regularization-based multi-task learning, which is what we mainly considered in this work. The general form of regularization-based MTL is:

$$\min_W \Big( \sum_{t=1}^{t'} \sum_{i=1}^{n} (y_{ti} - W_t^T x_{ti})^2 + \lambda \Omega(W) \Big) \qquad (1)$$

where $t$ denotes the $t$-th task and $i$ denotes the $i$-th sample in task $t$. Much work has been proposed, often introducing different cost functions and regularization terms.

Evgeniou and Pontil [16] assumed that the projection vectors of all tasks are close to each other and proposed the regularization term using a shared mean vector $\boldsymbol{w}_0$ and a small perturbation vector $\boldsymbol{v}_t$ to represent the projection vector of the $t$-th task $\boldsymbol{w}_t = \boldsymbol{w}_0 + \boldsymbol{v}_t$. This idea is intuitive and easy to implement, but the assumption is too strong to hold in real applications. Ando *et al*. [1] proposed Alternating Structure Optimization (ASO) based on a similar assumption that the projection model is the sum of a task specific component and a shared low dimensional subspace.

Processing high-dimensional feature datasets attracts a lot of research interests. Considering the task relatedness that different task models share a common set of features, Jacob *et al*. [10] and Liu *et al*. [15] introduced $\ell_1/\ell_q$-norm group lasso penalty as regularization to obtain a sparse projection matrix for feature selection. Ji and Ye [12] introduced trace norm as regularization and obtained a low-rank structure projection matrix to capture task relatedness. These approaches make the strong assumption that all tasks are related.

Considering the existence of outlier tasks, Jalali *et al*. [11] and Gong *et al*. [9] introduced an extra $\ell_1$ and $\ell_1/\ell_q$-norm regularization term individually into feature selection; Chen *et al*. [7] introduced an extra $\ell_1/\ell_q$-norm regularization term into low-rank subspace learning. These approaches learn a projection matrix as well as detect the outlier tasks.

Other multi-task learning approaches include assumptions that tasks have some special structure. For example, in [2][24], tasks in the same group are closer to each other than tasks in a different group; in [13], tasks from the same node are closer to each other and relatedness among the nodes depends on the depth in a tree; in [8], task relatedness depends on the edge weight between the two tasks in a graph representation.

The above MTL approaches are in general for classification, and there is little work on extending them for ranking applications. Note that, conceptually, one may use a MTL-based classifier for a ranking problem, if binary labels are also provided. This is the MTL method to be compared in our experiments. Such an approach is obviously unable to

employ the relative information given in the relative labels. Our proposed work attempts to learn a ranking function capturing multi-attribute/task correlation when only relative labels are available.

### 2.2. Relative Attributes Learning

Relative attribute learning is a fairly recent concept, which has drawn increasing attention. Relative attributes were first used by Parikh and Grauman [14] to learn a ranking function for each human-nameable attribute of an image. The relative "strength" of an attribute is measured by some distance metrics learned through SVM-like optimization using (relatively) labeled pairs. Relative attribute learning is applicable to zero-shot learning (detecting 'unseen' category) and image description in relative terms.

Parkash and Parikh [19] incorporated attribute feedback into the classification process. Employing attributes as the communication "language" between the human supervisor and the machine learner, their work allows supervisors to provide feedback to the learner for improved learning. Kovashka *et al*. [14] presented a feedback scheme for image search. Based on pre-trained relative attribute ranking functions, their system demonstrates an initial set of queried results and asks the user to provide relative attribute feedback. The system then updates the training set based on the feedback and provides new queried images utilizing newly trained relative attribute ranking functions.

Most of current relative attribute learning approaches only consider ranking attributes independently. The proposed work attempts to explicitly model potential correlation among the attributes of interest so as to achieve better ranking performance, especially when limited training data are available (and thus each individual attribute may have even fewer labeled pairs of training samples).

## 3. Proposed Approach

In this section, we first present the proposed formulation for relative multi-attribute learning that attempts to capture potential correlation among given attributes through a multi-task learning framework (Sect. 3.1), and then present an algorithm for finding solutions under this formulation (Sect. 3.2).

