Fusing Pointwise and Pairwise Labels for Supporting User-adaptive Image Retrieval

Lin Chen
Computer Sicence, Arizona State University
lin.chen.6@asu.edu

Peng Zhang*
Alibaba Group
minghua.zp@alibaba-inc.com

Baoxin Li
Computer Sicence, Arizona State University
baoxin.li@asu.edu

ABSTRACT

User-adaptive image retrieval/recommendation has drawn a lot of research interests in recent years, owing to fast development of various Web applications where retrieving images is a key enabling task. Existing challenges include the lack of user-adaptive training data, the ambiguity of user query and the real-time interactivity of a system. This paper proposes a hybrid learning strategy that fuses knowledge from both pointwise and pairwise training data into one framework for attribute-based, user-adaptive image retrieval. Under this framework, we develop an online learning algorithm for updating the ranking performance based on user feedback. Furthermore, we derive the framework into a kernel form, allowing easy application of kernel techniques. The proposed approach is evaluated on two image datasets and experimental results show that it achieves obvious performance gains over ranking and zero-shot learning from either type of training data independently. In addition, the online learning algorithm is able to deliver much better performance than batch learning, given the same elapsed running time, or can achieve better performance in much less time.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: [Miscellaneous]; I.4.8 [**Image Processing and Computer Vision**]: [Scene Analysis]

General Terms

Algorithms, Design, Experiment

Keywords

Adaptive Image Retrieval, Attribute Learning, Learning to Rank

*Effort performed while being a research fellow at ASU.

ICMR'15, June 23–26, 2015, Shanghai, China.

ACM 978-1-4503-3274-3/15/06 ...$15.00.
DOI: http://dx.doi.org/10.1145/2671188.2749358.

(a) **unstylish** (b) **stylish?** (c) **stylish**

(d) **more stylish?** (e) **more stylish?**

Figure 1: stylish and unstylish cars. Considering pointwise label, to most people (a) would be unstylish and (c) would be considered stylish. However, it is ambiguous to classify (b) to be stylish or unstylish. Considering pairwise label, people may have different preference to compare whether (d) or (e) is more stylish.

1. INTRODUCTION

The social media era has witnessed phenomenal growth of user-generated images on the Internet. The ever-growing number of images has brought about new challenges for efficient image retrieval, and in turn, for applications that rely on image retrieval. Conventional content-based image retrieval approaches learn some general ranking models purely based on the underlying images. In recent years, adaptive image retrieval [5][10][20] has emerged as a new trend, which intends to satisfy a user's specific requirements or preference. For example, in search of art images, some people like realism paintings while some may prefer abstract art. A retrieval engine being able to support such personalization would have the best potential to deliver what a user is really looking for. In practice, it is still difficult to learn a well generalized model due to the lack of user-adaptive training data. For example, in applications like on-line shopping, it is unreasonable to assume a user's personal preference data have been made available *a priori* for training the system. Often, desired is an on-line learning approach that accumulates such information over time interactively. Our approach adopts a model adaptation strategy and proposes a new ranking model and an online learning algorithm. Such

a method is especially proper for applications that utilize interactive input/feedback of a user in achieving adaptive image retrieval/recommendation.

Beneath the above challenge of personalization lie the fundamental problems of **semantic gap** and **intent gap** in general image retrieval. The semantic gap refers to the discrepancy between extractable low-level image features and high-level semantic concepts of images, while the intent gap refers to inadequacy of the representation of a query in expressing a user's true intent. In recent years, towards bridging the semantic gap, methods exploiting semantic attributes of visual objects have attracted significant attention in applications including object recognition [26][14][6], face verification [13] and image search [25][12][18]. Instead of using low-level features, these approaches describe images by high-level, human-nameable visual attributes, such as keep hair color, presence of beard or mustache, presence of eyeglasses, etc., to describe human faces.

In the meantime, towards bridging the intent gap, learning-to-rank approaches have been proposed. Recent literature on this regard includes three types of approaches distinguished by how the training data are used: pointwise, pairwise and listwise approaches. The first two types of approaches have been adopted in image ranking problems. Pointwise approaches [15][23] adopt category labels in the training samples to learn a ranking function. For example, to describe the car images in Fig. 1, the car in Fig. 1(c) is categorized as a "stylish" car and the one in Fig. 1(a) is labelled "unstylish". In a different way, pairwise approaches [2][24][7][8][3] learn a ranking function by taking comparative sample pairs for training. For example, most people would agree that the car in Fig. 1(c) looks more "stylish" than the one in Fig. 1(a). Such pair of samples with relative labels can be used to learn ranking functions for processing new images.

Pointwise data and pairwise data have different advantages and limitations in terms of data availability, labelling complexity and representational capability, as elaborated below.

Data availability In practical applications pointwise and pairwise labels are not always available for every data sample, especially considering the subjectivity of the labels. For example, in pointwise labelling, most people would agree that Fig. 1(c) is a "stylish" car and Fig. 1(a) is an "unstylish" car, but it would be difficult to tell whether Fig. 1(b) is a "stylish" car. Some people may think it is "stylish" because of the design of the headlights, while some others may deem the body design unattractive. Similarly, ambiguity also exists in pairwise data labelling. For example, comparing Fig. 1(d) and 1(e), people may have different opinions on which car is more "stylish" because of subjective preference. When ambiguity exists, it is better not to allow the data to be labeled so as not to produce noisy labels.

Labelling complexity In general, pairwise data may be more expensive to label. For example, given 10 images, we only need to label 10 samples to assign each image into one category. Also, category labels can be acquired from other sources such as image tags. On the other hand, to assign pairwise labels for all 10 images, we would need to compare 45 pairs to completely capture the ranking information. (Although the relative relation is transferable such as $(A \succ B) \& (B \succ C) \Rightarrow A \succ C$, it is difficult to discover those "key pairs" since we usually have to randomly pick pairs without any prior knowledge.) We note, however, that sometimes it is easier for a user to assign pairwise labels through comparison than having to give a pointwise label for a given image.

Representational capability Pairwise data tend to have stronger representational capability than pointwise data in ranking problems, as pointwise label only implies the relative order of data samples from different categories but not those from the same category. In contrast, pairwise labels already give the relative order of every training pairs, and thus contain more knowledge to learn a better ranking model.

