# Implied Feedback: Learning Nuances of User Behavior in Image Search

Devi Parikh
Virginia Tech
parikh@vt.edu

Kristen Grauman
University of Texas at Austin
grauman@cs.utexas.edu

## Abstract

*User feedback helps an image search system refine its relevance predictions, tailoring the search towards the user's preferences. Existing methods simply take feedback at face value: clicking on an image means the user wants things like it; commenting that an image lacks a specific attribute means the user wants things that have it. However, we expect there is actually more information behind the user's literal feedback. In particular, a user's (possibly subconscious) search strategy leads him to comment on certain images rather than others, based on how any of the visible candidate images compare to the desired content. For example, he may be more likely to give negative feedback on an irrelevant image that is relatively close to his target, as opposed to bothering with one that is altogether different. We introduce novel features to capitalize on such implied feedback cues, and learn a ranking function that uses them to improve the system's relevance estimates. We validate the approach with real users searching for shoes, faces, or scenes using two different modes of feedback: binary relevance feedback and relative attributes-based feedback. The results show that retrieval improves significantly when the system accounts for the learned behaviors. We show that the nuances learned are domain-invariant, and useful for both generic user-independent search as well as personalized user-specific search.*

## 1. Introduction

We often use image search to find images that match our visual mental model. For instance, you might see someone wearing a pair of black shoes that you would like to purchase. Or you may be on a dating website trying to find someone with the right looks. Or you may be a graphic designer seeking a specific illustration. Typically, you would use either keywords or a query image to initiate the search. Unfortunately, more often that not, the first round of results returned by today's image search engines will not be satisfactory. Hence, *feedback* plays a critical role in allowing a user to better communicate his needs.

Interactive image feedback can take various forms. In binary relevance feedback [4, 6, 13, 19, 26] the user clicks on a few of the images returned by the system, and conveys whether each one is relevant or not. This system can then reason about the similarity of images in the database to these marked references images and provide an updated set of (hopefully more relevant) results. The feedback can also be relative [10], where the user clicks on a reference image and specifies how what he is looking for is different from it. For instance, the user might say "I am looking for a downtown scene that is less congested than this." With access to a set of relative attribute models [17] (in this case for *congested*), the system can return a new set of results that satisfy the user-specified constraints. Any such feedback – be it based on binary relevance, relative attributes, or some other form – allows users to explicitly inject subjectivity into the search results, and thereby improve performance.

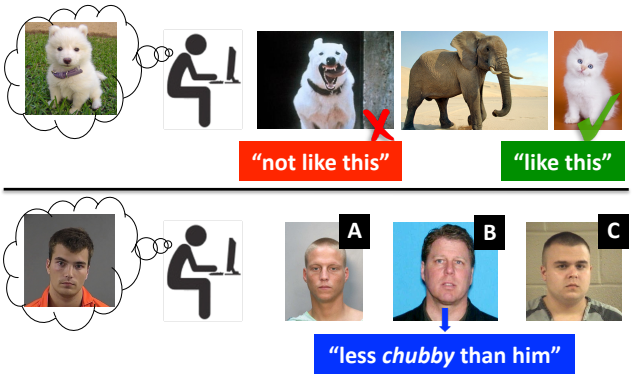



Figure 1: Choices made by users while providing feedback to an image search system reveal information about the desired target image beyond what is explicitly stated. See text.

However, there is more revealed by a user's feedback beyond what is explicitly stated. We hypothesize that when a user provides feedback, some beyond-the-obvious thought – be it conscious or subconscious – goes into the specific choices made. For instance, let's say you are looking for a picture of a white furry puppy to place on an advertisement for an animal shelter. Suppose the set of returned results contains pictures of a white furry kitten, an elephant, and a white angry dog (Figure 1, top). We suspect that you are more likely to click on the kitten and say, "I want something

like this", or on the white angry dog and say, "I don't want something like this." It is unlikely that you will instead provide feedback on the picture of the elephant, even though "I don't want something like this" is true for that image as well. Hence, what the user chooses to *not comment on* contains nuanced but valuable information that is not tapped into by existing work.

Consider another search scenario that allows for *relative* feedback. Someone witnessed a crime and is searching through mugshots of suspects at a local police station where you are the officer in charge. Based on the witness's description, you are only showing him pictures of white men as seen in Figure 1 (bottom). If he now tells you, "The person I saw is less chubby than Person B", is it likely that in the next round you will show him pictures of suspects less chubby than A? No. Rather, you will likely assume that the person the witness saw is *not* less chubby than Person A, otherwise he would have provided that (tighter) constraint.

The above scenarios are just two examples among many other possible ways in which humans communicate implicit information in a visual search task. We hypothesize that such nuances of user behavior also carry over to human-machine interactions, yet (unlike human listeners) machines do not exploit them in existing systems. Importantly, these subtleties need not stem from detailed knowledge of how the search engine works nor be explicitly taught to the user. Instead, these strategies seem to be evoked naturally, perhaps from an implicit assumption by the user that the search engine incorporates feedback in *some* reasonable fashion.

We propose to learn these subtle tendencies in user behavior, with the goal of improving image search. Whereas prior work concentrates on building richer interfaces to elicit more detailed (and thus possibly cumbersome) feedback from the user (e.g., [2, 10, 24]), we explore an orthogonal direction: how can we more richly model the information conveyed in *existing* modes of interaction?

