This CVPR2014 paper is the Open Access version, provided by the Computer Vision Foundation.
The authoritative version of this paper is available in IEEE Xplore.

Beyond Comparing Image Pairs: Setwise Active Learning for Relative Attributes

Lucy Liang and Kristen Grauman
University of Texas at Austin

Abstract

It is useful to automatically compare images based on their visual properties—to predict which image is brighter, more feminine, more blurry, etc. However, comparative models are inherently more costly to train than their classification counterparts. Manually labeling all pairwise comparisons is intractable, so which pairs should a human supervisor compare? We explore active learning strategies for training relative attribute ranking functions, with the goal of requesting human comparisons only where they are most informative. We introduce a novel criterion that requests a partial ordering for a set of examples that minimizes the total rank margin in attribute space, subject to a visual diversity constraint. The setwise criterion helps amortize effort by identifying mutually informative comparisons, and the diversity requirement safeguards against requests a human viewer will find ambiguous. We develop an efficient strategy to search for sets that meet this criterion. On three challenging datasets and experiments with "live" online annotators, the proposed method outperforms both traditional passive learning as well as existing active rank learning methods.

1. Introduction

While vision research has long focused on *categorizing* visual entities (e.g., recognizing objects in images, or activities in video), there is increasing interest in *comparing* them. For example, whereas the presence or absence of an attribute in an image may not be clear-cut, whether one image exhibits the attribute more or less than another may be more informative [27]. Similarly, while a user doing image search may have difficulty declaring certain images as entirely irrelevant, he may more easily decide whether one image is more or less relevant than another [13, 31, 14]. Recent work continues to discover new benefits of representing comparative visual properties [19, 32, 8, 21, 24, 17, 30, 2].

In such settings, methods to learn ranking functions are a natural fit. However, their training requirements take us into new territory, compared to familiar data collection. Training a classifier requires ground truth *labels* that hard-assign each instance to a category, and there are many existing category-labeled datasets and tools that make labeling efficient (e.g., ImageNet, Hollywood videos, etc.). In contrast, training a ranking function requires ground truth *comparisons* that relate one instance to another (e.g., person A is *smiling more* than person B; image X is *more relevant* than image Y), and thus far only modest amounts of comparative annotations are available.

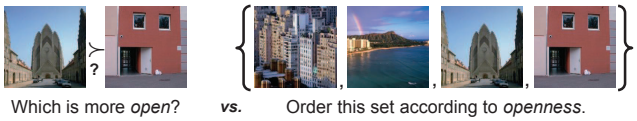


Figure 1. To learn relative attribute ranking functions, we propose an efficient active selection criterion that asks annotators to partially order a set of diverse yet informative images. Whereas a pairwise approach (left) gets just one bit of information, the setwise approach (right) amortizes annotator effort by getting (implicitly) all mutual comparisons.

How to best collect comparative image labels is not straightforward, in part due to immediate scaling issues. To make the problem concrete, suppose we have $15,000$ images to label. At 1 cent per image on Mechanical Turk, it would cost just $150 to label them all by category. In contrast, naively posting all pairs of comparisons on that same data would cost over 1 million dollars! Besides, intuitively, exhaustive pairwise comparisons should not be necessary to learn the concept, as some will be redundant.

Our goal is to leverage human supervision only where it is needed most when training relative attributes, such as *more/less bright*, *more/less feminine*, etc. To this end, we explore active learning for ranking functions. Active learning empowers the system to select those examples a human should label in order to most expedite learning. While its use for classification is fairly mature in both the learning and vision communities, it is much less studied for ranking.

Active rank learning presents three distinct technical challenges. First, hard comparisons for the system can also be hard for a human labeler due to their visual similarity. Second, restricting labeling tasks to solely paired comparisons can be wasteful; the human labeler spends time interpreting the attributes in two images, yet the system gets only

one bit of information in return (that is, which image has the property more than the other). Third, the quadratic number of possible comparisons poses a scalability challenge for any but the most simplistic criteria, since active selection typically entails scanning through all yet-unlabeled data to select the optimal request.

