In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2013.

Attribute Pivots for Guiding Relevance Feedback in Image Search

Adriana Kovashka Kristen Grauman
The University of Texas at Austin
{adriana, grauman}@cs.utexas.edu

Abstract

In interactive image search, a user iteratively refines his results by giving feedback on exemplar images. Active selection methods aim to elicit useful feedback, but traditional approaches suffer from expensive selection criteria and cannot predict informativeness reliably due to the imprecision of relevance feedback. To address these drawbacks, we propose to actively select "pivot" exemplars for which feedback in the form of a visual comparison will most reduce the system's uncertainty. For example, the system might ask, "Is your target image more or less crowded than this image?" Our approach relies on a series of binary search trees in relative attribute space, together with a selection function that predicts the information gain were the user to compare his envisioned target to the next node deeper in a given attribute's tree. It makes interactive search more efficient than existing strategies—both in terms of the system's selection time as well as the user's feedback effort.

1. Introduction

In image search, the user often has a mental picture of his or her desired content. For example, a shopper wants to retrieve those catalog pages that match his envisioned style of clothing; a witness wants to help law enforcement locate a suspect in a database based on his memory of the face. Therefore, a central challenge is how to allow the user to convey that mental picture to the system. Due to the well known semantic gap, one-shot retrieval is generally insufficient. Instead, an *interactive* approach lets the user help the system refine the top-ranked results via iterative feedback [3, 19, 13, 23, 11, 26, 5]. The most common form of interaction consists of binary relevance feedback, in which the user declares certain exemplars to be relevant or irrelevant, and then the system updates its relevance metric in response. With each round of feedback, the results are reranked, and the top-ranked images (ideally) gradually converge on the user's target.

While this basic pipeline is well established, an important question remains: On which images should the user give feedback? Typically, the system simply displays a screen full of top-ranked images, leaving a user free to provide feedback on any of them. This strategy has the appeal of simultaneously showing the current results and accepting feedback [26]. However, the images believed to be most *relevant* need not be most *informative* for reducing the system's uncertainty. As a result, this passive approach may fail to explore relevant portions of the feature space, and can waste interaction cycles eliciting redundant feedback.

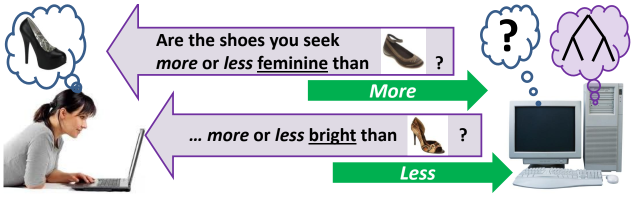
Are the shoes you seek
more or *less* feminine than
?
More
... *more* or *less* bright than
?
Less
?

Figure 1. Our image search approach actively requests feedback on selected images in terms of visual attribute comparisons. To formulate the optimal question to ask next, it unifies an entropy reduction criterion with binary search trees in attribute space.

Thus, methods to *actively* select exemplar images for user feedback are needed. The goal is to solicit feedback on those exemplars that would most improve the system's notion of relevance. Many existing methods exploit classifier uncertainty to find useful exemplars (e.g., [23, 11, 3, 26]). However, traditional approaches have two main limitations. First, the imprecision of binary relevance feedback ("Image X is relevant; image Y is not.") makes it difficult to reliably eliminate database images as irrelevant since the system does not know *what about* the images led to the user's response. This makes it ambiguous how to extrapolate relevance predictions to other images, which in turn clouds the active selection criterion. Second, existing active selection techniques add substantial computational overhead to the interactive search loop, since ideally they must scan all database images to find the most informative exemplars.

We introduce a novel approach that addresses these shortcomings. We propose to guide the user through a coarse-to-fine search using a relative attribute image representation. At each iteration of feedback, the user provides a *visual comparison* between the attribute in his envisioned target and a "pivot" exemplar, where a pivot separates all database images into two balanced sets. Furthermore, we show how to actively determine along which of multiple such attributes the user's comparison should next be requested, based on the expected information gain that would result. See Figure 1.

The approach works as follows. Given a database of images, we first construct a binary search tree for each relative attribute of interest (e.g., "pointiness", "shininess", etc.). Initially, the pivot exemplar for each attribute is the database image with the median relative attribute value. Starting at the roots of these trees, we predict the information gain that would result from asking the user how his target image compares to each of the current pivots. To compute the expected gain, we introduce methods to estimate the likelihood of the user's response given the feedback history. Then, among the pivots, the most informative comparison is requested, generating a question to the user such as, "Is your target image more, equally, or less pointy than this image?" Following the user's response, the system updates its relevance predictions on all images. It also moves the current pivot down one level within the selected attribute's tree (unless the response is "equally", in which case we no longer need to explore this tree). The procedure iterates until the user is satisfied with the top-ranked results.

In technical terms, our problem setting demands repeatedly estimating the total expected error reduction over all unlabeled database images, as a function of requesting any possible comparison from the human searcher. Whereas prior information-gain methods would require a naive scan through all database images for each iteration, the proposed attribute search trees allow us to limit the scan to just one image per attribute. Thus, our method is efficient both for the system (which analyzes a small number of candidates per iteration) and the user (who locates his content via a small number of well-chosen interactions).

