Image Ranking and Retrieval based on Multi-Attribute Queries

Behjat Siddiquie[1] Rogerio S. Feris[2] Larry S. Davis[1]
[1]University of Maryland, College Park [2]IBM T. J. Watson Research Center
{behjat,lsd}@cs.umd.edu rsferis@us.ibm.com

Abstract

We propose a novel approach for ranking and retrieval of images based on multi-attribute queries. Existing image retrieval methods train separate classifiers for each word and heuristically combine their outputs for retrieving multi-word queries. Moreover, these approaches also ignore the interdependencies among the query terms. In contrast, we propose a principled approach for multi-attribute retrieval which explicitly models the correlations that are present between the attributes. Given a multi-attribute query, we also utilize other attributes in the vocabulary which are not present in the query, for ranking/retrieval. Furthermore, we integrate ranking and retrieval within the same formulation, by posing them as structured prediction problems. Extensive experimental evaluation on the Labeled Faces in the Wild(LFW), FaceTracer and PASCAL VOC datasets show that our approach significantly outperforms several state-of-the-art ranking and retrieval methods.

1. Introduction

In the past few years, methods that exploit the semantic attributes of objects have attracted significant attention in the computer vision community. The usefulness of these methods has been demonstrated in several different application areas, including object recognition [5, 17, 24] face verification [16] and image search [22, 15].

In this paper we address the problem of *image ranking and retrieval based on semantic attributes*. Consider the problem of ranking/retrieval of images of people according to queries describing the physical traits of a person, including facial attributes (*e.g.* hair color, presence of beard or mustache, presence of eyeglasses or sunglasses etc.), body attributes (*e.g.* color of shirt and pants, striped shirt, long/short sleeves etc.), demographic attributes (*e.g.* age, race, gender) and even non-visual attributes (*e.g.* voice type, temperature and odor) which could potentially be obtained from other sensors. There are several applications that naturally fit within this attribute based ranking and retrieval framework. An example is criminal investigation. To locate a suspect, law enforcement agencies typically gather the physical traits of the suspect from eyewitnesses. Based on the description obtained, entire video archives from surveillance cameras are scanned manually for persons with similar characteristics. This process is time consuming and can be drastically accelerated by an effective image search mechanism.

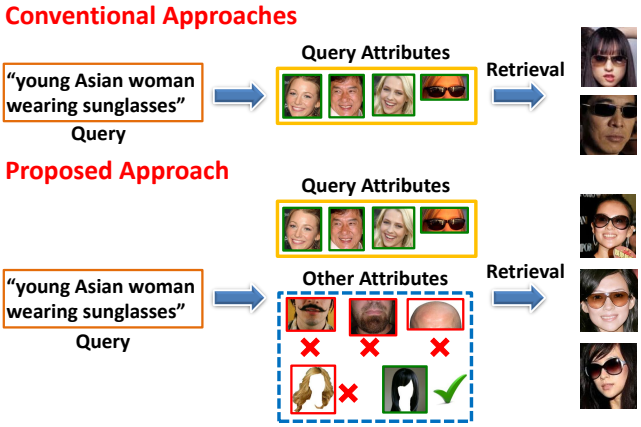


Figure 1: Given a multi-attribute query, conventional image retrieval methods such as [22, 15], consider only the attributes that are part of the query, for retrieving relevant images. On the other hand, our proposed approach also takes into account the remaining set of attributes that are not a part of the query. For example, given the query "*young Asian woman wearing sunglasses*", our system infers that relevant images are unlikely to have a *mustache*, *beard* or *blonde hair* and likely to have *black hair*, thereby achieving superior results.

Searching for images of people based on visual attributes has been previously investigated in [22, 15]. Vaquero *et al.* [22] proposed a video based surveillance system that supports image retrieval based on attributes. They argue that while face recognition is extremely challenging in surveillance scenarios involving low-resolution imagery, visual attributes can be effective for establishing identities over short periods of time. Kumar *et al.* have built an image search engine [15] where users can retrieve images of faces based on queries involving multiple visual attributes. However, these methods do not consider the fact that attributes are highly correlated. For example, a person who has a *mustache* is almost definitely a *male*, or a person who is *Asian* is unlikely to have *blonde hair*.

We present a new framework for multi-attribute image

retrieval and ranking, which retrieves images based not only on the words that are part of the query, but also considers the remaining attributes within the vocabulary that could potentially provide information about the query (Figure 1). Consider a query such as *"young Asian woman wearing sunglasses"*. Since the query contains the attribute *young*, pictures containing people with gray hair, which usually occurs in older people, can be discounted. Similarly pictures containing bald people or persons with mustaches and beards, which are male specific attributes, can also be discarded, since one of the constituent attributes of the query is *woman*. While an individual detector for the attribute *woman*, will implicitly learn such features, our experiments show that when searching for images based on queries containing fine-grained parts and attributes, explicitly modeling the correlations and relationships between attributes can lead to substantially better results.

In image retrieval, the goal is to return the set of images in a database that are relevant to a query. The aim of ranking is similar, but with additional requirement that the images be ordered according to their relevance to the query. For large scale datasets, it is essential for an image search application to rank the images such that the *most* relevant images are at the top. Ranking based on a single attribute can sometimes seem unnecessary; for example, for a query like "*beard*", one can simply classify images into people with beards and people without beards. For multi-attribute queries however, depending on the application, one can have multiple levels of relevance. For example, consider a query such as "*man wearing a red shirt and sunglasses*", since sunglasses can be easily removed, it is reasonable to assume that images containing men wearing a red shirt but without sunglasses are also relevant to the query, but perhaps less relevant than images of men with both a red shirt and sunglasses. Hence, we also consider the problem of ranking based on multi-attribute queries to improve the effectiveness of attribute based image search. Instead of treating ranking as a separate problem, we propose a structured learning framework, which integrates ranking and retrieval within the same formulation.