In light of these challenges, we explore a series of increasingly complex active selection criteria for learning to rank. We start with a pairwise margin-based criterion for ranking functions that selects pairs with high uncertainty. Then, we consider a setwise extension [37] that requests a partial order on multiple examples at once. Finally, we introduce a novel setwise criterion that both amortizes human perceptual effort and promotes diversity among selected images, thereby avoiding uninformative comparisons that may be too close for even humans to reliably distinguish. See Figure 1. In particular, our formulation seeks a set of examples that minimizes the mutual rank margin in attribute space, subject to a visual diversity constraint in the original image feature space. We show how to efficiently search for batches that meet this criterion.

We apply the methods to three challenging datasets. We demonstrate that with an active approach, a system can learn accurate relative attribute models with less human supervision. This in itself is a contribution, as no prior work examines active training of comparative visual models. Furthermore, we show that the proposed setwise strategy consistently outperforms the existing strategies, supporting our main novel technical contribution. We run results both in the standard offline setting, as well as in a "live" setting, where our approach pushes its active requests to Mechanical Turk workers and iteratively updates its model. The practical impact is significant: we reduce annotation costs by 39% compared to the status quo passive approach.

2. Related Work

Relative attributes Attributes are human-understandable properties reflecting textures (*spotted*), geometric properties (*boxy*), or parts (*has-legs*) [11, 20, 10]. By relaxing attributes to take on ordinal values, a relative attribute represents how images compare along some property (e.g., *smoother*, *less boxy*), and can be trained in a "learning to rank" framework [27, 8, 21]. Given a model for relative visual properties, new tasks become possible. In object recognition, category models can be trained with fewer examples ("He looks like Joe, but *chubbier*."), via transfer [27] or semi-supervised learning [30]. In image search, relevance feedback can explicitly refer to properties important to a user ("These shoes are *less formal* than the ones I want.") [17]. Whereas all prior work uses passive learning to train attribute models, we propose to actively learn them.

Collecting comparative image data Our active learning method asks annotators to order sets of images according to an attribute. At a high level, this relates to other interfaces requiring annotators to compare or contrast images. Researchers developing attribute lexicons ask humans to describe differences between sets of images to elicit plausible attributes [25, 28]. The "crowd kernel" method [32] builds a similarity matrix from crowdsourced data, and selects maximally informative triples of things for annotators to compare. It yields a fixed, human-created matrix capturing some (possibly non-describable) notion of visual similarity, and it does not generalize to new data. In contrast, we actively select comparisons on a describable property, so as to efficiently learn a predictive function that can estimate attribute strength in any new image.

Active learning for recognition and retrieval Active learning for recognition helps train a classifier with fewer labeled images (e.g., [16, 35, 38]), and can also incorporate attributes [3, 18, 2]. In image retrieval, active learning can identify images that should receive binary relevance feedback to reduce uncertainty [33, 29]. All of these methods request labels from users; none actively request visual comparisons. As discussed above, the challenges in active ranker learning are distinct from active classifier learning, and much less studied.

Diversity in active learning The need to inject "diversity" into active label selection criteria has been considered in classification work [4, 36, 29], to help ensure all parts of the feature space are explored. We propose a novel diversity bias for active *rank* learning. In contrast to prior work, here diversity also serves to avoid focusing only on comparisons that would be ambiguous to a human viewer.

