Deep Robust Subjective Visual Property Prediction in Crowdsourcing

Qianqian Xu[1] Zhiyong Yang[2,3] Yangbangyan Jiang[2,3]
Xiaochun Cao[2,3] Qingming Huang[1,4,5] Yuan Yao[6]

[1] Key Lab of Intell. Info. Process., Inst. of Comput. Tech., CAS, Beijing, 100190, China
[2] State Key Laboratory of Info. Security (SKLOIS), Inst. of Info. Engin., CAS, Beijing, 100093, China
[3] School of Cyber Security, University of Chinese Academy of Sciences, Beijing,100049, China
[4] School of Computer Science and Tech., University of Chinese Academy of Sciences, Beijing, 101408, China
[5] BDKM, University of Chinese Academy of Sciences, Beijing, 100190, China
[6] Department of Mathematics, Hong Kong University of Science and Technology, Hong Kong

xuqianqian@ict.ac.cn {yangzhiyong,jiangyangbangyan,caoxiaochun}@iie.ac.cn
qmhuang@ucas.ac.cn yuany@ust.hk

Abstract

*The problem of estimating subjective visual properties (SVP) of images (*e.g.*, Shoes A is more comfortable than B) is gaining rising attention. Due to its highly subjective nature, different annotators often exhibit different interpretations of scales when adopting absolute value tests. Therefore, recent investigations turn to collect pairwise comparisons via crowdsourcing platforms. However, crowdsourcing data usually contains outliers. For this purpose, it is desired to develop a robust model for learning SVP from crowdsourced noisy annotations. In this paper, we construct a deep SVP prediction model which not only leads to better detection of annotation outliers but also enables learning with extremely sparse annotations. Specifically, we construct a comparison multi-graph based on the collected annotations, where different labeling results correspond to edges with different directions between two vertexes. Then, we propose a generalized deep probabilistic framework which consists of an SVP prediction module and an outlier modeling module that work collaboratively and are optimized jointly. Extensive experiments on various benchmark datasets demonstrate that our new approach guarantees promising results.*

1. Introduction

In recent years, estimating subjective visual properties (SVP) of images [9, 19, 24] is gaining rising attention in computer vision community. SVP measures a user's subjective perception and feeling, with respect to a certain property in images/videos. For example, estimating properties of consumer goods such as shininess of shoes [9] improves customer experiences on online shopping websites; and estimating interestingness [8] from images/videos would be helpful for media-sharing websites (*e.g.*, Youtube). Measuring and ensuring good estimation of SVP is thus highly subjective in nature. Traditional methods usually adopt absolute value to specify a rating from 1 to 5 (or, 1 to 10) to grade the property of a stimulus. For example, in image/video interestingness prediction, 5 being the most interesting, 1 being the least interesting. However, since by definition these properties are subjective, different raters often exhibit different interpretations of the scales and as a result the annotations of different people on the same sample can vary hugely. Moreover, it is unable to concretely define the concept of scale (for example, what a scale 3 means for an image), especially without any common reference point. Therefore, recent investigations turn to an alternative approach with pairwise comparison. In a pairwise comparison test, an individual is simply asked to compare two stimuli simultaneously, and votes which one has the stronger property based on his/her perception. Therefore individual decision process in pairwise comparison is simpler than in the typical absolute value tests, as the multiple-scale rating is reduced to a dichotomous choice. It not only promises assessments that are easier and faster to obtain with less demanding task for raters, but also yields more reliable feedback with less personal scale bias in practice. However, a shortcoming of pairwise comparison is that it has more expensive sampling complexity than the absolute value tests, since the number of pairs grows quadratically with the number of items to be ranked.

With the growth of crowdsourcing [2] platforms such as MTurk, InnoCentive, CrowdFlower, CrowdRank, and AllOurIdeas, recent studies thus resort to using crowdsourcing tools to tackle the cost problem. However, since the participants in the crowdsourcing experiments often work in the absence of supervision, it is hard to guarantee the annotation quality in general [5]. If the experiment lasts too long, the raters always lose their patience and end the test in a hurry with random annotations. Worse, the bad users might even provide wrong answers deliberately to corrupt the system. Such contaminated decisions are useless and may deviate significantly from other raters' decisions thus should be identified and removed in order to achieve a robust SVP prediction result.

Therefore, existing approaches on SVP prediction are often split into two separate steps: the first is a standard outlier detection problem (*e.g.*, majority voting) and the second is a regression or learning to rank problem. However, it has been found that when pairwise local rankings are integrated into a global ranking, it is possible to detect outliers that can cause global inconsistency and yet are locally consistent, *i.e.*, supported by majority votes [14]. To overcome this limitation, [9] proposes a more principled way to identify annotation outliers by formulating the SVP prediction task as a unified robust learning to rank problem, tackling both the outlier detection and SVP prediction tasks jointly. Different from this work which only enjoys the limited representation power of the image low-level features, our goal in this paper is to leverage the strong representation power of deep neural networks to explore the SVP prediction issue from a deep perspective.

When it comes to deep learning, it is known that several kinds of factors can drive the deep learning model away from a perfect one, with the data perturbation issue as an typical example. Besides the notorious issue coming from the crowdsourcing process, deep learning is in itself known to be more vulnerable to contaminated data since the extremely high model complexity brings extra risks to overfit the noisy/contaminated data [20, 10, 30, 32, 15, 25, 22]. We believe that how to guarantee the robustness is one of the biggest challenges when constructing deep SVP prediction models. In this sense, we propose a deep robust model for learning SVP from crowdsourcing. As an overall summary, we list our main contributions as follows:

- A novel method for robust prediction of SVP is proposed. To the best of our knowledge, our framework offers the first attempt to carry out the prediction procedure with automatic detection of sparse outliers from a deep perspective.
- In the core of the framework lies the unified probabilistic model, which is used to formulate the generating process of the labels when outliers exist. Based on this model, we then propose a Maximum A Posterior (MAP) based objective function.
- An alternative optimization scheme is adopted to solve the corresponding model. Specifically, the network parameters could be updated from the gradient-based method with the back-propagation, whereas the outlier pattern could be solved from an ordinal gradient descent method or a proximal gradient method.

2. Related Work

2.1. Subjective visual properties

Subjective visual property prediction has gained rising attention in the last several years. It covers a large variety of computer vision problems, including image/video interestingness [8], memorability [16], and quality of experience [27] prediction, etc. When used as a semantically meaningful representation, the subjective visual properties are often referred to as relative attributes [29, 19]. The original SVP prediction approach treats this task as a learning-to-rank problem. The main idea is to use ordered pairs of training images to train a ranking function that will generalize to new images. Specifically, a set of pairs ordered according to their perceived property strength is obtained from human annotators, and a ranking function that preserves those orderings is learned. Given a new image pair, the ranker indicates which image has the property more. A naive way to learn the ranker is to resort to traditional pairwise learning-to-rank methods such as RankSVM [17], RankBoost [6], and RankNet [3], etc. However, these methods are not a natural fit in the scenarios with crowdsourced outliers. In [9], it proposes a unified robust learning to rank (URLR) framework to solve jointly both the outlier detection and learning to rank problems. Different from this line of research, we study the robust SVP prediction in the context of deep learning. Equipped with better feature representation power, we show both theoretically and experimentally that by solving both the outlier detection and ranking prediction problems jointly in a deep framework, we achieve better outlier detection and better ranking prediction.