Our approach works as follows. First, we collect training data: human subjects are given a target image to search for, and we record the feedback choices they make. We consider two possible modes of feedback: binary relevance feedback and relative attribute-based feedback. Then, we extract features that characterize the observed feedback interactions, in terms of which among the candidate images the user chooses to comment on, and how. Critically, these features capture users' choices that reflect their underlying search strategy – as opposed to features of the specific target image itself. Then, we learn a ranking function that, given the choices of a user, assigns a higher score to the true target image than to other distractor images in the database.

We stress that our strategy *learns* a model of implicit feedback. Thus, rather than hand code any search rules to exploit scenarios like the ones suggested above, our method will learn the nuances in user behavior that are useful for search. In this regard, the feedback features and ranking formulation we propose are important novel aspects of the work.

We conduct experiments on three domains – scenes, faces, and shoes – and we show that modeling the nuances of user behavior significantly improves image search with both binary relevance and relative attribute-based feedback. Moreover, we show that the model of user behavior learnt is not dataset-specific and can be successfully used across domains. Finally, we show that our model can be used to personalize search results by learning user-specific behaviors, leading to further improvements in search performance.

## 2. Related Work

**Image search:** Many efforts have been made in the computer vision and multimedia community to improve image search. Some approaches build intermediate representations that capture mid-level semantic concepts [5, 15, 18, 21, 23, 24] and help bridge the well known semantic gap. These semantic concepts or "attributes" can also be used to pose queries for image search [11, 20]. Statements about relative attributes can be used to refine search results [10]. Various other modes of feedback have been explored to improve interactive search, the most common being binary relevance feedback [4, 6, 13, 19, 22, 26]. We show that *both* binary relevance and attribute-based feedback are enhanced by the proposed implicit cues. While typically the exemplar images presented to a user are those currently ranked highest by the system, some methods actively select exemplars to elicit the most informative feedback [4, 6]. The goal of our work is orthogonal to these efforts. We wish to model the implicit information hidden in the explicit feedback provided by users in order to improve image search. In particular, the fact that we consider how to more deeply leverage *existing modes of feedback* is in stark contrast to prior work that explores novel forms of *deeper explicit feedback* [2, 10, 24].

**Reading between the lines:** Our idea can be thought of as reading between the lines of what the user is saying, and not simply taking the feedback at face value. This is related to an approach that uses the order in which a user tags objects in an image to better localize the objects [8]. It models how nuances of the image implicitly affect the order in which people name objects in a scene. Although for a completely different application, we are similarly interested in modeling the nuances involved in the complex subconscious strategies users may follow when providing the search engine feedback. This goal also relates to natural language processing research on pragmatics, which studies how people vary their text usage to convey more than their explicit words [7].

**Personalization:** Personalization of web services is receiving more attention as diverse information about users is available online. Some work looks specifically at personalizing image search. This can be viewed as modeling contextual information about a *specific* user beyond what is explicitly stated to improve search. For example, users can teach the machine to detect visual concepts that interest them [2], and user-generated meta-data on social networks can be used to personalize search results by learning user preferences [14]. Our work of modeling user behavior lends itself naturally to personalization. While our primary goal is to learn patterns in collective user behavior to improve search results in a domain- and user-independent manner, we also conduct experiments on learning user-specific behaviors for further improvements in search quality.

## 3. Approach

We model user behavior in two different feedback settings: binary relevance feedback and relative attribute-based feedback (Section 3.1). To gather training data, we conduct user studies and log the interactions – that is, the feedback choices made by users searching for a given target image. We extract features to describe each such interaction that capture not only the feedback the user provided, but also the feedback the user *could* have provided but did not (Section 3.3). We then learn a ranking function, which, given a set of user choices, assigns higher scores to true target images and lower scores to other distractor images in the database (Section 3.2). The learnt ranking function is used at test time; given a novel user's feedback choices, the system computes the likelihood of each image in the database being the target.

### 3.1. Image Search with Feedback

Let's say a user is searching through a database of $N$ images $D = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i, \ldots, \boldsymbol{x}_N\}$ for a specific target image $\boldsymbol{t} \in D$. That target could be something he has literally seen before (e.g., a specific person), or simply a rough idea of what he wants (e.g., a style of shoes). The search process starts by showing the user a set $P$ of $K$ images; $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_v, \ldots, \boldsymbol{p}_K\}$, $P \subset D$. These may be selected randomly, or by using keywords or a query image provided by the user. The user then examines these images. Assuming the target image is not one of them *i.e.* $\boldsymbol{t} \notin P$, the user provides the system feedback on one of the images $\boldsymbol{p}^* \in P$. The system uses this feedback to compute a relevance function $S(\boldsymbol{x}_i)$ that captures the likelihood of image $\boldsymbol{x}_i$ being the target image $\boldsymbol{t}$ (not known to the system). The system sorts all images in $D$ by $S(\boldsymbol{x}_i)$ (in descending order) and returns the top $K$ images as the new set of candidate reference images $P$. This revised set $P$ is shown to the user for further feedback, and the process continues.[1]

We now explain how $S(\boldsymbol{x}_i)$ is computed using two different modes of explicit user feedback. In Sections 3.2 and 3.3, we show how to model the nuances of user behavior for either of these feedback mechanisms to improve their effectiveness.