As pointwise and pairwise labels encapsulate information of different types/amounts and may have different availability, we set out to develop a new framework for fusing both types of training data for improved ranking performance. Most of current fusion approaches [16][19] only use pointwise labels and the fusion only appears in the cost function. To our best knowledge, the only work considering fusing pointwise and pairwise data is presented by Sculley [21] whose object function is simply a linear combination of loss functions from regression and ranking.

In this paper, towards supporting adaptive image retrieval, we propose a new ranking-based framework. Our approach uses visual attributes to describe images, which helps to partially overcome the semantic gap problem. To alleviate the problem of lacking adaptive training samples, our approach attempt to maximize the utilization of all available training data by fusing both pointwise and pairwise labelled data in training. Compared to [21], our approach is formulated as a soft margined SVM which is able to achieve better generalization performance. Furthermore, to support interactivity, which is one natural way of gathering adaptive training data on the fly, we derive an online learning algorithm which can incrementally acquire a user's online feedback to improve the performance of the model incrementally with additional amount of data. As will be demonstrated by experiments, the proposed framework is able to take advantage of both types of data and deliver better performance than the baseline approaches that use only one type of data for learning.

There are three key **contributions** of our work. (1) We propose a new ranking framework termed "hybrid ranking" which takes both pointwise and pairwise labelled data for learning. (2) We propose an online learning algorithm for our proposed hybrid ranking framework, which can better support applications like adaptive image ranking. (3) We derive our hybrid ranking framework into a kernel form so that different kernel functions (depending on the application) may be applied for better performance.

In the remaining of the paper, we first give a formal problem definition in Section 2. The proposed approach is presented in Section 3. Experiments and results are demonstrated in Section 4. We conclude the paper in Section 5.

2. PROBLEM DEFINITION

Adaptive Image Retrieval We consider the following adaptive image retrieval procedure illustrated in Fig. 2: Given a training dataset including both the attribute existence labels of images (pointwise) and the relative attribute strength labels of image pairs (pairwise), we first train a general image ranking functions (the "Offline trained model" in the figure). This is used to retrieve images for a user based on his/her query. Looking at the initial retrieval results, the user may interactively provide feedback,

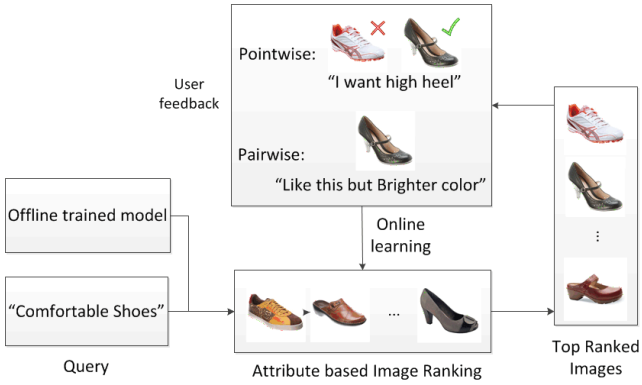


Figure 2: Adaptive image retrieval via on-line feedback.

in forms of newly labelled attribute existence labels and/or relative attribute strength labels. Such feedback is used as new training samples by an on-line-learning algorithm for updating the ranking function. As the feedback is specific to a user, the updated ranking function (and thus the retrieval engine) is presumably adapted to a user's preferences and hence achieving some level of personalization.

Hybrid Ranking To solve the above adaptive image retrieval task, we propose a hybrid ranking SVM framework. Given (1) the pointwise dataset $\mathcal{P}$ where each data sample $\boldsymbol{x}_i \in \mathcal{P}$ is assigned a category label and (2) the pairwise datasets including both the ordered pair set $\mathcal{O}$ and the unordered pair set $\mathcal{S}$ where for any pair $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{O}$, $\boldsymbol{x}_i \succ \boldsymbol{x}_j$ (e.g., $\boldsymbol{x}_i$ has a stronger attribute than $\boldsymbol{x}_j$), and for any pair $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S}$, $\boldsymbol{x}_i \sim \boldsymbol{x}_j$ (e.g., $\boldsymbol{x}_i$ has a similar attribute value to $\boldsymbol{x}_j$), we attempt to learn a ranking model taking both the pointwise and pairwise data into consideration. This hybrid ranking approach aims to capture as much information as possible from all available data so as to achieve better ranking performance especially when labelled data are scarce.

Notations In this paper, we represent scalars as lower case letters (e.g., x), vectors as bold face lower case letters (e.g., $\boldsymbol{x}$), matrices as capital letters (e.g., X) and sets as calligraphic capital letters (e.g., $\mathcal{X}$). The standard inner product between the vectors $\boldsymbol{u}\in\mathcal{R}^n$ and $\boldsymbol{v}\in\mathcal{R}^n$ is denoted as $\langle \boldsymbol{u}, \boldsymbol{v}\rangle = \sum_{i=1}^n u_i v_i$ with u_i/v_i the i-th element of $\boldsymbol{u}/\boldsymbol{v}$.

3. PROPOSED APPROACH

In this section we propose a general hybrid ranking SVM framework and an online algorithm to solve this problem.

3.1 Hybrid Ranking SVM

To make the best use of the available knowledge, we propose a hybrid ranking SVM which takes both pointwise and pairwise labelled data samples for learning.

The approach presented in [23] for ordinal regression learns a number of parallel hyperplanes by the large margin principle as a ranking model. One implementation of the approach tries to maximize a fixed margin for all the adjacent classes. Relative attributes [18] applies pairwise learning-to-rank approach on image attributes for image ranking. This approach learns a ranking function for each human-nameable attribute of an image. The relative "strength" of an attribute is measured by some distance metrics learned through SVM-like optimization using (relatively) labeled pairs. Both of these two SVM models aim to optimize a project direction $\boldsymbol{w}$, such that $\langle \boldsymbol{w}, \boldsymbol{w}\rangle$ (i.e., the inverse of the margin) is minimized subject to the separability constraints (modulo margin errors in the non-separable case).