We demonstrate our method applied to several realistic search tasks for shoes, people, and scenes. We quantify its advantages over conventional passive and active methods [8, 23]. The results strongly support our pivot-based approach as an efficient means to guide user feedback. For example, in a database of $\sim$15K images, a user can typically locate his exact target image with just 12 rounds of feedback, whereas the standard approach requires 21 rounds to reach the same level of accuracy.

Our main contributions are: (1) a new format for visual search in which the system guides the user through a series of informative visual comparisons, (2) an entropy reduction criterion that exploits the proposed attribute binary search trees for both efficiency and regularization, (3) a technique to predict the likelihood of a user's comparative response given their feedback history, and (4) a probabilistic formulation for using relative attribute feedback.

2. Related Work

Feedback in image search. The benefits of interactive feedback for image search are well studied [3, 19, 26, 5]. In practice, the images displayed to the user for feedback are usually those ranked best by the system's current relevance model. However, if a user is cooperative, it can be more valuable to present a mix of probable relevant and irrelevant examples for feedback. If feedback is binary, with the user labeling examples as relevant (positive) or irrelevant (negative), the selection can naturally be cast as an active learning problem: the best examples to show are those that the relevance classifier is most uncertain about [13, 23, 11, 26].

Notably, prior efforts to display the exemplar set that minimizes uncertainty were forced to resort to sampling or clustering heuristics due to the combinatorial optimization problem inherent when categorical feedback is assumed (e.g., [18, 3, 5]). In contrast, we show that eliciting *comparative* feedback on ordinal visual attributes naturally leads to an efficient sequential selection strategy, where each comparison is guaranteed to decrease the predicted relevance of half of the unexplored database images.

Attributes for image search. Visual attributes are semantic properties of objects (e.g., "fuzzy", "plastic") that serve as a middle ground between low-level features (e.g., color, texture) and high-level categories. Attributes (or "concepts", their counterpart in multimedia retrieval) are known to provide an effective representation for image search [15, 10, 20, 22, 4, 8, 25, 7], especially since they permit content-based keyword queries [10, 22, 7]. While often treated as categorical ("is smiling" vs. "is not smiling"), attributes can more generally be modeled as continuous or *relative* properties ("is smiling more than X") [16, 21].

While binary relevance feedback is most common, our recent work [8] shows how relative visual attributes are useful for feedback (e.g., "retrieve faces that are *smiling more* than this one"). While this work also uses relative attribute feedback, the similarity to [8] ends there. Whereas in [8] search proceeds in a standard passive manner, with the user offering feedback on images of his choosing among the top-ranked ones, our main idea is an actively guided search procedure based on a sequence of system-requested comparisons. This entails novel methods for active selection with binary attribute trees (Sec. 3.2) and user response prediction (Sec. 3.4). Furthermore, we refine the simple counting model of [8] to account for uncertainty in attribute predictions (Sec. 3.3).

Active testing and "20 questions". Active testing methods choose a series of useful "tests" (e.g., features to extract) or label requests ("does the bird have a yellow beak?") [6, 2]. In the case where a *human* answers the tests, attributes are well-suited to query for intermediate labels that will lead to the right category label, as shown for bird labeling [2]. Our work shares the spirit of rapidly reducing uncertainty through a sequence of useful questions. However, our aim is distinct. Active testing entails selecting queries to classify a single novel image efficiently, whereas we select queries to efficiently find a target in a database of images. Moreover, our approach solicits visual *comparisons*—key to eliminating irrelevant content in search—whereas prior work solicits traditional image labels.

Active classifier training with attributes. More distant from our work, other work investigates training classifiers with actively selected attribute labels. By modeling object and attribute relationships [24, 9, 14], one can request the most useful labels to refine the classifiers. Our goal is very different: we do active feedback requests for image search, not classification, and our approach requests visual comparisons, not attribute labels.

3. Approach

A user initiates a search with a multi-attribute query (e.g., "black high-heels") or a sample image (e.g., a snapshot of a pair of heels she saw). Our approach then refines the results. It interacts with the user through multiple-choice questions of the form: "Is the image you are looking for *more*, *less*, (or *equally*) A than image I?", where A is a semantic attribute and I is an exemplar from the database being searched. Our goal is to generate the series of such questions that will most efficiently narrow down the relevant images in the database, so that the user finds his target[1] in few iterations. To this end, at each iteration we will actively select a comparison for the user to provide, that is, the (A, I) pair which yields the expected maximal information gain. Rather than exhaustively search all database images as potential exemplars, however, we consider only a small number of *pivot* exemplars—the internal nodes of binary search trees constructed for each attribute. The output of the system is the list of database images, sorted by their predicted relevance.

After reviewing an existing method [16] to predict relative attribute strengths (Sec. 3.1), we explain how we construct attribute binary search trees (Sec. 3.2). Next we present our model of image relevance that accounts for the user's attribute-based feedback (Sec. 3.3). Finally, we introduce our active selection approach to determine which comparison should be requested next (Sec. 3.4).