Learning to rank The learning to rank problem has received much attention in the information retrieval and machine learning communities (e.g., [6, 15, 7, 22]). Many methods take a pairwise approach, in which constraints on a learning objective require satisfying comparisons for pairs of examples [6, 15]. Alternatively, a listwise approach defines a loss function in terms of ordered lists of instances [7]. We use a pairwise objective for training, and map setwise supervision into pairwise constraints. Most rank learning work addresses document retrieval, though applications to image retrieval are emerging [13, 31, 14]. In either case, their goal is to rank examples relevant to a user most highly, which is similar in spirit to classifying all relevant data confidently. In contrast, we want to actively learn relative visual properties, and the goal is to generalize to compare any novel pair. Like rank learning, some metric learning methods use relative comparisons for training [12], and could potentially also benefit from our ideas to focus human effort on useful comparisons. However, metrics are

less suited for attributes than rankers, since they can only report distances, not more/less decisions.

Active learning to rank Only a few prior methods exist for active rank learning, and none have been applied to visual data to our knowledge. Margin-based selection criteria seek pairs of instances whose estimated ranks are nearest under the current model [5, 37], while others seek examples expected to most influence the ranking function [9] or minimize expected loss [23]. We explore the suitability of margin-based criteria for attribute training, and we propose a new formulation that accounts for diversity.

3. Approach

We use active learning to efficiently gather comparative labeled data to train visual attribute models. We first describe the "learning to rank" approach we use to build the relative attribute models (Sec. 3.1). Then we overview the three active strategies we explore (Sec. 3.2). The first relies on a margin criterion to select pairs of images (Sec. 3.2.1). The second reasons about mutual margins between a set of images (Sec. 3.2.2). For the third, we propose a setwise criterion that promotes feature space exploration (Sec. 3.2.3).

3.1. Learning to rank visual attributes

Relative attributes, originally introduced in [27], compare images in terms of how strongly they exhibit a nameable visual property. Whereas categorical attributes use classifiers trained with labeled images, relative attributes use ranking functions trained with comparative labels.

Training objective We use a large-margin approach [15, 27] to model relative attributes, and briefly review it next. Given an attribute of interest (e.g., *fuzziness*), the method trains a ranking function r that will predict the relative strength of that attribute in an image. To learn r, it uses 1) a set of training images $I = \{i\}$, each of which is described by some image features $\boldsymbol{x}_i \in \Re^d$, together with 2) two sets of human-provided visual comparisons on those images. The first set $O = \{(i,j)\}$ consists of ordered pairs of images for which the first image i has the attribute more than the second image j. The second set $S = \{(i,j)\}$ consists of unordered pairs for which both images have the attribute to a similar extent. The ranking function takes the form[1]

$$r(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}. \tag{1}$$

Ideally, it should satisfy the maximum number of constraints specified by the training comparisons. That is, $\forall (i,j) \in O : \boldsymbol{w}^T \boldsymbol{x}_i > \boldsymbol{w}^T \boldsymbol{x}_j$, and $\forall (i,j) \in S : \boldsymbol{w}^T \boldsymbol{x}_i = \boldsymbol{w}^T \boldsymbol{x}_j$. While this is an NP hard problem, an approximate solution is [15, 27]:

$$\begin{aligned} \text{minimize} \quad & \left(\frac{1}{2}||\boldsymbol{w}||_2^2 + C\left(\sum \xi_{ij}^2 + \sum \gamma_{ij}^2\right)\right) \\ s.t. \quad & \boldsymbol{w}^T(\boldsymbol{x_i} - \boldsymbol{x_j}) \geq 1 - \xi_{ij}; \forall (i,j) \in O \\ & |\boldsymbol{w}^T(\boldsymbol{x_i} - \boldsymbol{x_j})| \leq \gamma_{ij}; \forall (i,j) \in S \\ & \xi_{ij} \geq 0; \gamma_{ij} \geq 0, \end{aligned} \tag{2}$$

where the constant C balances the regularizer and constraints. The learning objective is similar to the SVM classification problem, but on paired difference vectors. Basically, it aims to find the vector $\boldsymbol{w} \in \Re^d$ that will project data in such a way that 1) the orderings specified in the training set are satisfied, and 2) the margin between the nearest projected training points in O is maximal. See Figure 2.