While searching for images of people involves only a single object class (human faces), we show that our approach is general enough to be utilized for attribute based retrieval of images containing multiple object classes, and outperforms a number of different ranking and retrieval methods on three different datasets - LFW [11] and FaceTracer [15] for human faces and PASCAL [4] for multiple object categories.

There are three key **contributions** of our work: **(1)** We propose a single framework for image ranking and retrieval. Traditionally, *learning to rank* is treated as a distinct problem within information retrieval. In contrast, our approach deals with ranking and retrieval within the same formulation, where learning to rank or retrieve are simply optimizations of the same model according to different performance measures. **(2)** Our approach supports image retrieval and ranking based on multi-label queries. This is non-trivial, as the number of possible multi-label queries for a vocabulary of size L is 2^L. Most image ranking/retrieval approaches deal with this problem by learning separate classifiers for each individual label, and retrieve multi-label queries by heuristically combining the outputs of the individual labels. In contrast, we introduce a principled framework for training and retrieval of multi-label queries. **(3)** We also demonstrate that attributes within a single object category and even across multiple object categories are interdependent so that modeling the correlations between them leads to significant performance gains in retrieval and ranking.

2. Related Work

An approach that has proved extremely successful for document retrieval is *learning to rank* [12, 7, 18, 2], where a ranking function is learnt, given either the pairwise preference relations or relevance levels of the training examples. Similar methods have also been proposed for ranking images, [10]. Several image retrieval methods, which retrieve images relevant to a textual query, adopt a *visual reranking* framework [1, 6, 13, 23], which is a two stage process. In the first stage images are retrieved based purely on textual features like tags(*e.g.* in Flickr), query terms in webpages and image meta data. The second stage involves reranking or filtering these images using a classifier trained on visual features. A major limitation of these approaches is the requirement of textual annotations for the first stage of retrieval, which are not always available in many applications - for example the surveillance scenario described in the introduction. Another drawback of both the image ranking approaches as well as the *visual reranking* methods is that they learn a separate ranking/classification function corresponding to each query term and hence have to resort to ad-hoc methods for retrieving/ranking multi-word queries. A few methods have been proposed for dealing with multi-word queries. Notable among them are PAMIR [8] and TagProp [9]. However, these methods do not take into account the dependencies between query terms. We show that there often exist significant dependencies between query words and modeling them can substantially improve ranking and retrieval performance.

Recently, there have been several works which utilize an attribute based representation to improve performance of computer vision tasks. In [5], Farhadi *et al*. advocate an attribute centric approach for object recognition, and show that in addition to effective object recognition, it helps in describing unknown object categories and detecting unexpected attributes in known object classes. Similarly, Lampert *et al*. [17] learn models of unknown object categories from attributes based on textual descriptions. Kumar *et al*. [16] have shown that comparing faces based on facial attributes and other visual traits can significantly improve face verification. Wang and Mori [24] have demonstrated that recognizing attributes and modeling the interdependencies between them can help improve object recognition performance. In general, most of these methods exploit the fact that attributes provide a high level representation which is

compact and semantically meaningful.

Tsochantaridis *et al*. introduced Structured SVMs [21] to address prediction problems involving complex outputs. Structured SVMs provide efficient solutions for structured output problems, while also modeling the interdependencies that are often present in the output spaces of such problems. They have been effectively used for object localization [3] and modeling the cooccurrence relationships between attributes [24]. The structured learning framework has also been utilized for document ranking [18], which is posed as a structured prediction problem by having the output be a permutation of the documents. In this work, we employ structured learning to pose a single framework for ranking and retrieval, while also modeling the correlations between the attributes.

3. Multi Attribute Retrieval and Ranking

We now describe our Multi-Attribute Retrieval and Ranking(MARR) approach. Our image retrieval method is based on the concept of reverse learning. Here, we are given a set of labels $\mathcal{X}$, and a set of training images $\mathcal{Y}$. Corresponding to each label x_i ($x_i \in \mathcal{X}$) a mapping is learned to predict the set of images y ($y \subset \mathcal{Y}$) that contain the label x_i. Since reverse learning has a structured output (set of images) it fits well into the structured prediction framework. Reverse learning was recently proposed in [19], and was shown to be extremely effective for multi-label classification. The main advantage of reverse learning is that it allows for learning based on the minimization of loss functions corresponding to a wide variety of performance measures such as *hamming loss*, *precision* and *recall*. We build upon this approach in three different ways. First we propose a single framework for both retrieval and ranking. This is accomplished by adopting a ranking approach similar to [18], where the output is a set of images ordered by relevance, enabling integration of ranking and reverse learning within the same framework. Secondly, we facilitate training, as well as retrieval and ranking, based on queries consisting of multiple-labels. In [19], training and retrieval were performed independently for each label, whereas we explicitly utilize multi-labeled samples present in the training set for the purpose of learning our model. Finally, we model and learn the pairwise correlations between different labels(attributes) and exploit them for retrieval and ranking. We show that these improvements result in significant performance gains for both ranking and retrieval.