2.2. Learning with noisy data

Learning from noisy data has been studied extensively in recent years. Traditionally, such methods could be tracked back to statistical studies such as Majority voting, M-estimator [13], Huber-LASSO [27], and Least Trimmed Squares (LTS) [28], etc. However, these work do not have prediction (especially with the power of deep learning) ability for unseen samples. Recently, there is a wave to explore robust methods to learn from noisy labels, in the context of deep learning. Generally speaking, there are four types of existing methods: (**I**) robust learning based on probabilistic graphical models where the noisy patterns are often modeled as latent variables [30, 25]; (**II**) progressive and self-paced learning, where easy and clean examples are learned first, whereas the hard and noisy labels are progressively considered [10]; (**III**) loss-correction methods,

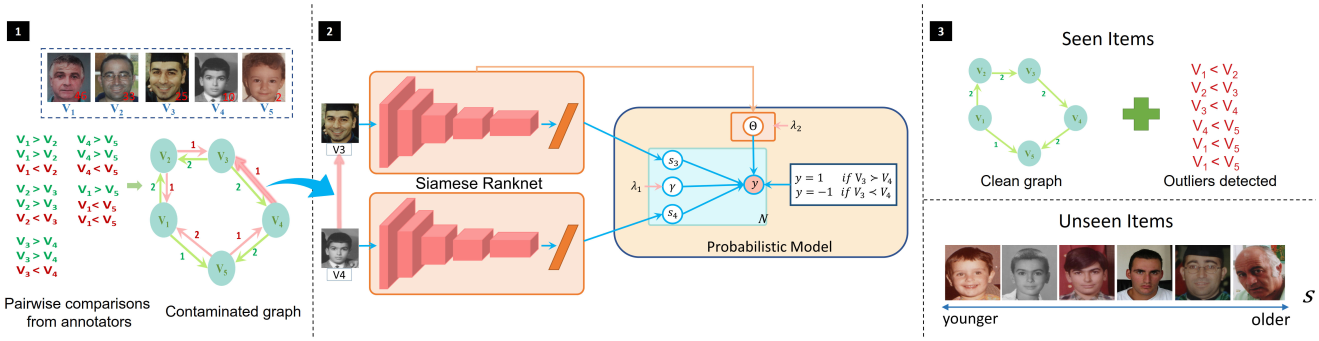

Figure 1: Overview of our approach. (1) Constructing a comparison graph from the crowdsourcing annotations, which is contaminated with outlier labels. (2) We propose a generalized deep probabilistic framework, where an outlier indicator γ is learned along with the network parameters Θ. (3) Our Framework will output a clean graph on the training set, where contaminated annotations are eliminated. Furthermore, our model could predict a rank-preserved score for each unseen instance. Best viewed in color.

where the loss function is corrected iteratively [22]; (**IV**) network architecture-based method, where the noisy patterns are modeled with specifically designed modules [15]. Meanwhile, there are also some efforts on designing deep robust models for specific tasks and applications: [20] proposes a method to learn from weak and noisy labels for semantic segmentation; [32] proposes a deep robust unsupervised method for saliency detection, etc.

Compared with these recent achievements, our work differs significantly in the sense that: a) We provide the first trial to explore the deep robust learning problem in the context of crowdsourced SVP learning. b) We adopt a pairwise learning framework, whereas the existing work all adopt instance-wise frameworks.

3. Methodology

3.1. Problem definition

Our goal in this paper is two-fold:

(a) We aim to learn a deep SVP prediction model from a set of sparse and noisy pairwise comparison labels. Specifically the ranking patterns should be preserved.

(b) To guarantee the quality of the model, we expect that all the noisy annotations could be detected and removed along with the training process.

We denote the id of two images in the ith pair as i_1 and i_2, and denote the corresponding image pair as $(\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2})$. More precisely, we are given a pool with n training images and a set of SVPs. In addition, for each SVP, we are given a set of pairwise comparison labels. Such pairwise comparison data can be represented by a directed multi-graph where multiple edges could be found between two vertexes. Mathematically, we denote the graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ is the set of vertexes which contains all the distinct image items occurred in the comparisons. $\mathcal{E}$ is the set of comparison edges. For a specific user with id j and a specific comparison pair i defined on two item vertexes i_1 and i_2, if the user believes that i_1 holds a stronger/weaker presence of the SVP, we then have an edge $(i_1, i_2, j)/(i_2, i_1, j)$, respectively. Equivalently we also denote this relation as $i_1 \overset{j}{\succ} i_2/i_2 \overset{j}{\succ} i_1$. Since multiple users take part in the annotation process, it is natural to observe multi-edges between two vertexes. Now we could denote the labeling results as a function $\mathcal{Y} : \mathcal{E} \to \{-1, 1\}$. For a given pair i and a rater j who annotates this pair, the corresponding label is denoted as y_{ij}, which is defined as:

$$\begin{cases} y_{ij} = & 1, \quad (i_1, i_2, j) \in \mathcal{E}; \\ y_{ij} = & -1, \quad (i_2, i_1, j) \in \mathcal{E}. \end{cases} \tag{1}$$

Now we present an example of the defined comparison graph. See step 1 in Figure 1. In this figure, the SVP in question is the age of the humans in the images. Suppose we have 5 images with ground truth ages (marked with red in the lower right corner of each image), we then have $\mathcal{V} = \{1, 2, \cdots, 5\}$. Furthermore, we have three users with id 1, 2, 3 who take part in the annotation. According to the labeling results shown in the lower left side, we have $\mathcal{E} = \{(1, 2, 1), (1, 2, 2), (2, 1, 3), \cdots, (1, 5, 1), (5, 1, 2), (5, 1, 3)\}$. As shown in this example, we would be most likely to observe both $i_1 \succ i_2$ and $i_2 \succ i_1$ for a specific pair i. This is mainly caused by the bad and ugly users who provide erroneous labels. For example for vertexes 1 and 2, the edge $(2, 1, 3)$ is obviously an abnormal annotation. With the above definitions and explanations, we are ready to introduce the input and output of our proposed model.

Input. The input of our deep model is the defined multi-graph $\mathcal{G}$ along with the image items, where each time a specific edge is fed to the network.

Output. As will be seen in the next subsection, our model will output the relative score s_{i_1} and s_{i_2} of the image pair along with an outlier indicator which could automatically remove the abnormal directions on $\mathcal{G}$. Note that learning s_{i_1} and s_{i_2} directly achieves our goal **(a)**, while detecting and removing outlier directions on the graph directly achieves goal **(b)**.

