The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

From Common to Special: When Multi-Attribute Learning Meets Personalized Opinions

Zhiyong Yang,[1,2] Qianqian Xu,[1] Xiaochun Cao,[1,2] Qingming Huang [3,4] *
[1]SKLOIS, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]University of Chinese Academy of Sciences, Beijing, China
[4]Key Lab of Intell. Info. Process., Inst. of Comput. Tech., CAS, Beijing, China
{yangzhiyong, xuqianqian, caoxiaochun}@iie.ac.cn, qmhuang@ucas.ac.cn

Abstract

Visual attributes, which refer to human-labeled semantic annotations, have gained increasing popularity in a wide range of real world applications. Generally, the existing attribute learning methods fall into two categories: one focuses on learning user-specific labels separately for different attributes, while the other one focuses on learning crowd-sourced global labels jointly for multiple attributes. However, both categories ignore the joint effect of the two mentioned factors: the personal diversity with respect to the global consensus; and the intrinsic correlation among multiple attributes. To overcome this challenge, we propose a novel model to learn user-specific predictors across multiple attributes. In our proposed model, the diversity of personalized opinions and the intrinsic relationship among multiple attributes are unified in a common-to-special manner. To this end, we adopt a three-component decomposition. Specifically, our model integrates a common cognition factor, an attribute-specific bias factor and a user-specific bias factor. Meanwhile Lasso and group Lasso penalties are adopted to leverage efficient feature selection. Furthermore, theoretical analysis is conducted to show that our proposed method could reach reasonable performance. Eventually, the empirical study carried out in this paper demonstrates the effectiveness of our proposed method.

Introduction

Visual attributes, which describe human labeled properties (like *open*, *fashionable*) for a given image, have shown its great potential as a mid-level semantic cue to enhance a variety of applications including face verification (Song, Tan, and Chen 2014), person re-identification(Su et al. 2016; 2017), and zero-shot learning (Ji et al. 2017; Wang et al. 2017; Zhang and Saligrama 2016), *etc*. Generally speaking, there are two types of attributes: i) binary attributes express whether a property is absent or present in a given image (like *A is/is not open*); ii) relative attributes show the strength of an attribute conveyed in one image with respect to another image (like *A is more/similarly/less open than B*) (Parikh and Grauman 2011).

On one hand, the attribute predictors are often trained with the crowd-sourced global labels. The justification of such an approach is that there is only one unique ground truth and the majority of annotators have to recognize this *"correct answer"* simultaneously. However, different annotators might very well have distinct preferences, such that participants of the crowdsourced experiments might vote under different criteria or conditions. It might be misleading to merely look at a global consensus while ignoring personal diversity. On the other hand, practical visual applications often involve simultaneously learning multiple attributes together. In such a case, different attributes may intrinsically share some common patterns. One reason is that these attributes convey similar semantic meaning. Another reason is that they use common subsets of low level image features. In that sense, training multiple attribute predictors independently might not be an appropriate protocol.

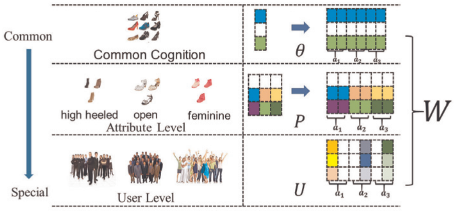


Figure 1: Illustration of the three-component decomposition. Here a_1, a_2 and a_3 are the three mentioned attributes: high heeled, open and feminine. We assume that, for each attribute, there are two annotators who labeled the corresponding images. Note that we extend $\boldsymbol{\theta}$ and $\boldsymbol{P}$ to match the size of $\boldsymbol{U}$.

Based on the discussion above, our goal is to solve two problems simultaneously in this paper: 1) learning user-specific attributes, 2) learning multiple attributes together with their shared information.

For 1), (Kovashka and Grauman 2013) regard user-specific attribute learning as an adaption process. In this work, a generate model is first trained based on a large pool of crowd-sourced labels. Then a small user-specific dataset is employed to adapt the generic model to user-specific predictors. Meanwhile, (Kovashka and Grauman 2015) argue that one attribute may fit to different shades (interpretations) for different groups of persons. Correspondingly, the authors proposed an automatic shade discovery method to leverage

*The corresponding author.

group-wise user-specific attributes. Though these existing works are designed to deal with user-specific attributes, they neglect the mutual interactions between different attributes.

Multi-task learning framework is well known as a standard solution for 2). Recently, many efforts have been made to improve multi-task learning. (Ando and Zhang 2005) proposed an alternating structure optimization algorithm to decompose the predictive model of each task into two components: the task-specific feature mapping and task-shared feature mapping. For robust multi-task learning, (Chen, Zhou, and Ye 2011) proposed a corresponding method with a low-rank structure and a column-wise sparse structure. In (Gong, Ye, and Zhang 2012), the low rank structure proposed in (Chen, Zhou, and Ye 2011) was replaced by a row-wise sparse structure to leverage selection of a common subset of features. Since the tasks from the same group are closer to each other than those from a different group, (Zhou, Chen, and Ye 2011) proposed a clustering based multi-task learning framework. Motivated by the fact that the tasks should be related in terms of subsets of features, (Xu et al. 2015) proposed a novel multi-task learning method via task-feature co-clustering. As for applications, (Huang et al. 2014) proposed a robust dynamic multi-task method for trajectory regression.

There are also some existing works that focus on applying multi-task frameworks to attribute learning. One typical way to do this is to extend the existed multi-task algorithms to match attribute learning. For instance, in (Chen, Zhang, and Li 2014) the model proposed in (Gong, Ye, and Zhang 2012) was generalized to learn multiple relative attributes with their shared information. Meanwhile, some works employ deep learning methods to solve this problem by partially sharing the learned weights among different attributes. (Ehrlich et al. 2016) proposed a multi-task restricted boltzman machine so as to learn a shared feature representation for multiple facial attribute learning. (Hsieh, Hsu, and Chen 2017) incorporates identity and human attributes in learning discriminative face representations through a multi-task method. A deep multi-task learning approach was proposed in (Han et al. 2017) to jointly estimate multiple heterogeneous attributes from a single face image. (Hand and Chellappa 2017) also proposed a multi-task deep convolutional neural network with an auxiliary network at the top to capture attribute relationships. Though these works have successfully improved attribute learning with multi-task models, as was mentioned previously in this section, they all employ crowdsourced labels to train attribute predictors and ignore the disagreement among users.