### 3.1. A Relative Multi-attribute Learning Framework

With the reasonable assumption that multiple attributes describing the same object should be related in some way and that only relatively-labelled data pairs are given, we propose to jointly learn multi-attribute ranking functions in the following general formulation of an optimization problem:

$$\begin{aligned}
\min_{W,\xi,\gamma} \Big( & \sum_{t=1}^{t'} \big(\frac{1}{2}\left\|\boldsymbol{w}_t\right\|^2 + \rho_1 \sum_{i,j\in\mathcal{O}} \xi_{ijt} + \rho_2 \sum_{i,j\in\mathcal{S}_t} \gamma_{ijt}\big) \\
& + \mu\Omega(W)\Big) \\
s.t.\ & \boldsymbol{w}_t^T(\boldsymbol{x}_i - \boldsymbol{x}_j) \geq 1 - \xi_{ijt}; \forall (i,j) \in \mathcal{O}_t; \\
& \left|\boldsymbol{w}_t^T(\boldsymbol{x}_i - \boldsymbol{x}_j)\right| \leq \gamma_{ijt}; \forall (i,j) \in \mathcal{S}_t; \\
& \xi_{ijt} \geq 0; \gamma_{ijt} \geq 0; \\
& t = 1, 2, ..., t'.
\end{aligned} \tag{2}$$

In this formulation, $W$ is the projection matrix with the $t$-th column $\boldsymbol{w}_t$ as the projection vector for the $t$-th attribute (task), $\Omega(W)$ is a regularization term, $\boldsymbol{x}_i$ is the feature vector of the $i$-th sample, $\mathcal{O}_t = \{(i,j)\}$ is the set of ordered pairs $(i,j)$ satisfying $\boldsymbol{w}_i^T\boldsymbol{x}_i > \boldsymbol{w}_j^T\boldsymbol{x}_j$, $\mathcal{S}_t$ is the set of similar pairs $(i,j)$ satisfying $\boldsymbol{w}_i^T\boldsymbol{x}_i \approx \boldsymbol{w}_j^T\boldsymbol{x}_j$, $\rho_1$, $\rho_2$ and $\mu$ are trade-off constants, $\xi_{ijt}$ and $\gamma_{ijt}$ are slack variables measuring the error of the distance of prior and similar pairs. By applying appropriate regularization terms, the attribute projection model $W$ is learned simultaneously.

Some existing MTL frameworks only consider the correlation among the tasks, but ignore potential outliers. They brutally enforce all tasks to be similar, though they may be not. In this study, we adopted the same regularization scheme as in [9] which is more robust to such outliers and effectively achieves joint feature learning based on the assumption that the same set of essential features may be shared across different attributes with existence of outlier tasks. This results in the following specialized problem

$$\begin{aligned}
\min_{W,\xi,\gamma} \Big( & \sum_{t=1}^{t'} \big(\frac{1}{2}\left\|\boldsymbol{w}_t\right\|^2 + \rho_1 \sum_{i,j\in\mathcal{O}} \xi_{ijt} + \rho_2 \sum_{i,j\in\mathcal{S}_t} \gamma_{ijt}\big) \\
& + \mu_1\|P\|_{1,2} + \mu_2\left\|Q^T\right\|_{1,2}\Big) \\
s.t.\ & W = P + Q; \\
& \boldsymbol{w}_t^T(\boldsymbol{x}_i - \boldsymbol{x}_j) \geq 1 - \xi_{ijt}; \forall (i,j) \in \mathcal{O}_t; \\
& \left|\boldsymbol{w}_t^T(\boldsymbol{x}_i - \boldsymbol{x}_j)\right| \leq \gamma_{ijt}; \forall (i,j) \in \mathcal{S}_t; \\
& \xi_{ijt} \geq 0; \gamma_{ijt} \geq 0; \\
& t = 1, 2, ..., t'.
\end{aligned} \tag{3}$$

where the first regularization term enforces a group Lasso penalty on row groups of $P$ in order to capture the shared features among the attributes. The second term enforces the same group Lasso penalty, but on column groups of $Q$ to discover the outlier tasks.

It should be noted that although joint-feature-learning regularization is adopted in this work as a case study, other types of regularization term could also be used. For example, one may choose to impose a low-rank constraint or some graph-based structure on the attributes, if the problem

warrants such assumptions. But the essence of the problem, to capture the potential correlation among the attributes, remains the same.