3.1. Retrieval

Given a multi-attribute query $\mathcal{Q}$, where $\mathcal{Q} \subset \mathcal{X}$, our goal is to retrieve images from the set $\mathcal{Y}$ that are relevant to $\mathcal{Q}$. Under the reverse learning formulation described above, for an input $\mathcal{Q}$, the output is the set of images y^* that contain all the constituent attributes in $\mathcal{Q}$. Therefore, the prediction function $f_w : \mathcal{Q} \rightarrow y$ returns the set y^* which maximizes the score over the weight vector w:

$$y^* = \arg\max_{y \subset \mathcal{Y}} w^T \psi(\mathcal{Q}, y) \tag{1}$$

here w is composed of two components; w^a for modeling the appearance of individual attributes and w^p for modeling the dependencies between them. We define $w^T\psi(\mathcal{Q}, y)$ as:

$$w^T\psi(\mathcal{Q}, y) = \sum_{x_i \in \mathcal{Q}} w_i^a \Phi_a(x_i, y) + \sum_{x_i \in \mathcal{Q}} \sum_{x_j \in \mathcal{X}} w_{ij}^p \Phi_p(x_j, y) \tag{2}$$

where

$$\Phi_a(x_i, y) = \textstyle\sum_{y_k \in y} \phi_a(x_i, y_k) \tag{3}$$

$$\Phi_p(x_j, y) = \textstyle\sum_{y_k \in y} \phi_p(x_j, y_k) \tag{4}$$

$\phi_a(x_i, y_k)$ is the feature vector representing image y_k for attribute x_i. $\phi_p(x_j, y_k)$ indicates the presence of attribute x_j in image y_k, which is not known during the test phase and hence $\phi_p(x_j, y_k)$ can be treated as a latent variable [24]. However, we adopt a simpler approach and set $\phi_p(x_j, y_k)$ to be the output of an independently trained attribute detector. In equation 2, w_i^a is a standard linear model for recognizing attribute x_i based on the feature representation $\phi_a(x_i, y_k)$ and w_{ij}^p is a potential function encoding the correlation between the pair of attributes x_i and x_j. By substituting (3) into the first part of (2), one can intuitively see that this represents the summation of the confidence scores of all the individual attributes x_i in the query $\mathcal{Q}$, over all the images $y_k \in y$. Similarly, the second(pairwise) term in (2) represents the correlations between the query attributes $x_i \in \mathcal{Q}$ and the entire set of attributes $\mathcal{X}$, over images in the set y. Hence, the pairwise term ensures that information from attributes that are not present in the query $\mathcal{Q}$, is also utilized for retrieving the relevant images.

Given a set of multi-label training images $\mathcal{Y}$ and their respective labels, our aim is to train a model w which given a multi-label query $\mathcal{Q} \subset \mathcal{X}$, can correctly predict the subset of images y_t^* in a test set $\mathcal{Y}_t$, which contain all the labels $x_i \in \mathcal{Q}$. Let $\mathbf{Q}$ be the set of queries; in general we can include all queries, containing a single attribute as well as multiple attributes, that occur in the training set. During the training phase, we want to learn w such that, for each query $\mathcal{Q}$, the desired output set of retrieved images y^*, has a higher score (equation 1) than any other set $y \in \mathcal{Y}$. This can be performed using a standard max-margin training formulation:

$$\begin{aligned} \arg\min_{w,\xi} \quad & w^T w + C \textstyle\sum_t \xi_t \\ \forall\ t \quad & w^T\psi(\mathcal{Q}_t, y_t^*) - w^T\psi(\mathcal{Q}_t, y_t) \geq \Delta(y_t^*, y_t) - \xi_t \end{aligned} \tag{5}$$

where C is a parameter controlling the trade-off between the training error and regularization, $\mathcal{Q}_t$ ($\mathcal{Q}_t \in \mathbf{Q}$) are the training queries, ξ_t is the slack variable corresponding to query $\mathcal{Q}_t$ and $\Delta(y_t^*, y_t)$ is the loss function. Unlike standard SVMs which use a simple 0/1 loss, we employ a complex loss function as it enables us to heavily(gently) penalize outputs y_t that deviate significantly(slightly) from the correct output y_t^*, measured based on the performance metric we want to optimize for. For example, we can define $\Delta(y_t^*, y_t)$ for optimizing training error based on different performance metrics as follows:

$$\Delta(y_t^*, y_t) = \begin{cases} 1 - \frac{|y_t \cap y_t^*|}{|y_t|} & \text{precision} \\ 1 - \frac{|y_t \cap y_t^*|}{|y_t^*|} & \text{recall} \\ 1 - \frac{|y_t \cap y_t^*| + |\bar{y}_t \cap \bar{y}_t^*|}{|\mathcal{Y}|} & \text{hamming loss} \end{cases} \quad (6)$$

Similarly, one can optimize for other performance measures such as F_β. This is the main advantage of the reverse learning approach, as it allows one to train a model optimizing for a variety of performance measures.

The quadratic optimization problem in Equation 5 contains $O(|\mathbf{Q}|2^{|\mathcal{Y}|})$ constraints, which is exponential in the number of training instances $|\mathcal{Y}|$. Hence, we adopt the constraint generation strategy proposed in [21], which consists of an iterative procedure that involves solving Equation 5, initially without any constraints, and then at each iteration adding the most violated constraint of the current solution to the set of constraints. At each iteration of the constraint generation process, the most violated constraint is given by:

$$\xi_t \geq \max_{y_t \subset \mathcal{Y}} \left[\Delta(y_t^*, y_t) - (w^T \psi(\mathcal{Q}_t, y_t^*) - w^T \psi(\mathcal{Q}_t, y_t))\right] \quad (7)$$

Equation 7 can be solved in $O(|\mathcal{Y}|^2)$ time, as shown in [19]. During prediction, we need to solve for 1, which again as shown in [19] can be efficiently performed in $O(|\mathcal{Y}| \log(|\mathcal{Y}|))$.