Note that, except learning global labels for multiple attributes, multi-task frameworks are also suitable for learning user specific attributes where global patterns are necessary for capturing the public opinion, and task-specific patterns are indispensable as well for capturing user bias toward that public opinion. With such belief in mind, different with most of the previous works which partially met the requirement of our goal, we propose a hierarchical multi-task framework where task relationships are modeled on both the attribute level and the user level.

The main contributions of this paper are two-fold:

- To match the hierarchical nature of the underlying problem, we propose a common-to-special decomposition of the model weights, which captures the general cognition patten, attribute level bias and user specific bias, respectively. An optimization method is established based on the accelerated proximal gradient method.
- Theoretical analysis is performed in this paper. The corresponding results show that our proposed algorithm could attain reasonable performance.

Methodology

In this section, we'll present an attribute learning method to learn user specific labels across multiple attributes. We first introduce the notations used in this paper. Secondly, we propose our model formulation, which includes a common-to-special decomposition of the model weights and the corresponding objective function. Thirdly, we introduce our optimization method based on the accelerated proximal gradient method. Finally, the theoretical analysis is carried out to show the performance bound of our method.

Notations

In this paper, scalars, vectors, and matrices are denoted as lowercase letters (a), bold lower case letters ($\boldsymbol{a}$), and bold upper case letters ($\boldsymbol{A}$). $\boldsymbol{X}_k$ denotes the kth row of $\boldsymbol{X}$. x_{ij} denotes the (i,j) entry of a matrix $\boldsymbol{X}$. $\mathbb{P}(\cdot)$ denotes a probability measure. $[a]$ denotes the set :$\{1, 2, \cdots, a\}$. Given an index set $\mathcal{I}$, $\boldsymbol{A}^{\mathcal{I}}$ denotes a matrix that contains all the corresponding rows of $\boldsymbol{A}$, while $\boldsymbol{a}^{\mathcal{I}}$ represents the vector that contains the corresponding elements of vector $\boldsymbol{a}$. $\|\cdot\|_p$ denotes the ℓ_p norm : $\|x\|_p = (\sum\limits_i x_i^p)^{(1/p)}$. $\|\cdot\|_{p,q}$ denotes the $\ell_{p,q}$ norm :$\Big(\sum\limits_i \big(\sum\limits_j (x_{ij}^q)^{(1/q)}\big)^p\Big)^{(1/p)}$. $\|\cdot\|$ denotes the ℓ_2 norm. $\langle\cdot,\cdot\rangle$ denotes the inner product for two matrices or two vectors. If $f(x) = o(g(x))$, we have $\lim\limits_{x \to +\infty} \dfrac{f(x)}{g(x)} = 0$. $\oplus$ denotes the direct sum of two matrices.

Model Formulation

Assume that we have n_a attributes to be evaluated, and that, for the ith attribute, we are given user-specific labels from n_{u_i} different workers. Then the training data could be represented as:

$$\mathcal{T} = \Big\{(\boldsymbol{X}^{(1,1)}, \boldsymbol{y}^{(1,1)}), \cdots, (\boldsymbol{X}^{(n_a,n_{u_{n_a}})}, \boldsymbol{y}^{(n_a,n_{u_{n_a}})})\Big\}$$

where n_{ij} is the number of images the jth user for the ith attribute labeled. The input feature is preprocessed such that

$$\sum_{k=1}^{n_{ij}} \left(x_{k,l}^{(i,j)}\right)^2 = 1 \qquad (1)$$

, where $x_{k,l}^{(i,j)}$ is the (k,l)th entry of $\boldsymbol{X}^{(i,j)}$.

For binary attributes, $\boldsymbol{X}^{(i,j)} \in \mathbb{R}^{n_{ij} \times d}$ is the corresponding feature matrix for the images that the jth user of the ith attribute labeled. Each row of $\boldsymbol{X}^{(i,j)}$ represents the low-level feature for a corresponding image. $\boldsymbol{y}^{(\boldsymbol{i},\boldsymbol{j})} \in \{-1,1\}^{n_{ij}}$ is the corresponding label vector [1]. If $y_k^{(i,j)} = 1$, then the user thinks that the corresponding attribute is present in the kth image, otherwise it will be labeled as -1.

For relative attributes, we need to solve a ranking problem. The corresponding users are given a set of image pairs $\{(\boldsymbol{x}_{1,k}^{(i,j)}, \boldsymbol{x}_{2,k}^{(i,j)})\}_{k=1}^{n_{ij}}$. Since we adopt linear models in this paper, we define the kth row of $\boldsymbol{X}^{(i,j)}$ as $\boldsymbol{X}_k^{(i,j)} = \boldsymbol{x}_{1,k}^{(i,j)} - \boldsymbol{x}_{2,k}^{(i,j)}$. For the kth pair $y_k^{(i,j)} = 1$ if the user thinks that the corresponding attribute has a stronger expression in the former image (*say 1 is more open than 2*); $y_k^{(i,j)} = 0$ if the user thinks that both images show similar strength for the current attribute (*say 1 is as open as 2*); $y_k^{(i,j)} = -1$ if the user thinks that the corresponding attribute has a weaker expression in the former image (*say 1 is less open than 2*).

As mentioned in the introduction section, our goal is to learn a predictor $\boldsymbol{f}^{(i,j)}$ for each of the personalized label vectors $\boldsymbol{y}^{(i,j)}$. In this paper, we assume that $f^{(i,j)}(\cdot)$ has a linear form :

$$\boldsymbol{f}^{(i,j)} = \boldsymbol{X}^{(\boldsymbol{i},\boldsymbol{j})}\boldsymbol{w}^{(\boldsymbol{i},\boldsymbol{j})} \tag{2}$$

where $\boldsymbol{w}^{(i,j)}$ is the corresponding model weight.