**Binary Relevance Feedback:** In this form of feedback, the user can select a reference image $\boldsymbol{p}^* \in P$ and state whether the reference image is relevant or irrelevant to the image that he is looking for. If the user says "What I want is like $\boldsymbol{p}^*$", then we have $S(\boldsymbol{x}_i) = -d(\boldsymbol{x}_i, \boldsymbol{p}^*)$, where $d$ captures the distance between two images in some feature space (e.g., texture, attributes). So the images that are most similar to the selected reference image are returned as the most relevant images in the next iteration. If the user says "What I want is not like $\boldsymbol{p}^*$", then $S(\boldsymbol{x}_i) = d(\boldsymbol{x}_i, \boldsymbol{p}^*)$, making the images most dissimilar from $\boldsymbol{p}^*$ to be the most relevant. This simple model captures the essence of standard prior models [19, 26] that continue to be used today (e.g., [25]), though of course one could elaborate the details, for example by using classifiers that use all accumulated feedback.

**Relative Attribute-based Feedback:** In this form of feedback, the user can select a reference image $\boldsymbol{p}^* \in P$ and state *how* it is different from what he is looking for, as proposed in [10]. The system is assumed to have access to a vocabulary of $M$ attributes $\{a_1, \ldots, a_m, \ldots, a_M\}$ (e.g., *shiny*, *chubby*). For each attribute, there is an associated pre-trained relative attribute predictor $r_m(\boldsymbol{x}_i)$, which estimates the extent to which the attribute $m$ is present in image $\boldsymbol{x}_i$. Following [17], we use a max-margin "learning to rank" approach to learn these functions, where the training data consists of image pairs whose relative strengths (more, less, equal) of the property are known. See [17] for details.

If the user feedback is "What I want is more $a_m$ than $\boldsymbol{p}^*$", then the explicit feedback method (which will serve as a baseline for our approach) computes $S(\boldsymbol{x}_i) = r_m(\boldsymbol{x}_i) - r_m(\boldsymbol{p}^*)$. Hence, the stronger the presence of attribute $m$ in an image, the more likely it is to be the target image. Similarly, if the user says "What I want is less $a_m$ than $\boldsymbol{p}^*$", then $S(\boldsymbol{x}_i) = r_m(\boldsymbol{p}^*) - r_m(\boldsymbol{x}_i)$. This relevance scoring strategy is similar to the one proposed in [10], but softer. In [10], $S(\boldsymbol{x}_i) = 1$ if $r_m(\boldsymbol{x}_i) > r_m(\boldsymbol{p}^*)$ and 0 otherwise for the first feedback statement, and vice-versa for the second feedback statement. We find the softer version to work better in practice and so it offers a stronger baseline in our experiments.

### 3.2. Relevance Ranking with Implied Feedback

We build our model of user behavior by collecting training data, that is, by observing users providing feedback in

[1] We describe our approach for the user providing a single statement of feedback on a single reference image for one iteration. However, as we show later, it can be extended to multiple statements and iterations.

search scenarios where the target image is known to us. The $l^{th}$ training interaction $\Omega_l = \langle P_l, \boldsymbol{p}_l^*, m_l^*, q_l \rangle$ is a 4-tuple consisting of (1) $P_l$, the $K$ candidate reference images visible to the user for that interaction, (2) $\boldsymbol{p}_l^*$, the user's choice of reference image, (3) $m_l^*$, his choice of feedback statement, and (4) $q_l$, the "polarity" of the feedback statement. In the case of attribute feedback, $m_l^*$ is the attribute the user chose to comment on, whereas in the case of binary relevance feedback, $m_l^*$ is simply constant, since the user only comments on the "attribute" of generic similarity. The polarity of the feedback $q_l$ refers to whether the user said "more" or "less" in the attribute case, and "similar" or "dissimilar" in the binary relevance case.

We represent each observed interaction with a feature vector $\phi(\boldsymbol{t}_l, \Omega_l)$ that depends on both the target image $\boldsymbol{t}_l$ for that interaction as well as the corresponding choices made by the user $\Omega_l$ while trying to find that target image $\boldsymbol{t}_l$ (feature details to be given in the next section). At test time when a user provides novel feedback $\Omega_{test}$, each image $\boldsymbol{x}_i$ in the database will be considered as a potential target image and will be paired with $\Omega_{test}$ and plugged into $\phi$ to extract the corresponding feature vector $\phi(\boldsymbol{x}_i, \Omega_{test})$.

We wish to learn the scoring function $S(\boldsymbol{x}) = \boldsymbol{w}^T \phi$ such that the score is highest when the true target image $\boldsymbol{t}_l$ is paired with the corresponding interaction $\Omega_l$ to form the arguments to $\phi$. The score should be lower if the same interaction $\Omega_l$ were to be paired with any other distractor (non-target) image from the dataset. In other words, we wish to learn $\boldsymbol{w}$ such that the following constraints are satisfied:

$$\begin{aligned} & S(\boldsymbol{t_l}) > S(\boldsymbol{x}_i) \qquad (1) \\ \implies \quad & \boldsymbol{w}^T \phi(\boldsymbol{t}_l, \Omega_l) > \boldsymbol{w}^T \phi(\boldsymbol{x}_i, \Omega_l), \quad \forall \boldsymbol{x}_i \neq \boldsymbol{t}_l, \forall l. \end{aligned}$$