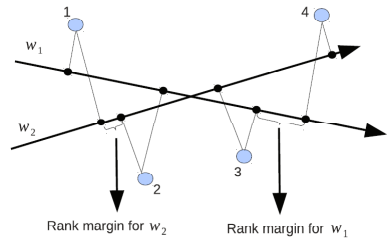


Figure 2. Here, two candidate vectors, $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ rank four points. $\boldsymbol{w}_1$ is the better candidate according to Eqn. 2 [15], because it yields the largest rank margin for the closest-ranked training pair.

We can apply the learned ranking function to compare new image pairs. Specifically, if $\boldsymbol{w}^T \boldsymbol{x}_p > \boldsymbol{w}^T \boldsymbol{x}_q$, we predict that image p has the attribute more strongly than image q. Note that a relative attribute predictor provides a 1D ordering of the image data; we will exploit this structure below when searching for sets of useful examples to compare.

Soliciting partial orders While the learning objective is expressed in terms of pairs, O and S can be deduced from any *partial ordering* of the images I in the training data with respect to attribute strength. To test our active setwise approach, we develop an intuitive interface to facilitate partial order requests on sets of images (where two or more images may be marked as equally strong). Rather than ask the annotator to assign a number to each image indicating its rank order, which can be tedious, we present a visual cascade. First, the user is shown all images in the set. Then, he must select all those that show the specified attribute most. The interface removes those image(s), then repeats the process with the remaining ones, until all images are accounted for. See Figure 3 (best on pdf).

Note that this cascaded interface obtains the exact same information as depicted in Figure 1 (right) using only mouse clicks. We find this is a relatively foolproof way to gather ordering information on multiple images, which is important when we do our live experiments with non-expert MTurk workers.

[1] The method is also kernelizable for non-linear ranking functions.

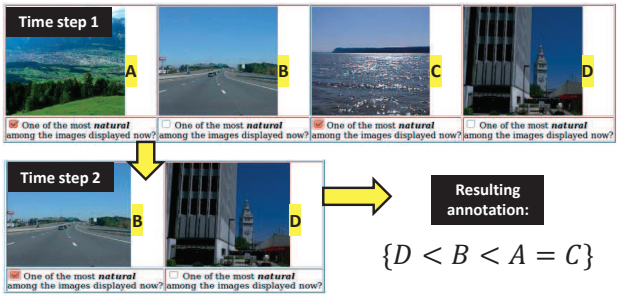


Figure 3. Our cascade user interface for requesting partial orders.

Aggregating annotator comparisons A strength of training with relative comparisons, as opposed to absolute ranks, is that annotators tend to be more consistent. For example, deciding if image i is *brighter* than image j is often easier than quantifying its absolute *brightness*. Nonetheless, some attributes may have different connotations to different observers, and some are less careful than others. Therefore, when our learning methods request new comparisons, we need to build in some resiliency to noisy annotator responses.

To this end, we use a simple but effective aggregation procedure. It accounts for the annotators' labels as well as their stated confidences. When giving an ordering, the annotator rates it as "very obvious", "somewhat obvious", or "subtle". We first assign a numerical rank to each image in the comparison.[2] For a pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ where the annotator finds image i has the attribute more than image j, we assign ranks 2 and 1, respectively. If the annotator says i and j are equal in the attribute, we assign ranks 1.5 to both i and j. (The scale of these constants is unimportant; they are just for aggregation.) Then, for each training image, we take a weighted average of the numerical ranks across all annotators, where the weight reflects annotator confidence. Specifically, we attribute twice the weight to "very obvious" comparisons as "somewhat obvious" ones, and three times the weight as "subtle" ones.

We take three further steps to eliminate outliers and improve robustness. First, if all annotators designate a comparison "subtle", we remove it as unreliable data. In the case of a partial order, we additionally gauge how consistent each annotator's numerical ranks are with the other annotators. Specifically, we use Kendall's τ rank correlation coefficient to compare each annotator's ranks to the average of all other annotators; if $\tau < 0.7$, we eliminate the annotator. We update the aggregated scores after any such removals. Finally, we apply mean shift clustering on the 1D rank scores to cluster those that are relatively close. Those that belong to the same cluster will form a similar pair. This accounts for fluctuations caused by uncertainty among annotators.