Now we are ready to introduce the modeling of $\boldsymbol{w}^{(i,j)}$. Note that our underlying problem could be comprehended in a common-to-special manner: there is a common pattern that captures the general cognition of a given object; an attribute-specific pattern is also necessary to express the attribute-level common pattern; finally, a user-specific factor is necessary to describe the personalized preference. As a result, we adopt a three-component additive decomposition of $\boldsymbol{w}^{(i,j)}$:

$$\boldsymbol{w}^{(i,j)} = \boldsymbol{\theta} + \boldsymbol{p}^{(i)} + \boldsymbol{u}^{(i,j)} \tag{3}$$

The practical meaning of these three components could be explained as follows:

- $\boldsymbol{\theta}$(General Cognition Factor): $\boldsymbol{\theta} \in \mathbb{R}^{d\times 1}$ is the global factor that captures the overall cognition for the given class of object (*say for shoes dataset, this factor captures the overall cognition about shoes*). θ is shared among all subtasks.
- $\boldsymbol{p}^{(i)}$(Attribute Specific Bias Factor): An attribute-specific factor that captures the bias of the i th attribute with respect to the global cognition. For mathematical convenience, we denote $\boldsymbol{P} = [\boldsymbol{p}^{(1)}, \cdots, \boldsymbol{p}^{(n_a)}]$, and we have $\boldsymbol{P} \in \mathbb{R}^{d\times n_a}$.
- $\boldsymbol{u}^{(i,j)}$ (User Specific Bias Factor): A user specific factor that captures the personal bias for the jth user of the ith attribute. In order to simplify the mathematical expressions, we define $\boldsymbol{U}^{(t)} = [\boldsymbol{u}^{(t,1)}, \cdots, \boldsymbol{u}^{(t,n_{u_t})}]$, $\boldsymbol{U} = [\boldsymbol{U}^{(1)}, \cdots, \boldsymbol{U}^{(n_a)}]$, thus we have $\boldsymbol{U} \in \mathbb{R}^{d\times n_u}$, where $n_u = \sum_{i=1}^{n_a} n_{u_i}$.

Since we adopt a linear form for $\boldsymbol{f}^{(i,j)}$, it is natural to assume that the real response $\boldsymbol{y}^{(i,j)}$ could be interpreted as the true predictor in our proposed model :${\boldsymbol{f}^*}^{(i,j)} = \boldsymbol{x}^{(i,j)}{\boldsymbol{w}^*}^{(i,j)}$ plus a Gaussian noise $\boldsymbol{\delta}^{(i,j)} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$:

$$\boldsymbol{y}^{(i,j)} = {\boldsymbol{f}^*}^{(i,j)} + \boldsymbol{\delta}^{(i,j)} \tag{4}$$

where ${\boldsymbol{w}^*}^{(i,j)} = \boldsymbol{\theta}^* + {\boldsymbol{p}^*}^{(i)} + {\boldsymbol{u}^*}^{(i,j)}$.

As for the objective function, we adopt the least square loss as our empirical loss: $L(\boldsymbol{W})$ and a general regularizer $\Omega(\cdot)$. We could thus formulate our problem as $(P1)$ as:

$$(P1): \ \min_{\boldsymbol{W}} \underbrace{\sum_{i=1}^{n_a}\sum_{j=1}^{n_{u_i}} \|\boldsymbol{y}^{(i,j)} - \boldsymbol{X}^{(i,j)}\boldsymbol{w}^{(i,j)}\|^2}_{L(\boldsymbol{W})} + \Omega(\boldsymbol{W})$$

where $\boldsymbol{W} := \{\boldsymbol{w}^{(i,j)}\}_{(i,j)}$ is the set of all parameters.

For our problem, user annotated labels are often limited. Furthermore, the low level features for an image are located in a high dimensional space. Then it is necessary to encourage sparse models to reduce model complexity. To preserve the relationship among subtasks, we also need to leverage a shared feature subset. Consequently, we penalize $\boldsymbol{\theta}$ and $\boldsymbol{P}$ with ℓ_1 norm and $\ell_{1,2}$ norm, respectively. Meanwhile, $\boldsymbol{u}^{(i,j)} \neq \boldsymbol{0}$ only when the corresponding user has a specific bias with respect to the popular opinion. We penalize $\boldsymbol{U}^\top$ with $\ell_{1,2}$ norm to leverage column-wise sparsity. Above all, $\Omega(\boldsymbol{W})$ could be represented as follows :

$$\Omega(\boldsymbol{W}) = \lambda_1\|\boldsymbol{\theta}\|_1 + \lambda_2\|\boldsymbol{P}\|_{1,2} + \lambda_3\|\boldsymbol{U}^\top\|_{1,2}$$

Putting all together, $(P1)$ could be reformed as :

$$\begin{aligned} \min_{\boldsymbol{W}} \sum_{i=1}^{n_a}\sum_{j=1}^{n_{u_i}} &\|\boldsymbol{y}^{(i,j)} - \boldsymbol{X}^{(i,j)}(\boldsymbol{\theta} + \boldsymbol{p}^{(i)} + \boldsymbol{u}^{(i,j)})\|^2 \\ &+ \lambda_1\|\boldsymbol{\theta}\|_1 + \lambda_2\|\boldsymbol{P}\|_{1,2} + \lambda_3\|\boldsymbol{U}^\top\|_{1,2} \end{aligned} \tag{5}$$

Figure 1 illustrates the expected structure of the three components in the proposed model. It could be seen that both the attribute level and the user level task correlations are included in our proposed model.

To end this section, we introduce two important mathematical properties of $(P1)$ as proposition 1 and proposition 2. Considering the limited space of this paper,please refer to our supplementary materials[2] for a detailed proof of proposition 1 and proposition 2.

Proposition 1 (Global Optimality). *$P1$ is jointly convex with respect to $\boldsymbol{\theta}$, $\boldsymbol{P}$, $\boldsymbol{U}$*

Proposition 2 (Lipschitz Continuous Gradient). *Given two arbitrary feasible solutions W and W', we have :*

$$\|\nabla L(\tilde{W}) - \nabla L(\tilde{W}')\| \leq \rho\|\tilde{W} - \tilde{W}'\|$$

where:

$$\rho = 6n_u\sqrt{(n_u + n_a + 1)}\max_{i,j}\left[\sigma_1\left(\boldsymbol{X}^{(i,j)}\right)\right]^2$$

[1] If not explained, $A^{(i,j)}$ denotes the corresponding variable of A for the jth user of the ith attribute; $A^{(i)}$ denotes the corresponding variable for the ith attribute.

[2] https://github.com/joshuaas/AAAI-18-Personalized-Multi-Attribute-Learning

$$\tilde{W} = [vec(\boldsymbol{\theta})^\top, vec(\boldsymbol{P})^\top, vec(\boldsymbol{U})^\top]^\top$$

$$\tilde{W}' = [vec(\boldsymbol{\theta'})^\top, vec(\boldsymbol{P'})^\top, vec(\boldsymbol{U'})^\top]^\top, n_u = \sum_{i=1}^{n_a} n_{u_i}$$

Optimization Method

According to proposition 1 and proposition 2, $L(\boldsymbol{W})$ is a convex and smooth function while $\Omega(\boldsymbol{W})$ is a convex non-smooth function. Similar as the related literatures (Beck and Teboulle 2009; Chen, Zhou, and Ye 2011; Gong, Ye, and Zhang 2012), the accelerated proximal gradient method is employed to solve $(P1)$.