3.2. Ranking

We now show that, with minor modifications, the proposed framework for image retrieval can also be utilized for ranking multi-label queries. In the case of image ranking, given a multi-attribute query $\mathcal{Q}$, where $\mathcal{Q} \subset \mathcal{X}$, our goal is to rank the set of images $\mathcal{Y}$ according to their relevance to $\mathcal{Q}$. Unlike image retrieval, where given an input $\mathcal{Q}$, the output is a subset of the test images, in the case of ranking the output of the prediction function $f_w : \mathcal{Q} \rightarrow z$, is a permutation z^*, of the set of images $\mathcal{Y}$:

$$z^* = \arg \max_{z \in \pi(\mathcal{Y})} w^T \psi(\mathcal{Q}, z) \quad (8)$$

where $\pi(\mathcal{Y})$ is the set of all possible permutations of the set of images $\mathcal{Y}$. For the case of ranking, we make a slight modification to ψ by having:

$$w^T \psi(\mathcal{Q}, z) = \sum_{x_i \in \mathcal{Q}} w_i^a \hat{\Phi}_a(x_i, z) + \sum_{x_i \in \mathcal{Q}} \sum_{x_j \in \mathcal{X}} w_{ij}^p \hat{\Phi}_p(x_j, z) \quad (9)$$

where

$$\hat{\Phi}_a(x_i, z) = \textstyle\sum_{z_k \in z} A(r(z_k)) \phi_a(x_i, z_k) \quad (10)$$

$$\hat{\Phi}_p(x_j, z) = \textstyle\sum_{z_k \in z} A(r(z_k)) \phi_p(x_j, z_k) \quad (11)$$

with $A(r)$ being any non-increasing function and $r(z_k)$ being the rank of image z_k. Suppose we care only about the ranks of the top K images, we can define $A(r)$ as:

$$A(r) = \max(K + 1 - r, 0) \quad (12)$$

This ensures that the lower(top) ranked images are assigned higher weights and since $A(r) = 0$ for $r > K$, only the top K images of the ranking are considered.

During the training phase, we are given a set of training images $\mathcal{Y}$ and the set of queries, $\mathbf{Q}$, that occur among them. Unlike many ranking methods, which simply divide the set of training images into two sets - *relevant* and *irrelevant* - corresponding to each query and just learn a binary ranking, we utilize multiple levels of relevance. Given a query $\mathcal{Q}$, we divide the training images into $|\mathcal{Q}| + 1$ sets based on their relevance. The most relevant set consists of images that contain all the attributes in the query $\mathcal{Q}$, and are assigned a relevance $\text{rel}(j) = |\mathcal{Q}|$, the next set consists of images containing any $|\mathcal{Q}|-1$ of the attributes which are assigned a relevance $\text{rel}(j) = |\mathcal{Q}| - 1$ and so on, with the last set consisting of images with none of the attributes present in the query and they are assigned relevance $\text{rel}(j) = 0$. This ensures that, in case there are no images containing all the query attributes, images that contain the most number of attributes are ranked highest. While we have assigned equal weights to all the attributes, one can conceivably assign higher weights to attributes involving race or gender which are difficult to modify and lower weights to attributes that can be easily changed(*e.g. wearing sunglasses*). We use a max-margin framework, similar to the one used in retrieval but with a different loss function, for training our ranking model:

$$\begin{aligned} \arg\min_{w,\xi} \quad & w^T w + C \textstyle\sum_t \xi_t \quad (13) \\ \forall\ t \quad & w^T \psi(\mathcal{Q}_t, z_t^*) - w^T \psi(\mathcal{Q}_t, z_t) \geq \Delta(z_t^*, z_t) - \xi_t \end{aligned}$$

where $\Delta(z^*, z)$ is a function denoting the loss incurred in predicting the permutation z instead of the correct permutation z^*, which we define as $\Delta(z^*, z) = 1 - \text{NDCG@100}(z^*, z)$. The *normalized discount cumulative gain*(NDCG) score is a standard measure used for evaluating ranking algorithms. It is defined as:

$$NDCG@k = \frac{1}{Z} \sum_{j=1}^{k} \frac{2^{\text{rel}(j)} - 1}{\log(1 + j)} \quad (14)$$

where $\text{rel}(j)$ is the relevance of the j^{th} ranked image and Z is a normalization constant to ensure that the correct ranking results in an NDCG score of 1. Since NDCG@100 takes into account only the top 100 ranked images, we set $K = 100$ in Equation (12).