In the following, let $\mathcal{I} = \{I_1, \dots, I_N\}$ denote the N images in the database, each of which has a corresponding image descriptor $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ (e.g., GIST, bag of words, etc.). Suppose we have an attribute vocabulary consisting of M properties $A_1, \dots, A_m, \dots, A_M$. For example, for a shoe shopping database, those properties might be "pointiness", "shininess", "heel height", etc. We use $A_m(I_i)$ to denote the true strength of an attribute m in image I_i—that is, as would be perceived by a human viewer.

3.1. Relative Attribute Predictions

In order to utilize attribute-based comparisons, we need to estimate the strength of each attribute in each database image. To this end, following [16], we learn one ranking function per attribute. For each attribute m, we obtain a set of ordered pairs $O_m = \{(I_i, I_j)\}$, for which each image I_i has greater strength of attribute m than image I_j does, as well as a set of unordered pairs $E_m = \{(I_i, I_j)\}$, for which both images in a pair exhibit the attribute equally. All such pairs come directly from comparative human judgments.

For each attribute m, we use its associated training pairs to learn a (possibly kernelized) ranking function: $a_m(I_i) = \boldsymbol{w}_m^T \boldsymbol{x_i}$, which maps the image descriptor $\boldsymbol{x_i}$ for image I_i to its real-valued attribute strength. The projection parameters $\boldsymbol{w}_m$ are optimized using a large-margin ranking objective. It aims to satisfy the ordered pair constraints above, such that $\boldsymbol{w}_m^T \boldsymbol{x}_i > \boldsymbol{w}_m^T \boldsymbol{x}_j, \forall (I_i, I_j) \in O_m$, and $\boldsymbol{w}_m^T \boldsymbol{x}_i \approx \boldsymbol{w}_m^T \boldsymbol{x}_j, \forall (I_i, I_j) \in E_m$, while at the same time maintaining a wide margin in the output ranks of the nearest training examples. See [16] for details.

These predicted attribute values $a_m(I_i)$ are what we can observe for image I_i. They are a function of (but distinct from) the "true" latent attribute strengths $A_m(I_i)$. We will refer to both below. Using standard features and kernels, we find that 75% of held-out human comparisons are preserved by attribute predictors trained with $\sim$200 pairs. Thus, they are quite reliable; more elaborate features [10] or learning algorithms [12] would likely improve them even further.

3.2. Attribute Binary Search Trees

For each attribute $m = 1, \dots, M$, we construct a binary search tree. The tree recursively partitions all the database images into two balanced sets, where the key at a given node is the median relative attribute value occurring within the set of images passed to that node. To build the m-th attribute tree, we start at the root with all database images, sort them by their predicted attribute values $a_m(I_1), \dots, a_m(I_N)$, and identify the median value. Let I_p denote the "pivot" image—the one that has the median attribute strength. Those images exhibiting the attribute less than I_p, i.e., all I_i such that $a_m(I_i) \leq a_m(I_p)$, are passed to the left child, while those exhibiting the attribute more, i.e., $a_m(I_i) > a_m(I_p)$, are passed to the right child. Then the splitting repeats recursively, each time storing the next pivot image and its relative attribute value at the appropriate node.

Note that both the relative attribute ranker training and the search tree construction are offline procedures; they are

[1] Throughout we use "target" to refer to the imagined visual content of the user. It could be a literal image s/he has seen before, or simply a mental model of content of interest.

performed once, before handling any user queries.

Already, one could imagine a search procedure that walks a user through one such attribute tree, at each successively deeper level requesting a comparison to the pivot, and then eliminating the appropriate portion of the database depending on whether the user says "more" or "less". However, there are two problems with such a simple approach. First, we cannot assume that the attribute predictions are identical to the attribute strengths a user will perceive; thus, a hard pruning of a full sub-tree is error-prone. Second, this approach fails to account for the variable information gain that could be achieved depending on *which attribute* is explored at any given round of feedback. Therefore, we propose a probabilistic representation of whether images satisfy the comparison constraints (Sec. 3.3), and we use the pivots to limit the pool of candidate images that are evaluated for their expected information gain (Sec. 3.4).

3.3. Predicting the Relevance of an Image

Now we explain how we predict the relevance of a database image, given the user's comparative feedback. Let $y_i \in \{1, 0\}$ denote the binary label for image I_i, which reflects whether it is relevant to the user (matches his target), or not. Let $\mathcal{F} = \{(I_{p_m}, r)_k\}_{k=1}^{T}$ denote the set of comparative constraints accumulated in the T rounds of feedback so far. The k-th item in $\mathcal{F}$ consists of a pivot image I_{p_m} for attribute m, and a user response $r \in$ {"more", "less", "equally"}. The final output of our search system will be a sorting of the database images $I_i \in \mathcal{I}$ according to their probability of relevance, given the image content and all user feedback: $P(y_i = 1|I_i, \mathcal{F})$.