According to proposition 1, $L(\cdot)$ is differentiable with Lipschitz continuous gradient. According to the basic mathematical properties of Lipschitz continuous functions, at iteration step k, for any reference point $\boldsymbol{W}^{ref_k} = (\boldsymbol{\theta}^{ref_k}, \boldsymbol{P}^{ref_k}, \boldsymbol{U}^{ref_k})$, $\exists \rho_k > 0$, such that :

$$\begin{aligned} L(\boldsymbol{W}) &\le L(\boldsymbol{W}^{ref_k}) + \left\langle \nabla_P L(\boldsymbol{W}^{ref_k}), \Delta \boldsymbol{P} \right\rangle \\ &+ \left\langle \nabla_\theta L(\boldsymbol{W}^{ref_k}), \Delta \boldsymbol{\theta} \right\rangle + \left\langle \nabla_U L(\boldsymbol{W}^{ref_k}), \Delta \boldsymbol{W} \right\rangle \\ &+ \frac{\rho_k}{2}\|\Delta \boldsymbol{P}\|_F^2 + \frac{\rho_k}{2}\|\Delta \boldsymbol{\theta}\|_2^2 + \frac{\rho_k}{2}\|\Delta \boldsymbol{U}\|_F^2 \\ &\overset{def}{=} \hat{L}_{\boldsymbol{W}^{ref_k},\rho}(\boldsymbol{W}) \end{aligned} \tag{6}$$

Following the Majorization-Minimization (MM) (Hunter 2004) scheme, at the k-th iteration, we could then solve $(P2)$ instead of updating the weights based on original problem:

$$(P2) : (\boldsymbol{\theta}^k, \boldsymbol{P}^k, \boldsymbol{U}^k) := \underset{\boldsymbol{\theta},\boldsymbol{P},\boldsymbol{U}}{\operatorname{argmin}} \hat{L}_{\boldsymbol{W}^{ref_k},\rho_k}(\boldsymbol{W}) + \Omega(W)$$

It is obvious that $\boldsymbol{\theta}, \boldsymbol{P}, \boldsymbol{U}$ are decoupled in $\hat{L}_{\boldsymbol{W}^{ref_k},\rho_k}(\boldsymbol{W})$. Hence, solving $(P2)$ is equivalent to solving the following three proximal subproblems simultaneously:

$$\boldsymbol{\theta}^k := \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{2}\left\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}^k\right\|^2 + \frac{\lambda_1}{\rho_k}\|\boldsymbol{\theta}\|_1 \tag{7}$$

$$\boldsymbol{P}^k := \underset{\boldsymbol{P}}{\operatorname{argmin}} \frac{1}{2}\left\|\boldsymbol{P} - \tilde{\boldsymbol{P}}^k\right\|^2 + \frac{\lambda_2}{\rho_k}\|\boldsymbol{P}\|_{1,2} \tag{8}$$

$$\boldsymbol{U}^k := \underset{\boldsymbol{U}}{\operatorname{argmin}} \frac{1}{2}\left\|\boldsymbol{U} - \tilde{\boldsymbol{U}}^k\right\|^2 + \frac{\lambda_3}{\rho_k}\|\boldsymbol{U}^\top\|_{1,2} \tag{9}$$

where

$$\tilde{\boldsymbol{\theta}}^k = \boldsymbol{\theta}^{ref_k} - \frac{1}{\rho_k}\nabla_\theta L(\boldsymbol{W}^{ref_k})$$

$$\tilde{\boldsymbol{P}}^k = \boldsymbol{P}^{ref_k} - \frac{1}{\rho_k}\nabla_P L(\boldsymbol{W}^{ref_k})$$

$$\tilde{\boldsymbol{U}}^k = \boldsymbol{U}^{ref_k} - \frac{1}{\rho_k}\nabla_{\boldsymbol{U}} L(\boldsymbol{W}^{ref_k})$$

All of these subproblems admit closed-form solutions :

$$\theta_i^k := sign(\tilde{\theta}_i^k)\left(\left|\tilde{\theta}_i^k\right| - \frac{\lambda_1}{\rho_k}\right)_+ \tag{10}$$

Algorithm 1: The accelerated proximal gradient method for solving $(P1)$

```
Input: $\mathcal{T}$, $\lambda_1$,$\lambda_2$,$\lambda_3$,$\rho_0 > 0$ ,$\eta > 1$
Output: $\boldsymbol{\theta}$,$\boldsymbol{P}$,$\boldsymbol{U}$
Initialize $\boldsymbol{\theta}^0$,$\boldsymbol{P}^0$,$\boldsymbol{U}^0$;
$\boldsymbol{\theta}^{ref} := \boldsymbol{\theta}^0$, $\boldsymbol{P}^{ref} := \boldsymbol{P}^0$,
  $\boldsymbol{U}^{ref} := \boldsymbol{U}^0$,$t_1 := 1$,$k := 1$;
while Not Converged do
    Solve $\boldsymbol{W}^k = (\boldsymbol{\theta}^k, \boldsymbol{P}^k, \boldsymbol{U}^k)$ with
      Eq.(10)-Eq.(12);
    Find the smallest $i_k$ such that when $\tilde{\rho} = \eta^{i_k}\tilde{\rho}_{k-1}$
      : $L(\boldsymbol{W^k}) \le \hat{L}_{\boldsymbol{W}^{ref_k},\tilde{\rho}}(\boldsymbol{W}^k)$ ;
    $\boldsymbol{\rho}_k := \tilde{\rho}$;
    update $\boldsymbol{W}^k$ again;
    $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$;
    $dt := \frac{t_k - 1}{t_{k+1}}$;
    $\boldsymbol{\theta}^{ref} := \boldsymbol{\theta}^k + dt(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{k-1})$;
    $\boldsymbol{P}^{ref} := \boldsymbol{P}^k + dt(\boldsymbol{P}^k - \boldsymbol{P}^{k-1})$;
    $\boldsymbol{U}^{ref} := \boldsymbol{U}^k + dt(\boldsymbol{U}^k - \boldsymbol{U}^{k-1})$;
    $k := k + 1$;
end
```

$$\boldsymbol{P}_i^k := \left(1 - \frac{\lambda 2}{\rho_k\|\tilde{\boldsymbol{P}}_i^k\|}\right)_+ \tilde{\boldsymbol{P}}_i^k \tag{11}$$

$$\left(\boldsymbol{u}^{(i,j)}\right)^k := \left(1 - \frac{\lambda 3}{\rho_k\left\|\left(\tilde{\boldsymbol{u}}^{(i,j)}\right)^k\right\|}\right)_+ \left(\tilde{\boldsymbol{u}}^{(i,j)}\right)^k \tag{12}$$

Furthermore, the nestrov's acceleration strategy is employed to update the reference point $\boldsymbol{W}^{ref}$. Integrating all the results, an efficient algorithm to solve $(P1)$ is introduced as Algorithm 1. According to the theoretical analyses proposed in (Beck and Teboulle 2009), $\boldsymbol{W}^k$ could converge to a global optimal solution with rate $\mathcal{O}(\frac{1}{k^2})$, which is the provable optimal rate for first-order methods.