While this is an NP hard problem [9], it is possible to approximate the solution with the introduction of non-negative slack variables. We directly adapt the formulation proposed in [9], except we use a quadratic loss function. This leads to the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{w}, \xi_{il}} \quad & \frac{1}{2}\left|\left|\boldsymbol{w}\right|\right|_2^2 + C \sum \xi_{il}^2 \qquad (2) \\ s.t. \quad & \boldsymbol{w}^T \phi(\boldsymbol{t}_l, \Omega_l) \geq \boldsymbol{w}^T \phi(\boldsymbol{x}_i, \Omega_l) + 1 - \xi_{il} \\ & \forall \boldsymbol{x}_i \neq \boldsymbol{t}_l, \forall l, \quad \xi_{il} \geq 0. \end{aligned}$$

Rearranging the constraints reveals that the above formulation is quite similar to the SVM classification problem, but on pairwise difference vectors, where $C$ is the trade-off constant between maximizing the margin and satisfying the pairwise relative constraints, and $\xi_{il}$ are slack variables. We solve the primal problem using Newton's method [3]. While we use a linear ranking function in our experiments, the method is also kernelizable. For computational reasons, instead of enforcing the pairwise constraints between every training target image and every other image in the dataset, we enforce them between every target image and a random sampling of 100 images from the dataset.

We stress that the learned function is parameterized by both the target image as well as the user feedback. During training, the target images are known, since we tell the user what image to search for. At test time, however, the target is unknown. What *is* known is the user's interaction with the system $\Omega_{test}$, which includes the candidate reference images the user saw. Therefore, to rank the results at test time, each database image $\boldsymbol{x}_i$ is considered as the potential target in turn and scored accordingly by $S(\boldsymbol{x}_i) = \boldsymbol{w}^T \phi(\boldsymbol{x}_i, \Omega_{test})$, for $i = 1, \ldots, N$. Our method's most confident guess for the target is the database image scored highest by $S(\boldsymbol{x}_i)$.

### 3.3. Features to Capture Implicit Feedback

We now describe our intuitions regarding plausible user behavior when using both types of feedback. Motivated by these intuitions, we design features describing each user interaction $\phi(\boldsymbol{t}_l, \Omega_l)$. We drop the subscript $l$ from the interactions for clarity of notation. The features for any image $\boldsymbol{x}_i \in D$ can be computed the same way by replacing $\boldsymbol{t}$ with $\boldsymbol{x}_i$ in the following. We stress that all the hypotheses below are simply possible behaviors that we want our features to expose to the rank learning algorithm. Ultimately, their impact will be entirely learned, and not hand-coded by us.

**Features for Binary Relevance Feedback:** Recall the puppy example above (Figure 1, top). Perhaps to provide negative feedback, users may click on images that are different enough from the target images to not be satisfactory, but not so different that using them as feedback is barely informative. To capture this sweet spot, we propose the following five features to characterize the interaction. They capture the distance of the selected reference image from the target image relative to the min, max and average distances of all candidate reference images from the target image, as well as relative to the visual diversity spanned by the available candidate reference images: $\phi(\boldsymbol{t}, \Omega) = [d(\boldsymbol{t}, \boldsymbol{p}^*), \quad \frac{\min_{\boldsymbol{p} \in P} d(\boldsymbol{t}, \boldsymbol{p})}{d(\boldsymbol{t}, \boldsymbol{p}^*)}, \quad \frac{d(\boldsymbol{t}, \boldsymbol{p}^*)}{\max_{\boldsymbol{p} \in P} d(\boldsymbol{t}, \boldsymbol{p})}, \quad \frac{d(\boldsymbol{t}, \boldsymbol{p}^*)}{\frac{1}{K} \sum_{\boldsymbol{p} \in P} d(\boldsymbol{t}, \boldsymbol{p})}, \quad \frac{d(\boldsymbol{t}, \boldsymbol{p}^*)}{\max_{\boldsymbol{p}_1, \boldsymbol{p}_2 \in P} d(\boldsymbol{p}_1, \boldsymbol{p}_2) - \min_{\boldsymbol{p}_1, \boldsymbol{p}_2 \in P} d(\boldsymbol{p}_1, \boldsymbol{p}_2)}]$. These features reflect not only how the target relates to the selected reference image, but also how it relates to the reference images that *the user also saw but declined to comment on*. In this way, we capture implicit cues about the user's choice.

We expect the distribution of these features to be different depending on whether the user says "like this" or "not like this" (i.e., the value of $q$). For example, when giving negative feedback, the user may comment on an image that is not too similar to the target, but is also not too dissimilar (the puppy example). In contrast, when giving positive feedback, the user may very well comment on the reference

image that is most similar to the target. Hence, we learn separate scoring functions for the two feedback statements. At test time, depending on the user's feedback, we use the appropriate scoring function. Note that the baseline approach defined in Section 3.1 essentially uses just the first feature in this list (or its negative, depending on $q$).