In the situation that the training data are very limited, learning $\boldsymbol{w}$ based on both pointwise and pairwise datasets jointly would become a necessity in order to achieve reasonable performance. To fuse information from both types of data, the margins assigned to them should be different. To this end, we introduce a new superparameter ρ representing the margin corresponding to the pairwise data. We propose the hybrid ranking approach as follows:

$$\begin{aligned}
\min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{\xi},\boldsymbol{\zeta},\boldsymbol{\eta}} & \frac{1}{2}\|\boldsymbol{w}\|^2 + c_1\tau_1 \sum_i \sum_j (\xi_i^j + \xi_i^{*j+1}) \\
& + c_2\tau_2 \sum \zeta_{ij} + c_2\tau_3 \sum \eta_{ij} \\
s.t. \quad & \boldsymbol{w}\cdot\boldsymbol{x}_i^j - b_j \leq -1 + \xi_i^j, \\
& \boldsymbol{w}\cdot\boldsymbol{x}_i^{j+1} - b_j \geq 1 - \xi_i^{*j+1}, \\
& \boldsymbol{w}\cdot(\boldsymbol{x}_i - \boldsymbol{x}_j) \geq \rho - \zeta_{ij}, \forall (i,j) \in \mathcal{O}, \\
& |\boldsymbol{w}\cdot(\boldsymbol{x}_i - \boldsymbol{x}_j)| \leq \eta_{ij}, \forall (i,j) \in \mathcal{S}, \\
& \xi_i^j \geq 0, \xi_i^{*j} \geq 0, \zeta_{ij} \geq 0; \eta_{ij} \geq 0
\end{aligned}$$

where $\boldsymbol{x}_i^j \in \mathcal{R}^n$ is an object (feature vector) with $j = 1, ..., k-1$ denoting the class number, $i = 1, ..., i_j$ is the index within class j, and k is the total number of classes. $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is sample pairs, ξ_i^j and ξ_i^{*j} are non-negative slack variables measuring the degree of misclassified data, ζ_{ij} and η_{ij} are soft margin slack variables for pairwise ranking, c_1 and c_2 are super parameters controlling the weight for the pointwise and pairwise data, τ_i is the weight function penalizing different training datasets according to the data size. Specifically, let n_1, n_2 and n_3 denote the data sizes of the pointwise, ordered and unordered pairwise datasets respectively, then $\tau_i = \frac{n_i}{\sum_{j=1}^3 n_j}, i = 1, 2, 3$. Note that if only pointwise data are provided then the framework is equivalent to regression, and if only pairwise data are provided the framework becomes pairwise ranking.

In the following discussion, we focus on the image retrieval task which can be simplified as a hybrid ranking model with "binary type " of pointwise label (i.e., existence/non-existence of certain attribute). For clarity, in the following, we use $\boldsymbol{x}_i^{\mathcal{P}}$ to denote the i-th pointwise training sample $x_i \in \mathcal{P}$, and $\boldsymbol{x}_i^{\mathcal{O}}(\boldsymbol{x}_i^{\mathcal{S}})$ denotes the difference of the i-th ordered(unordered) pairwise training sample as $x_p - x_q$ for any $(x_p, x_q) \in \mathcal{O}(\mathcal{S})$. Then the ranking model can be formulated as the following primal form of the hybrid learning problem:

$$\begin{aligned}
\min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{\xi},\boldsymbol{\zeta},\boldsymbol{\eta}} & \frac{1}{2}\|\boldsymbol{w}\|^2 + c_1\tau_1 \sum_i \xi_i \\
& + c_2\tau_2 \sum \zeta_i + c_2\tau_3 \sum \eta_i \\
s.t. \quad & y_i \cdot (\boldsymbol{w}\cdot\boldsymbol{x}_i^{\mathcal{P}} + b) \geq 1 - \xi_i, \\
& \boldsymbol{w}\cdot\boldsymbol{x}_i^{\mathcal{O}} \geq \rho - \zeta_i, \\
& \left|\boldsymbol{w}\cdot\boldsymbol{x}_i^{\mathcal{S}}\right| \leq \eta_i, \\
& \xi_i \geq 0, \zeta_i \geq 0; \eta_i \geq 0.
\end{aligned} \tag{1}$$

This formulation can be solved by quadratic programming.

Algorithm 1 The Mini-Batch Online Learning Algorithm

Input:
1. Training set $\mathcal{A}$ with data type flags;
2. Parameters ρ, c_1, c_2, $\tilde{t}$, k;

Output: $\boldsymbol{w}_{\tilde{t}}$;

Set $\boldsymbol{w}_0 = 0$;
for $t = 1, 2, ..., \tilde{t}$ **do**
  Choose $\mathcal{A}_t \subseteq \mathcal{A}$ where $|\mathcal{A}_t| = k$ uniformly at random;
  Set $\mathcal{A}_t^{\mathcal{P}} = \{i \in \mathcal{A}_t : (x_i, y_i) \in \mathcal{P} \wedge y_i \langle \boldsymbol{x}_i, \boldsymbol{w}_t \rangle < 1\}$, $n_1 = |\mathcal{A}_t^{\mathcal{P}}|$;
  Set $\mathcal{A}_t^{\mathcal{O}} = \{i \in \mathcal{A}_t : (x_i, y_i) \in \mathcal{O} \wedge \langle \boldsymbol{x}_i, \boldsymbol{w}_t \rangle < \rho\}$, $n_2 = |\mathcal{A}_t^{\mathcal{O}}|$;
  Set $\mathcal{A}_t^{\mathcal{S}} = \{i \in \mathcal{A}_t : (x_i, y_i) \in \mathcal{S}$, $n_3 = |\mathcal{A}_t^{\mathcal{S}}|$;
  Set $\eta_t = \frac{1}{(n_1+n_2+n_3)t}$;
  Set $\tau_1 = \frac{n_1}{\sum_{j=1}^{3} n_j}$, $\tau_2 = \frac{n_2}{\sum_{j=1}^{3} n_j}$, $\tau_3 = \frac{n_3}{\sum_{j=1}^{3} n_j}$;
  Set $\boldsymbol{w}_t \leftarrow (1 - \eta_t)\boldsymbol{w}_{t-1} + \eta_t(c_1\tau_1 \sum_{i \in \mathcal{A}_t^{\mathcal{P}}} y_i \boldsymbol{x}_i + c_2\tau_2 \sum_{i \in \mathcal{A}_t^{\mathcal{O}}} \boldsymbol{x}_i) + c_2\tau_3 \sum_{i \in \mathcal{A}_t^{\mathcal{S}}} \operatorname{sgn} \langle \boldsymbol{x}_i, \boldsymbol{w}_t \rangle \boldsymbol{x}_i)$;
end for
return $\boldsymbol{w}_t$;

3.2 Mini-Batch Online Learning Algorithm

We now propose an online learning algorithm for the hybrid ranking SVM for adaptive image retrieval. In the retrieval application, we first train a general ranking function for the user. Based on the retrieval results, the user may provide feedback (new category and relative labels) according to their preferences. Then our online learning approach will update the ranking function based on the newly labelled data to make the ranking results better fit to the user's personal needs.