Theoretical Analysis

Following (Gong, Ye, and Zhang 2012), we will propose the performance bound of our algorithm based on assumption 1. Considering the limited space of this paper, the detailed proofs are attached to our supplementary materials as mentioned before.

Here we define a set $\mathcal{N}(\boldsymbol{A})$ for a matrix (vector) $\boldsymbol{A}$ as the indexes for zero rows (entries): $\mathcal{N}(\boldsymbol{A}) = \{i : \boldsymbol{A}_i = \boldsymbol{0}\}$, and $\mathcal{N}_\perp(A)$ as the complement of $\mathcal{N}(\boldsymbol{A})$: $\mathcal{N}_\perp(A) = \{i : \boldsymbol{A}_i \neq \boldsymbol{0}\}$. $|\mathcal{N}(\boldsymbol{A})|$ is the number of zero rows (entries) of a matrix(vector) $\boldsymbol{A}$, and we have similar definition for $|\mathcal{N}_\perp(\boldsymbol{A})|$. Now we provide the basic assumption of the main result as assumption 1.

Assumption 1. *It is defined that :* $\boldsymbol{X} \overset{def}{=} \oplus_{i,j} \boldsymbol{X}^{(i,j)}$,

$$\bar{\boldsymbol{W}} \overset{def}{=} [\boldsymbol{w}^{(1,1)^\top}, \boldsymbol{w}^{(1,2)^\top}, \cdots, \boldsymbol{w}^{(n_a, n_{un_a})^\top}]^\top$$

Let $0 \leq n_\theta \leq d$ *be the upper bound of* $|\mathcal{N}_\perp(\boldsymbol{\theta}^*)|$, $0 \leq n_p \leq d$ *be the upper bound of* $|\mathcal{N}_\perp(\boldsymbol{P}^*)|$, $0 \leq n_{u,a} \leq n_u$ *be the upper bound of* $|\mathcal{N}_\perp(\boldsymbol{U^*}^\top)|$. *We assume that there exists nonnegative* κ_θ, κ_p, $\kappa_{u,a}$*:*

$$\kappa_\theta = \min_{\Gamma_\theta, \Gamma_P, \Gamma_U} \frac{\|X\Gamma_{\overline{W}}\|}{\sqrt{n_{min}n_u}\|\Gamma_\theta^{\mathcal{N}_\perp(\theta)}\|_2}$$

$$\kappa_p = \min_{\Gamma_\theta, \Gamma_P, \Gamma_U} \frac{\|X\Gamma_{\overline{W}}\|}{\sqrt{n_{min}n_u}\|\Gamma_{\boldsymbol{P}}^{\mathcal{N}_\perp(P)}\|_F}$$

$$\kappa_{u,a} = \min_{\Gamma_\theta, \Gamma_P, \Gamma_U} \frac{\|X\Gamma_{\overline{W}}\|}{\sqrt{n_{min}n_u}\|(\Gamma_U)^{\top\mathcal{N}_\perp(U^\top)}\|_F}$$

where $\Gamma_\theta \in \mathbb{R}^d$ *is a function of* $\boldsymbol{\theta}$*;* $\Gamma_P \in \mathbb{R}^{d\times n_a}$ *is a function of* $\boldsymbol{P}$*;* $\Gamma_U \in \mathbb{R}^{d\times n_u}$ *is a function of* $\boldsymbol{U}$. *Define* $\Gamma_{w^{(i,j)}} = \Gamma_\theta + \Gamma_{p^{(i)}} + \Gamma_{u^{(i,j)}}$, *then :* $\Gamma_{\overline{W}} = [\Gamma^\top_{w^{(1,1)}}, \Gamma^\top_{w^{(1,2)}}, \cdots, \Gamma^\top_{w^{(n_a,n_{un_a})}}]^\top$. *Furthermore, it is assumed that the following inequalities hold :* $\|\Gamma_{\boldsymbol{\theta}}^{\mathcal{N}(\theta)}\|_1 \leq \beta_\theta \|\Gamma_{\boldsymbol{\theta}}^{\mathcal{N}_\perp(\theta)}\|_1$, $\|\Gamma_{\boldsymbol{P}}^{\mathcal{N}(P)}\|_{1,2} \leq \beta_p \|\Gamma_{\boldsymbol{P}}^{\mathcal{N}_\perp(P)}\|_{1,2}$, $\|\Gamma_U^{\top\mathcal{N}(U^\top)}\|_{1,2} \leq \beta_{u,a}\|\Gamma_U^{\top\mathcal{N}_\perp(U^\top)}\|_{1,2}$. *where* β_θ, β_p, $\beta_{u,a}$ *are positive scalars.*

Note that the assumption on κ_θ, κ_p, $\kappa_{u,a}$ is based on the restricted eigenvalue assumption (Bickel, Ritov, and Tsybakov 2008), which has been widely used in existing multi-task literatures (Chen, Zhou, and Ye 2011; Gong, Ye, and Zhang 2012).

According to the notations in assumption 1, the squared error between the predicted value $\boldsymbol{x}^\top\boldsymbol{w}$ and the real value $\boldsymbol{f}$ could be formed as : $\|\boldsymbol{X}\overline{\boldsymbol{W}} - \boldsymbol{F}\|$, where $\boldsymbol{F}$ is defined as :

$$\boldsymbol{F} = [\boldsymbol{f^*}^{(1,1)^\top}, \boldsymbol{f}^{*(1,2)^\top}, \cdots, \boldsymbol{f}^{*(n_a, nu_{n_a})^\top}]^\top$$

Let $\hat{\boldsymbol{W}} = (\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{P}}, \hat{\boldsymbol{U}})$ be an optimal solution of $(P1)$. According to Eq.(4), we define $\boldsymbol{W}^* = (\boldsymbol{\theta}^*, \boldsymbol{P}^*, \boldsymbol{U}^*)$. Our main result could be presented as Theorem 1.