In the case of ranking, the max-margin problem (Equation 13) again contains an exponential number of constraints and we adopt the constraint generation procedure, where the most violated constraint is iteratively added to the optimization problem. The most violated constraint is given by:

$$\xi_t \geq \max_{z_t \in \pi(\mathcal{Y})} \left[\Delta(z_t^*, z_t) - (w^T \psi(\mathcal{Q}_t, z_t^*) - w^T \psi(\mathcal{Q}_t, z_t))\right] \quad (15)$$

which, after omitting terms independent of z_t and substituting Equations (9),(10),(14) can be rewritten as:

$$\arg \max_{z_t \in \pi(\mathcal{Y})} \sum_{k=1}^{100} A(z_k) W(z_k) - \sum_{k=1}^{100} \frac{2^{\text{rel}(z_k)} - 1}{\log(1+k)} \quad (16)$$

where

$$W(z_k) = \sum_{x_i \in \mathcal{Q}_t} w_i^a \phi_a(x_i, z_k) + \sum_{x_j \in \mathcal{Q}_t} \sum_{x_j \in \mathcal{X}} w_{ij}^p \phi_p(x_j, z_k) \quad (17)$$

Equation (16) is a linear assignment problem in z_k and can be efficiently solved using the Kuhn-Munkres algorithm [14]. During prediction, Equation (8) needs to be solved, which can be rewritten as:

$$\arg \max_{z \in \pi(\mathcal{Y})} \sum_{k} A(r(z_k)) W(z_k) \quad (18)$$

Since $A(z_j)$ is a non-increasing function, ranking can be performed by simply sorting the samples according to the values of $W(z_k)$.

4. Experiments and Results

Implementation Details: Our implementation is based on the "Bundle Methods for Regularized Risk Minimization" BMRM solver of [20]. In order to speed up the training, we adopt the technique previously used in [24, 3], which involves replacing $\phi_a(x_i, y_k)$ in Equations (3),(10) by the output of the binary attribute detector of attribute x_i for the image y_k. This technique is also beneficial during retrieval, as pre-computing the output scores for different attributes can be done offline, significantly speeding up retrieval and ranking.

4.1. Evaluation

Retrieval: We compare our image retrieval approach to two state-of-the-art methods: Reverse Multi-Label Learning (RMLL) [19] and TagProp [9]. Neither of these methods explicitly model the correlations between pairs of attributes and in the case of multi-label queries we simply sum up the per-attribute confidence scores of the constituent attributes. In case of TagProp, we use the σML variant which was shown to perform the best [9]. Furthermore, for multi-label queries, we found that adding up the probabilities of the individual words gave better results and hence we sum up the output scores, instead of multiplying them as done in [9]. In case of RMLL and MARR we optimize for the hamming loss by setting the loss function as defined in (6).

Ranking: In case of ranking, we compare our approach against several standard ranking algorithms including rankSVM [12], rankBoost [7], Direct Optimization of Ranking Measures(DORM) [18] and TagProp [9], using code that was publicly available[1]. Here again, for ranking multi-attribute queries, we add up the output scores obtained from the individual attribute rankers.

[1]rankSVM `www.cs.cornell.edu/People/tj/svm_light/svm_rank.html`; rankBoost `http://www-connex.lip6.fr/~amini/SSRankBoost/`; DORM `http://users.cecs.anu.edu.au/~chteo/BMRM.html`; TagProp `http://lear.inrialpes.fr/people/guillaumin/code.php\#tagprop`

We perform experiments on three different datasets (1) Labeled Faces in the Wild(LFW) [11] (2) FaceTracer [15] and (3) PASCAL VOC 2008 [4]. We point out that there is an important difference between these datasets. While the LFW and FaceTracer datasets consist of multiple attributes within a single class *i.e. human faces*, the PASCAL dataset contains multiple attributes across multiple object classes. This enables us to evaluate the performance of our algorithm in two different settings.

4.2. Labeled Faces in the Wild (LFW)

We first perform experiments on the Labeled Faces in the Wild(LFW) dataset [11]. While, LFW has been primarily used for face verification, we use it for evaluation of ranking and retrieval based on multi-attribute queries. A subset consisting of 9992 images from LFW was annotated with a set of 27 attributes (Table 1). We randomly chose 50% of these images for training and the remaining were used for testing.

Asian	Goatee	No Beard
Bald	Gray Hair	No Eyewear
Bangs	Hat	Senior
Beard	Indian	Sex
Black	Kid	Short Hair
Black Hair	Lipstick	Sunglasses
Blonde Hair	Long Hair	Visible Forehead
Brown Hair	Middle Aged	White
Eyeglasses	Mustache	Youth

Table 1: **List of Attributes**

We extract a large variety of features for representing each image. Color based features include *color histograms, color corelograms, color wavelets* and *color moments*. Texture is encoded using *wavelet texture* and *LBP histograms*, while shape information is represented using *edge histograms, shape moments* and SIFT based visual words. To encode spatial information, we extract feature vectors of each feature type from individual grids of five different configurations (Fig. 2) and concatenate them. This enables localization of individual attribute detectors, for example, the attribute detector for *hat* or *bald* will give higher weights to features extracted from the topmost grids in the configurations *horizontal parts* and *layout* (Fig. 2).

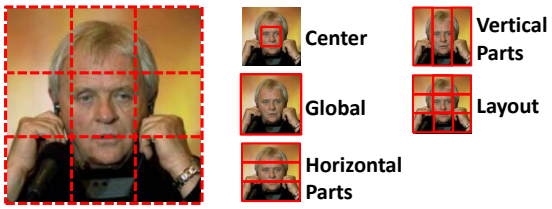


Figure 2: **Facial Feature Extraction:** Images are divided into a 3×3 grid(*left*) and features are extracted from five different configurations(*middle*,*center*).