The constrained quadratic programming problem of Eq. (1) can be cast as an unconstraint empirical loss minimization with a penalty term for the norm of the classifier that can be learned in the following form:

$$\begin{aligned} \min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 &+ c_1\tau_1 \sum_{i \in \mathcal{P}} \ell_1(\boldsymbol{w}; (\boldsymbol{x}_i^{\mathcal{P}}, y_i^{\mathcal{P}})) \\ &+ c_2\tau_2 \sum_{i \in \mathcal{O}} \ell_2(\boldsymbol{w}; \boldsymbol{x}_i^{\mathcal{O}}) + c_2\tau_3 \sum_{i \in \mathcal{S}} \ell_3(\boldsymbol{w}; \boldsymbol{x}_i^{\mathcal{S}}) \end{aligned} \tag{2}$$

where

$$\begin{aligned} &\ell_1(\boldsymbol{w}; (\boldsymbol{x}_i^{\mathcal{P}}, y_i^{\mathcal{P}})) = \max\{0, 1 - y_i^{\mathcal{P}} \langle \boldsymbol{x}_i^{\mathcal{P}}, \boldsymbol{w} \rangle\}, \\ &\ell_2(\boldsymbol{w}; \boldsymbol{x}_i^{\mathcal{O}}) = \max\{0, \rho - \langle \boldsymbol{x}_i^{\mathcal{O}}, \boldsymbol{w} \rangle\}, \\ &\ell_3(\boldsymbol{w}; \boldsymbol{x}_i^{\mathcal{S}}) = \left| \langle \boldsymbol{x}_i^{\mathcal{S}}, \boldsymbol{w} \rangle \right|. \end{aligned}$$

Inspired by the Pegasos algorithm [22], we also considered the mini-batch algorithm which utilize k ($1 \leq k \leq m$) examples at each iteration. We initiate the model by setting $\boldsymbol{w}_0$ to the zero vector. In iteration t of the algorithm, given a training set $\mathcal{A}$ with m samples of both pointwise and pairwise data (a flag bit is used to identify the type of the data), we choose a subset $\mathcal{A}_t \subseteq \mathcal{A}$ with k examples uniformly at random among the training subset. Thus we will optimize the following approximate objective function:

$$\begin{aligned} f(\boldsymbol{w}; A_t) =& \frac{1}{2}\|\boldsymbol{w}\|^2 + c_1\tau_1 \sum_{i \in \mathcal{A}_t} \ell_1(\boldsymbol{w}; (\boldsymbol{x}_i^{\mathcal{P}}, y_i^{\mathcal{P}})) \\ &+ c_2\tau_2 \sum_{i \in \mathcal{A}_t} \ell_2(\boldsymbol{w}; \boldsymbol{x}_i^{\mathcal{O}}) + c_2\tau_3 \sum_{i \in \mathcal{A}_t} \ell_3(\boldsymbol{w}; \boldsymbol{x}_i^{\mathcal{S}}) \end{aligned} \tag{3}$$

We employ the stochastic gradient methods in our algorithm. The sub-gradient of Eq. (3) at iteration t is given by

$$\begin{aligned} \nabla_t =& \boldsymbol{w}_t - c_1\tau_1 \sum_{i \in \mathcal{A}_t} \chi_{\Re^+}(1 - y_i^{\mathcal{P}} \langle \boldsymbol{x}_i^{\mathcal{P}}, \boldsymbol{w}_t \rangle) y_i^{\mathcal{P}} \boldsymbol{x}_i^{\mathcal{P}} \\ &- c_2\tau_2 \sum_{i \in \mathcal{A}_t} \chi_{\Re^+}(\rho - \langle \boldsymbol{x}_i^{\mathcal{O}}, \boldsymbol{w}_t \rangle) \boldsymbol{x}_i^{\mathcal{O}} \\ &- c_2\tau_3 \sum_{i \in \mathcal{A}_t} \operatorname{sgn} \langle \boldsymbol{x}_i^{\mathcal{S}}, \boldsymbol{w}_t \rangle \boldsymbol{x}_i^{\mathcal{S}} \end{aligned} \tag{4}$$

where $\chi_A(x)$ is the eigenfunction and $\operatorname{sgn}(x)$ the symbolic function. Then the weight vector can be updated by

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \nabla_t$$

with the step size $\eta_t = \frac{1}{(n_1+n_2+n_3)t}$. After a predetermined number $\tilde{t}$ of iterations, we output the final $\boldsymbol{w}_{\tilde{t}}$ as the learned projection model. The pseudocode of our algorithm is given in Algorithm (1). It can be shown that our proposed framework have the same convergence property with [22] and thus we can terminate the procedure at a random stopping time and in at least half of the cases the last hypothesis is an accurate solution.

3.3 Kernelization

We further derive the framework into the kernel form which strengthens our approach to learn non-linear model. Note that although the derivation is based on the online learning form, it can be generalized to batch learning since we are considering mini-batch learning in this paper.