**Features for Relative Attribute-based Feedback:** This form of feedback provides the user with more options for richer feedback, providing more opportunities to learn the nuances of user behavior. The explicit feedback baseline (defined in Section 3.1) can be thought of as looking at one simple feature (i) $q \cdot (\text{sign}(t_{m^*} - p^*_{m^*}))$ [10] or the softer version (ii) $q \cdot (t_{m^*} - p^*_{m^*})$ where $t_m$ is shorthand for $r_m(\boldsymbol{t})$, the strength of the attribute $a_m$ in image $\boldsymbol{t}$ (and similarly for $p_m$). The direction of feedback $q$ is $+1$ if the user said "more" and $-1$ if the user said "less". The expressions (i) and (ii) form our first two features for $\phi(\boldsymbol{t}, \Omega)$ in the relative feedback case.

Now we propose novel implicit features motivated by plausible hypotheses about user behavior. As described in the crime witness example in the introduction, one hypothesis is that the target image usually lies between the chosen reference image and another candidate reference image closest to it along the chosen attribute (Figure 1, bottom). If all candidate reference images in $P$ are sorted by the chosen attribute $m^*$, let $\boldsymbol{p}^+$ denote the reference image ranked consecutively to the chosen reference image $\boldsymbol{p}^*$ in the direction $q$ of the feedback. Our third feature that captures this hypothesis is: $\log\left(\frac{|t_{m^*} - p^*_{m^*}|}{p^+_{m^*} - p^*_{m^*}}\right)$, where the log helps control the spread of feature values.

Another hypothesis is that relative to the entire range of the attribute values spanned by images in $P$, perhaps the target image is usually close to the chosen reference image along the chosen attribute. This is captured by: $\frac{|p^*_{m^*} - t_{m^*}|}{\max_{p \in P} p_{m^*} - \min_{p \in P} p_{m^*}}$. Or, maybe users pick the reference image and attribute that allow the target image to be as close to the reference image as possible, as captured by the following ratio: $\frac{\min_{p \in P, m \in \{1,\ldots,M\}} |p_m - t_m|}{|p^*_{m^*} - t_{m^*}|}$. From here on we use $\min_{p,m}$ as shorthand for $\min_{p \in P, m \in \{1,\ldots,M\}}$. Several versions of this feature can be computed by replacing the $\min$ operator with $\max$ or average operators across both the choice of reference images and attributes, or just one and not the other. This gives us 8 more features (see supp.).

Further, maybe users pick the attribute and reference image such that the target image falls in the smallest interval formed by any two consecutive candidate reference images when sorted by the strength of the chosen attribute. This is captured by: $\frac{\min_{p,m} |p_m - p^+_m|}{|p^*_{m^*} - p^+_{m^*}|}$. For any candidate reference image $\boldsymbol{p}$ and attribute $m$, $p^+_m$ is the value of the $m^{th}$ attribute in a candidate reference image closest to $\boldsymbol{p}$ along $m$, while ensuring that $t_m \in [p_m, p^+_m]$. Again, different versions of this feature can be obtained by varying the $\min$ operator. This gives us 6 more features (total 19 so far).


Figure 2: Perhaps the user means "I want a scene *overall like this, but* with more of the expanding space property".

Finally, another hypothesis is that when a user says "What I want is more $a_m$ than $\boldsymbol{p}^*$", perhaps he really means "I want something like $\boldsymbol{p}^*$ but more $a_m$" (see Figure 2). In this case the user would pick the reference image and attribute such that the reference image has a high difference from the target image along the chosen attribute, but is similar to the target image along the remaining attributes. That is, $\frac{|p^*_{m^*} - t_{m^*}|}{\max_{m \neq m^*} |p^*_m - t_m|}$ is high. A user may make choices that optimize this value, as captured by: $\frac{\frac{|p^*_{m^*} - t_{m^*}|}{\max_{m \neq m^*} |p^*_m - t_m|}}{\max_{p,m} \frac{|p_m - t_m|}{\max_{m' \neq m} |p_{m'} - t_{m'}|}}$. A variety of such features can be computed by replacing the various $\max$ operators by averages in different combinations (spelled out in supp. file). This gives us a total of 31 features that we use to learn the single scoring function defined in Section 3.2.

**Discussion:** Overall, the proposed features capture both what the user *did* say in the feedback, as well as what he chose *not* to say, in light of all candidate reference images available. While they are fairly complex, they rely on intuitive hypotheses of user behavior. Moreover, we do not assume that the user optimizes these complex features while making choices. Our hypothesis is simply that users have soft inclinations towards some of these high-level strategies, and learning a model in this feature space will help capture those inclinations, leading to improved image search results. If any of our hypotheses are false, the model can learn to ignore the corresponding features.

Instead of learning the nuances of user behavior, one might envision simply *instructing* the user to behave in a certain way. For instance, we could explain to users how the system works, and tell them the optimal strategy of providing feedback to converge on their desired result quickly. However, this line of attack has several flaws. First, it may not be practical to effectively convey this information to layman users using search engines. Second, the optimal strategies may be unnatural or too complex, making the user experience unpleasant and inefficient. Finally, we suspect that in spite of being instructed to follow a certain strategy, natural human tendencies and biases would creep in. Instead of treating these biases as noise, we take the approach of treating them as an informative signal. Instead of forcing users to adapt to a system, we take the approach of having