### 3.2. An Optimization Algorithm

We now turn to the problem of finding an solution under the proposed formulation. Without loss of generality, our following discussion is in terms of a general regularization term $\Omega(W)$. In general, solving the constrained optimization problem of function (2) is difficult especially since common multi-task regularization terms are typically non-differentiable. In this study, we propose an algorithm based on Block Coordinate Descent (BCD) principles. In this approach, we introduce a slack variable $\tilde{W}$ which is similar to $W$ so that the original problem may be solved by two alternating processes, focusing on a new cost function and the regularization term respectively. That is, we first convert the original problem into

$$\min_{W,\tilde{W},\xi,\gamma} \Big( \sum_{t=1}^{t'} (\frac{1}{2} \|\boldsymbol{w}_t\|^2 + \rho_1 \sum_{i,j\in\mathcal{O}} \xi_{ijt} + \rho_2 \sum_{i,j\in\mathcal{S}_t} \gamma_{ijt}) + \lambda \Big\| W - \tilde{W} \Big\| + \mu\Omega(\tilde{W}) \Big) \tag{4}$$

in which the norm $\left\| W - \tilde{W} \right\|$ enforces a similar solution of $W$ and $\tilde{W}$.

We divide ranking and task coupling into separate steps by iteratively updating $W$ and $\tilde{W}$ in the following two separate problems:

**Optimization of $W$** For a fixed $\tilde{W}$, the optimal $W$ can be obtained via solving:

$$\begin{aligned} \min_{W,\xi,\gamma} \Big( & \sum_{t=1}^{t'} (\frac{1}{2} \|\boldsymbol{w}_t\|^2 + \rho_1 \sum_{i,j\in\mathcal{O}_t} \xi_{ijt} + \rho_2 \sum_{i,j\in\mathcal{S}_t} \gamma_{ijt}) \\ & + \frac{\lambda}{2} \Big\| W - \tilde{W} \Big\|_F^2 \Big) \\ s.t.\ \ & \boldsymbol{w}_t^T (\boldsymbol{x}_i - \boldsymbol{x}_j) \geq 1 - \xi_{ijt}; \forall (i,j) \in \mathcal{O}_t; \\ & \left| \boldsymbol{w}_t^T (\boldsymbol{x}_i - \boldsymbol{x}_j) \right| \leq \gamma_{ijt}; \forall (i,j) \in \mathcal{S}_t; \\ & \xi_{ijt} \geq 0; \gamma_{ijt} \geq 0; \\ & t = 1, 2, ...t'. \end{aligned} \tag{5}$$

where we used the Frobenius norm on $\left\| W - \tilde{W} \right\|$ for facilitating the solution. This problem focuses on capturing relative ranking information by encoding multi-attribute information into one quadratic optimization process. The second term enforces the projection weight matrix $W$ to be close to the given "multi-task" weight matrix $\tilde{W}$.

**Optimization of $\tilde{W}$** For a fixed $W$, the optimal $\tilde{W}$ can be obtained via solving:

$$\min_{\tilde{W}} \Big( \Big\| \tilde{W}^T - W^T \Big\| + \mu\Omega(\tilde{W}) \Big) \tag{6}$$

This problem enforces a joint learning regularization constraints $\Omega(\tilde{W})$ to the projection weight matrix to capture the correlation information among the attributes. The first term penalizes the difference to make sure the learned "multi-task" weight matrix $\tilde{W}$ is close to the given projection weight $W$.

The overall optimization algorithm is summarized in Table 1.

**Algorithm 1: Alternating Optimization**

**Input**: Data feature set $X$, training ranking pairs set $E$ (prior) and $F$ (similar), parameters $\rho_1$, $\rho_2$, $\lambda$, $\tilde{\lambda}$, $\mu$.
**Output**: Projection matrix $M$.

1: Initiate $\tilde{W}$ as random matrix, $W$ as zero matrix, $\lambda = 0.05\tilde{\lambda}$;
2: **while** $\frac{\|W-\tilde{W}\|_F^2}{\|W\|_F^2} > 10^{-10}$, **do**:
3: Optimize function (5), update matrix $W$;
4: Optimize function (6), update matrix $\tilde{W}$;
5: Set $\lambda = \lambda + 0.05\tilde{\lambda}$;
6: **end while**