Instead of considering predictors which are linear functions of the training instances $\boldsymbol{x}$ themselves, we consider predictors which are linear functions of some implicit mapping $\phi(\boldsymbol{x})$ of the instances. Then the original optimization problem can be redefined as:

$$\begin{aligned} \min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 &+ c_1\tau_1 \sum_{i \in \mathcal{P}} \ell_1(\boldsymbol{w}; (\phi(\boldsymbol{x_i}^{\mathcal{P}}), y_i^{\mathcal{P}})) \\ &+ c_2\tau_2 \sum_{i \in \mathcal{O}} \ell_2(\boldsymbol{w}; \phi(\boldsymbol{x}_i^{\mathcal{O}})) + c_2\tau_3 \sum_{i \in \mathcal{S}} \ell_3(\boldsymbol{w}; \phi(\boldsymbol{x}_i^{\mathcal{S}})) \end{aligned} \tag{5}$$

where

$$\begin{aligned} &\ell_1(\boldsymbol{w}; (\phi(\boldsymbol{x}_i^{\mathcal{P}}), y_i^{\mathcal{P}})) = \max\{0, 1 - y_i^{\mathcal{P}} \langle \phi(\boldsymbol{x}_i^{\mathcal{P}}), \boldsymbol{w} \rangle\}, \\ &\ell_2(\boldsymbol{w}; \phi(\boldsymbol{x}_i^{\mathcal{O}})) = \max\{0, \rho - \langle \phi(\boldsymbol{x}_i^{\mathcal{O}}), \boldsymbol{w} \rangle\}, \\ &\ell_3(\boldsymbol{w}; \phi(\boldsymbol{x}_i^{\mathcal{S}})) = |\langle \phi(\boldsymbol{x}_i^{\mathcal{S}}), \boldsymbol{w} \rangle|. \end{aligned} \tag{6}$$

Next we will directly derive the primal problem into the kernel form. For each t, let x_j represents the data sample, and let

$$\begin{aligned} &\alpha_t[j] = |\{t' \leq t : i_{t'} = j \wedge y_j^{\mathcal{P}} \langle \boldsymbol{w}_{t'}, \phi(\boldsymbol{x}_j^{\mathcal{P}}) \rangle < 1\}|, \\ &\beta_t[j] = |\{t' \leq t : i_{t'} = j \wedge \langle \boldsymbol{w}_{t'}, \phi(\boldsymbol{x}_j^{\mathcal{O}}) \rangle < \rho\}|, \\ &\gamma_t[j] = \sum_j \operatorname{sgn} \langle \phi(\boldsymbol{x}_j^{\mathcal{S}}), \boldsymbol{w}_{t'} \rangle, \forall j \in \{t' \leq t : i_{t'} = j\}, \end{aligned}$$

then Eq. (5) and (6) can be rewritten as

$$\boldsymbol{w}_{t+1} = \frac{1}{\lambda t}(c_1\tau_1 \sum_{j=1}^{n_1} \alpha_{t+1}[j] y_j^{\mathcal{P}} \phi(\boldsymbol{x}_j^{\mathcal{P}}) + c_2\tau_2 \sum_{j=1}^{n_2} \beta_{t+1}[j]\phi(\boldsymbol{x}_j^{\mathcal{O}}) + c_3\tau_3 \sum_{j=1}^{n_3} \gamma_{t+1}[j]\phi(\boldsymbol{x}_j^{\mathcal{S}})).$$

According to the Representer Theorem [9], the optimal solution to Eq. (2) can be expressed as a linear combination of the training instances, thus we can rewrite $\boldsymbol{w}$ as:

$$\boldsymbol{w} = \sum_{j=1}^{n_1} \alpha[j]\phi(\boldsymbol{x}_j^{\mathcal{P}}) + \sum_{j=1}^{n_2} \beta[j]\phi(\boldsymbol{x}_j^{\mathcal{O}}) + \sum_{j=1}^{n_3} \gamma[j]\phi(\boldsymbol{x}_j^{\mathcal{S}}),$$

Let $\boldsymbol{\vartheta}$ be the whole parameter vector and $\mathcal{D}$ be the whole training dataset include all three types of labeled data

$$\boldsymbol{\vartheta} = [\alpha[1\cdots n_1], \beta[1\cdots n_2], \gamma[1\cdots n_3]],$$
$$\mathcal{D} = [\phi(\boldsymbol{x}_{[1\cdots n_1]}^{\mathcal{P}})^T, \phi(\boldsymbol{x}_{[1\cdots n_2]}^{\mathcal{O}})^T, \phi(\boldsymbol{x}_{[1\cdots n_3]}^{\mathcal{S}})^T]^T,$$

and $\boldsymbol{d}_i$ is the i-th in $\mathcal{D}$, $n = n_1 + n_2 + n_3$, then the objective function can be written in the following kernel form through a kernel operator $K(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle$, yielding the inner products after the mapping $\phi(\cdot)$:

$$\begin{aligned}
\min_{\boldsymbol{\vartheta}} \frac{1}{2} \sum_{i,j=1}^{n} \boldsymbol{\vartheta}[i]\boldsymbol{\vartheta}[j] K(\boldsymbol{d}_i, \boldsymbol{d}_j) \\
+ c_1\tau_1 \sum_{i=1}^{n_1} \max\{0, 1 - y_i^{\mathcal{P}} \sum_{j=1}^{n} \boldsymbol{\vartheta}[j] K(\boldsymbol{x}_i^{\mathcal{P}}, \boldsymbol{d}_j)\} \\
+ c_2\tau_2 \sum_{i=1}^{n_2} \max\{0, \rho - \sum_{j=1}^{n} \boldsymbol{\vartheta}[j] K(\boldsymbol{x}_i^{\mathcal{O}}, \boldsymbol{d}_j)\} \\
+ c_2\tau_3 \sum_{i=1}^{n_3} |\sum_{j=1}^{m} \boldsymbol{\vartheta}[j] K(\boldsymbol{x}_i^{\mathcal{S}}, \boldsymbol{d}_j)|
\end{aligned}$$

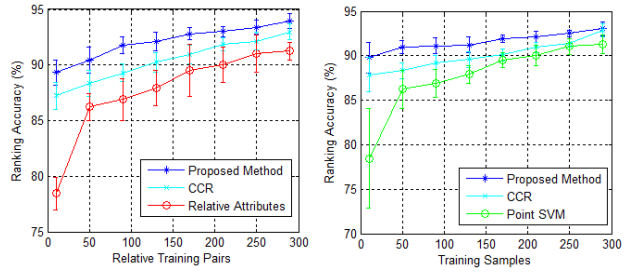
4. EXPERIMENTS

We evaluate our approach on two datasets with augmented relative attribute labels: (1) the **Outdoor Scene Recognition(OSR)** dataset [17][18] with 2688 images and 7 attributes, and the **Shoes** dataset [1][11] with 14568 images and 10 attributes. We directly use the features provided with the dataset of 512-dimensional gist descriptor for the **OSR** and 960-dimensional gist descriptor plus 20-dimensional color histogram for the **Shoes**.