Let $S_{k,i} \in \{0, 1\}$ be a binary random variable representing whether image I_i satisfies the k-th feedback constraint. For example, if the user's k-th comparison yields response $r =$ "more", then $S_{k,i} = 1$ if the database image I_i has attribute m more than the corresponding pivot image I_{p_m}. The probability of relevance is thus the probability that all T feedback comparisons in $\mathcal{F}$ are satisfied:

$$P(y_i = 1|I_i, \mathcal{F}) = \sum_{k=1}^{T} \log P(S_{k,i} = 1|I_i), \qquad (1)$$

where we use a sum of log probabilities rather than a product for numerical stability.

The probability that the k-th individual constraint is satisfied given that the user's response was r for pivot I_{p_m} is:

$$P(S_{k,i} = 1|I_i) = \begin{cases} P(A_m(I_i) > A_m(I_p)) & \text{if } r = \text{"more"} \\ P(A_m(I_i) < A_m(I_p)) & \text{if } r = \text{"less"} \\ P(A_m(I_i) = A_m(I_p)) & \text{if } r = \text{"equally"}. \end{cases}$$

To estimate these probabilities, we map the attribute predictions $a_m(\cdot)$ to probabilistic outputs, by adapting Platt's method [17] to the paired classification problem implicit in the large-margin ranking objective. Specifically, this yields:

$$P(A_m(I_i) > A_m(I_p)) = \frac{1}{1 + \exp(\alpha_m(a_m(I_i) - a_m(I_p)) + \beta_m)} \qquad (2)$$

$$P(A_m(I_i) = A_m(I_p)) = \frac{1}{1 + \exp(\gamma_m|a_m(I_i) - a_m(I_p)| + \delta_m)}, \qquad (3)$$

where the sigmoid parameters are learned using the sets O_m and E_m from above. In particular, to learn α_m and β_m, we use pairs with "more" judgments from O_m as positive paired-instances, and "less" judgments as negative instances. For γ_m and δ_m, we use "equally" pairs from E_m as positive labels, and both "more" and "less" responses from O_m as negative instances. Note $P(A_m(I_i) < A_m(I_p)) = 1 - P(A_m(I_i) > A_m(I_p))$. When computing the user response likelihoods in Sec. 3.4, we normalize these values so the three probabilities ("more"/"less"/"equally") sum to 1.

Our probabilistic model of relevance accounts for the fact that predicted attributes can deviate from true perceived attribute strengths. In contrast, prior work using relative attribute feedback [8] makes hard decisions, simply counting how many predicted attribute values satisfy the user's constraints to measure relevance. We find that a hard pruning of images on irrelevant branches of an attribute tree eliminates the true target for 93% of the queries, clearly supporting the proposed probabilistic formulation.

3.4. Actively Selecting an Informative Comparison

The proposed binary trees serve to guide the active exemplar selection and reduce its computational overhead, rather than completely eliminate images from consideration. Our system maintains a set of M current pivot images (one per attribute tree) at each iteration, denoted $\mathcal{P} = \{I_{p_1}, \dots, I_{p_M}\}$. The pivots are initially the root pivot images from each tree. During active selection, our goal is to identify the pivot in this set that, once compared by the user to his target, will most reduce the entropy of the relevance predictions on all database images. Note that selecting a pivot corresponds to selecting both an image as well as an attribute along which we want it to be compared; I_{p_m} refers to the pivot for attribute m.

Entropy reduction objective. Given the feedback history $\mathcal{F}$, we want to predict the information gain across all N database images for each pivot in $\mathcal{P}$. We will request a comparison for the pivot that most reduces the total relevance entropy over all images—or equivalently, the pivot that minimizes the expected entropy when used to augment the current set of feedback constraints.

The entropy based on the feedback thus far is:

$$H(\mathcal{F}) = -\sum_{i=1}^{N} \sum_{\ell} P(y_i = \ell|I_i, \mathcal{F}) \log P(y_i = \ell|I_i, \mathcal{F}), \qquad (4)$$

where $\ell \in \{0, 1\}$. Let R be a random variable denoting the user's response, $R \in \{$"more", "less", "equally"$\}$. We select the next pivot for comparison as:

$$I_p^* = \underset{I_{p_m} \in \mathcal{P}}{\arg\min} \sum_r P(R = r | I_{p_m}, \mathcal{F}) \; H(\mathcal{F} \cup (I_{p_m}, r)). \quad (5)$$

The basic idea of expected error reduction was first proposed in [18] for active learning in text classification, and variations have been explored in vision tasks (e.g., [2, 9, 14]). Our formulation is novel in that we survey only the attribute pivots, exploiting the special structure of rankable visual properties for substantial computational savings. In contrast, existing work resorts to sampling heuristics [3], approximations [5], or simply small data pools [9] to make the problem tractable.

Furthermore, as we will show in the results, the pivots also enhance selection *accuracy*, by essentially isolating those images likely to impact relevance predictions. Intuitively, if a user has ruled out a subtree ("Target is bluer than image with blueness X."), it is likely redundant (low info gain) to ask how the target compares to more data on that path ("Is target bluer than image with blueness $X - Y$?"), i.e., ask the user to comment on something even less blue than the previous exemplar.

User response likelihood. Optimizing Eqn. 5 requires estimating the likelihood of each of the three possible user responses to a question we have not issued yet. We develop three possible strategies to estimate it. In each case, we use cues from the available feedback history to form a "proxy" for the user, essentially borrowing the probability that a new constraint is satisfied from previously seen feedback.