Table 1. Projection Matrix Alternating Optimization Algorithem

In implementation, the first problem given in (5) can by solved by first converting it to its dual form problem, which is a typical quadratic optimization problem. While interested readers may find the derivation in the supplemental material, we list the dual form below for completeness:

$$\begin{aligned} \min_{a_{st},d_{st}} & \Big( \frac{1}{2} \boldsymbol{x}^T Y^T Y \boldsymbol{x} + (\lambda Y^T \tilde{\boldsymbol{w}} - (1+\lambda)\boldsymbol{e})^T \boldsymbol{x} \Big) \\ s.t.\ & A\boldsymbol{x} \leq \boldsymbol{b} \\ with\ & \boldsymbol{x} = (\boldsymbol{x}_1; \boldsymbol{x}_2; \cdots; \boldsymbol{x}_t)^T; \\ & Y = \Sigma(Y_t); \\ & \tilde{\boldsymbol{w}} = [\tilde{\boldsymbol{w}}_1; \tilde{\boldsymbol{w}}_2; \cdots; \tilde{\boldsymbol{w}}_t]; \\ & \boldsymbol{e} = [\boldsymbol{e}_1; \boldsymbol{e}_2; \cdots; \boldsymbol{e}_t]; \\ & A = [E_{|Y|\times|Y|}; -E_{|Y|\times|Y|}]; \\ & \boldsymbol{b}_t = [\underbrace{\rho_1, \rho_1, \cdots, \rho_1}_{|\mathcal{O}_t|}, \underbrace{\rho_2, \rho_2, \cdots, \rho_2}_{|\mathcal{S}_t|}]^T. \\ & \tilde{\boldsymbol{b}}_t = [\underbrace{0, 0, \cdots, 0}_{|\mathcal{O}_t|}, \underbrace{\rho_2, \rho_2, \cdots, \rho_2}_{|\mathcal{S}_t|}]^T. \\ & \boldsymbol{b} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_t, \tilde{\boldsymbol{b}}_1, \tilde{\boldsymbol{b}}_2, \cdots, \tilde{\boldsymbol{b}}_t]; \\ & t = 1, 2, \cdots, t'. \end{aligned} \tag{7}$$

In essence, the problem of (5) is similar to regular relative attribute learning, and the problem of (6) is similar to MTL, and thus there convergence behavior is well-understood. In our implementation, to facilitate convergence, we set a small value for $\lambda$ in Equation (5) at the beginning. Then in each iteration afterwards, we increase $\lambda$ gradually until it reaches a specified large threshold.

Therefore, the weight of the second term becomes larger and larger which ensures the cost $\left\|W - \tilde{W}\right\|_F^2$ would decrease after each iteration. The algorithm terminates when $W \approx \tilde{W}$ is reached.

We have 4 (since $\lambda$ and $\tilde{\lambda}$ are correlated) hyper parameters, all having limited search space. $\lambda$ mainly enforces the proper convergence and doesnt impact much on ranking. Experiments also showed $\rho 1$ and $\rho 2$ do not influence ranking result much. These parameters is selected via cross-validation. Specifically, we first find a suitable parameter search space by binary search or subgradient approach. For example, $\mu$ can be searched in a space ranging from achieving a desired minimal sparsity to a maximal sparsity. Then we adjust the parameters one by one while fixing the other parameters according to the performance of cross-validation.

## 4. Experiments

In this section, we tested our proposed framework in one synthetic dataset and two real datasets. We first experimented on synthetic dataset to show how well the correlation among the attributes are captured in our new proposed attribute learning framework. Then we test the framework on two real datasets including **Outdoor Scene Recognition (OSR) Dataset** [17] and **Shoes** [5]. We compare our framework with two alternative approaches. The first approach is relative attribute [18] which learns a ranking function for each attribute independently. The second approach is based on multi-task learning work [9] [7], by which we trained classifiers and used the classification score to rank the attributes. We tested both the ranking accuracy of learned ranking function and classification accuracy of zero-shot learning in the experiments.

We implemented the program on Matlab and employed the multi-task learning solver package **MALSAR** developed by Zhou *et al*. [25]. Hyper parameters $\mu_1$, $\mu_2$, $\rho_1$, $\rho_2$ and $\tilde{\lambda}$ are determined by cross validation as we discussed previously. Let $x_{ij}^t$ represents the $(i,j)$-th entry in the data matrix $X_t$ of the $t$-th attributes, where $i$ indexes $d$ dimensions and $j$ indexes $n$ data samples, we normalize the experiment data to satisfy:

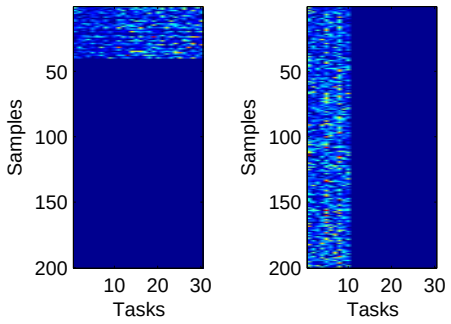
$$\sum_{j=1}^{n} \left(x_{ij}\right)^2 = 1; \forall i \in \mathcal{N}_d \tag{8}$$