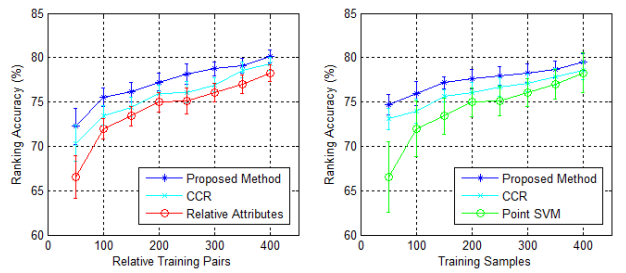
4.1 Accuracy of Hybrid Ranking

We first demonstrate that the proposed approach is capable of utilizing information from both type of labeled data with the comparison of three baseline approaches: Relative Attributes [18], pointwise SVM for ordinal regression [23] and CRR [21] which optimizes regression and ranking simultaneously. We compute the average ranking accuracy with standard deviation by running 10 rounds of each implemented approach. The average ranking accuracy is evaluated by the frequency of correctly ranked pairs. The parameters are selected by cross validation of 5 randomly selected small subsets per dataset.

Tables 1 and 2 demonstrate the average ranking accuracy with standard deviation of each attribute per dataset. Pointwise and pairwise training samples are randomly selected as 100 for **OSR** and 200 for **Shoes** with the rest of



(a) OSR



(b) Shoes

Figure 3: Learning curve of average ranking accuracy and corresponding standard deviation with regard to different number of pairwise or pointwise training samples on both dataset.

OSR and 6000 samples (due to memory limitation) from **Shoes** for test. The result shows that our approach apparently outperforms all baseline approaches in average ranking accuracies while generating lower standard deviations.

Fig. 3 illustrates how the ranking accuracy changes with different size of training samples on the attribute "Natural" of **OSR** and "Open" of **Shoes**. Specifically, the result in Fig. 3 Left is achieved by keeping the pointwise data size fixed at 100 and increasing the size of pairwise data. The blue curve shows the average ranking accuracy and standard deviation of our approach and the red and cyan curves shows the result of baseline approaches. In Fig. 3 Right, results of our approach (blue curve) compared with baseline approaches are shown where the size of training pairs is fixed at 100 and the size of training samples increases gradually. The results show that, in both configurations, our approach achieves obvious higher ranking accuracy and lower standard deviation than all baseline approaches. Besides, the result implies that, when fewer training samples were used, a higher performance gain was observed. For example, the best performance gain is 11% compared with Relative Attributes and 14% compared with pointwise SVM in "natural", when only 10 training samples or pairs are fed.

Fig. 4 shows some examples of ranked image pairs. In each column, the top image is more "natural" than the bottom image according to the ranking groundtruth. Fig. 4(a) is the comparison of our hybrid approach with Relative Attributes. The first three pairs (columns) are correctly ranked by our approach but incorrectly ranked by Relative Attributes, e.g., coast is more natural than highway, forest is more natural than inside city, mountain is more natural than buildings. The last three pairs are incorrectly ranked by our

Attribute Name	Hybrid Ranking(%)	CCR(%)	Relative Attribute(%)	Pointwise SVM(%)
Natural	**91.56±0.89**	88.75±0.90	87.09±2.08	88.86±2.24
Open	**88.50±0.62**	87.25±0.82	86.83±1.76	85.72±1.26
Perspective	**83.40±0.78**	82.23±0.81	80.20±1.69	80.56±1.73
Size-large	**72.93±1.04**	70.62±1.21	67.89±1.55	65.17±3.53
Diagonal-plane	**80.35±1.15**	78.42±1.08	76.25±1.76	76.61±2.73
Depth-cloth	**87.06±0.87**	85.24±0.95	84.27±1.66	82.32±1.51
Average	**83.97±0.80**	82.08±0.96	80.42 ±1.75	79.87±1.78

Table 1: Ranking accuracies and standard deviation of 6 attributes on the OSR dataset when the number of training samples and pairs are 100 for each attribute.

Attribute Name	Proposed Method(%)	CRR(%)	Relative Attribute(%)	Pointwise SVM(%)
Point at the front	**82.25±0.79**	81.42±0.82	80.61±1.08	79.13±1.53
Open	**76.02±0.83**	74.24±0.80	71.72±0.88	69.10±1.78
Bright in color	**64.40±0.76**	62.43±0.75	59.38±1.86	58.63±0.76
Covered with ornaments	**71.19±0.58**	70.02±0.61	68.88±0.72	59.10±3.87
Shiny	**79.60±0.52**	78.28±0.68	76.94±0.66	74.12±1.30
High at the heel	**80.71±0.75**	78.93±0.77	77.43±0.99	76.59±1.60
Long on the leg	**75.19±0.92**	73.32±0.82	72.44±1.07	69.08±2.19
Formal	**75.78±0.90**	74.45±0.91	72.37±1.10	70.76±1.57
Sporty	**80.24±0.90**	78.25±0.95	77.39±0.90	68.98±1.70
Feminine	**83.37±0.68**	82.42±0.70	81.38±0.71	81.86±1.50
Average	**76.88±0.76**	75.38±0.78	73.85±1.00	70.74±1.78

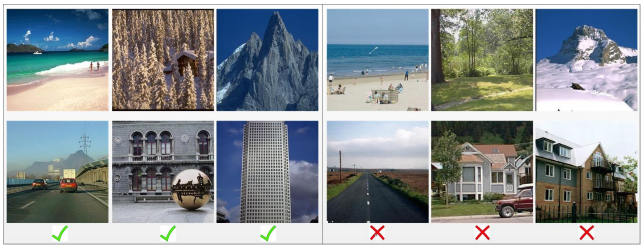
Table 2: Ranking accuracies and standard deviation of 10 attributes on the Shoes dataset when the number of training samples and pairs are 200 for each attribute.

approach but correctly ranked by Relative Attributes. The reason for the incorrect classification may be that our approach assigned a wrong category label to the scene. For instance, our approach also classified the bottom image of the fifth column as forest because of a tree appears in the scene. Fig. 4(b) illustrates the comparison of our approach with pointwise SVM. The first three pairs are correctly ranked by our approach but incorrectly ranked by pointwise SVM, e.g., coast is more natural than street, mountain is more natural than open county and forest is more natural than inside city. The last three pairs are incorrectly ranked by our approach but correctly ranked by pointwise SVM.

