Attribute Feedback

Hanwang Zhang[†], Zheng-Jun Zha[†], Shuicheng Yan[§], Jingwen Bian[†] and Tat-Seng Chua[†]
[†] School of Computing, National University of Singapore
[§] Dept. of Electrical and Computer Engineering, National University of Singapore
[†]{hanwang, zhazj, bian_jingwen, chuats}@comp.nus.edu.sg
[§]eleyans@nus.edu.sg

ABSTRACT

This work presents a new interactive Content Based Image Retrieval (CBIR) scheme, termed Attribute Feedback (AF). Unlike traditional relevance feedback purely founded on low-level visual features, the Attribute Feedback system shapes users' information needs more precisely and quickly by collecting feedbacks on intermediate level semantic attributes. At each interactive iteration, AF first determines the most informative binary attributes for feedbacks, preferring the attributes that frequently (rarely) appear in current search results but are unlikely (likely) to be users' interest. The binary attribute feedbacks are then augmented by a new type of attributes, "affinity attributes", each of which is off-line learnt to describe the distance between user's envisioned image(s) and a retrieved image with respect to the corresponding affinity attribute. Based on the feedbacks on binary and affinity attributes, the images in corpus are further re-ranked towards better fitting the users' information needs. Extensive experiments on two real-world image datasets well demonstrate the superiority of the proposed scheme over other state-of-the-art relevance feedback based CBIR solutions.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval Model

General Terms

Algorithms, Experimentation, Performance

Keywords

Image Search, Attribute Feedback, Relevance Feedback

1. INTRODUCTION

With the growing volume of on-line images as well as the increasing requirements in many multimedia applications, Content Based Image Retrieval (CBIR) has attracted significant attention in both academia and industry [19, 26, 33]. Most CBIR systems allow users to specify their information needs by means of uploading an example image as query, called Query by Example (QBE). CBIR systems retrieve images based on their visual similarities to the query example. A major challenge in CBIR is the well-known "*semantic gap*" [9] between the low-level visual features and high-level semantic meanings that interpret users' search intent.

To address this problem, relevance feedback has been introduced into CBIR [35]. Relevance feedback involves users in the search loop. Typically, users are asked to label the top images returned by the search model as "relevant" or "irrelevant". The feedbacks are then used to refine the search model. Through iterative feedback and model refinement, relevance feedback attempts to capture users' information needs and improve search results gradually. Although relevance feedback has shown encouraging potential in CBIR, its performance is usually unsatisfactory due to the following problems. First, relevance feedback relies on the search system to infer users' search intent from their "relevant" and/or "irrelevant" feedbacks, essentially based on the low-level visual features of the relevant or irrelevant images. It suffers from the "semantic gap" between users' intent and low-level visual cues and thus it is usually ineffective in narrowing down the search to target [24]. Second, the retrieval results are usually unsatisfactory, where the top results may contain rare or even no relevant samples. With no/rare relevant samples, most relevance feedback approaches are not able to perform well or even no longer applicable [32, 21].

In this paper, we move beyond the conventional relevance feedback (RF), and propose a novel interactive search scheme, named Attribute Feedback (AF), for content based image retrieval. Attribute refers to the **visual properties** (e.g., "round" as shape, "metallic" as texture), **components** (e.g., "has wheel", "has leg") and **functionalities** (e.g., "can fly", "man-made") of objects. As illustrated in Figure 1, AF allows users to deliver search intent by providing feedbacks on binary attributes to state which attributes are in their search intent and which are not. For example, when users are searching for "dog" images, they may give positive feedbacks on attributes "snout" and "leg" as well as negative feedbacks on "cloth" and "wheel". Here, attributes act as the bridge connecting user intent and visual features. The binary feedbacks compose a clear semantic description of users' search intent, such as "*has snout and leg, has no wheel and cloth*". However, the binary feedbacks on some attributes, which are shared by many categories, are not discriminative enough. For example, the feedback "*has snout*" is not discriminative enough to distinguish users' target (e.g., "dog") from other animals, since "snout" is shared by many animal categories (e.g., "dog" and "crocodile"). Hence, we propose a new type of attributes, termed Affinity Attribute. *Affinity attributes refer to attributes that are shared by many categories and have large variance in appearances*. For each affinity attribute of interest, AF allows users to give affinity judgments on the images containing this attribute to indicate which images are similar

MM'12, October 29–November 2, 2012, Nara, Japan.

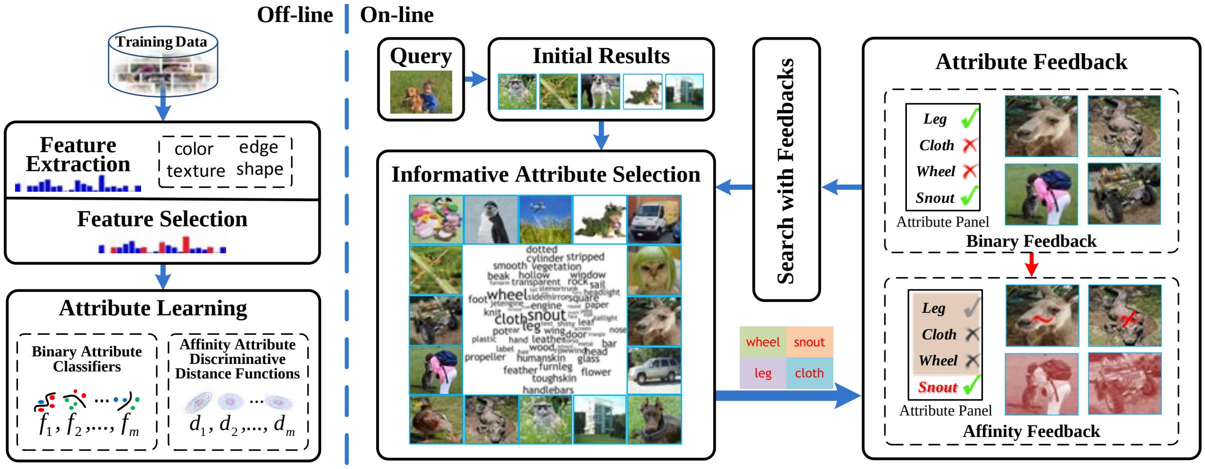


Figure 1: The flowchart of the proposed Attribute Feedback (AF) framework. The user's intent is "*find me dogs*" though the query image at hand is "*a boy and a dog*". ✓ and $\times$ denote "yes/no" feedbacks on the binary attributes. Users may further give affinity feedbacks for a binary attribute (e.g., "snout"). By that time, images without such binary attribute will be shadowed. $\sim$ and $\nsim$ denote the "similar/dissimiar" feedbacks on affinity attributes with the context of the referenced attributes (e.g., "snout"). By collecting user's binary and affinity attribute feedbacks iteratively, AF shapes user's intent precisely and quickly, leading to search results that well fit user's intent.

or dissimilar to their envisioned target images with respect to this attribute. As shown in Figure 1, for the attribute "snout", user can make affinity feedbacks to state that the "snout" in their envisioned images is similar to that in the retrieved "alpaca" image but dissimilar to that in the "crocodile" image. Such affinity feedbacks further reveal users' search intent. Different from traditional relevance feedback that simply states which images are relevant or irrelevant, AF helps users to specify their search intent more precisely through the binary and affinity attribute judgements. Therefore, AF permits the search system to quickly narrow down the search to users' information needs with less interaction efforts. Moreover, even when the top search results contain no relevant sample, some of the results might be partially similar to users' envisioned images on certain attribute(s). For example, the retrieved "alpaca" image is irrelevant to users' target, i.e., "dog", but is similar to "alpaca" on the attribute "snout". By accumulating users' feedbacks on such attribute(s), AF can push the search closer to users' target gradually. Hence, AF is expected to be able to overcome the no/rare relevant sample problem.