### 4.1. Experiments with Synthetic Data

In order to test whether our framework can capture the relatedness among the attributes, we construct the synthetic datasets in the following way. The total attribute (task) number is $t = 30$. For the $i$-th attribute, we generate the data set $X_i \in \mathcal{R}^{d \times n}$ containing $n = 200$ samples and $d$ dimensions for each sample. Each entry of $X_i$ is drawn from the normal distribution $N(0, 25)$. The groundtruth projection matrices $P \in \mathcal{R}^{d \times n}$ and $Q \in \mathcal{R}^{d \times n}$ are drawn from $N(0, 64)$. We set the first 10 columns of $Q$ non-zero and they indicate outlier tasks. We also draw a noise vector $\boldsymbol{\delta}_i \in R^n$ from $N(0, 1)$. Thus, the final ranking score for data set $X_i$ is computed as $\boldsymbol{y}_i = X_i^T(P + Q) + \boldsymbol{\delta}_i$.

We run the experiments 4 rounds with the feature dimension $d$ increasing from 50 to 200 with step size 50. In the first round, all 50 dimensions are set as shared intrisic features, which means all 50 rows of $P$ are set non-zeros. Then 50 more zero rows are added into $Q$ in each round afterwards till $d$ reaches to 200. In this setup, the first 50 dimensions of feature (first 50 rows of $P$) represent the selected joint features among the attributes.

Through cross validation, during each round of our experiment the best ranking performance is always achieved while the first 50 dimensions are selected as joint features (the first 50 rows of learned projection matrix $P$ are non-zeros) and the first 10 attributes are detected as outliers (the first 10 columns of learned projection matrix $Q$ are non-zeros). Figure 2 demonstrates the learned projection matrices $P$ and $Q$ when $d$ reaches 200 as the parameters are set as $\mu_1 = 9.3$, $\mu_2 = 20.7$, $\rho_1 = \rho_2 = 300$, and $\tilde{\lambda} = 500$. The result shows that when $d = 200$, the first 50 rows of $P$ are selected as the joint features and the first 10 columns of $Q$ are detected as outlier attributes, which are all non-zeros. This result matches the groundtruth we have constructed previously, which suggests that our approach is able to capture the inherent relatedness of the projection model.




(a) $P$ (b) $Q$

Figure 2. Projection matrices $P$ (a) and $Q$ (b) learned by our framework on synthetic data. Blue color represents zero entry while other colors represent non-zero entry. Results show the first 50 rows of $P$ are selected as selected shared features and the first 10 columns of $Q$ are detected as outlier tasks.

### 4.2. Experiments with Real Data

We compare our framework with the baseline methods on the following two data sets:

**OSR** This dataset includes 2688 color outdoor scene images from 8 categories. There are totally 29,000 objects

with each image contains $256 \times 256$ pixels. Image features are described as the 512-dimensional gist descriptor. We used the same attributes and labels defined in [18].

**Shoes** This dataset includes 14765 images collected from like.com containing 10 categories of shoes. Same image features (960-dimensional gist descriptor plus 30-dimensional color histogram), attributes and labels are adopted from [14]. We randomly selected 6000 images (600 images per category) as our experiment data in this paper.

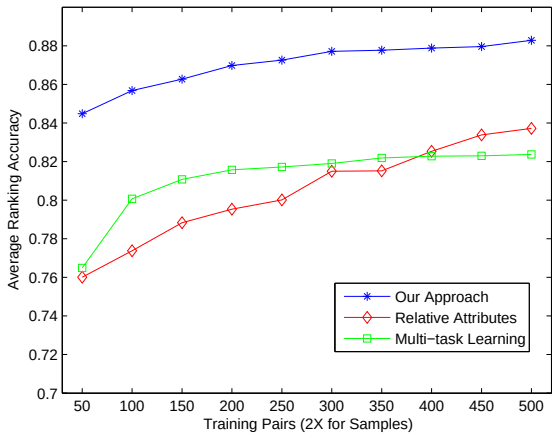
#### 4.2.1 Ranking Accuracy

We computed an average ranking accuracy (the frequency of correctly ranked pairs) by running 5 rounds of each implemented approach. By cross validation, parameters of our framework are set as $\mu_1 = 60$, $\mu_2 = 20$, $\rho_1 = \rho_2 = 300$, $\tilde{\lambda} = 400$ on **OSR** during which the projection matrix is learned after 17 iterations. On **Shoes**, parameters are set as $\mu_1 = 3$, $\mu_2 = 50$, $\rho_1 = \rho_2 = 300$, $\tilde{\lambda} = 500$ and the projection model is got through 15 iterations. For the baseline relative attributes approach, we adopted the same parameter setup which is reported in [18] as the optimal parameters.