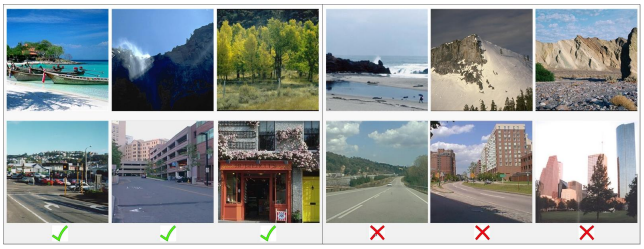
Based on these experiments, the potential performance gains of our approach appear to come from the extra information captured from different types of labels. In particular, the smaller standard deviation may result from the joint use of both information sources that help to denoise the training process.

4.2 Zero-Shot Learning

To further evaluate the proposed approach, we now consider the popular application of zero-shot learning. Given some training samples from some "seen" categories and some "unseen" categories without any training samples, zero-shot learning would predict the category labels of new samples. We compare our approach with the baseline approach Relative Attributes since [18] has shown that this approach outperforms most of the state of the art approaches on this regard. We followed the same parameter prediction rules of unseen categories as [4]. We adopted the same super parameters in Sect. 4.1 for model training. The average ranking accuracies with corresponding standard deviations are re-



(a) Hybrid v.s. Relative Attribute

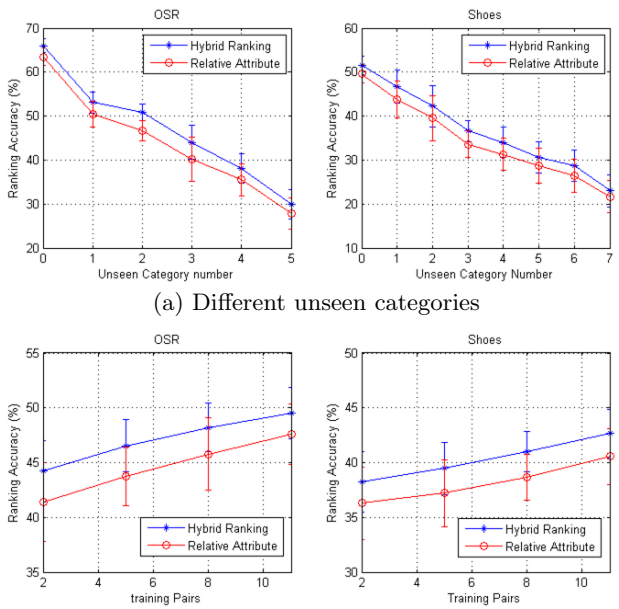


(b) Hybrid v.s. Pointwise SVM

Figure 4: Samples illustrating the ranking results. In groundtruth the top image is more "natrual" than the bottom image. The left three column is correctly ranked by the proposed approach while incorrectly ranked by the baseline. The right three is inverse.

Accuracy	70%	72.5%	75%	77.5%	80%	82.5%	85%
Online Learning	0.003 s	0.004 s	0.006 s	0.009 s	0.042 s	0.105 s	0.156 s
Batch Learning	0.006 s	0.098 s	0.104 s	0.231 s	0.451 s	1.558 s	6.308 s

Table 3: Elapsed times in order to achieve the same ranking accuracy (the less the better). The first row shows given ranking accuracies, and the second/third row shows the times needed for online/batch learning respectively to achieve the corresponding accuracy.



(a) Different unseen categories

(b) Different training pairs

Figure 5: Learning curve of zero-shot learning accuracy with regard to different unseen category numbers and training sample size on both dataset.

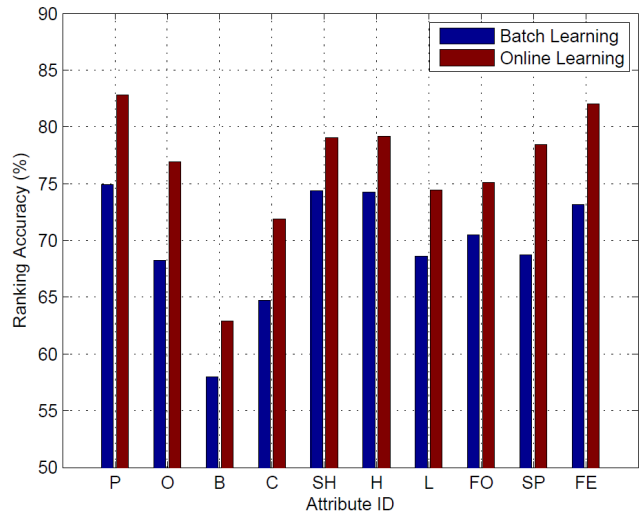


Figure 6: Average ranking accuracy of 10 attributes on the Shoes dataset by running the algorithms for 0.1 seconds. In each group the first bar is the result of batch learning and the second bar is the result of online learning.

ported by running each experiment 10 rounds. Assuming the data follows a Gaussian distribution, we estimated the mean and the covariance matrix of each (seen and unseen) category and assigned the category label of the new sample through maximum likelihood.

Fig. 5(a) shows the accuracy as a function of the number of unseen categories. For each seen category 30 images are left out for category parameter prediction, and 10 pointwise and 10 pairwise labelled samples are randomly picked for training. Results show that the ranking accuracy decreases as the number of unseen category increases. Our approach outperforms the baseline approach by around 3%.

Fig. 5(b) shows how the accuracy changes with the number of training pairs. In each run, 2 unseen categories and 30 images from the other seen categories are left out. 10 pointwise labelled samples are randomly picked for hybrid approach. Results show that ranking accuracies of both approaches increase with the increase of training pairs. Our approach yields performance gains by around 3% compared with the baseline approach.