For the first strategy, which we call *All Relevant*, we use all relevant database images as the proxy. The assumption is that the images that are relevant to the user thus far are (on the whole) more likely to satisfy the user's next feedback than those that are irrelevant. This is reminiscent of active classifier training, where posteriors estimated with the current classifier are used as weights in the expected entropy reduction of acquiring a new label. Ideally we would average the $P(S_{c,i} = 1|I_i)$ values among only the relevant images I_i, where c indexes the candidate new feedback for a (yet unknown) user response R. Of course, we can only *predict* relevance, so we compute the weighted probability of each possible response R:

$$P_{all}(R = r | I_{p_m}, \mathcal{F}) = \frac{1}{N} \sum_{i=1}^{N} P(y_i = 1 | I_i, \mathcal{F}) P(S_{c,i} = 1 | I_i), \quad (6)$$

where the *all* subscript stands for *All Relevant*.

The second strategy, which we call *Most Relevant*, is similar, but uses only our current best guess for the target image as the proxy:

$$P_{most}(R = r | I_{p_m}, \mathcal{F}) = P(S_{c,b} = 1 | I_b), \quad (7)$$

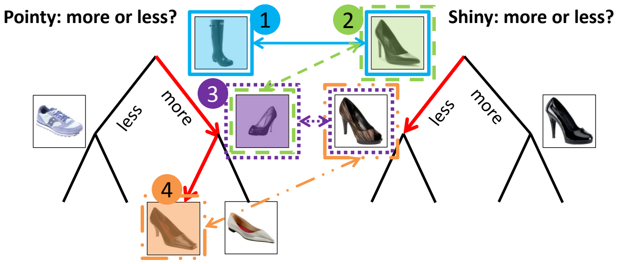

Figure 2. We request feedback on images that elicit the most information, using binary search trees to focus the active selection. In this sketch, $M = 2$ attribute trees are shown. Images with the same color outline are the pairs considered at each round, and the number in this color marks the image chosen at this round. Red arrows denote the user's responses. Here, first the user is asked to compare his target to the boot pivot (1) in terms of pointiness; then he is asked to compare it to (2) in terms of shininess, followed by (3) in terms of pointiness, and so on. Best viewed in color.

where I_b is the database image that maximizes $P(y_i = 1 | I_i, \mathcal{F})$, for $i = 1, \ldots, N$.

The third strategy, which we call *Similar Question*, examines all previously answered feedback requests, and copies the answer from the question that is most similar to the new one. We define question similarity in terms of the Euclidean distance between the pivot images' descriptors plus the similarity of the two attributes involved in either question. We quantify the latter by the Kendall's τ correlation between the ranks they assign to a set of validation images. For example, this reflects that "feminine" and "heel height" are more aligned than "feminine" and "grayness". Let r_k^* denote the response to the most similar question k found in the history $\mathcal{F}$ for the new pivot I_{p_m} under consideration. Then we have:

$$P_{question}(R = r | I_{p_m}, \mathcal{F}) = \begin{cases} 1 & \text{if } r = r_k^* \\ 0 & \text{otherwise} . \end{cases} \quad (8)$$

We evaluate all three likelihood strategies in the results.

Recap of interaction loop. At each iteration, we present the user with the pivot selected with Eqn. 5 and request the specified attribute comparison. In order for the user to monitor the search progress and stop if an image similar to his target has been found, we also show him the current top-ranked images. If further feedback is given, we first update $\mathcal{F}$ with the user's new image-attribute-response constraint. Then we either replace the pivot in $\mathcal{P}$ for that attribute with its appropriate child pivot (i.e., the left or right child in the binary search tree if the response is "less" or "more", respectively) or terminate the exploration of this tree (if the response is "equally"). Note that this means that the set of pivots consists of pointers into the binary trees at *varying* levels. See Figure 2. This is because our active selection criterion considers which attribute will most benefit from more refined feedback at any point in time.

Finally, the approach iterates until the user is satisfied with the top-ranked results, or until all of the attribute trees have bottomed out to an "equally" response from the user (in which case, our method can gain no further knowledge about the target given the available attribute vocabulary).

The cost of our selection method per round of feedback is $O(MN)$, where M is the size of the attribute vocabulary, N is the database size, and $M \ll N$. In contrast, a traditional information gain approach would scan all database items paired with all attributes, requiring $O(MN^2)$ time.

4. Experiments

We validate with three public datasets: **Shoes** [1], with the attributes from [8] (14,658 images and 10 attributes); outdoor **Scenes** (2,688 images and 6 attributes); and PubFig celebrity **Faces** [10] (772 images and 11 attributes). We concatenate GIST and color features for Shoes and Faces, and GIST alone for Scenes. To train the relative attributes $a_m(\cdot)$ and fit the sigmoid parameters in Sec. 3.3, we use the human judgment data provided online by [8], with about 200 image pairs per attribute. See supp. file for details.