3.2. A deep robust SVP prediction model

In contrast to traditional methods, we propose a deep robust SVP ranking model in this paper. According to step 2 in Figure 1, we employ a deep Siamese [4, 21] convolutional neural network as the ranking model to calculate the relative scores for image pairs. In this model, the input is an edge in the graph $\mathcal{G}$ together with the image pair $(\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2})$. Each branch of the network is fed with an image and outputs the corresponding scores $s(\boldsymbol{x}_{i_1})$ and $s(\boldsymbol{x}_{i_2})$. Then we propose a robust probabilistic model based on the difference of the scores. As a note for the network architecture, we choose an existing popular CNN architecture, ResNet-50 [11], as the backbone of the Siamese network. Such residual network is equipped with shortcut connections, bringing in promising performance in image tasks.

With the network given, we are ready to elaborate a novel probabilistic model to simultaneously prune the outliers and learn the network parameters for SVP prediction. In our model, the noisy annotations are treated as a mixture of reliable patterns and outlier patterns. More precisely, to guarantee the performance of the whole model, we expect $s(\boldsymbol{x}_{i_1}), s(\boldsymbol{x}_{i_2})$, *i.e.*, the scores returned by the network to capture the reliable patterns in the labels. Meanwhile, we introduce an outlier indicator term $\gamma(y_{ij})$ to model the noisy nature of the annotations. During the training process, our prediction is an additive mixture of the reliable score and the outlier indicator.

To see how the inclusion of $\boldsymbol{\gamma}$ could help us detect and remove outlier, one should realize that, since y_{ij} must be either 1 or -1, there are only two distinct values for $\gamma(y_{ij})$, with one for each direction. If we can learn a reasonable $\gamma(y_{ij})$ such that $\gamma(y_{ij}) \neq 0$ only if the corresponding direction is not reliable, we can then remove the contaminated directions in $\mathcal{G}$ and obtain a clean graph. To illustrate it in an easier way, let us back to step 1 in Figure 1. According to the lower left contents, we have three annotations for pair (V_1, V_2). We have two distinct $\gamma(y_{ij})$ for these annotations: For the correct direction, we have a $\gamma(1)$ for $(1, 2, 1)$ and $(1, 2, 2)$; For the contaminated direction, we have a different gamma with value $\gamma(-1)$ for $(2, 1, 3)$. Now if we can learn $\gamma(y_{ij})$ in a way that $\gamma(1) = 0$ and $\gamma(-1) \neq 0$, then we can easily detect the contaminated direction $(2, 1)$.

Given the clarification above, our next step is to propose a probabilistic model of the labels based on the outlier indicator $\boldsymbol{\gamma}$, the network parameters Θ, and the predicted scores $s(\cdot)$. Specifically, we model the conditional distribution of the annotations along with the prior distribution of $\boldsymbol{\gamma}$ and Θ in the following form:

$$y_{ij} \mid \boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2}, \Theta, \gamma(y_{ij}) \overset{i.i.d}{\sim} f(y_{ij}, s(\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \Theta) + \gamma(y_{ij})),$$

$$\gamma(y_{ij}) \mid \lambda_1 \overset{i.i.d}{\sim} h(\gamma(y_{ij}), \lambda_1), \quad \Theta \mid \lambda_2 \sim g(\Theta, \lambda_2).$$

- $s(\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2}, \Theta) = s(\boldsymbol{x}_{i_1}, \Theta) - s(\boldsymbol{x}_{i_2}, \Theta)$ is the relative score of the annotation, which will be directly learned from the deep learning model with the parameter set Θ. As mentioned above, $s(\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2}, \Theta)$ are expected to model the reliable pattern in the annotations. The prior distribution of Θ is assumed to be associated with a p.d.f. (probability density function) $p(\Theta \mid \lambda_2) = g(\Theta, \lambda_2)$ (λ_2 is a predefined hyperparameter), which is denoted as g in short.

- $\gamma(y_{ij})$ is the outlier indicator which induces unreliability. Since only outliers have a nonzero indicator, we model the randomness of $\gamma(y_{ij})$ with an i.i.d sparsity-inducing prior distribution (*e.g.*, Laplacian distribution) with the p.d.f. being $p(\gamma(y_{ij})|\lambda_1) = h(\gamma(y_{ij}), \lambda_1)$ (λ_1 denotes the hyperparameter), which is denoted as h_{ij} in short.

- As we have mentioned above, the noisy prediction $s(\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2}, \Theta) + \gamma(y_{ij})$ is an additive mixture of the reliable score and outlier indicator.

- $f(y_{ij}, s(\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \Theta) + \gamma(y_{ij}))$ is the conditional p.d.f. of the labels, which is denoted as f_{ij} in short.

Let $\boldsymbol{\gamma} = \{\gamma(y_{ij})\}_{(i_1,i_2,j)\in\mathcal{E}}$, $\boldsymbol{y} = \{y_{ij}\}_{(i_1,i_2,j)\in\mathcal{E}}$. Now our next step is to construct a loss function for this probabilistic model. According to the Maximum A Posterior (MAP) rule in statistics, a reasonable solution of the parameters should have a large posterior probability $P(\Theta, \boldsymbol{\gamma} \mid \boldsymbol{y}, \boldsymbol{X}, \lambda_1, \lambda_2)$. In other words, with high probability, the parameters ($\boldsymbol{\gamma}, \Theta$ *in our model*) should be observed after seeing the data ($\boldsymbol{y}, \boldsymbol{X}$ *in our model*) and the predefined hyperparameters (λ_1, λ_2). This motivates us to *maximize* the posterior probability in our objective function. Furthermore, to simplify the calculation of the derivatives, we adopt an equivalent form where the negative log posterior probability is *minimized*:

$$\min_{\Theta,\boldsymbol{\gamma}} -\log\left(P(\Theta, \boldsymbol{\gamma} \mid \boldsymbol{y}, \boldsymbol{X}, \lambda_1, \lambda_2)\right).$$

Following the Bayesian rule, one has:

$$\begin{aligned} & P(\Theta, \boldsymbol{\gamma} \mid \boldsymbol{y}, \boldsymbol{X}, \lambda_1, \lambda_2) \\ = & \frac{P(\boldsymbol{y}| \, \boldsymbol{X}, \Theta, \boldsymbol{\gamma}) \cdot P(\Theta| \, \lambda_1) \cdot P(\boldsymbol{\gamma}|\lambda_2) \cdot P(\boldsymbol{X})}{\int_\Theta \int_{\boldsymbol{\gamma}} P(\boldsymbol{X}, \, \boldsymbol{y}| \, \Theta, \boldsymbol{\gamma}) \cdot P(\Theta| \, \lambda_1) \cdot P(\boldsymbol{\gamma}|\lambda_2) d\Theta d\boldsymbol{\gamma}}. \end{aligned}$$