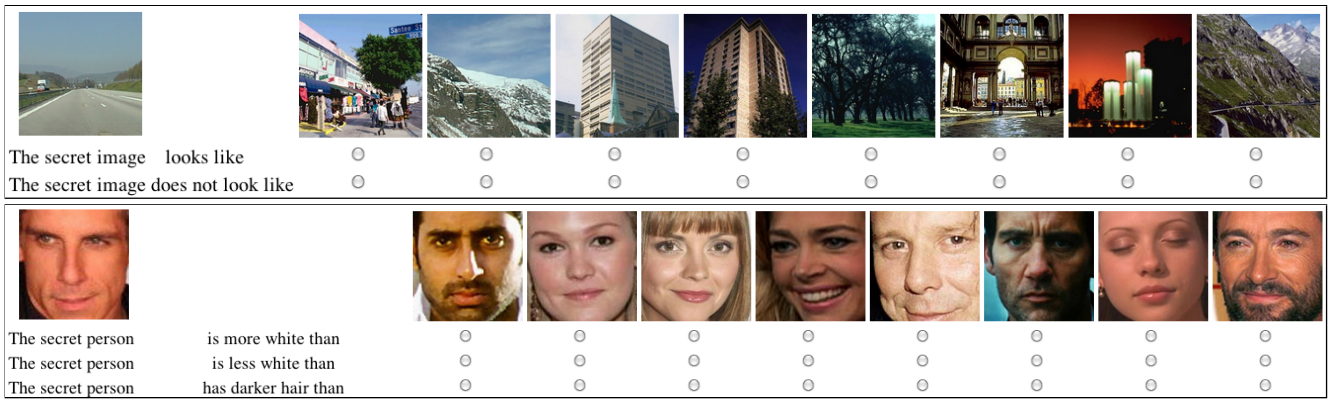



Figure 3: User interfaces for two forms of feedback: binary relevance (top) relative attribute-based (bottom; a subset of attributes are shown for illustration)

our systems adapt to user behavior.

## 4. Results

We conduct experiments on three domains: scenes, faces, and shoes. We use the Outdoor Scene Recognition dataset [16] with 2688 images from 8 categories, 900 random images of the 60 celebrities that comprise the development set of the Public Figures Face Dataset [12], and 1000 random images from all 10 categories in the Shoes dataset [1]. Our search task is to find a specific target image from the entire database, making the notion and number of categories somewhat irrelevant to the task at hand.

**Data Collection:** We collect our user data on Amazon Mechanical Turk. Subjects were shown one target image, and a set of $K = 8$ random images as candidate reference images. The task was disguised as a game between two players. Subjects were told that their goal is to help their partner guess what the "secret" (target) image is by giving him clues (see supp.). Subjects were allowed to pick a reference image and provide a feedback statement (clue) for that image (Figure 3). Our game-like interface is intended to bolster the realism of the data. The game aspect encourages the user to care about the quality of his response, much as he would if doing a search for his own purposes. In contrast, if he were to think he is simply participating in a data collection effort, it could dilute the very nuances in behavior that we are interested in modeling.

For relative attribute-based feedback, subjects were allowed to comment on one of three attributes for scenes [16] (natural, open and expanding space)[2], 10 attributes for faces [12] (*e.g.* masculine-looking, white, dark hair, young, chubby) and 10 attributes for shoes [10] (*e.g.* pointy at the front, open, bright in color). We train all relative attribute predictors offline on a held-out set of images. See supp. for attribute lists and details about the training procedure.

We collected on average 1200 interactions for each dataset from a total of about 60 subjects. For each interaction, we log the true target[3] image $\boldsymbol{t}$ and the tuple $\Omega$. Unless specified otherwise, we use 100 random interactions for training and 100 random interactions for testing. We ensure that the training target images and users do not overlap with the testing target images and users (except for experiments where we learn a user-specific behavior model).

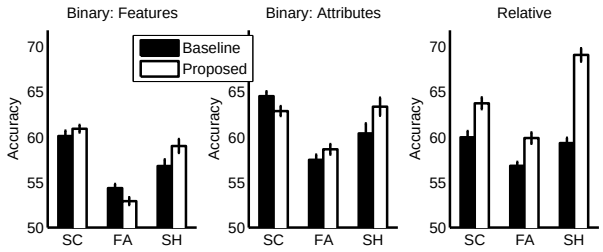



Figure 4: Our approach significantly outperforms baselines on three datasets and two modes of feedback. (SC: scenes, FA: faces, SH: shoes)

Our performance metric is the percentile rank of the target image according to $S(\boldsymbol{x}_i)$, since a good search result will place the desired image near the top of the list. We subtract the raw rank of the target from the total number of images in the dataset, divide by the number of images in the dataset and multiply by 100 to get an accuracy measure between 0 and 100. Higher percentiles are better. We report accuracy averaged across 20 random train/test splits. For binary relevance feedback, we compute the distance $d$ between images using low-level raw image features (gist and color histograms, details in supp.) as well as the mid-level relative attribute prediction scores.

**Binary Relevance Feedback:** We compare our proposed approach of modeling user behavior in binary relevance feedback to the traditional approach [19, 26] (Section 3.1) that simply takes what the user says at face value. If the user says "what I want is (not) like this", it sorts all images in the database in (descending) ascending order of their distance from the selected reference image. In Figure 4 (left

[2] the only three from [16] we find to be reliably understood by users.

[3] We use this info at training time to learn our ranking function, and at test time to evaluate the search results.