The flowchart of the proposed Attribute Feedback framework is illustrated in Figure 1. In the off-line part, we learn a set of binary classifiers, each of which predicts the presence of an attribute in an image. For each affinity attribute, we learn a discriminative distance function, which computes the distance between two images with respect to the corresponding affinity attribute. At each on-line feedback iteration, a set of informative attributes are selected and presented for feedbacks. In particular, the informative attributes refers to the attributes on which users' feedbacks can drastically enhance the subsequent search results. An attribute is considered as informative if it frequently (rarely) appears in current search results but is unlikely (likely) to be the users' interests. We propose a statistical attribute selection approach to select the most informative attributes. The approach simultaneously exploits both the search results at the current and previous iterations (see Section 4.1). In particular, we maintain a set of probabilistic models, each of which infers the candidacy of an attribute being selected. At the beginning of every feedback iteration, the posterior probabilities of these models are up-to-date based on the current search results and previous informative posteriors. The attributes with high posterior probabilities are then selected as informative attributes. After obtaining users' binary and affinity attribute feedbacks, a search model is then executed to update the search results.

To the best of our knowledge, this work presents the first attempt towards exploring attribute feedbacks for content based image retrieval. The main contributions of this paper are summarized as follows:

- We propose a novel interactive search scheme named Attribute Feedback for content based image retrieval. AF enables the search system to quickly narrow down the search to users' target based on their binary and affinity feedbacks. Moreover, AF performs well in case of the no/rare relevant sample problem that often exists in real-world CBIR.

- We develop an informative attribute selection approach, which simultaneously takes into account the current and previous search results.

- We define a new type of attribute, named affinity attribute. A discriminative distance function is learnt for each attribute to make effective use of users' affinity feedbacks.

The rest of this paper is organized as follows. Section 2 reviews the related works. Section 3 describes the attribute learning technologies, including classifier learning for binary attributes and discriminative distance function learning for affinity attributes. Section 3 elaborates the proposed AF framework, including informative attribute selection and image search with attribute feedbacks. Experimental results and analysis are given in Section 4, followed by the conclusions and future works in Section 5.

2. RELATED WORK

2.1 Attributes

Attributes are human-nameable *intermediate* semantic descriptors, refer to the visual properties (e.g., "round" as shape, "metallic" as texture), components (e.g., "has wheel", "has leg") and functionalities (e.g., "can fly", "man-made") of objects [6, 7, 13, 12]. As opposed to low-level visual features, e.g., the bag of visual words representation, an attribute has a semantic meaning and it is relatively easier than full objects (e.g., "car", "dog") to recognize for a machine [6, 7]. A popular attribute learning method is to train a binary classifier (e.g., SVM) for each attribute by training samples with and without such attribute. Then, the presence (confidence) of the attribute in an image can be predicted by the binary (probabilistic) output of the classifier. Attributes learned by such method are well-known as binary attributes [6, 13, 12]. Binary attributes form a general semantic base for various object categories [12, 27] and enable knowledge transfer across object categories [5], or even from known-categories to unknown-categories [13]. However, when objects share almost the same attribute vocabulary, using merely binary attributes are not sufficient to discriminate and describe the objects precisely.

There are several works on extending the discriminative and descriptive ability of attributes [6, 12, 16]. As a pioneering work, Farhadi et.al [6] exhaustively trained thousands of classifiers by random splitting the training data and choosing some of them which were good at distinguishing objects (e.g., attributes that "cat" and "dog" have but "sheep" and "horse" do not) as *discriminative* attributes. Kumar et al. [12] defined a new set of binary attributes called *similes* for face verifications. Similes are exclusive classifiers specialized for one category, e.g., "the Angelina Jolie's mouth". However, such category-dependent attributes are contrary to the spirit of attributes, i.e., knowledge that is generalizable and transferrable. Recently, Parikh and Grauman [16] proposed a new idea in describing and naming attributes called *relative* attributes, which describe the strength of the attribute presence in a relative way, e.g., "while A and B are both *shiny*, A is *shinier* than B". Instead of trained by binary classifiers, relative attributes are learned by ranking functions (i.e., the ranking SVM). The output of a ranking function indicates the relative presence of the corresponding attribute.

Attributes are also revealing the power in image search [12, 4, 18, 27]. The works in [12, 4, 27] focused on composing semantic feature (index) vectors by the confidence scores of binary attribute classifiers. Images are then retrieved by such features (index) instead of low-level features (index). In a different approach, Siddiquie et al. [18] proposed a structural SVM based approach for image search using multi-attribute text-based queries specified by users. Their approach explicitly models the correlations of attributes that are or not parts of the query. The output of the structural SVM is then considered as the ranking results of retrieved images.

2.2 Relevance Feedback

Relevance Feedback (RF) is the key technique to narrow down the *semantic gap* in CBIR by exploiting user interactions [35, 3]. Users are encouraged to label the retrieved images as being either "relevant" or "irrelevant". Users' feedbacks are then exploited by a relevance feedback algorithm to refine the search model. Through iterative feedbacks and refinement, relevance feedback attempts to capture users' search intent and improve the search results gradually. A wealth of methods has been proposed to learn a relevance feedback model based on users' feedbacks [17, 34, 8, 20, 10, 23]. At each feedback iteration, the model is updated using the labeled images as training samples. For example, Query Point Movement (QPM) [17] method gradually modifies the low-level visual features of the query image to make them more similar to "relevant" feedbacks and more dissimilar to "irrelevant" ones. Hence, the query feature is moving towards the search region of users' intent. Guo et al. [8] proposed to use SVM as the RF model. In each feedback loop, a SVM classifier is trained by the labeled samples and images in the database are further ranked according to the response from the classifiers. Zhou and Huang [34] developed Biased Discriminant Analysis (BDA) to learn a low-dimensional subspace from feedbacks in each loop. Then, all the images in the database are embedded into the subspace and ranked according to their Euclidean distance to the mean feature vector of the "relevant" images. Different from the above RF methods that present top retrieved images for user labeling, SVMactive [20] method actively selects images with high uncertainty for labeling in each feedback iterations. Please refer to [35, 3, 15] for more comprehensive reviews on RF. As aforementioned, the traditional relevance feedback suffers from the gap between users' search intent and low-level visual features and thus it is usually ineffective in narrowing down the search to users' target.

Recently, Kovashka et al. [11] proposed to collect user feedbacks like *"show me shoes more formal than these and shinier than those"* in terms of relative attributes [16]. A ranking SVM score function learned in relative attribute training can be used to whittle away images not meeting users' descriptions. Compared to our work, their approach aims to improve text-based image search with query-by-word by collecting text feedbacks while our work is targeting at CBIR with query-by-example. Moreover, binary and affinity attribute feedbacks proposed in this paper offer more general attribute descriptions of users' intent. For example, relative attributes are capable of describing adjectives like "shiny" ("shinier") but unable to describe nouns like "eye". However, we can always use binary and affinity attributes to describe precisely what we do and do not want (e.g., "A should have *ear* but should not have *feather*") and what are similar or not (e.g., "A is *shiny* like B" or "A's *eye* is dissimilar to B's").

3. ATTRIBUTE LEARNING

We refer attributes to the visual properties (e.g., "round" as shape, "metallic" as texture), components (e.g., "has wheel", "has leg") and functionalities (e.g., "can fly", "man-made") of objects [6, 13, 12]. In this work, we exploit the visual property and component attributes. The presence of the attributes can be predicted by the output of binary classifiers, e.g., a binary classifier for "wheel" can be used to predict whether an object has "wheel". Thus, we call such attributes the *binary attributes*. Binary attributes are good at in describing objects, especially unseen ones. For example, even we have no training samples of "cat" (unseen category), we can still describe "cat" as an animal that has "ear", "eye" and "furry" by using those binary attribute classifiers trained by seen categories. Although binary attributes offer users a way to describe their desired objects across categories, some attributes are not discriminative enough in distinguishing users' target with other objects. For example "ear" is not discriminative enough to distinguish different animals because it is shared by many animals.