Evaluation metrics. In order to quantify accuracy precisely, we tell the user which image to search for. That is, for a given search session, the user is instructed to give feedback by comparing the target we specify to the various methods' selected exemplars. We report the **percentile rank** each method assigns to the target at each iteration, defined as the fraction of database images ranker lower than the target. Higher percentile ranks are better; the ideal method would rank the target at the top of the search results page after very few iterations of feedback. Additionally, we measure the **NDCG@40 correlation** between the method's full ranking and the ground truth ranking. Higher correlations are better. To define the ground truth ranking, we sort all database images according to their perceptual distance (a learned metric on attributes and low-level features) from the target, following [8]. The two metrics give complementary information: while rank reveals how the exact target image ranks, NDCG reveals how many images very similar to the target are found among the top-ranked results.

Baselines. We compare our method ACTIVE ATTRIBUTE PIVOTS against the following six methods:

- ATTRIBUTE PIVOTS is a simplified version of our method that uses the proposed attribute trees to select candidate images, but cycles among the attributes in a round-robin fashion.
- ACTIVE ATTRIBUTE EXHAUSTIVE uses entropy to select questions like our method, but it evaluates all possible MxN candidate questions.
- TOP selects the image that has the current highest probability of relevance and pairs it with a random attribute. This method represents traditional interactive methods that assume an "impatient" user for whom feedback exemplars and search results must be one and the same. It is similar in spirit to [8].
- PASSIVE selects a random image paired with a random attribute for its question.
- ACTIVE BINARY FEEDBACK does not use statements about the relative attribute strength of images, but rather asks the user whether the exemplar is similar to the target. This popular method uses a binary SVM to rank images, and treats similar images as positives and dissimilar images as negatives. It actively chooses the image whose decision value is closest to 0, as in [23].
- PASSIVE BINARY FEEDBACK works as above, but randomly selects the images for feedback.

Relative feedback methods use the same relevance prediction function and only differ in the feedback they gather.

4.1. Results with Feedback by Simulated Users

To thoroughly test the methods, we first conduct experiments where we simulate the user's responses.[2] We generate the response for, "Is the target image *more, equally, or less* m than I_{p_m}?" using the difference in the predicted attribute values for the target and I_{p_m}. For a response of "equally", we use a threshold derived from the training data. By extrapolating a sparse set of real human judgments through a learned ranking function, we can perform large-scale comparisons and isolate the impact of our idea from the impact of the attribute rankers' precision.

We initialize all attribute search methods with the same feedback constraint. For ACTIVE BINARY FEEDBACK, we respond with "similar" if the target and exemplar images are within one standard deviation of the distances used for the ground truth ranking. We initialize this method with one positive and one negative image by peeking at the distances between the target image and a pool of 40 images. We add Gaussian noise to the relevance predictions of all methods in order to reflect the discrepancy between perceived and predicted attributes. See supp. for more details. We show all results over 200 randomly chosen queries (target images).

Comparison of likelihood models. Figure 3 compares the three proposed methods of predicting the user response. *Most Relevant* consistently outperforms the other two methods on all but the Scenes. This suggests that our best guess at the target tends to be a sufficient proxy, having a fairly similar attribute signature. *All Relevant* is slightly weaker, indicating that isolating the most relevant instance gives a

[2] The protocol is related to standard validation for active learning, where the algorithm receives the labels for those examples it queries, even if a human is not answering "live" in the loop. Note, gathering all possible comparisons in advance would cost *\$2B* if paying Turkers 1 cent each!

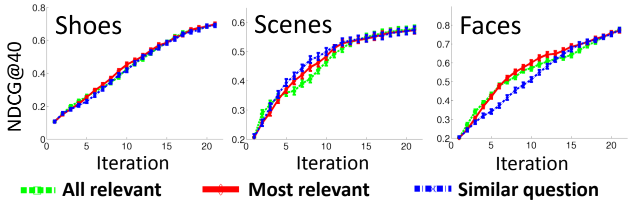

Figure 3. Comparing the proposed models for the likelihood of a user's response (higher curves are better). Best viewed in color.

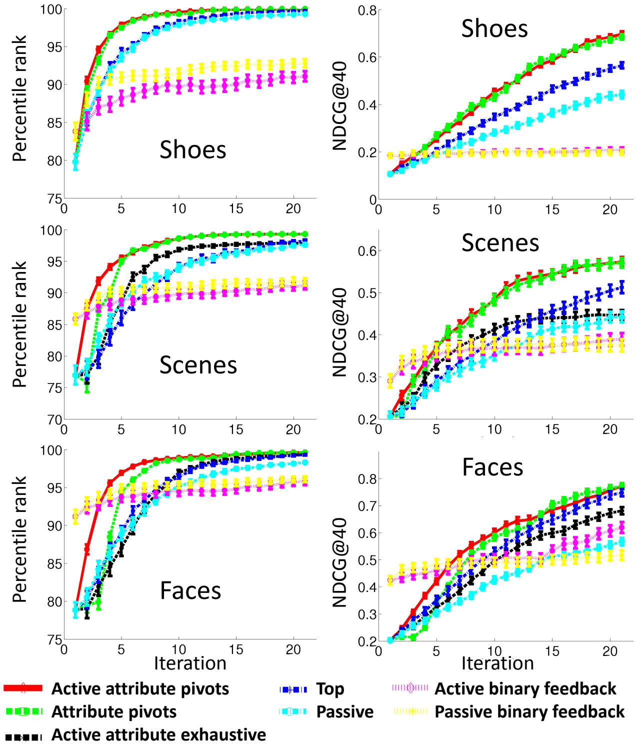

Figure 4. Comparison to existing interactive search methods (higher and steeper curves are better). Best viewed in color.