Figure 5 plots the NDCG scores, as a function of the ranking truncation level K, for different ranking methods. From the figure, it is clear that MARR (our approach) is significantly better than the other methods for all three types of queries, at all values of K. At a truncation level of 10 (NDCG@10), for single, double and triple attribute queries, MARR is respectively, 8.9%, 7.7% and 8.8% better than rankBoost [7], the second best method. The retrieval results are shown in Figure 3. In this case, we compare the mean areas under the ROC curves for single, double and triple attribute queries. Here MARR is 7.0%, 6.7% and 6.8% better than Reverse Multi-Label Learning (RMLL [19]), for single, double and triple attribute queries respectively. Compared to TagProp [9], MARR is 8.8%, 10.1% and 11.0% better for the three kinds of queries. Some qualitative results, for different kinds of queries are shown in Figure 4.

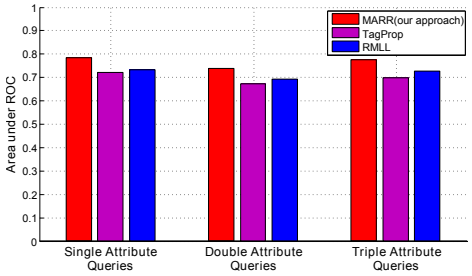


Figure 3: Retrieval Performance on the LFW dataset.

Figure 6 shows the weights learnt by the MARR ranking model on the LFW dataset. Each row of the matrix represents Equation 9 for a single-attribute query, with the diagonal elements representing w_i^a and the off-diagonal entries representing the pairwise weights w_{ij}^p. As expected, the highest weights are assigned to the diagonal elements underlining the importance of the individual attribute detectors. Among the pairwise elements, the lowest weights are assigned to attribute pairs that are mutually exclusive such as (*White*,*Asian*), (*Eyeglasses*,*No-Eyewear*) and (*Short-Hair*,*Long-Hair*). Rarely co-occuring attribute pairs like (*Kid*,*Beard*), and (*Lipstick*,*Sex*) (*Sex* is 1 for male and 0 for female) are also assigned low weights. Pairs of attributes such as (*Middle-aged*,*Eyeglasses*) and (*Senior*,*Gray-Hair*) that commonly co-occur are given relatively higher weights. Also note that the weights are asymmetric, for example, a person who has a beard is very likely to also have a mustache, but not the other way round. Hence while retrieving images for the query "*mustache*", the presence of a beard is a good indicator of a relevant image, but not vice-versa, and this is reflected in the weights learnt.

4.3. FaceTracer Dataset

We next evaluate our approach on the FaceTracer Dataset [15]. We annotated about 3000 images from the dataset with the same set of facial attributes (Table 1) that was used on LFW. We represent each image by the same feature set and compare the performance of the ranking models learnt on the LFW training set. Figure 7 summarizes the results. One can observe that the performance of each method drops when compared to LFW. This is due to the difference in the distributions of the two datasets. For example, the FaceTracer dataset contains many more images of babies and small children compared to LFW. However, MARR still outperforms all the other methods and its NDCG@10 score is 5.0%, 8.1% and 11.6% better than the second best method(rankBoost) for single, double and triple attribute queries respectively, demonstrating the robustness of our approach.