To this end, we introduce a new kind of attributes called *affinity attributes*, which refer to the attributes that are shared by many categories and have large variance in appearances. We learn a distance function for each affinity attribute in order to quantitatively describe how close the affinity of two images for the referenced affinity attribute. As illustrated in Figure 2, given three images of, say, two "sheep" and one "monkey". The "sheep" images are more similar than the "monkey" and "sheep" images with respect to the affinity attribute "ear". The distance function will presumably judge the "ear" of two "sheep" closer while that of the "sheep" and the "monkey" farther. Therefore, affinity attributes enable further comparisons between images who share common binary attributes and thus endow more discriminative information than binary attributes. Besides, the introduction of affinity attributes remains the size of attribute vocabulary unchanged.

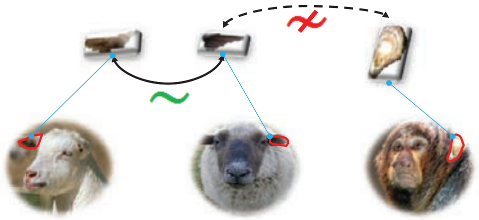


Figure 2: Motivation of Affinity Attributes. The two *sheep* images are similar on the affinity attribute "ear", while the *monkey* image is dissimilar to them on "ear".

Next, we will describe the details in learning binary and affinity attributes.

3.1 Binary Attribute Learning

Suppose we have an image collection $\mathcal{X} = \{x_1, x_2, ..., x_N\}$, where $x_i \in \mathbb{R}^n$ is a n-dimensional feature representation of image i. Moreover, we have defined a vocabulary of m binary attributes $\mathcal{A} = \{a_1, a_2, ..., a_m\}$, where $a_i \in \{0, 1\}$ denotes the presence of the i-th attribute. The binary classifier of attribute a_i predicts the presence of this attribute in an image x_j, with the probability $P(a_i = 1|x_j)$.

Given a training set of images which are labeled with attributes, one can easily train the binary classifiers such as SVM in [13]. However, due to the large visual variance of some attributes (e.g., the "wing" of a plane looks much differently from the "wing" of a bird), standard training methods cannot be directly applied because they may be trained by irrelevant feature dimensions. For example, the "wing" classifier may be unfortunately trained based on the feature dimensions related to "sky". Therefore, we need to select feature dimensions that are mostly informative and effective for the target attribute.

In this paper, we apply the feature selection method as described in [6]. The selected feature dimensions $\phi_a(x_i)$ of attribute a for image x_i are the union of feature dimensions that are most discriminative for sub-categories. For example, in order to train the "wing" classifier, we first collect "bird" with and without "wing" as training samples, and train a preliminary *linear* classifier (e.g., $\langle w_{bird}, x_i \rangle$), called "bird wing". Then, we may train another preliminary classifier called "plane wing" and so on. By doing so, we finally obtain a set of parameters (e.g., w_{bird}, w_{plane}) of such preliminary classifiers. Particularly, due to the success of ℓ_1-norm in signal representation [28] and image annotation [30, 31], we use the ℓ_1-norm regression model as the preliminary classifiers to learn sparse parameters. The features are then selected by pooling the union of indices of the sparse non-zeros entries in those parameters. Formally, for attribute a, we denote I_a as the union of the non-zero parameter indices and then the selected feature dimensions of attribute a for x_i is $\phi_a(x_i) = x_j(I_{a_i})$, where $x(I_a)$ returns the selected dimensions of x_i according to I_a. Once we have selected the feature dimensions, we can apply the standard training methods to learn an overall binary attribute classifier (e.g., SVM) of a, i.e., $P(a = 1|\phi_a(x_i))$. For the sake of simplicity, from now on and throughout this paper, we always denote $x_i \leftarrow \phi_a(x_i)$ when the context is clear.

3.2 Affinity Attribute Learning

For each affinity attribute a, we learn a distance function which computes the affinity between two images with respect to the referenced attribute. In particular, we define the distance function as a Mahalanobis distance metric as:

$$d_a(x_i, x_j) = \sqrt{(x_i - x_j)^T \mathbf{M}_a (x_i - x_j)}, \tag{1}$$

where the $\mathbf{M}_a$ is a semi-definite symmetric matrix, i.e., $\mathbf{M}_a \succeq \mathbf{0}$. Denote $x_i \sim_a x_j$ (or $x_i \nsim_a x_j$) as "image x_i is similar (or dissimilar) to image x_j with respect to attribute a". In order to characterize the affinity of two images with respect to attribute a, the distance metric $d_a(\cdot)$ should measure the distance of x_i and x_j closer than the distance of x_k and x_j if $x_i \sim_a x_j$ and $x_k \nsim_a x_j$, i.e.,

$$\begin{aligned} & d_a(x_j, x_k) - d_a(x_i, x_j) \geq c, \\ & s.t. \;\; x_i \sim_a x_j, \; x_k \nsim_a x_j, \end{aligned} \tag{2}$$

where $c > 0$ is a margin constant.

Given the training sample pairs $\tilde{\mathcal{S}}_a = \{(x_i, x_j)|x_i \sim_a x_j\}$ as the similar set and $\tilde{\mathcal{D}}_a = \{(x_j, x_k)|x_j \nsim_a x_k\}$ as the dissimilar set, we want to learn $\mathbf{M}_a$ in a discriminative way by favoring similar images to be closer while penalizing the distance of dissimilar ones that are not sufficiently farther than the distance of similar ones. Therefore, we have the following semi-definite programming (SDP) problem:

$$\begin{aligned} \min_{\mathbf{M}_a} & \sum_{i,j} d_a(x_i, x_j) + \lambda \sum_{i,j,k} \xi_{ijk}, \\ s.t. \;\; & d_a(x_j, x_k) - d_a(x_i, x_j) \geq 1 - \xi_{ijk}, \\ & \xi_{ijk} \geq 0, \; \mathbf{M} \succeq \mathbf{0}, \\ & (x_i, x_j) \in \tilde{\mathcal{S}}_a, \; (x_j, x_k) \in \tilde{\mathcal{D}}_a. \end{aligned} \tag{3}$$

where $\lambda > 0$ is a trade-off parameter and ξ_{ijk} is the slack variable. In order to avoid over-fitting and to shorten the training time, we first reduce the dimensions of the attribute feature vectors to d (e.g., $d = 200$) using PCA and then learn the distance matrix of size $d \times d$. Since it is expensive to label the attribute-level training pairs, in our implementation, $\tilde{\mathcal{S}}_a$ and $\tilde{\mathcal{D}}_a$ can be easily collected by randomly sampling image pairs with positive attribute labels from the same and distinct categories.

4. ATTRIBUTE FEEDBACK

In this section, our goal is to assist the user to find more target images via interactive feedbacks on attributes. With users in the loop, the system will learn how to describe the target images precisely using both binary and affinity attributes. The attribute descriptions will then serve as semantic cues for the system to find relevant results.

Without loss of generality, suppose we are at the t-th feedback iteration. The system displays the image set $\mathcal{R}_t$ (e.g., top 100 results) to the user and a binary attribute set $\mathcal{C}_t$ on the attribute panel (as illustrated in Figure 1). Then, through $\mathcal{C}_t$, the user will response to the system which attributes are positive (expected to appear) or negative (expected to disappear) in $\mathcal{R}_t$. Furthermore, when a positive binary attribute coincides with the name of an affinity attribute, the user may also give affinity feedbacks on images by telling the system whether an image in $\mathcal{R}_t$ is similar or dissimilar to her target image with respect to the referenced attribute.