It then becomes clear that $P(\Theta, \boldsymbol{\gamma}|\boldsymbol{y}, \boldsymbol{X}, \lambda_1, \lambda_2)$ is not directly tractable. Fortunately, since $\boldsymbol{X}, \boldsymbol{y}$ are given and we only need to optimize Θ and $\boldsymbol{\gamma}$, the tedious term $\frac{P(\boldsymbol{X})}{\int_\Theta \int_{\boldsymbol{\gamma}} P(\boldsymbol{X},\, \boldsymbol{y}|\ \Theta, \boldsymbol{\gamma}) \cdot P(\Theta|\ \lambda_1) \cdot P(\boldsymbol{\gamma}|\lambda_2) d\Theta d\boldsymbol{\gamma}}$ becomes a constant, which suggests that:

$$\begin{aligned} & P(\Theta, \boldsymbol{\gamma} \mid \boldsymbol{y}, \boldsymbol{X}, \lambda_1, \lambda_2) \\ \propto & \prod_{(i,j)\in\mathcal{D}} p(y_{ij} \mid \boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \gamma(y_{i,j}), \Theta) \cdot p(\gamma(y_{ij}) \mid \lambda_1) \cdot p(\Theta \mid \lambda_2) \\ = & \prod_{(i,j)\in\mathcal{D}} g \cdot h_{ij} \cdot f_{ij}. \end{aligned} \tag{2}$$

where $\mathcal{D} : \{(i, j) : (i_1, i_2, j) \in \mathcal{E} \ or \ (i_2, i_1, j) \in \mathcal{E}\}$. This implies that our loss function could be simplified as:

$$\min_{\Theta, \boldsymbol{\gamma}} \sum_{(i,j)\in\mathcal{D}} -\left(\log(f_{ij}) + \log(h_{ij})\right) - \log(g).$$

With the general framework given, we provide two specified models with different assumptions on the distributions:

- **Model A**: If the prior distribution of $\gamma(y_{ij})|\lambda_1$ is a Laplacian distribution with a zero location parameter and a scale parameter of $\frac{1}{\lambda 1}$: $Lap(0, \frac{1}{\lambda_1}) = \frac{\lambda_1}{2}\exp(-\frac{|\gamma|}{1/\lambda_1})$; the prior distribution of Θ is an element-wise Gaussian distribution $\mathcal{N}(0, \frac{1}{2\lambda_2})$; and y_{ij} conditionally subjects to a Gaussian distribution $\mathcal{N}(s(\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \Theta) + \gamma(y_{ij}), 1)$, then the problem becomes:

  $$\min_{\Theta, \boldsymbol{\gamma}} \sum_{(i,j)\in\mathcal{D}} \frac{1}{2}(y_{ij} - s(\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \Theta) - \gamma(y_{ij}))^2 + \lambda_1 \|\boldsymbol{\gamma}\|_1 + \lambda_2 \sum_{\theta\in\Theta} \theta^2,$$

  where $\|\boldsymbol{\gamma}\|_1 = \sum_{(i,j)\in\mathcal{D}} |\gamma(y_{ij})|$.

- **Model B**: If we adopt the same assumption as above, except that we assume that y_{ij} conditionally subjects to a Logistic-like distribution, then the problem could be simplified as:

  $$\min_{\boldsymbol{\gamma}, \Theta} \sum_{(i,j)\in\mathcal{D}} \log(1 + \Delta_{ij}) + \lambda_1 \|\boldsymbol{\gamma}\|_1 + \lambda_2 \sum_{\theta\in\Theta} \theta^2,$$

  where $\Delta_{ij} = \exp(-y_{ij}(s(\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \Theta) + \gamma(y_{ij})))$.

3.3. Optimization

With the model and network clarified, we then introduce the optimization method we adopt in this paper. Specifically, we employ an iterative scheme where $\boldsymbol{\gamma}$ and the network parameters Θ are alternatively updated until the convergence is reached.

3.3.1 Fix $\boldsymbol{\gamma}$, Learn Θ

When fixing $\boldsymbol{\gamma}$, we see that Θ could be solved from the following subproblem:

$$\min_{\Theta} -\sum_{(i,j)\in\mathcal{D}} \log(f_{ij}) - \log(g)$$

Since Θ only depends on the network, one could find an approximated solution by updating the network. For **Model A**, this subproblem becomes:

$$\min_{\Theta} \sum_{(i,j)\in\mathcal{D}} \frac{1}{2}(y_{ij} - s(\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \Theta) - \gamma(y_{ij}))^2 + \lambda_2 \sum_{\theta\in\Theta} \theta^2.$$

Similarly, for **Model B**, we come to a subproblem in the form:

$$\min_{\Theta} \sum_{(i,j)\in\mathcal{D}} \log(1 + \Delta_{ij}) + \lambda_2 \sum_{\theta\in\Theta} \theta^2.$$

3.3.2 Fix Θ, Learn $\boldsymbol{\gamma}$

Similarly, when Θ is fixed, we could solve $\boldsymbol{\gamma}$ from:

$$\min_{\boldsymbol{\gamma}} \sum_{(i,j)\in\mathcal{D}} -\left(\log(f_{ij}) + \log(h_{ij})\right)$$

This is a simple model of $\boldsymbol{\gamma}$ which does not involve the network. For **Model A**, this subproblem becomes:

$$\min_{\boldsymbol{\gamma}} \sum_{(i,j)\in\mathcal{D}} \frac{1}{2}(y_{ij} - s(\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \Theta) - \gamma(y_{ij}))^2 + \lambda_1 \|\boldsymbol{\gamma}\|_1.$$

It enjoys a closed-form solution with the proximal operator of ℓ_1 norm:

$$\gamma(y_{ij}) = \max(|c_{ij}| - \lambda_1, 0) \cdot \text{sign}(c_{ij}), \tag{3}$$

where

$$c_{ij} = y_{ij} - s(\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \Theta).$$

For **Model B**, this subproblem becomes:

$$\min_{\boldsymbol{\gamma}} \sum_{(i,j)\in\mathcal{D}} \log(1 + \Delta_{ij}) + \lambda_1 \|\boldsymbol{\gamma}\|_1.$$

Generally, there is no closed-form solution for this subproblem. In this paper, we adopt the proximal gradient method [1] to find a numerical solution.

4. Experiments

In this section, experiments are exhibited on three benchmark datasets (see Table 1) which fall into two categories: (1) experiments on human age estimation from face images (Section 4.1), which can be considered as synthetic experiments. With the ground truth available, this set of experiments enables us to perform in-depth evaluation of the significance of our proposed method, (2) experiments on estimating SVPs as relative attributes (Section 4.2 and 4.3).

Table 1: Dataset summary.

Dataset	**No.Pairs**	**No.Images**	**No.Classes**
FG-Net Face Age Dataset	15,000	1002	1
LFW-10 Dataset[23]	29,454	2000	10
Shoes Dataset [18]	87,946	14,658	7

Table 2: Experimental results on Human age dataset.