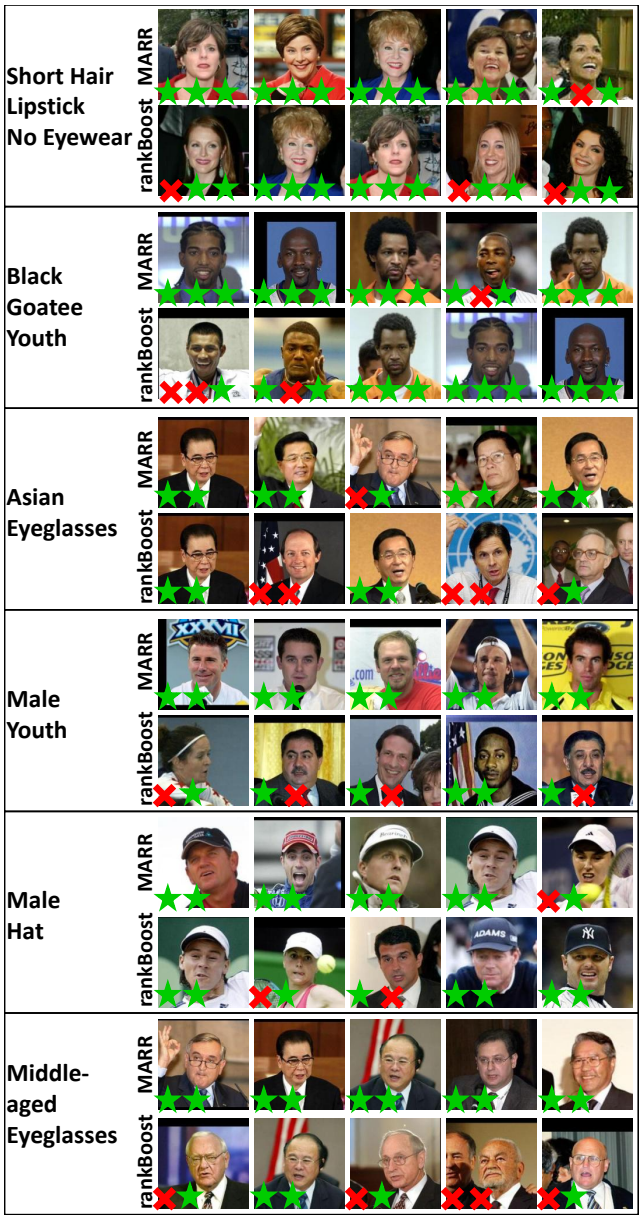
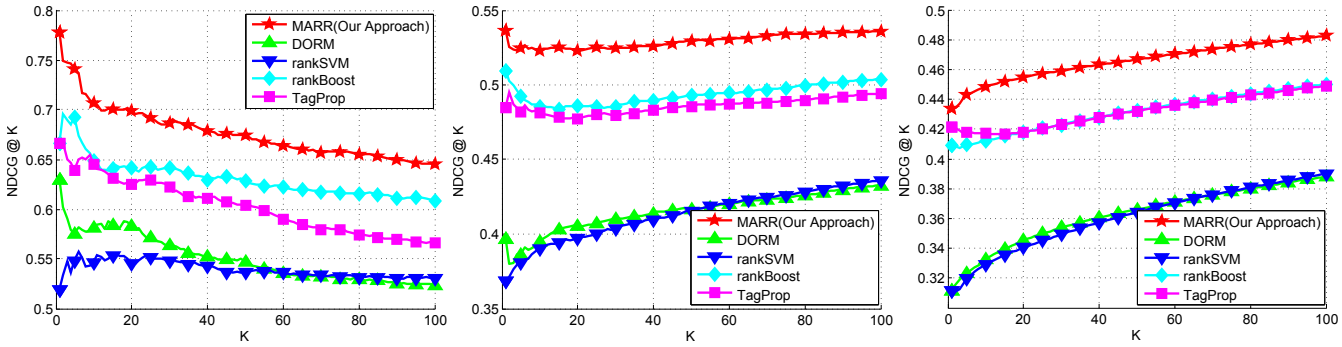
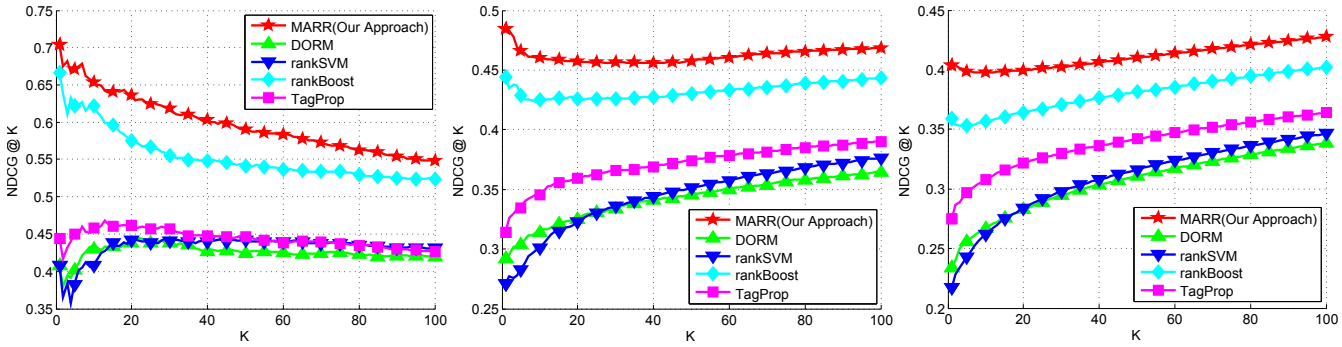


Figure 4: **Qualitative results:** Sample multi-label ranking results obtained by MARR and RankBoost(the second best method) for different queries on the LFW dataset. A *green star*(*red cross*) indicates that the image contains(does not contain) the corresponding attribute.



(a) Single Attribute Queries (b) Double Attribute Queries (c) Triple Attribute Queries

Figure 5: Ranking Performance on the LFW dataset



(a) Single Attribute Queries (b) Double Attribute Queries (c) Triple Attribute Queries

Figure 7: Ranking Performance on the FaceTracer dataset

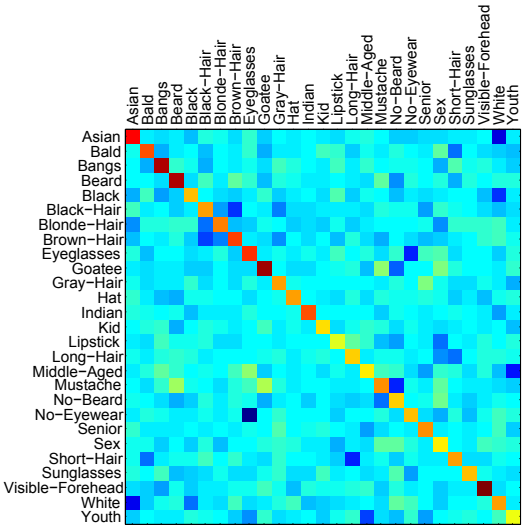


Figure 6: Classifier weights learnt on the LFW dataset, red and yellow indicate high values while blue and green denote low values. (best viewed in color).