Meanwhile, from the system's perspective, it maintains the sets $\mathcal{A}_t^+$ and $\mathcal{A}_t^-$ that record all the positive and negative binary attribute feedbacks accumulated from the $(t-1)$-th iteration, respectively. For the affinity attribute feedbacks, the system records $\mathcal{S}_t^l = \{x_i|x_i \sim_{a_l} x^*\}$ that accumulates the images similar to the user's envisioned image x^* with respect to the affinity attribute a_l. Correspondingly, the set $\mathcal{D}_t^l = \{x_i|x_i \nsim_{a_l} x^*\}$ accumulates the dissimilar images. Let the sets $\{\mathcal{S}_t^l\}_{l=1}^m$ and $\{\mathcal{D}_t^l\}_{l=1}^m$ be the feedbacks of similar and dissimilar images collected through all the attributes. We denote $\mathcal{T}_t = \{\mathcal{A}_t^+, \mathcal{A}_t^-, \{\mathcal{S}_t^l\}, \{\mathcal{D}_t^l\}\}$ as the attribute descriptions of the target image in terms of attribute feedbacks collected thus far. The system will then exploit such descriptions $\mathcal{T}_t$ to refine the search results.

The attribute feedback system has two key components as follows:

- *Informative Attribute Selection*. This component focuses on selecting attributes from the attribute vocabulary to form $\mathcal{C}_t$. The selected attributes should be informative for users to give feedbacks.

- *Search with Feedbacks*. Given the gathered user's attribute feedbacks $\mathcal{T}_t$, this component is targeted at retrieving images to form $\mathcal{R}_{t+1}$.

Next, we introduce the use of a Bayesian framework to select the most informative attributes for the first component. For the second one, we propose a discriminative model as the score function to search images with feedbacks.

4.1 Informative Attribute Selection

A straightforward approach is to let $\mathcal{C}_t = \mathcal{A}$, i.e., forcing the user to label all the attributes. However, it is unreasonable for users to complete such tedious task. Another method is to use only the most "apparent" attributes in the query image. However, using only the attributes extracted from one query cannot comprehensively describe the user's information needs. For example, although the attribute "wheel" is missing from a query image "car", the "wheel" is still highly informative for a user to find a "car".

We argue that an attribute is informative if it frequently (rarely) appears in search results $\mathcal{R}_t$ but is unlikely (likely) to be the users' interests. Such cases suggest there is a big difference between retrieved results at present and the user's search intent. Therefore, if an attribute frequently (rarely) appears in the results, a negative (positive) feedback on this attribute will drastically improve the subsequent results. We should also note that some trivially missing attributes should not be counted as informative. For example, it is absurd to let the user give feedback on the attribute "engine" when she is looking for "monkey" since "engine" hardly appears in "monkey" albeit it rarely appears in retrieved results.

We propose a Bayesian framework based on the currently and previously retrieved results gathered by $\mathcal{Y}_t = \{\mathcal{R}_s\}_{s=1}^t$. Suppose a binary random variable c_i is associated to each attribute $a_i \in \mathcal{A}$: $c_i = 1$ if $a_i \in \mathcal{C}_t$ and $c_i = 0$ otherwise. Our framework maintains m parallel Bayesian systems $p_t(c_i) = P(c_i = 1|\mathcal{Y}_t)$ which is the posterior probability of the informativeness of a_i based on $\mathcal{Y}_t$. Ordered by such probabilities in $p_t(c_i)$, the system will select up to K attributes to form $\mathcal{C}_t$.

Applying the Bayes' rule, we obtain the updating rule for model $p_t(c_i)$ as:

$$p_t(c_i) = \frac{P(\mathcal{R}_t|c_i = 1)p_{t-1}(c_i)}{P(\mathcal{R}_t|c_i=1)p_{t-1}(c_i) + P(\mathcal{R}_t|c_i=0)(1-p_{t-1}(c_i))}. \quad (4)$$

For initialization, given the initial query image x_q, we set $p_0(c_i)$ to the output of classifier $P(a_i|x_q)$ and $\mathcal{R}_1$ be the initial retrieved results by any search model (e.g., low-level feature matching).

The above derivation makes use of the basic statistical assumption:

$$P(\mathcal{R}_t|c_i, \mathcal{Y}_{t-1}) = P(\mathcal{R}_t|c_i), \quad (5)$$

which means that given c_i, the current search results $\mathcal{R}_t$ is independent of the history $\mathcal{Y}_{t-1}$. In other word, we say that c_i is a sufficient statistic for $\mathcal{Y}_{t-1}$ since $\mathcal{Y}_{t-1}$ only affects the distribution of $\mathcal{R}_t$ through c_i. This assumption stems from the intuition that the inference of c_i is evident from $\mathcal{Y}_t$ and thus it is evident from $\mathcal{Y}_{t-1}$.

Now we show how to compute $P(\mathcal{R}_t|c_i)$. We assume the images in $\mathcal{R}_t$ contain users' intent and use conditional entropy of a_i given the images in $\mathcal{R}_t$ to delineate frequency (or rarity) of a_i. Formally, we define the probabilities $P(\mathcal{R}_t|c_i = 1)$ and $P(\mathcal{R}_t|c_i = 0)$ of the form:

$$P(\mathcal{R}_t|c_i = 1) = \frac{\psi^-(H(a_i|\mathcal{R}_t))P(a_i = 1|\mathcal{R}_t)}{\sum_i \psi^-(H(a_i|\mathcal{R}_t))P(a_i = 1|\mathcal{R}_t)}, \quad (6a)$$

$$P(\mathcal{R}_t|c_i = 0) = \frac{\psi^+(H(a_i|\mathcal{R}_t))P(a_i = 0|\mathcal{R}_t)}{\sum_i \psi^+(H(a_i|\mathcal{R}_t))P(a_i = 0|\mathcal{R}_t)}, \quad (6b)$$

where $\psi^-(\cdot)$ and $\psi^+(\cdot)$ are monotonically decreasing and increasing functions, which will be defined later (in Eq. (8)). Here, $P(a_i|\mathcal{R}_t)$ is used as a regularization term to prevent trivial results that attributes are unlikely to appear in the $\mathcal{R}_t$ albeit the entropy of them is also very large. The probability $P(a_i = 1|\mathcal{R}_t) = 1 - P(a_i = 0|\mathcal{R}_t)$ is the average of the probabilistic outputs of the classifier of a_i, i.e., $P(a_i = 1|\mathcal{R}_t) = \sum_{x_j \in \mathcal{R}_t} P(a_i = 1|x_j)/|\mathcal{R}_t|$.

We compute the conditional entropy $H(a_i|\mathcal{R}_t)$ as:

$$\begin{aligned} H(a_i|\mathcal{R}_t) &= -\sum_{x_j \in \mathcal{R}_t} P(x_j)H(a_i|x_j) \\ &= -\frac{1}{|\mathcal{R}_t|}\sum_{x_j \in \mathcal{R}_t}\sum_{b\in\{0,1\}} P(a_i = b|x_j)\log P(a_i = b|x_j), \end{aligned} \quad (7)$$

where we assume the prior $P(x_j) = 1/|\mathcal{R}_t|$.

Note the maximum and minimum of $H(a_i|\mathcal{R}_t)$ is 1 and 0, respectively. Therefore, typical definitions of $\psi^-(\cdot)$ and $\psi^+(\cdot)$ can be:

$$\psi^-\left(H(a_i|\mathcal{R}_t)\right) = 1 - H(a_i|\mathcal{R}_t),\ \psi^+\left(H(a_i|\mathcal{R}_t)\right) = H\left(a_i|\mathcal{R}_t\right). \quad (8)$$

The intuition of these definitions is consistent with our argument on attribute informativeness. Frequently (rarely) appeared attributes (small $H(a_i|\mathcal{R}_t)$) and low users' interests (small normalization value in Eq. (6a)) result in large $P(\mathcal{R}_t|c_i = 1)$ and small $P(\mathcal{R}_t|c_i = 0)$, i.e., large informativeness $p(c_i)$.

The time complexity of updating all the Bayesian systems is $\mathcal{O}(m)$, where m is the size of the attribute vocabulary. Hence, the informative attribute selection can be applied in real-time systems. Next, we will introduce how to retrieve images based on the user's attribute feedbacks.