Theorem 1 (Performance Bounds)**.** *Define* $\alpha = 2\sigma\sqrt{dn_u + t}$, *choose* λ_1, λ_2, λ_3 *as :* $\lambda_1 \geq n_u\alpha$, $\lambda_2 \geq \tilde{n}\alpha$, $\lambda_3 \geq \alpha$ *, where* $n_u = \sum_{i=1}^{n_a} n_{u_i}$ *and* $\tilde{n} = \sqrt{\sum_i n_{u_i}^2}$. *Given Assumption 1, let*

$$\zeta = \frac{\lambda_1\sqrt{n_\theta}}{\kappa_\theta} + \frac{\lambda_2\sqrt{n_p}}{\kappa_p} + \frac{\lambda_2\sqrt{n_{u,a}}}{\kappa_{u,a}}$$

for t > 0, *we have :*

$$\mathbb{P}\left(\frac{1}{n_{min}n_u}\|\boldsymbol{X}\overline{\boldsymbol{W}} - \boldsymbol{F}\|^2 \leq (\frac{2\zeta}{n_{min}n_u})^2\right) \geq \delta(t) \quad (13)$$

$$\mathbb{P}\left(\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_1 \leq \frac{2(\beta_\theta + 1)\sqrt{n_\theta}}{\kappa_\theta n_{min} n_u}\zeta\right) \geq \delta(t) \quad (14)$$

$$\mathbb{P}\left(\|\hat{\boldsymbol{P}} - \boldsymbol{P}^*\|_{1,2} \leq \frac{2(\beta_p + 1)\sqrt{n_p}}{\kappa_p n_{min} n_u}\zeta\right) \geq \delta(t) \quad (15)$$

$$\mathbb{P}\left(\|\hat{\boldsymbol{U}}^\top - \boldsymbol{U^*}^\top\|_{1,2} \leq \frac{2(\beta_{u,a} + 1)\sqrt{n_{u,a}}}{\kappa_{u,a} n_{min} n_u}\zeta\right) \geq \delta(t) \quad (16)$$

where $\delta(t) = 1 - \frac{1}{\sqrt{2\pi Z_{dn_u}(t)}} exp\left(-\frac{Z_{dn_u}(t)}{2}\right)$*;* $n_{min} = \min_{i,j} n_{ij}$ *and* $Z_{dn_u}(t) = t - dn_u log(1 + \frac{t}{dn_u})$.

Remark 1. *According to theorem 1, if*

$$\zeta = o\left(n_{min}n_u \min\left\{1, \frac{\kappa_\theta}{\sqrt{n_\theta}}, \frac{\kappa_p}{\sqrt{n_p}}, \frac{\kappa_{u,a}}{\sqrt{n_{u,a}}}\right\}\right)$$

We have : $\mathbb{E}(\boldsymbol{X}^{(i,j)}\hat{\boldsymbol{w}}^{(i,j)} - \boldsymbol{f^*}^{(i,j)}) \to 0$, $\hat{\boldsymbol{\theta}} \overset{\ell_1}{\to} \boldsymbol{\theta}^*$, $\hat{\boldsymbol{P}} \overset{\ell_{1,2}}{\to} \boldsymbol{P}^*$, *and* $\hat{\boldsymbol{U}}^\top \overset{\ell_{1,2}}{\to} \boldsymbol{U^*}^\top$ *hold with high probability when* $n_{min} \to \infty$. *Furthermore, though the proof of theorem 1 uses standard techniques developed in (Gong, Ye, and Zhang 2012),* $\delta(t)$ *in theorem 1 is a tighter probability bound than* $\left(1 - exp(-\frac{Z_{dn_u}(t)}{2})\right)$ *proposed in (Gong, Ye, and Zhang 2012) and* $\left(1 - n_u exp(-\frac{Z_{dn_u}(t)}{2})\right)$ *proposed in (Chen, Zhou, and Ye 2011), for sufficiently large t.*

According to theorem 1 and remark 1, we see that our proposed method could both leverage good performance and estimate the parameters well with high probability.

Experiment

Now in this section, we show our experiment results on a simulated dataset, and two real world datasets respectively.

Experiment Setting

For each subtask, we randomly split the corresponding samples into a training subset and test subset, with 40% and 80% of the samples selected as training set respectively. For each involved algorithm, the hyper-parameters are tuned based on a 3 fold cross validation on the training set, and the average performance of the test set on 5 different splits are recorded. It is important to note that, different with the setting of (Kovashka and Grauman 2013), we will not use any extra dataset for pre-training in this paper. Furthermore, the training data is preprocessed according to Eq.(1)

Simulated Dataset

Data Generation For simulated dataset, our goal is to verify that the proposed algorithm could reach reasonable performance based on our theoretical analysis. We here define a regression problem for this dataset. To this end, we generate simulated features and continuous user scores (but not discrete labels) for 5 attributes, and all n_{u_i}s are fixed as 10. Furthermore, we set the dimensionality d as 50. For the (i,j)th subtask, 300 samples are generated such that $\boldsymbol{X}^{(i,j)} \sim \mathcal{N}(0, 4\boldsymbol{I})$ and $\boldsymbol{y}^{(i,j)} = \boldsymbol{X}^{(i,j)}\boldsymbol{w}^{(i,j)} + \mathcal{N}(0, \boldsymbol{I})$. To leverage group sparsity of $\boldsymbol{W}$: $\boldsymbol{\theta}$ is generated as $\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{1}, 4 * \boldsymbol{1})$, and the first 15 elements are set as zero; $\boldsymbol{P}$ is generated as $\boldsymbol{P} \sim \mathcal{N}(\boldsymbol{1}, 5\boldsymbol{I})$ and the 20-35 th rows of $\boldsymbol{P}$ are set as $\boldsymbol{0}$; $\boldsymbol{U}$ is generated as $\boldsymbol{U} \sim \mathcal{N}(\boldsymbol{1}, 10\boldsymbol{I})$ and the first 2 columns of each $\boldsymbol{U}^{(t)}$ are set as $\boldsymbol{0}$.

Competitors and Evaluation Metric To verify the effectiveness of our proposed algorithm, we compare our algorithm with four benchmark algorithms for regression: Support Vector Regression (SVR), Lasso regression (Lasso), Ridge regression (Ridge), and the Elastic Net. Meanwhile, all of these four benchmark algorithms are employed in a user-exclusive manner : the predictors for all the subtasks are trained separately as independent tasks. To evaluate the generalized performance of the algorithms, for a given attribute, we adopt $\overline{NMSE}$, the average value of normalized mean square error (NMSE) on all users for a given attribute, as the evaluation metric.