Method/Dataset	Shoes	Scenes	Faces
Active attribute pivots (Ours)	0.05	0.01	0.01
Active attribute exhaustive	656.27	28.20	3.42

Table 1. Selection time for one iteration of our method vs. the exhaustive active baseline, in seconds.

"cleaner" likelihood than attempting to refine it with our uncertainty about each relevant instance. *Similar Question* performs the best for a fraction of the iterations on Scenes, but does poorly on Faces. This is likely because we cannot estimate attribute similarity reliably due to the distinct face attributes (e.g., face "chubbiness" has no strongly correlated attributes, whereas scene "openness" does). In all remaining results, we use the *Most Relevant* method.

Comparison to existing methods. Figure 4 compares all methods on all three datasets. Overall, our method finds the target image most efficiently. Not only does it outperform traditional passive selection (Passive), but it also substantially improves over the Top approach. This shows that relative attribute feedback alone (the contribution of [8]) does not offer the most efficient search; rather, our idea to actively elicit comparisons is essential. We also see that our full active approach outperforms the round-robin variant of our method (Attribute pivots), with an average percentile rank 7.6% better after only 3 iterations. This shows actively interleaving the trees allows us to focus on attributes that better distinguish the relevant images.

Our method also outperforms Active attribute exhaustive.[3] This shows that the attribute trees serve as a form of regularization, helping our method focus on those comparisons that *a priori* may be most informative. Furthermore, our method is orders of magnitude faster (see Table 1).

The results confirm the striking advantage of attribute feedback compared to binary relevance feedback. Binary feedback has an advantage only in the first few iterations, likely because we generously initialize it with 2 feedback statements. We find that both feedback modes require similar user time: 6.4 *s* for relative, and 5.5 *s* for binary, and so the trends remain if we plot rank as a function of user time (see supp). Interestingly, we find that Passive binary feedback is actually stronger than its active counterpart for this data. This is likely because images near the decision boundary were often negative, whereas the passive approach samples more diverse instances.

In practical terms, we are interested in how many iterations it takes to get the target in the top 40 most relevant images, since that is how many images fit on a typical search page (e.g., on Google). On average our method uses 12, 10, and 4 iterations to place the target in the top 40 for Shoes, Scenes, and Faces, vs. 21, 21, and 9 iterations for Top. Thus, our method saves a user up to 70 seconds per query.

4.2. Results with Live Users

Next, we test our method "live" in real time with Mechanical Turk workers. We compare its performance against our Attribute pivots and the strongest baseline, Top. We issue 50 queries for Shoes-1k (a random 1000-image subset of Shoes), Scenes, and Faces-Unique (1 image for each of 200 individuals from the original PubFig dataset [10], using the 6 most predictable attributes). All methods share one simulated feedback statement at iteration 0, which we do not plot. See supp. for details. Note, this experiment is only possible because our method can make decisions in real time, unlike the exhaustive active method.

Figure 5 shows the results. Consistent with the results above, we see that typically our method ranks the target image better than the baselines. We achieve a 100-200 raw

[3] The exhaustive baseline was too expensive to run on all 14K Shoes. On a 1000-image subset, it does similarly as on other datasets; see supp.

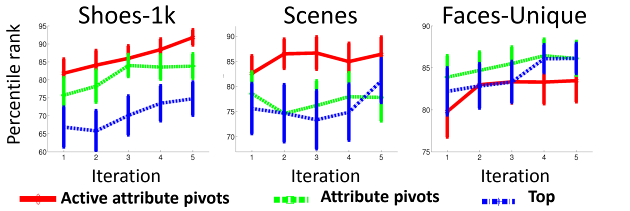

Figure 5. Our method makes quick and reliable choices, allowing the MTurk users to more efficiently find the target.

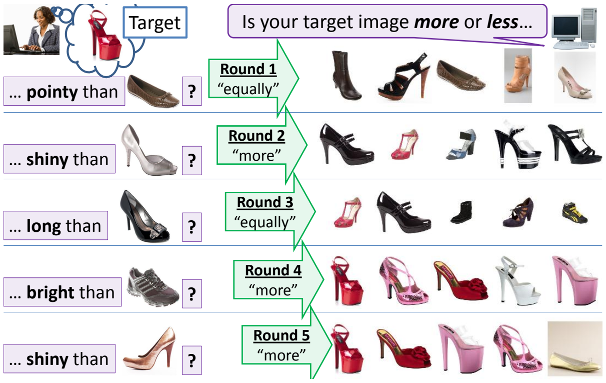

Figure 6. Using the user's feedback on the left, we retrieve the images on the right at the top of the results list.

rank improvement on two datasets, and a negligible 0-10 raw rank loss on Faces. This is a very encouraging result, given the noise inherent in MTurk responses (in spite of our best efforts at qualification tests) and the difficulty of predicting all attributes reliably. Our informativeness predictions on Faces-Unique are imprecise since the facial attributes are difficult for both the system and humans to compare reliably (e.g., it is hard to say who among two white people is whiter). This difficulty seems to hurt all methods, judging by their flatter curves. Since the rank metric does not give any credit for finding an image very close to the target, we also asked a separate set of workers to judge whether any of the top 10 ranked images were "very similar" to the target. For Shoes-1k, our full method takes only 1.9 iterations on average to find one that is very similar, whereas our ATTRIBUTE PIVOTS require 2.4 and TOP requires 3.15.