4.2 Search with Feedbacks

The search model aims to score an image x in the image collection by its relevance to the user's intent in terms of the attribute feedbacks, $\mathcal{T}_t = \{\mathcal{A}_t^+, \mathcal{A}_t^-, \{\mathcal{S}_t^i\}, \{\mathcal{D}_t^i\}\}$. The score function is a discriminative model of $\mathcal{T}_t$ given x:

$$P(\mathcal{T}_t|x) \propto P(\mathcal{B}_t|x)\Psi^+(\{\mathcal{S}_t^i\}, x)\Psi^-(\{\mathcal{D}_t^i\}, x), \quad (9)$$

where $\mathcal{B}_t = \{a_i|a_i \in \mathcal{A}_t^+ \cup \mathcal{A}_t^-,\ s.t.\ \mathcal{S}_t^i = \phi,\ \mathcal{D}_t^i = \phi\}$ is the set of attributes without any affinity feedbacks (i.e., pure binary feedbacks).

The first term $P(\mathcal{B}_t|x)$ in Eq. (9) measures the relevance of image x to the user's target image with respect to binary feedbacks. By assuming that the attributes in $\mathcal{B}_t$ to be conditionally independent of x, we have:

$$P(\mathcal{B}_t|x) = \prod_{a_i \in \mathcal{B}_t \cap \mathcal{A}_t^+} P(a_i = 1|x) \prod_{a_i \in \mathcal{B}_t \cap \mathcal{A}_t^-} P(a_i = 0|x), \tag{10}$$

The second and third terms in Eq. (9) are positive and negative potential functions that jointly model the relevance of image x with respect to the affinity attribute feedbacks. If $x_j \in \mathcal{S}_t^i$ (or $x_j \in \mathcal{D}_t^i$), the presence of a_i in x, i.e., $P(a_i = 1|x)$, should be enhanced if the affinity of x_j and x is close (or far) with respect to a_i. Thus, we define the two potential functions as:

$$\begin{cases} \Psi^+(\{\mathcal{S}_t^i\}, x) = \prod_i P(a_i = 1|x)^{1-\psi(\mathcal{S}_t^i, x)}, \\ \Psi^-(\{\mathcal{D}_t^i\}, x) = \prod_i P(a_i = 1|x)^{\psi(\mathcal{D}_t^i, x)}, \end{cases} \tag{11}$$

where $\psi(\mathcal{S}_t^i, x)$ is defined as:

$$\psi(\mathcal{S}_t^i, x) = exp\left(- \frac{\sum\limits_{x_j \in \mathcal{S}_t^i} d_{a_i}(x, x_j)}{2\sigma^2} \right). \tag{12}$$

Here, σ is the Gaussian normalization parameter, $d_{a_i}(x, x_j)$ is the distance function of affinity attribute a_i in Eq. (1) and $\psi(\mathcal{D}_t^i, x)$ can be defined similarly. We should force $d_{a_i}(x, x_j) = 0$ if a_i is absent in x. The absence of a_i can be determined if $P(a_i = 1|x)$ is smaller than $\sum_j P(a_i = 1|x_j)/(|\mathcal{S}_t^i| + |\mathcal{D}_t^i|)$, where $x_j \in \mathcal{S}_t^i \cup \mathcal{D}_t^i$. Note that this threshold is reasonable because users will always give affinity feedbacks on images (gathered by $\mathcal{S}_t^i \cup \mathcal{D}_t^i$) where the reference attribute a_i is present.

The search complexity with attribute feedbacks is $\mathcal{O}(Nm)$, which is linear with the size N of the image corpus. For large-scale databases, the time complexity of the model $P(\mathcal{B}_t|x)$, $\psi(\mathcal{S}_t^i, x)$ and $\psi(\mathcal{D}_t^i, x)$ can be reduced using any advanced CBIR search or indexing technologies [29, 25].

5. EXPERIMENTS

In this section, we systematically evaluate the proposed Attribute Feedback framework. We first investigate the attribute learning accuracy and then evaluate the performance of CBIR with attribute feedbacks.

5.1 Data and Methodology

5.1.1 Dataset Description

We conducted experiments over two real-world image datasets: the Pascal-Yahoo! image corpus [6] and the Web Image collection downloaded from the Microsoft Bing image search engine.

Pascal-Yahoo!. This dataset was used in [6]. It contains 15,339 images collected from Pascal VOC 2008 (12,695 images) and Yahoo! image search engine (2,644 images). The images are from 32 object categories (20 in Pascal and 12 in Yahoo!) and all of them were annotated with 64 pre-defined attributes, such as shapes "round", materials "wool" and parts "wheel", among which 39 attributes are considered as affinity attributes, such as "eye", "leg", etc. All the images were split into two subsets [6]. In particular, 6,340 images from Pascal were used to train the classifiers for all the 64 binary attribute and learn the discriminative distance functions for the 39 affinity attributes. The rest of 6,355 images in Pascal together with the 2,644 images in Yahoo! were used for image retrieval. We randomly selected 10 images from each category as experimental queries. This gives rise to 320 queries in total.

Web Image[1]. We collected another web image corpus from the Microsoft Bing image search engine with 120 popular text queries in *Animal* and *Object* domains. For the ambiguous queries, we were interested in only one of its multiple facets. Take the query "apple" as example, we only downloaded the images of "apple, the fruit" by using "apple fruit" as the query. In total, 102,657 images were collected. We invited 21 annotators to manually remove the irrelevant images. 76,303 relevant images were eventually kept as our experimental data. There are around 300-800 images within one query category. We defined 67 semantic attributes by referring to the 64 attributes in the Pascal-Yahoo! dataset. In particular, we followed the naming protocol of attributes in [6], except that we remove "2D_Boxy" and "3D_Boxy" since they were very ambiguous to our annotators. Moreover, we added 5 more attributes, such as "Transparent" and "ToughSkin". Among the 67 binary attributes, 39 attributes were considered as affinity attributes. We randomly selected 50 categories with 25,203 images as the development set for attribute learning, and the remaining 70 categories with 51,100 images as the test set for retrieval. Due to the high cost of manual labeling, we only manually annotated the 67 attributes on the development set. The annotation followed the same criteria as in [6]. 80% of images in the development set were used as training samples in attribute learning, and the remaining 20% of images were used for testing. For retrieval, we randomly selected 10 images from each of the 70 categories as the experimental queries, and obtained 700 queries in total.

5.1.2 Visual Features

We used four types of features, including color and texture, which are good for material attributes; edge, which is useful for shape attributes; and scale-invariant feature transform (SIFT) descriptor, which is useful for part attributes. We used a bag-of-words style representation for each of these four features.

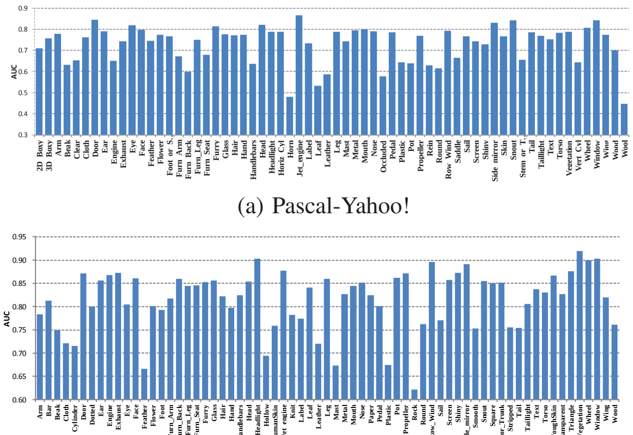
Color descriptors were densely extracted from each pixel as the 3-channel LAB values. The color descriptors of each image were then quantized into a 128-bin histogram. Texture descriptors were computed for each pixel as the 48-dimensional responses of texton filter banks [14]. The texture descriptors of each image were then quantized into a 256-bin histogram. Edges were found using the standard canny edge detector and their orientations were quantized into 8 unsigned bins. This gives rise to a 8-bin edge histogram for each image. SIFT descriptors [22] were densely extracted from the 8×8 neighboring block of each pixel with 4-pixel step size. The descriptors were quantized into a 1,000-dimensional bag-of-words feature. Afterward, for each image, we concatenated these four type of features into a 1,392-dimensional vector, which was used in image retrieval. Since attributes usually appear in one or more regions in an image, we further split each image into 2×3 grids and extracted the above four kinds of features from each grid respectively. Finally, we obtained a 9,744-dimensional feature for each image, consisting of a 1,392×6-dimensional feature from the grids and a 1,392-dimensional feature from the whole image. The 9,744-dimensional feature was used in attribute learning.