[2]For clarity, we describe it for pairs, but it generalizes to partial orders.

Figure 4. Which image in each pair exhibits more *diagonal plane*? Focusing on either extreme—low or high rank margins—can thwart active learning, requesting comparisons that are too hard or easy for both the human and learning algorithm.

After aggregation, we disregard the temporary numerical ranks, turning the (now more robust) orders back into comparative judgments.

3.2. Active learning to rank

We follow a pool-based active learning strategy to train relative attribute rankers. At each iteration, the system must examine a pool $\mathcal{P}$ of unlabeled images and predict what comparison will most benefit its current ranking functions. After it makes a selection, the comparison is posed to annotators, and their (aggregated) comparisons are used to augment the training sets O and S. Then, the learned attribute rankers are retrained, and the process repeats.

In the following, we explain a series of three increasingly complex active selection criteria that we investigate for active relative attribute learning.

3.2.1 Pairwise margin criterion

Intuitively, the large margin criterion in Eqn. 2 prefers confident orderings, in that it favors projections that keep the closest pair of training instances as far away as possible. Accordingly, a natural active selection criterion is to identify unlabeled pairs of images for which the rank margin is lowest. That is, the best pair to compare is:

$$(i^*, j^*) = \underset{i,j \in \mathcal{P}}{\operatorname{argmin}} \, |\boldsymbol{w}^T \boldsymbol{x}_i - \boldsymbol{w}^T \boldsymbol{x}_j|, \qquad (3)$$

where $\mathcal{P}$ denotes the set of unlabeled images, and $\boldsymbol{w}$ is the relative attribute ranking function trained with all data acquired so far. See Figure 5(a). This criterion is analogous to the well known "simple margin" criterion for SVM classifiers [34], which requests labels on examples close to the decision hyperplane. For ranking, the margin is instead the distance between ranks. And, rather than request a label for a single instance, the learner requests a *comparison* for a *pair* of instances.

The low margin selection criterion is sensible and can be effective in practice, but it has two potential weaknesses. First, it may select examples that are not only hard to distinguish for the machine, but also the human annotator. As a result, some low margin pairs can be wasted requests: human labelers either disagree on the correct ordering or simply label them as "equal", which will have little impact on

the learning objective. Reversing the criterion to request comparisons on *high* margin pairs would ensure more distinct images, but is prone to uninformative requests, since distant examples are often already captured by the ranker learned with minimal labeled pairs. See Figure 4. The second weakness is its restriction to requesting solely *pairwise* comparisons. The human labeler spends time interpreting the images' attributes, whose visual differences may be subtle. Yet, then the system gets only one bit of information in return—namely, which image exhibits the attribute more. Ideally, we would get more human insight for the "perceptual effort" invested.

3.2.2 Setwise margin criterion

In light of the latter shortcoming, we next consider the *setwise* margin criterion proposed in [37]. It selects a set of examples whose mutual margin distances are low. Specifically, the best set $\mathcal{S}^*$ is:

$$\mathcal{S}^* = \underset{\mathcal{S} \subseteq \mathcal{P}}{\operatorname{argmin}} \sum_{(i,j) \in \mathcal{S}} |\boldsymbol{w}^T \boldsymbol{x}_i - \boldsymbol{w}^T \boldsymbol{x}_j|, \tag{4}$$

where $|\mathcal{S}| = k$ is a parameter to the algorithm. The selected set should be useful, in that all respective pairs within it are currently ambiguous to the learned ranking function (i.e., close in rank). See Figure 5(b).