Table 1: Performance comparison for the simulated dataset with 40% samples selected as training data

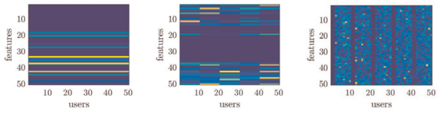
SVR	Ridge	Lasso	Elastic Net	ours
1.000	0.830	2.99E-05	9.96E-05	**2.70E-05**
1.000	0.830	3.20E-05	9.92E-05	**2.82E-05**
1.000	0.830	3.15E-05	1.07E-04	**2.78E-05**
0.998	0.829	4.20E-05	1.14E-04	**3.75E-05**
1.005	0.834	2.88E-05	9.34E-05	**2.72E-05**
1.001	0.830	3.28E-05	1.03E-04	**2.95E-05**

Table 2: Performance comparison for the simulated dataset with 80% samples selected as training data

SVR	Ridge	Lasso	Elastic Net	ours
1.005	0.884	2.27E-05	5.19E-05	**2.11E-05**
1.009	0.888	2.37E-05	5.27E-05	**2.17E-05**
1.009	0.887	2.30E-05	5.14E-05	**2.12E-05**
1.010	0.889	2.79E-05	5.64E-05	**2.60E-05**
1.019	0.896	2.03E-05	4.79E-05	**1.89E-05**
1.010	0.889	2.35E-05	5.20E-05	**2.18E-05**

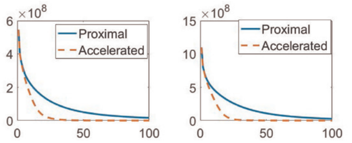
Performance Comparison According to Table 1 and Table 2, we could draw the following conclusions. The six rows in these two tables records the performance for attributes 1-5 and their average, respectively. On one hand, due to the inability to leverage sparse parameters, we see that performance of SVR and Ridge couldn't outperform the other three algorithms. On the other hand, our proposed algorithm reaches the best performance based on $\overline{NMSE}$, which verifies the effectiveness of our proposed algorithm. T

Parameters Recovery Now we show the ability of our algorithm to recover the structured parameters. With the same simulated dataset, we select 80% the samples as training data. According to theorem 1, we set $t = 10$, $\lambda_1 = 2n_u\alpha$, $\lambda_2 = 2.5\tilde{n}\alpha$, $\lambda_3 = 32\alpha$. Figure 2 shows the resulting parameters of our algorithm. Note that we extend θ and P to $\mathbb{R}^{d\times n_u}$, so that they could match the size of U. As is shown in this figure, we conclude that all of these three sets of parameters could roughly reach their expected structure.



(a) θ (b) P (c) U

Figure 2: Figures for the magnitude of θ,P and U. For comparison convenience, both θ and P are extended to $\mathbb{R}^{d\times n_u}$ matrix



(a) 40% (b) 80%

Figure 3: Effect of nestrov's Acceleration with (a) 40% random selected samples as training data and (b) 80% samples as training data. The y axis represents the average loss function of $(P1)$ on 10 repetitions, and the x axis represents the iteration number.

Effect of Nestrov's Acceleration Strategy Next, we conduct empirical study the on the effect of Nestrov's Acceleration. To do this, we randomly select 40% and 80% samples as training data respectively. For each ratio, we run our algorithm 10 times for 100 iterations with different initial parameters. The average loss function per iteration both with and without the nestorv's acceleration strategy are presented in Figure 3. According to this figure, we conclude that, for both ratios, the accelerated algorithm starts to converge before the 50th iteration, which is much earlier than that of the ordinary proximal algorithm. We thus draw the conclusion that this acceleration strategy could successfully leverage faster convergence of our algorithm.

Shoes Dataset with Binary Attributes

Dataset Description For attribute learning, we use the shoes Dataset (Kovashka and Grauman 2013; Kovashka, Parikh, and Grauman 2015) which contains 14,658 online shopping images. Here we choose 6 user labeled attributes from the original dataset: bright, ornate, shiny, high, long, formal. For each of the attribute, user specific binary labels were collected on 60 images from 10 workers (Kovashka and Grauman 2013). In other words, we have $n_a = 6$, $n_{u_i} = 10$ and $n_{ij} = 60$. To form the feature of the images, we concatenate the GIST and color histograms provided by the original dataset.

Competitors and Evaluation Metric To show the effectiveness of our proposed algorithm on binary personal attribute learning, we compare our algorithm with three kinds of algorithms: global algorithms, user-specific algorithms and multi-task algorithms. For global algorithms, during the training phase, user specific labels are first processed to

Table 3: Performance comparison for the binary attribute dataset

algorithm	accuracy	
	40%	80%
SVM	0.6278	0.6756
MLP	0.5057	0.4997
user exclusive	0.6549	0.6913
user adaptive	0.6771	0.6956
rMTFL-G	0.6421	0.6941
rMTFL-U	0.6659	0.7056
ours	**0.6894**	**0.7121**

global labels via majority voting. For valid set and test set the user specific labels are used directly for performance evaluation. This kind of algorithms include: the Support Vector Machine(SVM), and Multi Layer Perceptron with single hidden layer (MLP). For user specific algorithms, we employ the user exclusive method where one SVM classifier is trained for each user independently and the user adaptive method proposed in (Kovashka and Grauman 2013). For multi-task algorithms, we employ rMTFL (Gong, Ye, and Zhang 2012), which shares similar model assumption as the proposed model, as the benchmark, and both the global version (rMTFL-G) and the user specific version (rMTFL-U) are considered. The setting of rMTFL-G is the same as that of the other global algorithms, except that the global classifiers for different attributes are trained in a multi-task manner. While for rMTFL-U, we regard all the user-specific classifiers as subtasks of rMTFL. To evaluate the generalized performance of different algorithms, the average value of the classification accuracy among all users is adopted as the performance metric.

Results Table 3 shows the average performance on 5 splits for all these algorithms when 40% and 80% of the samples are chosen as training data respectively. We could observe that our proposed algorithm reaches the best average accuracy.

Table 4: Performance comparison for the relative attribute dataset

algorithm	accuracy	
	40%	80%
rel_attr	0.4797	0.5195
RankNet	0.4791	0.4721
RankBoost	0.4669	0.5251
user exclusive	0.4753	0.5303
user adaptive	0.4777	0.5336
rMTFL-G	0.4807	0.5074
rMTFL-U	0.4838	0.5433
ours	**0.5119**	**0.5546**

Shoes Dataset with Relative Attributes

Dataset Description Here we use the same shoes dataset as the binary attribute experiment. The only difference is that the user specific labels are collected based on relative attribute between a pair of shoes images. For this task, we choose 6 attributes : pointy, bright, ornate, shiny, sporty and feminine. Similar as the previous subsection, we have $n_a = 6$, $n_{u_i} = 10$, $n_{ij} = 60$.