5.1.3 Experimental Setting

In our implementation, we employed the publicly available toolbox L1_LOGREG[2] to learn the ℓ_1-norm regression functions for feature selection and adopted LIBSVM [1] for training the binary attribute classifiers. The regularization parameter of ℓ_1-norm regression was set to 0.01 empirically and the parameters of SVM classifiers were determined through five-fold cross validation. For

[1] http://www.comp.nus.edu.sg/~hanwang/data.html

[2] http://stanford.edu/~boyd/l1_logreg/index.html



(a) Pascal-Yahoo!

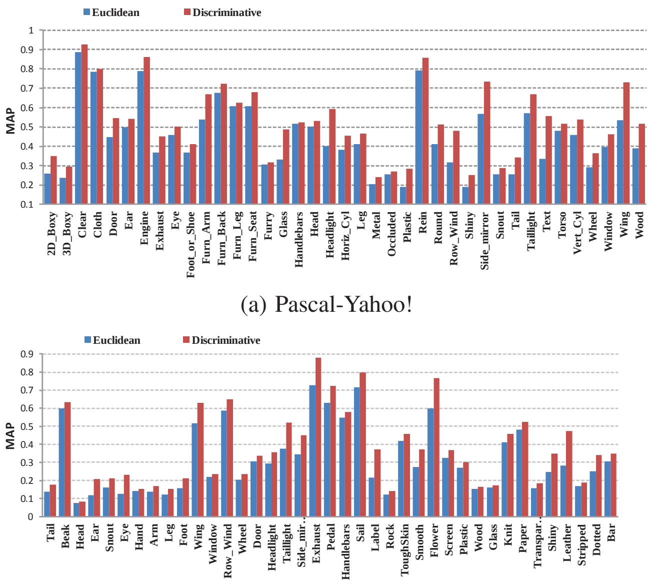
(b) Web Image

Figure 3: Classification performance of (a) the 64 attributes on the Pascal-Yahoo! dataset and (b) the 67 binary attributes on the Web Image dataset.

learning the discriminative distance functions of affinity attributes, we used the SDPLIB[3], where the tradeoff parameter λ was set by five-fold cross validation.

To evaluate the effectiveness of the proposed Attribute Feedback (**AF**) approach, we compared it against four popular relevance feedback methods and one active learning approach: (a) **QPM** [17]. Query Point Movement method gradually refines the query point based on user's "relevance" and "irrelevance" feedbacks in each feedback iteration. The images in database are ranked according to their Euclidean distances to the refined query. (b) **SVM** [8]. This approach learns a SVM classifier from user's "relevance" and "irrelevance" feedbacks and ranks images according to SVM's outputs on them; (c) **BDA** [34]. Biased Discriminant Analysis algorithm learns a low-dimensional subspace from user's feedbacks. All the images in database are embedded into this subspace and ranked according to their Euclidean distance to the mean feature vector of the "relevant" images; and (d) **SVMactive** [20]. Different from the above relevance feedbacks methods that present top retrieved images for user labeling, SVMactive method actively selects images with high uncertainty for labeling in each feedback iteration. The labeled images are then utilized to learn a SVM classifier, which is in turn used to rank the images in the database. The parameter settings of these baseline methods were identical to those in the corresponding papers.

We conducted the retrieval process for each query as follows. For each submitted query, the search system generated initial search results based on low-level visual features. The number of feedbacks per round is widely set as 10-20 in leteratures [35]. We set it to 20 as in [20, 34]. For QPM, SVM and BDA baseline methods, the top 20 returned images were presented for users to label at each feedback iteration. For the SVMactive approach, 20 images were selected by this approach and presented to users for labeling. The label information was then used to update the search results. In the proposed attribute feedback (AF) method, ten informative attributes, including binary and affinity attributes, were presented for user feedback at each iteration. For each affinity attribute with positive feedback, user further made comparative judgements on some images from the top 20 ones to state which images are similar or dissimilar to their intent with respect to the referenced attribute. For the sake of fair comparison, we constrained the number of attribute feedbacks in each iteration to be the same in the baseline methods. That is, the total number of feedbacks on binary attributes and comparative judgments on affinity attribute was limited to 20.



(a) Pascal-Yahoo!

(b) Web Image

Figure 4: Performance of the discriminative distance functions for the 39 affinity attributes on (a) the Pascal-Yahoo! dataset and (b) the Web Image dataset.

5.1.4 *Performance Metric*

We adopted the widely used metric AUC (area under ROC curve) value for evaluating the accuracies of attribute classifiers. AUC describes the probability that a randomly chosen positive sample will be ranked higher than a randomly chosen negative sample. We used the mean Average Precision (MAP) for evaluating the distance functions of affinity attributes. In particular, we collected all the testing and training images containing a certain affinity attribute. For each testing image, all the training images were ranked according to their distances to the testing image. We used the Average Precision (AP) to evaluate the ranking list for each testing image and averaged the APs over all the testing images to obtain the MAP. The distances were computed using the distance function learned for this attribute. We used precision in the top K results as the basic evaluation metric for all the feedback methods. When the top K images are considered and there are R relevant images, the precision within the top K images is defined to be R/K.

5.2 Experimental Results

5.2.1 *Evaluations on Attribute Learning*

Figure 3 shows the AUC values of the 64 and 67 binary attribute classifiers on the Pascal-Yahoo! and the Web Image datasets, respectively. The detection accuracy is comparable to the state-of-the-art methods. The average AUC on the Pascal-Yahoo! dataset is 0.76 comparable to 0.73 in [6]. On the Web Image dataset, the average AUC is 0.82 and can be also considered reliable. Moreover, AF is flexible to use any binary attribute learning method.

Figure 4 details the MAP of all the 39 affinity attributes com-

[3] http://euler.nmt.edu/~brian/sdplib/sdplib.html

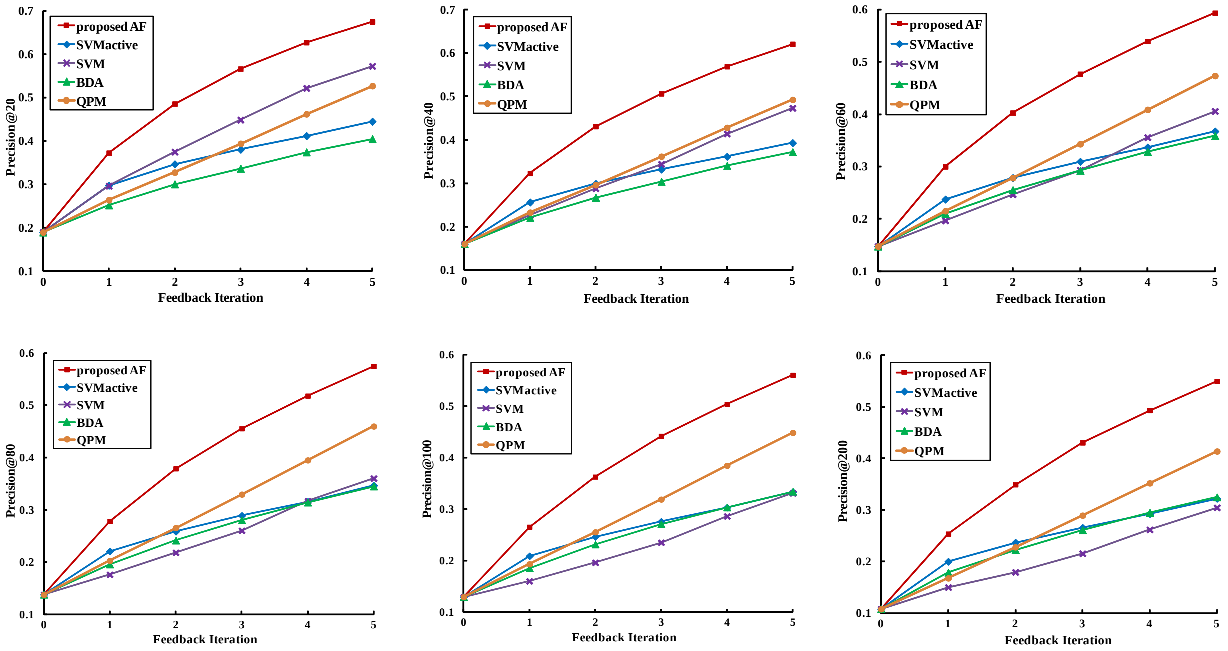


Figure 5: Precision@20, 40, 60, 80, 100, 200 over 320 queries with AF and the four baseline methods at five feedback iterations on the Pascal-Yahoo! dataset.