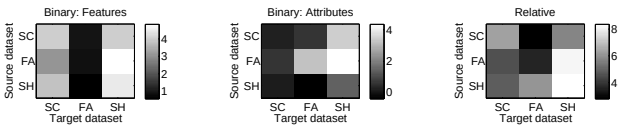



Figure 5: Our models can be trained on different source datasets (SC: scenes, FA: faces, SH: shoes) and effectively applied to other target datasets. Values are improvement in performance over the baseline approach for different pairs of source-target datasets.

two plots) we see that our proposed approach improves performance in most cases, whether using low-level (left) or mid-level features (right). Using state-of-the-art facial appearance features and more attributes for scenes may help overcome the decrease in performance for faces (with features) and scenes (with attributes) respectively.

**Relative Attribute-based Feedback:** We compare to the existing WhittleSearch method [10] (Section 3.1) which, if the user says "what I want is more (less) colorful than this", sorts all images in descending order of how much more (less) colorful they are than the selected reference image. In Figure 4 (right plot) we see significant gains across the board by modeling the nuances of user behavior. On the shoes dataset, we see an improvement of as much as 10 points on the absolute scale. On a dataset of 1000 images, this corresponds to the rank of the true target image improving by 100 spots on average. The shoes dataset has fluid category boundaries. This makes it a rich and particularly realistic testbed for visual search. As a consequence, it is more amenable to eliciting nuances in user behavior, resulting in our dramatic gains in performance. We observe similar improvements using NDCG as the evaluation metric and the ground truth relevance from [10]; this metric accounts not only for the target's rank, but also the rank of images that look similar to it.

**Cross-dataset:** While the above results are trained and tested per dataset, we are also interested in generalizing the user behavior models across domains. We next train our ranking function using interactions from user studies conducted on one dataset and use it to sort the images of a different dataset. We conduct experiments with all pairs of datasets, giving us a $3 \times 3$ performance matrix. Figure 5 shows these matrices for both binary relevance (using both low-level and mid-level features) and relative attribute-based feedback. We display the improvement over the baseline approach; an improvement of 5 means that the rank is 5 percentiles better. We see that our learnt models generalize to other datasets seamlessly. This demonstrates that what our model is learning is truly tendencies of users, and not specifics or biases of a dataset or the attribute predictors.

**User-specific:** While the primary goal of our work is to learn user- and domain-independent models of user behavior, our model naturally lends itself to learning *user-specific* tendencies as well. This allows, for example, a search engine to personalize its reactions to user feedback, given the user's prior search history. We next conduct experiments using interactions from only one user to train our model, and then use held-out interactions from the same user to test our model. The target images do not overlap between the train and test sets. Here we use 50 interactions each for training and testing, taking data from only those subjects who provided us at least 100 interactions (averages 5.5 subjects per dataset). For a fair comparison, we re-train the user-independent models using 50 interactions (instead of 100). Figure 6 shows the results. We see that learning the tendencies of a specific user usually further improves search accuracy. Unmotivated workers can add noise to our data collected via uncontrolled real user studies on Mechanical Turk. Results for user-specific studies can be especially sensitive to this, which may explain the decline in performance for faces with relative attribute-based feedback.

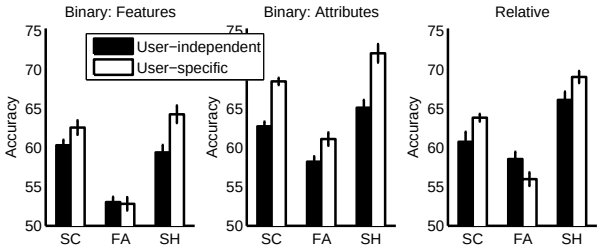



Figure 6: Our models can be used to learn user-specific behaviors for further improvement in performance. (SC: scenes, FA: faces, SH: shoes)

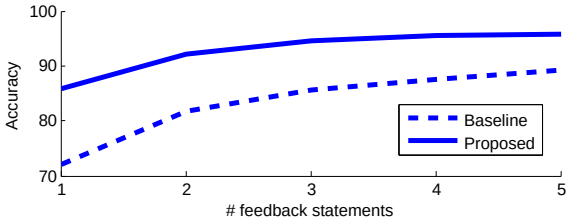
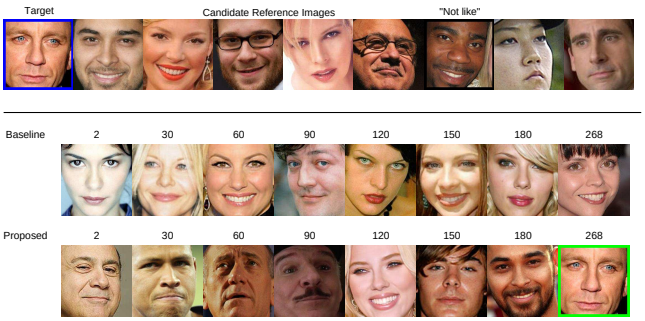



Figure 7: Preliminary result using multiple feedback statements on 5 query shoe images using relative attribute-based feedback.

**Multiple Feedback Statements:** For clarity, our approach is presented and tested in the setting where a user provides one feedback statement on one reference image in a single iteration. However, it naturally extends to handle multiple iterations and/or multiple statements. An example result is shown in Figure 7. We compute the scoring function $S(\boldsymbol{x}_i)$ for each statement individually, and then compute the combined scoring function for multiple statements (possibly gathered across multiple iterations) by summing the individual scoring functions. Our approach continues to outperform the baseline. We leave more elaborate combination strategies as future work.