Algorithm	**ACC**	**F1**	**Prec.**	**Rec.**	**AUC**
Maj-LS	.5555	.4673	.4369	.5022	.5650
LS-with γ	.5594	.4729	.4414	.5093	.5759
Maj-Logistic	.5421	.4687	.4264	.5205	.5489
Logistic-with γ	.5585	.4743	.4410	.5131	.5735
Maj-RankNet [3]	.5611	.4804	.4445	.5227	.5792
Maj-RankBoost [6]	.5425	.5991	.6458	.5587	.4507
Maj-RankSVM [17]	.5838	.3858	.4517	.3367	.5665
Maj-GBDT [7]	.5827	.3880	.4504	.3408	.5619
Maj-DART [26]	.5940	.3668	.4648	.3029	.5690
URLR [9]	.5765	.4633	.5748	.5131	.5762
LS-Deep-w/o γ	.7313	.6694	.6407	.7008	.8060
Logit-Deep-w/o γ	.7439	.6818	.6584	.7070	.8168
LS-Deep-with γ	**.7967**	**.7414**	**.7323**	**.7508**	**.8784**
Logit-Deep-with γ	**.7917**	**.7370**	**.7228**	**.7518**	**.8739**

4.1. Human age dataset

In this experiment, we consider age as a subjective visual property of a face. The main difference between this SVP with the other SVPs evaluated so far is that we do have the ground truth, *i.e.*, the person's age when the picture was taken. This enables us to perform in-depth evaluation of the significance of our proposed framework.

Dataset The FG-NET [1] image age dataset contains 1002 images of 82 individuals labeled with ground truth ages ranging from 0 to 69. The training set is composed of the images of 41 randomly selected individuals and the rest used as the test set. For the training set, we use the ground truth age to generate the pairwise comparisons, with the preference direction following the ground-truth order. To create sparse outliers, a random subset (*i.e.*, 20%) of the pairwise comparisons is reversed in preference direction. In this way, we create a paired comparison graph, possibly incomplete and imbalanced, with 1002 nodes and 15,000 pairwise comparison samples.

Competitors We compare our method **Model A** and **Model B** with 10 competitors. Note that **Model A** is the least square based deep model, while **Model B** is a logistic regression based deep model. In the following experiments, we give **Model A** an alias as **LS-Deep**, and give **Model B** an alias as **Logit-Deep**:

1) **Maj-LS**: This method uses majority voting for outlier pruning and least squares problem for learning to rank.

2) **LS-with** γ: To test the improvement of merely adopting the robust model, we jointly employ the linear regression model and our proposed robust mechanism as a baseline.

3) **Maj-Logistic**: This method stands for another baseline in our work, where the majority voting is adopted for label processing followed with the logistic regression.

4) **Logistic-with** γ: Again, to test the improvement of merely adopting the robust model, we jointly employ the logistic regression model and our proposed robust mechanism as a baseline.

5) **Maj-RankSVM** [17]: We record the performance of RankSVM to show the superiority of the representation learning.

6) **Maj-RankNet** [3]: To show the effectiveness of using a deeper network, we compare our method with the classical RankNet model preprocessed by the majority voting.

7) **Maj-RankBoost** [6]: Besides the deep learning framework, it is also known that the ensemble-based methods could also serve a model for hierarchical learning and representation. In this sense, we compare our method with the RankBoost model, one of the most classical ensemble method.

8) **Maj-GBDT** [7]: Gradient Boosting Decision Tree (GBDT) has gained surprising improvements in many traditional tasks. Accordingly, we compare our methods with GBDT to show its strength.

9) **Maj-DART** [26]: Recently, the well-known drop-out trick has also been applied to ensemble-based learning, be it the DART method. We also record the performance of DART to show the superiority of our method.

10) **URLR** [9]: URLR is a unified robust learning to rank framework which aims to tackle both the outlier detection and learning to rank jointly. We compare our algorithm with this method to show the effectiveness of using a generalized probabilistic model and a deep architecture.

Ablation: To show the effectiveness of the proposed probabilistic model, we additionally add two competitors as the ablation. Note that the key element to detect outlier is the factor γ. In this way, the ablation competitors are formed with γ eliminated:

1) **LS-Deep-w/o** γ: This is a partial implementation of **LS-Deep**, where the factor γ is removed.

2) **Logit-Deep-w/o** γ: This is a partial implementation of **Logit-Deep**, where the factor γ is removed.

Evaluation metrics Because the ground-truth age is available, we adopt ACC, Precision, Recall, F1-score and AUC as the evaluation metrics to demonstrate the effectiveness of our proposed method.

Implementation Details For the four deep learning methods, the learning rate is set as 10^{-4}, and λ_2 is set as 10^{-3}. For LS-Deep-with γ, λ_1 is set as 1.2. For Logit-Deep-with γ, λ_1 is set as 0.6.

Comparative Results In all the non-deep competitive experiments, we adopt LBP as the low-level features. Looking at the five-metrics results in Table 2, we see that our method (marked with red and green color) consistently outperforms all the benchmark algorithms by a significant mar-

[1] http://www.fgnet.rsunit.com/

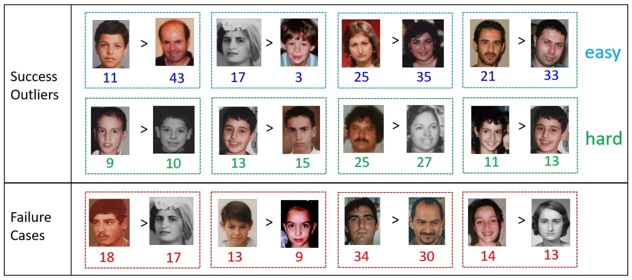

Figure 2: Outlier examples detected on Human age dataset.

gin. This validates the effectiveness of our method. In particular, it can be observed that: (1) LS-with γ (or Logistic-with γ) is superior to Maj-LS (or Maj-Logistic) because the global outlier detection is better than local outlier detection (*i.e.*, Majority voting). (2) The performance of deep methods is better than all non-deep methods, interestingly even the ablation baseline methods without γ give better results than traditional methods with outlier detection, which suggests the strong representation power of deep neural networks in SVP prediction tasks. (3) It is worth mentioning that our proposed Deep-with γ methods successfully exhibit roughly $5\% - 8\%$ improvement on all the five-metrics than Deep-without γ methods, demonstrating the superior outlier detection ability of our proposed framework. (4) Our proposed two models **A** (*i.e.*, LS-Deep-with γ) and **B** (*i.e.*, Logit-Deep-with γ) show comparable results on this dataset, while model **A** holds the lead by a slight margin.

Moreover, we visualize some examples of outliers detected by model **A** in Figure 2, while results returned by model **B** are very similar. It can be seen that those in the blue/green boxes are clearly outliers and are detected correctly by our method. For better illustration, the ground-truth age is printed under each image. Moreover, blue boxes show pairs with a large age differences while green boxes illustrate samples with subtle age differences, which indicates that our method not only can detect the easy pairs with a large age gap, but also can handle hard samples with small age gap (*e.g.*, within only 1-2 years difference). Four failure cases are shown in red boxes, in which our method treats the images on the left are older than the right one as an outlier, but the ground truth agrees with the annotation. We can easily find that this often occurs on pairs with small age differences, which indicates that our methods may occasionally lose its power when meeting highly competitive or confused pairs.