pared using the original Euclidean distance and the learned discriminative Mahalanobis distance. As we can see from the results, the discriminative ability of the affinity attributes is significantly improved across query categories. More specifically, affinity attributes improve the discriminative ability of attributes which are even ambiguous to human beings. For example, on the Web Image dataset, the "Taillight" can be correctly classified into different vehicles with over 0.5 MAP while only 0.37 is achieved when directly using Euclidean distance. Unfortunately, the "Head" attribute on the Web Image is not well improved. This may due to the large within-attribute variance of "Head".

5.2.2 *Evaluations on Attribute Feedback*

Figure 5 illustrates the retrieval performance on the Pascal-Yahoo! dataset by the proposed AF approach and the four state-of-the art relevance feedback methods, i.e., SVMactive, SVM, BDA and QPM. In particular, the average precision over 320 experimental queries at top 20, 40, 60, 80, 100, and 200 search results in five feedback iterations are reported here. Figure 6 shows the retrieval performance over 700 queries on the Web Image dataset. From the results, we can derive the following key observations.

- The proposed AF approach outperforms the other four methods at every iteration and on the entire scope (i.e., top 20, 40, 60, 80, 100, 200 results) on both datasets. For example, consider the precision@20 on the Pascal-Yahoo! dataset. At the first feedback iteration, AF achieves 25.2%, 25.6%, 47.8% and 41.1% relative improvements over SVMactive, SVM, BDA and QPM, respectively. The relative improvements at the last iteration are 51.7%, 17.9%, 66.8%, 28.0%, respectively. On the Web Image dataset, AF achieves 39.6%, 22.9%, 33.2%, 45.4% relative improvements at the first iteration and 181.4%, 67.0%, 154.0%, 99.0% relative improvements at the last iteration.

- On both datasets, AF significantly reduces the interaction efforts while it achieves comparable performance to the other four methods. For example, consider the precision at top 20 results on the Pascal-Yahoo! dataset, AF obtains a comparable performance at the 2nd, 3rd, 1st, 3rd iteration compared to SVMactive, SVM, BDA, QPM at the last round, respectively. That is to say, AF can reduce labeling efforts by about 60%, 40%, 80%, 40% as compared to the four methods. For the top 20 results on the Web Image dataset, AF achieves a better performance at the 1st, 3rd, 1st, 2nd iteration as compared to the four methods at the last iteration.

The above results demonstrate the superiority of AF over the traditional relevance feedback methods. The reasons behind this are three folds. First, feedbacks on binary attributes empower users to specify search intent more precisely than traditional "relevant" and "irrelevant" feedbacks on images. Second, the informative attribute selection aids the users to feedback on attributes of their interests efficiently. Third, feedbacks on affinity attributes offer the users to feedback on the more fine-grained and discriminative semantics. Therefore, AF can shape users' intent more precisely and quickly.

Figure 8 presents the detailed results over all the query categories on the Web Image dataset at the first iteration. Due to page limitation, we here only report the precision@20 performance at the first iteration on the Web Image dataset. We can see that AF outperforms the other four methods on most of the query categories. While the traditional RF methods purely based on low-level visual features may suffer from the large within-category variance especially on the Web Image dataset, the proposed AF exploits intermediate-level semantic attributes and thus can well model users' intent by collecting feedbacks on attributes.

At last, we investigate the performance of AF in the case of no relevant samples in the initial search results. Since the Pascal-Yahoo! dataset is of small within-class variance, there is no such case happened in the experiments on the Pascal-Yahoo! dataset. However, on the Web Image dataset, this situation is fairly common. In particular, among the 70 query categories, there are 127 queries from 23 categories suffered from this problem. Figure 7

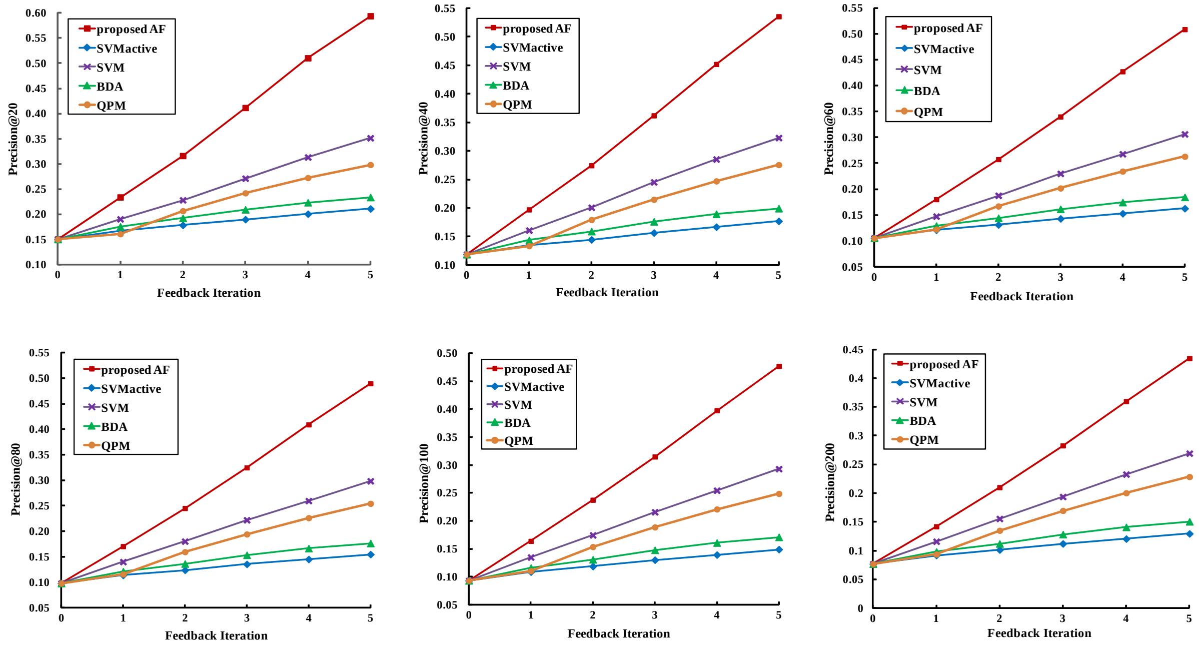


Figure 6: Precision@20, 40, 60, 80, 100, 200 over 700 queries with AF and the four baseline methods at five feedback iterations on the Web Image dataset.

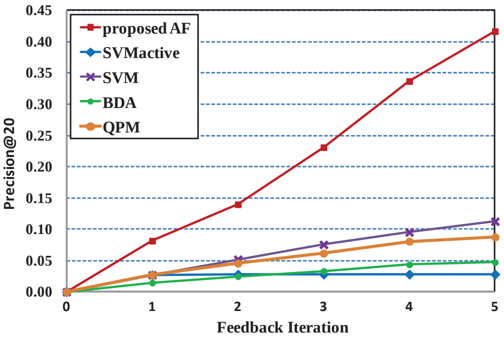


Figure 7: Average Precision@20 by the proposed AF and the four baseline methods with queries that have no relevant samples in the initial search results on the Web Image dataset.

illustrates the performance of the five feedback algorithms with the 127 queries that have no relevant samples in the initial search results on the Web Image dataset. When faced with such situation, the only positive example that the traditional RF methods can use is the query image. Such insufficient cues severely degrade the performance. We can see that AF significantly outperforms the other four methods. This demonstrates the capability of AF in handling no relevant sample problem in CBIR.