Figure 6 shows an example search done by an MTurker. Notice how our method generates useful comparison questions across the different attributes, quickly converging on top-ranked images that look like the target.

Conclusion Today's visual search systems place the burden on the user to initiate useful feedback by labeling images as relevant. In contrast, our system *actively* guides the search based on visual *comparisons*, helping a user navigate the image database via relative semantic properties. Compared to existing active and passive methods, our pivot-based formulation is both more efficient (by orders of magnitude) and more accurate in practice. Results with both simulated and live users confirm that we can rapidly pinpoint the visual target using a series of well-chosen comparative queries. In future work, we plan to explore ways to personalize results given a user's prior search sessions.

Acknowledgement This research was supported in part by ONR YIP grant N00014-12-1-0754.

References

[1] T. Berg, A. Berg, and J. Shih. Automatic Attribute Discovery and Characterization from Noisy Web Data. In *ECCV*, 2010.

[2] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. Visual Recognition with Humans in the Loop. In *ECCV*, 2010.

[3] I. Cox, M. Miller, T. Minka, T. Papathomas, and P. Yianilos. The Bayesian Image Retrieval System, PicHunter: Theory, Implementaion and Psychophysical Expts. *IEEE Trans on Image Proc*, 2000.

[4] M. Douze, A. Ramisa, and C. Schmid. Combining Attributes and Fisher Vectors for Efficient Image Retrieval. In *CVPR*, 2011.

[5] M. Ferecatu and D. Geman. Interactive Search for Image Categories by Mental Matching. In *ICCV*, 2007.

[6] D. Geman and B. Jedynak. Model-Based Classification Trees. *IEEE Transactions on Information Theory*, 1998.

[7] A. Kovashka and K. Grauman. Attribute Adaptation for Personalized Image Search. In *ICCV*, 2013.

[8] A. Kovashka, D. Parikh, and K. Grauman. WhittleSearch: Image Search with Relative Attribute Feedback. In *CVPR*, 2012.

[9] A. Kovashka, S. Vijayanarasimhan, and K. Grauman. Actively Selecting Annotations Among Objects and Attributes. In *ICCV*, 2011.

[10] N. Kumar, P. Belhumeur, and S. Nayar. Facetracer: A Search Engine for Large Collections of Images with Faces. In *ECCV*, 2008.

[11] B. Li, E. Chang, and C.-S. Li. Learning Image Query Concepts via Intelligent Sampling. In *ICME*, 2001.

[12] S. Li, S. Shan, and X. Chen. Relative forest for attribute prediction. In *ACCV*, 2012.

[13] S. D. MacArthur, C. E. Brodley, and C.-R. Shyu. Relevance Feedback Decision Trees in Content-Based Image Retrieval. In *IEEE Wkshp on Content-Based Access of Image and Video Libs*, 2000.

[14] T. Mensink, J. Verbeek, and G. Csurka. Learning Structured Prediction Models for Interactive Image Labeling. In *CVPR*, 2011.

[15] M. Naphade, J. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-Scale Concept Ontology for Multimedia. *IEEE Transactions on Multimedia*, 2006.

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

[17] J. C. Platt. Probabilistic Output for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances in Large Margin Classifiers*, 1999.

[18] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, 2001.

[19] Y. Rui, T. Huang, M. Ortega, and S. Mehrotra. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval. *IEEE Transactions on Circuits and Video Technology*, 1998.

[20] W. Scheirer, N. Kumar, P. Belhumeur, and T. Boult. Multi-Attribute Spaces: Calibration for Attribute Fusion and Similarity Search. In *CVPR*, 2012.

[21] A. Shrivastava, S. Singh, and A. Gupta. Constrained Semi-Supervised Learning Using Attributes and Comparative Attributes. In *ECCV*, 2012.

[22] B. Siddiquie, R. Feris, and L. Davis. Image Ranking and Retrieval Based on Multi-Attribute Queries. In *CVPR*, 2011.

[23] S. Tong and E. Chang. Support Vector Machine Active Learning for Image Retrieval. In *ACM Multimedia*, 2001.

[24] C. Zhang and T. Chen. An Active Learning Framework for Content Based Information Retrieval. *IEEE Trans on Multimedia*, 2002.

[25] H. Zhang, Z.-J. Zha, S. Yan, J. Bian, and T.-S. Chua. Attribute Feedback. In *ACM Multimedia*, 2012.

[26] X. Zhou and T. Huang. Relevance Feedback in Image Retrieval: A Comprehensive Review. *ACM Multimedia Systems*, 2003.