4.2. LFW-10 dataset

Dataset The LFW-10 dataset [23] consists of 2,000 face images, taken from the Labeled Faces in the Wild [12] dataset. It contains 10 relative attributes, like smiling, big eyes, etc. Each pair was labeled by 5 people. For example, given a specific attribute, the user will choose which one to be stronger in the attribute. As the goal of our paper is to predict SVP from noisy labels, we do not conduct any pro-precessing steps to meet the agreement of labels as [31]. The resulting dataset has 29,454 total annotated sample pairs, on average 2945 binary pairs per attribute.

Table 3: Experimental results (ACC) of 10 attributes on LFW-10 dataset.

Algorithm	**Bald**	**D.Hai**	**B.Eye**	**GLook**	**Masc.**	**Mouth**	**Smile**	**Teeth**	**Foreh.**	**Young**	**Aver.**
Maj-LS	.4767	.5368	.4787	.4788	.5588	.4774	.5220	.5073	.4759	.5162	.5029
LS-with γ	.5805	.6400	.5506	.5932	.6009	.5097	.5178	.5198	.5680	.5911	.5672
Maj-Logistic	.6123	.6716	.5146	.5890	.6253	.5032	.5031	.5322	.5724	.6599	.5784
Logistic-with γ	.6059	.6400	.5640	.6038	.6275	.5269	.5073	.5405	.5724	.6437	.5832
Maj-RankNet [3]	.6123	.6421	.5551	.6208	.6275	.5097	.5304	.5468	.5899	.6275	.5862
Maj-RankBoost [6]	.5996	.7053	.5236	.5975	.6231	.5097	.5199	.5094	.6053	.6032	.5797
Maj-RankSVM [17]	.4852	.6526	.4180	.5805	.5588	.4882	.5283	.5156	.5482	.6397	.5415
Maj-GBDT [7]	.5551	.6253	.4899	.5466	.5721	.4903	.5094	.5198	.5965	.6235	.5528
Maj-DART [26]	.5508	.6337	.4899	.5339	.5698	.4989	.5597	.5364	.5943	.6134	.5581
URLR [9]	.5889	.6538	.6505	.5258	.5614	.6319	.5311	.4968	.5446	.5570	.5742
LS-Deep-w/o γ	.5932	.7095	.5551	.6081	.5543	.5742	.6436	.6133	.5746	.6741	.6100
Logit-Deep-w/o γ	.5551	.6758	.5124	.6335	.6253	.5806	.6038	.6175	.5724	.6235	.6000
LS-Deep-with γ	**.6335**	**.7684**	**.5551**	**.6377**	**.6253**	**.7312**	**.7421**	**.7547**	**.6469**	**.7308**	**.6826**
Logit-Deep-with γ	**.6631**	**.7726**	**.5798**	**.6419**	**.5965**	**.7032**	**.7358**	**.7069**	**.6075**	**.6862**	**.6694**

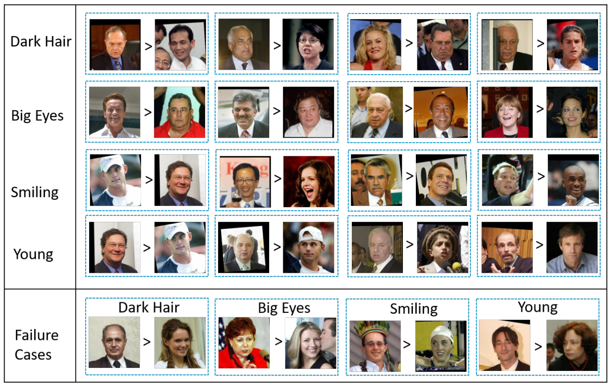

Figure 3: Outlier examples of 4 representative attributes on LFW-10 dataset.

Implementation Details In competitive experiments, we adopt GIST as the low-level features. For the four deep learning methods, the learning rate is set as 10^{-4}, and λ_2 is set as 10^{-3}. For LS-Deep-with γ, λ_1 is set as 1.2. For Logit-Deep-with γ, λ_1 is set as 0.5.

Comparative Results Table 3 reports the summary ACC for each attribute. The following observations can be made: (1) Our deep-methods always outperform traditional non-deep methods and ablation baseline methods for all experiment settings with higher average ACC on all attributes (0.6826 vs. 0.6100 and 0.6694 vs. 0.6000 on two models, respectively). (2) The performance of other methods is in general consistent with what we observed in the Human age experiments.

Moreover, Figure 3 gives some examples of the pruned pairs of 4 randomly selected attributes. In the success cases, the left images are (incorrectly) annotated to have more of the attribute than the right ones. However, they are either wrong or too ambiguous to give consistent answers, and as such are detrimental to learning to rank. A number of failure cases (false positive pairs identified by our models) are also shown. Some of them are caused by unique viewpoints (*e.g.*, for 'dark hair' attribute, the man has sparse scalp, so it is hard to tell who has dark hair more); others are caused by the weak feature representation, *e.g.*, in the 'young' attribute example, as 'young' would be a function of multiple subtle visual cues like face shape, skin texture, hair color, etc., whereas something like baldness or smiling has a better visual focus captured well by part-based features.

4.3. Shoes dataset

Dataset The Shoes dataset is collected from [18] which contains 14,658 online shopping images. In this dataset, 7 attributes are annotated by users with a wide spectrum of interests and backgrounds. For each attribute, there are at least 190 users who take part in the annotation, and each user is assigned with 50 images. Note that the dataset actually uses binary annotations rather than pairwise annotations (1 for Yes, -1 for No). We then randomly sample positive annotations and negatives annotations from each user's records to form the pairs we need. For each attribute, we randomly select such 2000 distinct pairs, finally yielding a volume of 87,946 total personalized comparisons.

Implementation Details In competitive experiments, we concatenate the GIST and color histograms provided by the original dataset as the low-level features. For the LS-based deep methods, the learning rate is set as 10^{-3}. For the Logit-based deep methods, the learning rate is set as 10^{-5}. λ_2 is set as 10^{-3} for all four methods. For LS-Deep-with γ, λ_1 is set as 1.2. For Logit-Deep-with γ, λ_1 is set as 0.8.

Comparative Results Similar to the Human age and LFW-10 datasets, Table 4 again shows that the performance of our proposed deep models is significantly better than that of other competitors. Moreover, some outlier detection examples are shown in Figure 4. In the top four rows with successful detection examples, the right images clearly have more of the attribute than the left ones, however are incorrectly annotated by crowdsourced raters. The failure cases are caused by the invisibility (*e.g.*, for 'comfortable' attribute, though the transparent rain-boots itself is flat, there is in fact a pair of high-heeled shoes inside with red color); others are caused by different visual definitions of attributes (*e.g.*, for 'open' attribute, it has multiple shades of meaning, *e.g.*, peep-toed (open at toe) vs. slip-on (open at heel) vs. sandal-like (open at toe and heel)); The remaining may be caused by ambiguity: both images have this attribute with similar degree. This thus corresponds to a truly ambiguous case which can go either way.

Table 4: Experimental results (ACC) of 7 attributes on Shoes dataset.