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new interactive CBIR scheme, named Attribute Feedback. Unlike traditional relevance feedback that purely based on low-level features, AF allows user to provide feedbacks on intermediate-level semantic attributes and thus can shape user's search intent more precisely and quickly. At each interactive iteration, a set of informative attributes are selected and presented for user feedbacks. A statistical informative attribute selection approach has been proposed. In order to offer more discriminative attribute feedbacks, we proposed a new type of attributes, called "affinity attributes", each of which is learnt off-line to describe the distance between user's envisioned image(s) and a retrieved image with respect to the corresponding affinity attribute. Based on users' feedbacks on binary and affinity attributes, a discriminative search model is then updated towards better fitting users' search intent. We have conducted extensive experiments on two real-world image datasets: the Pascal-Yahoo! corpus containing 15,339 images within 32 categories; and the Web Image collection with 76,303 images of 120 categories. The experimental results demonstrated that the proposed AF scheme outperforms other state-of-the-art relevance feedback based CBIR approaches.

Our future work focuses on the algorithmic and systematic aspects of AF. First, we will incorporate inter-attribute dependencies such as inclusive and exclusive attribute co-occurrence [2] for informative attribute selection. Second, for the robustness of the binary attribute detections, we will refine the attribute classifiers based on the user feedback logs. Last but not the least, we will conduct a comprehensive user study on the practical usefulness of AF in large-scale Web images search.

Acknowledgments

This work was supported by NUS-Tsinghua Extreme Search (NExT) project under the grant number: R-252-300-001-490.

7. REFERENCES

[1] C. Chang and C. Lin. LIBSVM: A library for support vector machines. *ACM TIST*, 2011.

[2] X. Chen, X. Yuan, S. Yan, Y. Rui, and T. Chua. Towards multi-semantic image annotation with graph regularized exclusive group lasso. In *MM*, 2011.

[3] R. Datta, D. Joshi, J. Li, and J. Wang. Image retrieval: Ideas,

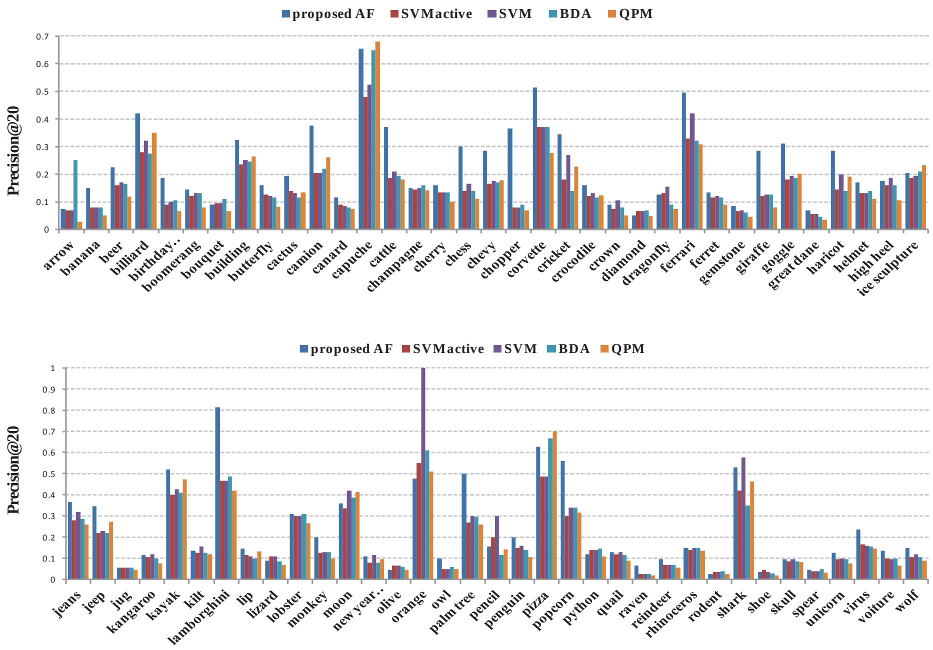


Figure 8: **Detailed retrieval performance over the 70 testing query categories at the first iteration on the Web Image dataset.**

influences, and trends of the new age. *ACM Computing Surveys*, 2008.

[4] M. Douze, A. Ramisa, and C. Schmid. Combining attributes and fisher vectors for efficient image retrieval. In *CVPR*, 2011.

[5] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *CVPR*, 2010.

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

[7] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2008.

[8] G. Guo, A. Jain, W. Ma, and H. Zhang. Learning similarity measure for natural image retrieval with relevance feedback. *TNN*, 2002.

[9] J. Hare, P. Lewis, P. Enser, and C. Sandom. Mind the gap: Another look at the problem of the semantic gap in image retrieval. *Multimedia Content Analysis, Management, and Retrieval*, 2006.

[10] T. Huang and X. Zhou. Image retrieval with relevance feedback: From heuristic weight adjustment to optimal learning methods. In *ICIP*, 2001.

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

[12] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Describable visual attributes for face verification and image search. *TPAMI*, 2011.

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

[14] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 2001.

[15] M. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *TOMCCAP*, 2006.

[16] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011.

[17] Y. Rui, T. Huang, and S. Mehrotra. Content-based image retrieval with relevance feedback in mars. In *ICIP*, 1997.

[18] B. Siddiquie, R. Feris, and L. Davis. Image ranking and retrieval based on multi-attribute queries. In *CVPR*, 2011.

[19] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *TPAMI*, 2000.

[20] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *MM*, 2001.

[21] S. Uchihashi and T. Kanade. Content-free image retrieval based on relations exploited from user feedbacks. In *ICME*, 2005.

[22] A. Vedaldi and B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *MM*, 2010.

[23] M. Wang and X. Hua. Active learning in multimedia annotation and retrieval: A survey. *TIST*, 2011.

[24] Z.-J. Zha., L. Yang, T. Mei, M. Wang, and Z.-F. Wang. Visual query suggestion. In *MM*, 2009.

[25] W. Zhou., Y. LU, H. Li, Y. Song, and Q. Tian. Spatial Coding for Large Scale Partial-Duplicate Web Image Search. In *MM*, 2010.

[26] M. Wang, K. Yang, X.-S. Hua, and H. Zhang. Towards a relevant and diverse search of social images. *TMM*, 2010.

[27] X. Wang, K. Liu, and X. Tang. Query-specific visual semantic spaces for web image re-ranking. In *CVPR*, 2011.

[28] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan. Sparse representation for computer vision and pattern recognition. *PIEEE*, 2010.

[29] Z.-J. Zha, M. Wang, Y.-T. Zheng, Y. Yang, R. Hong, and T.-S. Chua. Interactive video indexing with statistical active learning. *TMM*, 2012.

[30] F. Wu, Y. Han, Q. Tian, and Y. Zhuang. Multi-label boosting for image annotation by structural grouping sparsity. In *MM*, 2010.

[31] Y. Yang, Y. Yang, Z. Huang, H. Shen, and F. Nie. Tag localization with spatial correlations and joint group sparsity. In *CVPR*, 2011.

[32] J. Yuan, Z.-J. Zha, Y. Zheng, M. Wang, X. Zhou, and T.-S. Chua. Utilizing related samples to enhance interactive concept-based video search. *TMM*, 2011.

[33] Z.-J. Zha., L. Yang, T. Mei, M. Wang, Z.-F. Wang, T.-S. Chua, and X.-S. Hua. Visual query suggestion: Towards capturing user intent in internet image search. *TOMCCAP*, 2010.

[34] X. Zhou and T. Huang. Small sample learning during multimedia retrieval using biasmap. In *CVPR*, 2001.

[35] X. Zhou and T. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 2003.