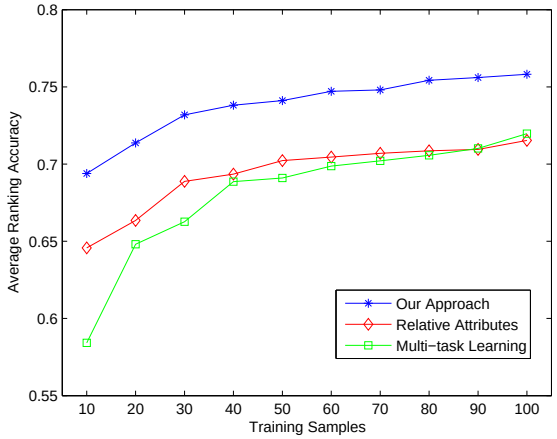
We first experimented the approaches on **OSR** dataset. Labeled training pairs are randomly left out for each attribute. The number of training pairs of each attribute increased from 50 to 500 with step size 50. For the baseline multi-task classification approach, we left 100 to 1000 training samples out for comparison. Since $n$ training pairs would select at most $2n$ training samples, the training set left for multi-task classification gains no less information than the other two ranking approaches. Figure 3(a) illustrates the average ranking accuracies as a function of increased number of training pairs with the similar standard deviation among three approaches around $\pm 1.2\%$. The result show that the accuracies of all three approaches increase with growing size of training data. The accuracy achieved by our framework (blue curve) outperforms the baseline results by $5\% \sim 11\%$. The best performance gain is achieved when the number of training pairs gets to 50. Table 2 details the ranking accuracies of all 6 attributes on **OSR** when the number of training pairs is 50. According to the result, other than "Depth-cloth", accuracies of all attributes achieved by our framework are obviously higher than the competing results and the best performance gain is $18\%$ in attribute "natural". We also analyzed on the $P$ and $Q$ matrices, where $P$ includes 150 shared dimensions and the outlier task in $Q$ is shown as the attribute "Size-large". This agrees with our observations that object in different sizes are randomly appeared in pictures of different classes.

The implemented approaches are then tested on **Shoes** in which a different training sets selection scheme is applied. Instead of leaving training pairs out, we left some training samples out (ranging from 10 to 100 in number), and the training pairs are selected merely from the left training set. Figure 3(b) depicts the average ranking accuracies as a function of the size of training data with similar standard deviation among three approaches around $\pm 0.6\%$. This experiment shows similarly that our proposed framework (blue curve) outperforms the other approaches by $4\% \sim 5\%$. The highest performance gain is got when the number of training samples is 10. Table 3 describes the ranking accuracies on **Shoes** in all 10 attributes when the 60 samples are left out for training. In all of the attributes, better ranking accuracies are achieved by our proposed framework. The best performance gain is $8.5\%$ in attribute "Pointy at the front".