Algorithm	Comf.	Fash.	Form.	Pointy	Brown	Open	Ornate	Aver.
Maj-LS	.7300	.7825	.7325	.7897	.6950	.7331	.7300	.7418
LS-with γ	.8150	.8125	.7975	.7860	.7275	.7444	.7625	.7779
Maj-Logistic	.7600	.7850	.7475	.7970	.6900	.7068	.7175	.7434
Logistic-with γ	.8375	.8175	.7825	.7934	.7250	.7444	.7525	.7790
Maj-RankNet [3]	.7425	.7850	.7200	.7860	.6925	.7444	.7300	.7429
Maj-RankBoost [6]	.7525	.7300	.7275	.7675	.6975	.6955	.6725	.7204
Maj-RankSVM [17]	.7425	.7925	.7925	.8081	.6850	.7331	.7200	.7534
Maj-GBDT [7]	.7075	.7325	.7425	.8007	.6750	.7519	.7550	.7379
Maj-DART [26]	.6900	.7275	.7375	.8376	.6975	.7857	.7125	.7412
URLR [9]	.8200	.8150	.7900	.7860	.7325	.7444	.7550	.7775
LS-Deep-w/o γ	.7100	.8075	.7400	.7749	.7725	.7669	.7050	.7538
Logit-Deep-w/o γ	.7100	.8025	.7500	.8044	.7525	.7857	.6975	.7575
LS-Deep-with γ	**.8500**	**.8550**	**.8125**	**.8044**	**.8250**	**.7782**	**.8300**	**.8222**
Logit-Deep-with γ	**.8550**	**.8500**	**.8200**	**.8339**	**.8125**	**.7481**	**.8325**	**.8217**

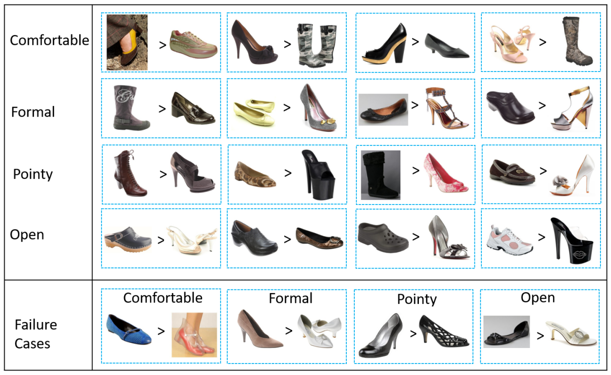

Figure 4: Outlier examples of 4 representative attributes on Shoes dataset.

5. Conclusion

This work explores the challenging task of SVP prediction from noisy crowdsourced annotations from a deep perspective. We present a simple but effective general probabilistic model to simultaneously predict rank preserving scores and detect the outliers annotations, where an outlier indicator γ is learned along with the network parameters Θ. Practically, we present two specific models with different assumptions on the data distribution. Furthermore, we adopt an alternative optimization scheme to update γ and Θ iteratively. In our empirical studies, we perform a series of experiments on three real-world datasets: Human age dataset, LFW-10, and Shoes. The corresponding results consistently show the superiority of our proposed model.

6. Acknowledgments

This work was supported in part by National Basic Research Program of China (973 Program): 2015CB351800 and 2015CB85600, in part by National Natural Science Foundation of China: 61620106009, U1636214, 61861166002, 61672514, and 11421110001, in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013, in part by Beijing Natural Science Foundation (4182079), in part by Youth Innovation Promotion Association CAS, and in part by Hong Kong Research Grant Council (HKRGC) grant 16303817.

References

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. 5

[2] S. Branson, G. Van Horn, and P. Perona. Lean crowdsourcing: Combining humans and machines in an online system. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7474–7483, 2017. 1

[3] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *International Conference on Machine Learning*, pages 89–96, 2005. 2, 6, 7, 8

[4] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 539–546, 2005. 4

[5] F. Daniel, P. Kucherbaev, C. Cappiello, B. Benatallah, and M. Allahbakhsh. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Computing Surveys*, 51(1):7, 2018. 2

[6] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4(Nov):933–969, 2003. 2, 6, 7, 8

[7] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. 6, 7, 8

[8] Y. Fu, T. M. Hospedales, T. Xiang, S. Gong, and Y. Yao. Interestingness prediction by robust learning to rank. In *European Conference on Computer Vision*, pages 488–503, 2014. 1, 2

[9] Y. Fu, T. M. Hospedales, T. Xiang, J. Xiong, S. Gong, Y. Wang, and Y. Yao. Robust subjective visual property prediction from crowdsourced pairwise labels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):563–577, 2016. 1, 2, 6, 7, 8

[10] B. Han, I. W. Tsang, L. Chen, P. Y. Celina, and S.-F. Fung. Progressive stochastic learning for noisy labels. *IEEE Transactions on Neural Networks and Learning Systems*, (99):1–13, 2018. 2

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 4

[12] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Workshop on faces in 'Real-Life' Images: detection, alignment, and recognition*, 2008. 7

[13] P. Huber. *Robust Statistics*. New York: Wiley, 1981. 2

[14] X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye. Statistical ranking and combinatorial Hodge theory. *Mathematical Programming*, 127(6):203–244, 2011. 2

[15] I. Jindal, M. Nokleby, and X. Chen. Learning deep networks from noisy labels with dropout regularization. In *IEEE International Conference on Data Mining*, pages 967–972, 2016. 2, 3

[16] P. Jing, Y. Su, L. Nie, and H. Gu. Predicting image memorability through adaptive transfer learning from external sources. *IEEE Transactions on Multimedia*, 19(5):1050–1062, 2017. 2

[17] T. Joachims. Optimizing search engines using clickthrough data. In *ACM International Conference on Knowledge Discovery and Data Mining*, pages 133–142, 2002. 2, 6, 7, 8

[18] A. Kovashka and K. Grauman. Discovering attribute shades of meaning with the crowd. *International Journal of Computer Vision*, 114(1):56–73, 2015. 6, 8

[19] A. Kovashka and K. Grauman. Attributes for image retrieval. In *Visual Attributes*, pages 89–117. Springer, 2017. 1, 2

[20] Z. Lu, Z. Fu, T. Xiang, P. Han, L. Wang, and X. Gao. Learning from weak and noisy labels for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(3):486–500, 2017. 2, 3

[21] M. Norouzi, D. J. Fleet, and R. R. Salakhutdinov. Hamming distance metric learning. In *Annual Conference on Neural Information Processing Systems*, pages 1061–1069, 2012. 4

[22] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu. Making deep neural networks robust to label noise: A loss correction approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952, 2017. 2, 3

[23] R. N. Sandeep, Y. Verma, and C. Jawahar. Relative parts: Distinctive parts for learning relative attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3614–3621, 2014. 6, 7

[24] H. Squalli-Houssaini, N. Q. Duong, M. Gwenaëlle, and C.-H. Demarty. Deep learning for predicting image memorability. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2371–2375, 2018. 1

[25] A. Vahdat. Toward robustness against label noise in training deep discriminative neural networks. In *Annual Conference on Neural Information Processing Systems*, pages 5601–5610, 2017. 2