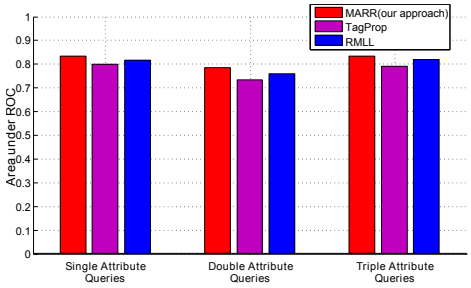
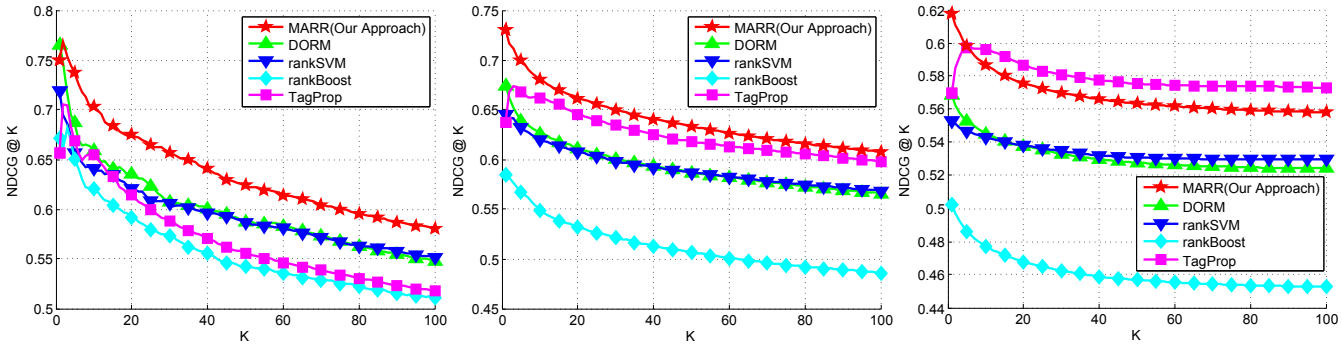


Figure 8: Retrieval Performance on the PASCAL dataset.

4.4. PASCAL

Finally, we experiment on the PASCAL VOC 2008 [4] trainval dataset, which consists of 12695 images comprising 20 object categories. The training set consists of 6340 images, while the validation set consisting of 6355 images is used for testing. Each of these images have been labeled with a set of 64 attributes [5]. We use the set of features used in [5], with each image being represented by a feature vector comprised of edge information and color, HOG and texton based visual words.

Figure 9 plots the ranking results on the PASCAL dataset. We can observe that MARR substantially outperforms all other ranking methods except TagProp, for all the three kinds of queries. Compared to TagProp, MARR is significantly better for single attribute queries(7.4% improvement in NDCG@10) and marginally better for double attribute queries(2.4% improvement in NDCG@10),



(a) Single Attribute Queries (b) Double Attribute Queries (c) Triple Attribute Queries

Figure 9: Ranking Performance on the PASCAL dataset.

while TagProp is marginally better than MARR for triple attribute queries(1.5% improvement in NDCG@10). The retrieval results are shown in Figure 8, here, MARR outperforms TagProp by about 5% and Reverse Multi-Label Learning(RMLL [19]) by about 2%.

5. Conclusion

We have presented an approach for ranking and retrieval of images based on multi-attribute queries. We utilize a structured prediction framework to integrate ranking and retrieval within the same formulation. Furthermore, our approach models the correlations between different attributes leading to improved ranking/retrieval performance. The effectiveness of our framework was demonstrated on three different datasets, where our method outperformed a number of state-of-the-art approaches for both ranking as well as retrieval. In future, we plan to explore image retrieval/ranking based on more complex queries such as scene descriptions, where a scene is described in terms of the objects present, along with their attributes and the relationships among them.

Acknowledgements: The authors thank Michele Merler and Gang Hua for providing facial feature extraction code and the attribute labels for the LFW/FaceTracer datasets. This research was partially supported by the ONR MURI grant N000141010934.

References

[1] T. Berg and D. Forsyth. Animals on the web. *CVPR*, 2006.

[2] C. J. C. Burges, R. Ragno, and Q. V. Le. Learning to rank with nonsmooth cost functions. *NIPS*, 2006.

[3] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. *ICCV*, 2009.

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results.

[5] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. *CVPR*, 2009.

[6] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories using google's image search. *ICCV*, 2005.

[7] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *JMLR*, 2003.

[8] D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. *IEEE PAMI*, 2008.

[9] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Discriminative metric learning in nearest neighbor models for image auto-annotation. *ICCV*, 2009.

[10] Y. Hu, M. Liu, and N. Yu. Multiple-instance ranking: learning to rank images for image retrieval. *CVPR*, 2008.

[11] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, 2007.

[12] T. Joachims. Optimizing search engines using clickthrough data. *KDD*, 2002.

[13] J. Krapac, M. Allan, J. Verbeek, and F. Jurie. Improving web image search results using query-relative classifiers. *CVPR*, 2010.

[14] H. M. Kuhn. The hungarian method for the assignement problem. *Naval Research Logistics Quarterly*, 1955.

[15] N. Kumar, P. Belhumeur, and S. Nayar. Facetracer: A search engine for large collections of images with faces. *ECCV*, 2008.

[16] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. *ICCV*, 2009.

[17] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. *CVPR*, 2009.

[18] Q. V. Le and A. J. Smola. Direct optimization of ranking measures. http://arxiv.org/abs/0704.3359, 2007.

[19] J. Petterson and T. S. Caetano. Reverse multi-label learning. *NIPS*, 2010.

[20] C. Teo, S. Vishwanathan, A. Smola, and Q. Le. Bundle methods for regularized risk minimization. *JMLR*, 2010.

[21] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. *ICML*, 2004.

[22] D. A. Vaquero, R. S. Feris, D. Tran, L. Brown, A. Hampapur, and M. Turk. Attribute-based people search in surveillance environments. *WACV*, 2009.

[23] G. Wang and D. Forsyth. Object image retrieval by exploiting online knowledge resources. *CVPR*, 2008.

[24] Y. Wang and G. Mori. A discriminative latent model of object classes and attributes. *ECCV*, 2010.