Both of these two experiments show that the more lim-

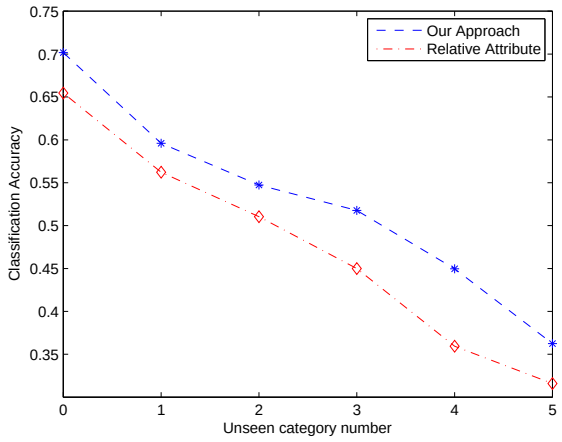



(a) **OSR**

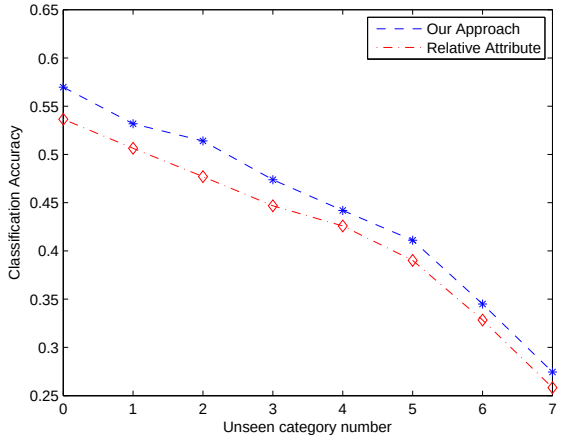



(b) **Shoes**

Figure 3. Average ranking accuracy of **OSR** and **Shoes** datasets as the increased number of training pairs and samples. Our framework (blue) outperforms the compared approaches by more than $5\%$ (450 pairs) to $11\%$ (50 pairs) on **OSR** and by more than $4\%$ (100 samples) to $5\%$ (10 samples) on **Shoes**.




(a) **OSR**




(b) **Shoes**

Figure 4. Classification accuracies of zero-shot learning on **OSR** and **Shoes**. The number of unseen categories increases from 0 to 5 for **OSR** and from 0 to 7 for **Shoes**. Our framework (blue) outperforms the competing approach (green) by $4\%$ to $9\%$ on **OSR** and by $2\%$ to $4\%$ on **Shoes**.

ited size of training dataset it is, the more benefits our proposed framework can gain from the relatedness among the attritutes.

#### 4.2.2 Zero-shot Learning

Finally, to show that the learned multi-attribute predictor captures intrinsically useful information for the underlying problem, we apply it to the task of zero-shot learning. Given training data from some 'seen' categories and some 'unseen' categories without any training data, zero-shot learning tries to learn a classifier to predict the category label of a new sample. We choose relative attribute as the comparing approach which has been shown to be the state-of-the-art work in [18]. We also adopted the same optimal parameters setup used in 4.2.1. We compute the average classification accuracies by running the experiment 5 rounds and in each round we randomly selected 400 training pairs for each seen categories to learn the projection model. Same as in [18], we also assumed the data follows Gaussian distribution model and estimated the mean $\mu$ and the covariance matrix $\Sigma$ through maximum likelihood estimation. Given a test image $i$ and its corresponding ranking score vector $\tilde{x}_i$, we assigned the category label according to the maximum likelihood.

For the estimation of $\mu$ and $\Sigma$ for unseen categories, we also adopted the similar schemes but added one more rule which we believe can better estimate the model: let $a_i^{(t)}$ and $a_j^{(t)}$ represent the $t$-th attribute value from the unseen category $i$ and seen category $j$, we set $\mu_i^{(t)} = \frac{1}{n}\sum_{j=1}^{n}\mu_j^{(t)}$ and $\Sigma_i^{(t)} = \frac{1}{n}\sum_{j=1}^{n}\Sigma_j^{(t)}$.

Figure 4 shows the classification accuracies of zero-shot learning on **OSR** and **Shoes**. For **OSR**, the number of unseen categories increases from 0 to 5 while the total category number is 8 and the parameters of seen categories are estimated by randomly selected 30 samples; for **Shoes**, the number of unseen categories increases from 0 to 7 while the total category number is 10 and the parameters of seen categories are estimated by randomly selected 100 samples. The unseen categories are also randomly selected during each test round for both datasets. The result shows that the classification accuracies decrease as the number of unseen category increasing for both two datasets. On **OSR**, the accuracy of our framework outperforms the competing approaches by $4\%\sim9\%$ and best performance gain got as the unseen category number is $4$. On **Shoes**, our classification accuracy is $2\%\sim4\%$ better than the results from the competing approach and the best performance gain is achieved when the unseen category number gets to 2.

## 5. Conclusions

In this paper, we proposed a framework for relative multi-attribute prediction through multiple task learning. By employing a multi-task learning framework for learning multiple attributes with only relative labels, our proposed framework is able to capture the intrinsic relatedness among the different attributes. The proposed method was evaluated on two public datasets **OSR** and **Shoes** with the comparison with the baseline approaches of relative attribute and multi-task learning. Through the experiments on image ranking and zero shot learning, we demonstrated that our method obviously outperforms the baseline methods in both ranking and classification capacities.