[26] R. K. Vinayak and R. Gilad-Bachrach. DART: dropouts meet multiple additive regression trees. In *International Conference on Artificial Intelligence and Statistics*, 2015. 6, 7, 8

[27] Q. Xu, J. Xiong, Q. Huang, and Y. Yao. Robust evaluation for quality of experience in crowdsourcing. In *ACM Conference on Multimedia*, pages 43–52, 2013. 2

[28] Q. Xu, M. Yan, C. Huang, J. Xiong, Q. Huang, and Y. Yao. Exploring outliers in crowdsourced ranking for qoe. In *ACM Conference on Multimedia*, pages 1540–1548, 2017. 2

[29] X. Yang, T. Zhang, C. Xu, S. Yan, M. S. Hossain, and A. Ghoneim. Deep relative attributes. *IEEE Transactions on Multimedia*, 18(9):1832–1842, 2016. 2

[30] J. Yao, J. Wang, I. W. Tsang, Y. Zhang, J. Sun, C. Zhang, and R. Zhang. Deep learning from noisy image labels with quality embedding. *IEEE Transactions on Image Processing*, 2018. 2

[31] A. Yu and K. Grauman. Just noticeable differences in visual attributes. In *IEEE International Conference on Computer Vision*, pages 2416–2424, 2015. 7

[32] J. Zhang, T. Zhang, Y. Dai, M. Harandi, and R. Hartley. Deep unsupervised saliency detection: A multiple noisy labeling perspective. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9029–9038, 2018. 2, 3

Supplementary Materials

1. Ablation Study

In the supplementary materials, we perform extra experiments to show whether joint end-to-end feature learning and robust ranking is better than other stage-wise deep robust ranking alternatives. We perform the following ablation studies on the three datasets:

- **pretrained+URLR**: In this baseline, we feed the pretrained feature extracted from Resnet-50 to a traditional robust learning to rank model URLR (a brief introduction of URLR could be found in the main paper). This baseline could show us the power of our method against the pre-trained deep feature.
- **noise+finetuned logit+URLR**: In this baseline, we feed the noisy annotations to a finetuned Resnet-50 network and minimize the cross entropy loss function (logit function). After the training phase, we obtain the finetuned features from the network, which are then fed to URLR. This experiment shows us whether the noisy data is sufficient for a good feature representation. Moreover, it tells us whether our proposed method outperforms finetuned features learned from noisy labels.
- **noise+finetuned l2+URLR**: This baseline is the same as the previous one except that the loss function is changed to the squared error loss.
- **major+finetuned logit+URLR**: In this baseline, we first perform a majority voting on the annotations and use the voted results to train a finetuned Resnet-50 network and minimize the cross entropy loss function (logit function). After the training phase, we obtain the finetuned features from the network, which are then fed to URLR. This experiment shows us whether the majority voting procedure could remove the noises and lead to a good feature representation. Moreover, it tells us whether our proposed method outperforms finetuned features learned from voted labels.
- **major+finetuned l2+URLR**: This baseline is the same as the previous one except that the loss function is changed to the squared error loss.

The ablation results for the three datasets are recorded in Tab.1a-1c, and we have the following findings regarding the results: 1) The finetuned feature merely gains a slight improvement with respect to the pre-trained feature. In fact, without the robust learning mechanism, the vanilla finetuning process (with raw/voting data) could not disentangle the contaminated patterns from the learned features. This weakens the power of traditional robust learning methods (URLR). 2) There is only a minor difference between the raw-data-based results and the majority voting-data-based results. This shows that the majority voting process fails to improve the robustness of the resulting model. As a justification, majority voting tackles the inconsistency results at a local level (removing minority directions independently). However, the higher-order/global inconsistency is totally neglected. 3) For URLR, filtering out outliers from the dataset alters the distribution of the positive/negative labeled instances. This directly results in a larger distribution gap between the training set and test set. Correspondingly, we observe a clearly worsened AUC generalization ability on the age dataset for all the five ablation methods. To sum up, it is vital to do joint end-to-end feature learning and robust ranking.

Table 1: Ablation studies on three datasets.

(a) Ablation studies on Human age dataset.

Algorithm	**ACC**	**F1**	**Prec.**	**Rec.**	**AUC**
pretrained+URLR	.7244	.6536	.6381	.6700	.7144
noise+finetuned logit+URLR	.7382	.6733	.6492	.6994	.7319
noise+finetuned l2+URLR	.7380	.6774	.6489	.7086	.7326
major+finetuned logit+URLR	.7391	.6741	.6544	.6949	.7310
major+finetuned l2+URLR	.7381	.6730	.6530	.6943	.7301
LS-Deep-with γ	**.7967**	**.7414**	**.7323**	**.7508**	**.8784**
Logit-Deep-with γ	**.7917**	**.7370**	**.7228**	**.7518**	**.8739**

(b) Ablation studies on Shoes dataset.

Comf.	**Fash.**	**Form.**	**Pointy**	**Brown**	**Open**	**Ornate**	**Aver.**
.8317	.8299	.8021	.7976	.8042	.7598	.8008	.8037
.8448	.8291	.8030	.8216	.7958	.7278	.8437	.8094
.8492	.8446	.8142	.8011	.8097	.7405	.8358	.8135
.8471	.8434	.8078	.7912	.8268	.7690	.8325	.8168
.8655	.8646	.8294	.8398	.7814	.7217	.7925	.8135
.8500	**.8550**	**.8125**	**.8044**	**.8250**	**.7782**	**.8300**	**.8222**
.8550	**.8500**	**.8200**	**.8339**	**.8125**	**.7481**	**.8325**	**.8217**

(c) Ablation studies on LFW-10 dataset.

Algorithm	**Bald**	**D.Hai**	**B.Eye**	**GLook**	**Masc.**	**Mouth**	**Smile**	**Teeth**	**Foreh.**	**Young**	**Aver.**
pre trained+URLR	.5424	.6295	.5213	.6356	.6519	.5699	.6059	.6133	.5746	.6781	.6030
noise +fine tuned logit +URLR	.6695	.6105	.5393	.6059	.6231	.6452	.6373	.6653	.5439	.6802	.6231
noise +fine tuned l2 +URLR	.6568	.6968	.5011	.6377	.6341	.5484	.6059	.6050	.6206	.6781	.6195
major+fine tuned logit+URLR	.6144	.7242	.4989	.6314	.5854	.6301	.6604	.6881	.5987	.6599	.6305
major+fine tuned l2+URLR	.6250	.7495	.5213	.6144	.6009	.6323	.6688	.6445	.6140	.6781	.6361
LS-Deep-with γ	**.6335**	**.7684**	**.5551**	**.6377**	**.6253**	**.7312**	**.7421**	**.7547**	**.6469**	**.7308**	**.6826**
Logit-Deep-with γ	**.6631**	**.7726**	**.5798**	**.6419**	**.5965**	**.7032**	**.7358**	**.7069**	**.6075**	**.6862**	**.6694**