4.3 Online Learning Evaluation

In this subsection, we compare the performance of the proposed online learning algorithm with the batch learning algorithm on the shoes dataset. For the online learning algorithm, the super parameters are set as $c_1 = 0.2$, $c_2 = 3$, $\rho = 0.1$. In training, we first construct a data pool mixed with both pointwise and pairwise data, and then randomly pick one data sample without considering the specific label type from the data pool for training. For batch learning, we construct the training dataset as half pointwise and half pairwise samples.

Fig. 6 illustrates the average ranking accuracy of 10 attributes by running both implemented approaches 10 rounds for a small time interval T (T=0.1 second in this experiment), simulating very limited training data availability. In each group, the first bar (blue) shows the result of batch learning and the second bar (red) shows the result of online learning. The results shows that in the same elapsed time of 0.1 second, the online learning algorithm clearly outperforms batch learning. The highest performance gain is obtained on the attribute "Sporty" by 9.69% and the lowest performance gain is for the attribute "Formal" by 4.63%.

Table 3 collects the elapsed time after both approaches achieved the same ranking performance from 70% to 85%. The results show that the batch learning approach takes longer time than online learning to achieve the same accuracy. With the ranking accuracy increased, the time difference become much more obvious. For example, batch learning takes double time (0.006s vs 0.003s) than online learning

to achieve the accuracy of 70%, and takes 40 times (6.308s vs 0.156s) more time to achieve the accuracy of 85%.

5. CONCLUSIONS

We proposed a hybrid ranking framework for supporting adaptive, attribute-based image retrieval. We evaluated the proposed approach on two image datasets. The results show that through capturing the information from both relative attribute strength (pairwise) and absolute attribute scale (pointwise), our method is able to achieve better ranking performance than Relative Attribute and pointwise SVM, which are current leading approaches that learn the ranking function purely based on either pairwise or pointwise data. We also proposed an online learning algorithm for the proposed framework and derived the formulation into the kernel form. The experiments of online learning and batch learning show that our online learning algorithm can achieve much better ranking performance than batch learning given the same running time or can achieve better performance in much less time. The results also suggest that the less training data are available, the more relative performance gains can be obtained by our approach than independent learning.

6. ACKNOWLEDGMENTS

The work was supported in part by a grant from National Science Foundation and a grant from Army Research Office. Any opinions and conclusions expressed in this material are those of the author(s) and do not necessarily reflect the view of sponsors.

7. REFERENCES

[1] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *Proc. of ECCV'10*, pages 663–676, Berlin, Heidelberg, 2010. Springer-Verlag.

[2] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proc. of ICML '05*, pages 89–96. ACM, 2005.

[3] Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon. Adapting ranking svm to document retrieval. In *Proc. of SIGIR '06*, pages 186–193. ACM, 2006.

[4] L. Chen, Q. Zhang, and B. Li. Predicting multiple attributes via relative multi-task learning. In *Proc. of CVPR '14*, pages 1027–1034, July 2014.

[5] N. Elahi, R. Karlsen, and E. J. Holsbø. Personalized photo recommendation by leveraging user modeling on social network. In *Proc. of IIWAS '13*, pages 68:68–68:71. ACM, 2013.

[6] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Proc. of CVPR '09*, pages 1778–1785, June 2009.

[7] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, Dec. 2003.

[8] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. of KDD '02*, pages 133–142. ACM, 2002.

[9] G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1):82 – 95, 1971.

[10] A. Kovashka and K. Grauman. Attribute adaptation for personalized image search. In *Proc. of ICCV '13*, pages 3432–3439, Dec 2013.

[11] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Image search with relative attribute feedback. In *Proc. of CVPR'12*, pages 2973–2980, June 2012.

[12] N. Kumar, P. Belhumeur, and S. Nayar. Facetracer: A search engine for large collections of images with faces. In *Proc. of ECCV '08*, pages 340–353. Springer-Verlag, 2008.

[13] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Attribute and simile classifiers for face verification. In *Proc. of CVPR '09*, pages 365–372, Sept 2009.

[14] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Proc. of CVPR '09*, pages 951–958, June 2009.

[15] P. Li, Q. Wu, and C. J. Burges. Mcrank: Learning to rank using multiple classification and gradient boosting. In *Proc. of NIPS '08*, pages 897–904. Curran Associates, Inc., 2008.

[16] T. Moon, A. Smola, Y. Chang, and Z. Zheng. Intervalrank: Isotonic regression with listwise and pairwise constraints. In *WSDM*, 2010.

[17] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42(3):145–175, May 2001.

[18] D. Parikh and K. Grauman. Relative attributes. In *Proc. of ICCV '11*, pages 503–510, Nov 2011.

[19] C. Renjifo and C. Carmen. The discounted cumulative margin penalty: Rank-learning with a list-wise loss and pair-wise margins. In *MLSP*, Sept 2012.

[20] J. Sang, C. Xu, and D. Lu. Learn to personalized image search from the photo sharing websites. *Multimedia, IEEE Transactions on*, 14(4):963–974, Aug 2012.

[21] D. Sculley. Combined regression and ranking. In *Proc. of SIGKDD'10*, pages 979–988, New York, NY, USA, 2010. ACM.

[22] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proc. of ICML '07*, pages 807–814. ACM, 2007.

[23] A. Shashua and A. Levin. Ranking with large margin principle: Two approaches. In *Proc. of NIPS '03*, pages 961–968. MIT Press, 2003.

[24] M.-F. Tsai, T.-Y. Liu, T. Qin, H.-H. Chen, and W.-Y. Ma. Frank: A ranking method with fidelity loss. In *Proc. of SIGIR '07*, pages 383–390. ACM, 2007.

[25] D. Vaquero, R. Feris, D. Tran, L. Brown, A. Hampapur, and M. Turk. Attribute-based people search in surveillance environments. In *Proc. of WACV'09*, December 2009.

[26] Y. Wang and G. Mori. A discriminative latent model of object classes and attributes. In *Proc. of ECCV'10*, pages 155–168. Springer-Verlag, 2010.