**Acknowledgement**: The work was supported in part by a grant (#1135616) from the National Science Foundation. Any opinions expressed in this material are those of the au-

| Attribute Name | Our Approach | Relative Attributes | Multi-task Learning |
|---|---|---|---|
| Natural | **90.42**% | 72.90% | 85.33% |
| Open | **88.62**% | 83.18% | 79.44% |
| Perspective | **83.64**% | 77.67% | 78.17% |
| Size-large | **80.15**% | 71.96% | 61.05% |
| Diagonal-plane | **84.08**% | 74.21% | 71.73% |
| Depth-cloth | 82.65% | 76.70% | **83.21**% |
| Average | **84.93**% | 76.10% | 76.49% |

Table 2. Ranking accuracies of each attribute on **OSR** when the number of training pairs are 50 of each attribute for our approach and relative attributes, 100 samples of each attribute for multi-task learning.

| Attribute Name | Our Approach | Relative Attributes | Multi-task Learning |
|---|---|---|---|
| Pointy at the front | **82.90**% | 74.52% | 72.10% |
| Open | **76.41**% | 72.52% | 65.33% |
| Bright in color | **56.55**% | 55.24% | 53.17% |
| Covered with ornaments | **67.66**% | 65.72% | 51.15% |
| Shiny | **78.59**% | 75.03% | 72.71% |
| High at the heel | **76.23**% | 70.67% | 70.87% |
| Long on the leg | **74.53**% | 71.91% | 64.60% |
| Formal | **73.59**% | 70.03% | 60.61% |
| Sporty | **79.88**% | 72.39% | 69.30% |
| Feminine | **81.51**% | 76.29% | 68.45% |
| Average | **74.79**% | 70.43% | 64.83% |

Table 3. Ranking accuracies of each attribute on **Shoes** when the 60 training samples of each attributes are left for training for each approach. Training pairs are generated from these 60 samples for our approach and relative attributes.

thors and do not necessarily reflect the views of the NSF.

## References


[1] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, Dec. 2005.

[2] F. R. Bach. Clustered multi-task learning: a convex formulation.

[3] B. Bakker and T. Heskes. Task clustering and gating for bayesian multitask learning. *J. Mach. Learn. Res.*, Dec. 2003.

[4] D. A. Baxter and J. H. Byrne. Simulator for neural networks and action potentials. November 2007.

[5] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *Proc.*, ECCV 2010.

[6] R. Caruana. Multitask learning, 1997.

[7] J. Chen, J. Zhou, and J. Ye. Integrating low-rank and group-sparse structures for robust multi-task learning. In *Proc.*, KDD '11, 2011.

[8] X. Chen, S. Kim, Q. Lin, J. G. Carbonell, and E. P. Xing. Graph-structured multi-task regression and an efficient optimization method for general fused lasso, 2010.

[9] P. Gong, J. Ye, and C. Zhang. Robust multi-task feature learning. In *Proc.*, KDD '12, 2012.

[10] L. Jacob and G. Obozinski. Group lasso with overlap and graph lasso.

[11] A. Jalali, P. D. Ravikumar, and S. Sanghavi. A dirty model for multiple sparse regression. *CoRR*, 2011.

[12] S. Ji and J. Ye. An accelerated gradient method for trace norm minimization.

[13] S. Kim and E. P. Xing. Tree-guided group lasso for multi-task regression with structured sparsity.

[14] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Image search with relative attribute feedback. In *Proc.*, CVPR 2012.

[15] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient l2,1-norm minimization, 2009.

[16] C. A. Micchelli and M. Pontil. Regularized multi-task learning. 2004.

[17] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 2001.

[18] D. Parikh and K. Grauman. Relative attributes. In *Proc.*, CVPR 2011, 2011.

[19] A. Parkash and D. Parikh. Attributes for classifier feedback. In *Proc.*, ECCV 2012. 2012.

[20] D. L. Silver and R. E. Mercer. The task rehearsal method of sequential learning.

[21] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 2007.

[22] K. Yu, V. Tresp, and S. Yu. A nonparametric hierarchical bayesian framework for information filtering. In *Proc.*, SIGIR '04, 2004.

[23] T. Zhang and J. S. Liu. Nonparametric hierarchical bayes analysis of binomial data via bernstein polynomial priors. *Canadian Journal of Statistics*, 2012.

[24] J. Zhou, J. Chen, and J. Ye. Clustered multi-task learning via alternating structure optimization. In *Advances in Neural Information Processing Systems 24*. 2011.

[25] J. Zhou, J. Chen, and J. Ye. *MALSAR: Multi-tAsk Learning via StructurAl Regularization*, 2011.