**Qualitative Results:** Finally, Figure 8 shows example searches for the two modes of feedback. In (a): while searching for a target face (blue outline), the user clicks on the reference image (black outline) and says "Not like this". The baseline approach identifies images most differ-




(a) Binary Relevance Feedback


(b) Relative Attribute-based Feedback

Figure 8: Real examples comparing our method to the baseline for both modes of feedback. In each example, Top: feedback given by MTurk user for the target image on the left. Middle: top ranked images of the baseline. Bottom: our result. Numbers indicate ranks. Best viewed in color.

ent from the reference image to be most relevant and places the target image at rank 833 out of 900. Our approach, on the other hand, prioritizes images that are different enough from the reference image (*e.g.* different race) but still bear similarities to it (*e.g.* same gender, similar age, etc.), placing the target image at rank 268 (outlined in green). In (b): while searching for the target pair of shoes (blue outline), the user clicks on the reference image (black outline) and says "What I want is shinier than this". The existing approach [10] assumes that the shinier the shoe relative to the reference image, the more relevant the image is. As seen from the top ranked retrieved images in the middle row, the baseline returns very shiny shoes. Our approach, on the other hand, has learnt the nuances of user behavior and infers that since there were shinier shoes available in the candidate reference images, but the user did not click on them, what the user must want are shoes that are shinier than the chosen reference image, but not by much. Our top retrieved results are more like the target image than the baseline's. Specifically, the true target image is ranked 93 by our approach, as compared to 953 out of 1000 by the baseline.

**Summary of Results:** Overall, the results quite consistently support our main claim: implicit cues are embedded in existing forms of feedback, and they ought to be learned and exploited for interactive image retrieval, an important problem in computer vision. Whether using traditional binary relevance feedback [4, 6, 13, 19, 22, 26] or a more recent form of attribute feedback [10], our method offers notable gains in search accuracy, yet requires no additional overhead on the part of the user. Further, we have shown that our features and learning formulation are general enough to support both cross-domain use (i.e., implicit cues learned with faces helps do better search for shoes) as well as user-specific personalization.

**Acknowledgements** This research is supported in part by a Google Faculty Research Award (DP) and ONR YIP N00014-12-1-0754 (KG).

## References

[1] T. Berg, A. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010.
[2] S. Bissol, P. Mulhem, and Y. Chiaramella. Towards personalized image retrieval. In *Intl Wkshp on Adaptive Multimed Retrieval*, 2004.
[3] O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 2007.
[4] I. Cox, M. Miller, T. Minka, T. Papathomas, and P. Yianilos. The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments. *Trans on Img Proc*, Jan 2000.
[5] M. Douze, A. Ramisa, and C. Schmid. Combining attributes and fisher vectors for efficient image retrieval. In *CVPR*, 2011.
[6] M. Ferecatu and D. Geman. Interactive search for image categories by mental matching. In *ICCV*, 2007.
[7] E. Hovy. Pragmatics and natural language generation. *Artificial Intelligence*, 43(2):153–197, May 1990.
[8] S. J. Hwang and K. Grauman. Reading between the lines: Object localization using implicit cues from image tags. *PAMI*, 2012.
[9] T. Joachims. Optimizing search engines using clickthrough data. In *SIGKDD*, 2002.
[10] A. Kovashka, D. Parikh, and K. Grauman. WhittleSearch: Image search with relative attribute feedback. In *CVPR*, 2012.
[11] N. Kumar, P. Belhumeur, and S. Nayar. Facetracer: A search engine for large collections of images with faces. In *ECCV*, 2010.
[12] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009.
[13] T. Kurita and T. Kato. Learning of personal visual impression for image database systems. In *ICDAR*, 1993.
[14] K. Lerman, A. Plangprasopchok, and C. Wong. Personalizing image search results on flickr. In *AAAI Wkhp on Intelligent Techniques for Web Personalization*, 2007.
[15] M. Naphade, J. Smith, J. Tesic, S. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-scale concept ontology for multimedia. *IEEE Multimedia*, 13(3), 2006.
[16] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 2001.
[17] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011.
[18] N. Rasiwasia, P. Moreno, and N. Vasconcelos. Bridging the gap: Query by semantic example. *Trans Multimedia*, 9(5), Aug 2007.
[19] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Trans on Circuits and Video Technology*, 1998.
[20] B. Siddiquie, R. S. Feris, and L. S. Davis. Image ranking and retrieval based on multi-attribute queries. In *CVPR*, 2011.
[21] J. Smith, M. Naphade, and A. Natsev. Multimedia semantic indexing using model vectors. In *ICME*, 2003.
[22] K. Tieu and P. Viola. Boosting image retrieval. In *CVPR*, 2000.
[23] X. Wang, K. Liu, and X. Tang. Query-specific visual semantic spaces for web image re-ranking. In *CVPR*, 2011.
[24] E. Zavesky and S.-F. Chang. Cuzero: Embracing the frontier of interactive visual search for informed users. In *ACM MIR*, 2008.
[25] H. Zhang, Z. Zha, S. Yan, J. Bian, and T. Chua. Attribute feedback. In *ACM MM*, 2012.
[26] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8:536–544, 2003.