Competitors and Evaluation Metric For relative attribute learning, we also adopt the aforementioned three kinds of algorithms as benchmarks. The global models include: ranksvm models for relative attribute learning (Parikh and Grauman 2011) (rel_attr), ranknet (Burges et al. 2005), and rankboost (Freund et al. 2003). For the user exclusive model we train one rel_attr model for each user independently. And user adaptive model is the same as (Kovashka and Grauman 2013) except that we do not use any extra data for pre-training. rMTFL-G and rMTFL-U are the same as that used in the previous subsection except that the input feature for an image pair is processed as what mentioned in the "Model Formulation" section to fit these algorithms to ranking problems. We use the average ranking accuracy among all users for all tasks as our evaluation metric.

Results Table 4 show the performance comparison when 40% and 80% of the samples are chosen as training data respectively. It is concluded that our proposed algorithm could reach reasonable improvements with respect to the benchmarks, which demonstrate its effectiveness.

Conclusion

In this paper, we propose a hierarchical multi-task model for user specific attribute learning across multiple attributes with a common-to-special decomposition of the model weights. Specifically, our model weights include a common cognition factor, an attribute-specific factor and a user specific factor. The well-known accelerated proximal gradient method is employed to solve this model. Based on assumption 1, we prove theoretically that the proposed algorithm could both leverage good performance and estimate the true parameters well with high probability. The experiment results further verify the effectiveness of our proposed model.

Acknowledgments

This work was supported by National Key Research and Development Plan (No.2016YFB0800603). The research of Zhiyong Yang and Qingming Huang was supported in part by National Natural Science Foundation of China: 61332016, U1636214, 61650202 and 61620106009, in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013. The research of Qianqian Xu was supported in part by National Natural Science Foundation of China (No. 61672514, 61390514, 61572042, No.U1605252), CCF-Tencent Open Research Fun.

References

Ando, R. K., and Zhang, T. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research* 6:1817–1853.

Beck, A., and Teboulle, M. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences* 2(1):183–202.

Bickel, P. J.; Ritov, Y.; and Tsybakov, A. B. 2008. Simultaneous analysis of lasso and dantzig selector. *Annals of Statistics* 37(4):1705–1732.

Burges, C. J. C.; Shaked, T.; Renshaw, E.; Lazier, A.; Deeds, M.; Hamilton, N.; and Hullender, G. N. 2005. Learning to rank using gradient descent. In *proceedings of the the Twenty-Second International Conference onMachine Learning*, 89–96.

Chen, L.; Zhang, Q.; and Li, B. 2014. Predicting multiple attributes via relative multi-task learning. In *proceedings of the Conference on Computer Vision and Pattern Recognition, 2014*, 1027–1034.

Chen, J.; Zhou, J.; and Ye, J. 2011. Integrating low-rank and group-sparse structures for robust multi-task learning. In *proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 42–50.

Ehrlich, M.; Shields, T. J.; Almaev, T.; and Amer, M. R. 2016. Facial attributes classification using multi-task representation learning. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016*, 752–760.

Freund, Y.; Iyer, R. D.; Schapire, R. E.; and Singer, Y. 2003. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research* 4:933–969.

Gong, P.; Ye, J.; and Zhang, C. 2012. Robust multi-task feature learning. In *proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12*, 895–903.

Han, H.; Jain, A. K.; Shan, S.; and Chen, X. 2017. Heterogeneous face attribute estimation: A deep multi-task learning approach. *arxiv* 1706.00906.

Hand, E. M., and Chellappa, R. 2017. Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification. In *proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 4068–4074.

Hsieh, H.; Hsu, W. H.; and Chen, Y. 2017. Multi-task learning for face identification and attribute estimation. In *proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2017*, 2981–2985.

Huang, A.; Xu, L.; Li, Y.; and Chen, E. 2014. Robust dynamic trajectory regression on road networks: A multi-task learning framework. In *proceedings of the International Conference on Data Mining,2014*, 857–862.

Hunter, D. R. 2004. Mm algorithms for generalized bradley-terry models. *Annals of Statistics* 32(1):384–406.

Ji, Z.; Sun, Y.; Yu, Y.; Guo, J.; and Pang, Y. 2017. Semantic softmax loss for zero-shot learning. *arxiv* 1705.07692.

Kovashka, A., and Grauman, K. 2013. Attribute adaptation for personalized image search. In *proceedings of the International Conference on Computer Vision 2013*, 3432–3439.

Kovashka, A., and Grauman, K. 2015. Discovering attribute shades of meaning with the crowd. *International Journal of Computer Vision* 114(1):56–73.

Kovashka, A.; Parikh, D.; and Grauman, K. 2015. Whittlesearch: Interactive image search with relative attribute feedback. *International Journal of Computer Vision* 115(2):185–210.

Parikh, D., and Grauman, K. 2011. Relative attributes. In *proceedings of theIEEE International Conference on Computer Vision, 2011*, 503–510.

Song, F.; Tan, X.; and Chen, S. 2014. Exploiting relationship between attributes for improved face verification. *Computer Vision and Image Understanding* 122:143–154.

Su, C.; Zhang, S.; Xing, J.; Gao, W.; and Tian, Q. 2016. Deep attributes driven multi-camera person re-identification. In *proceedings of the 14th European Conference on Computer Vision,2016*, 475–491.

Su, C.; Zhang, S.; Yang, F.; Zhang, G.; Tian, Q.; Gao, W.; and Davis, L. S. 2017. Attributes driven tracklet-to-tracklet person re-identification using latent prototypes space mapping. *Pattern Recognition* 66:4–15.

Wang, Y.; Kwok, J. T.; Yao, Q.; and Ni, L. M. 2017. Zero-shot learning with a partial set of observed attributes. In *proceedings of the International Joint Conference on Neural Networks*, 3777–3784.

Xu, L.; Huang, A.; Chen, J.; and Chen, E. 2015. Exploiting task-feature co-clusters in multi-task learning. In *proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 1931–1937.

Zhang, Z., and Saligrama, V. 2016. Zero-shot learning via joint latent similarity embedding. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6034–6042.

Zhou, J.; Chen, J.; and Ye, J. 2011. Clustered multi-task learning via alternating structure optimization. In *proceedings of the 25th Annual Conference on Neural Information Processing Systems*, 702–710.