To efficiently optimize Eq. 4, the authors of [37] exploit the 1D ordering induced by the ranking function. First, all unlabeled instances in $\boldsymbol{x}_i \in \mathcal{P}$ are sorted by their rank values $r(\boldsymbol{x}_i)$ using the current model. Then, the cumulative margin of each contiguous set of k sorted items is evaluated in succession. That is, we start with a set of the k lowest ranked instances, and record their summed pairwise margin distances. Then, we repeatedly shift the lowest ranked instance out of the set and replace it with the next higher ranked instance not already in the set. At the same time, we incrementally compute the current set's cumulative margin, by subtracting and adding the paired margins for the lowest and highest instance, respectively. The operation requires only $O(|\mathcal{P}|)$ time.

Since this criterion chooses a set rather than a pair, we ask an annotator to provide a partial ordering. For comparative annotation tasks, requesting partial orders on small sets is appealing because it will amortize effort in examining the images. We will get substantial supervision from a partial order of k items—implicitly, all k-choose-2 comparisons are revealed—yet, with small enough batches of k, the mental load on the annotator remains modest.

To be concrete, we found that across all three datasets in our tests, the average time an annotator takes to compare two pairs is nearly identical to the time he takes to fully order a set of $k = 4$ examples, namely, 3.72 s for the 4-set, and 3.57 s for two pairs. Moreover, the time that would be required to *explicitly* compare 6 independent pairs (4 choose 2) is about 3 times what is required to do the full ordering of 4 (which *implicitly* relates all 6 pairs). Thus, the setwise selection and partial order interface stand to give us more value for annotator effort.

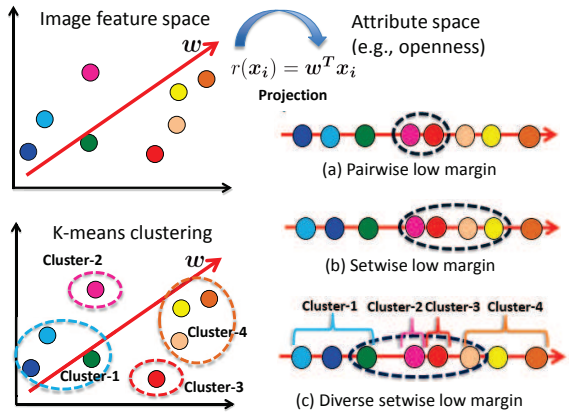


Figure 5. Overview of all three selection criteria. The rank function r projects the high-dimensional image descriptors to a 1D ordering that preserves the training attribute comparisons. (a) The pairwise low margin active selection method requests labels for those pairs with minimal rank margins. (b) The setwise low margin method generalizes that to select a set of instances whose mutual rank margins are low. (c) The proposed diverse setwise low margin further accounts for the diversity of the selected set *in image space*. Here, we see that the chosen set (dashed black ellipse) has not only low mutual rank margins, but is also composed of diverse examples spanning the K-means clusters in image feature space (bottom left). Best viewed in color.

3.2.3 Diverse setwise margin criterion

While the setwise criterion amortizes annotator effort more effectively than requesting a series of individual pairs, it still can suffer from the ambiguity issue described above. By definition, the mutually close set of examples may be hard for a human annotator to compare relatively.

Thus we introduce a new approach called the *diverse setwise low margin* (DSLM) criterion. Our goal is to select the image examples that minimize the setwise margin, subject to a visual diversity constraint. To capture diversity, we first cluster all the image descriptors $\boldsymbol{x}_i$ (e.g., GIST, color) in $\mathcal{P}$. This establishes the primary modes among the unlabeled examples. Let c_i denote the cluster to which image i belongs. Our selection objective is:

$$\mathcal{S}^* = \underset{\mathcal{S} \subseteq \mathcal{P}}{\operatorname{argmin}} \sum_{(i,j) \in \mathcal{S}} |\boldsymbol{w}^T \boldsymbol{x}_i - \boldsymbol{w}^T \boldsymbol{x}_j|, \quad s.t. \quad c_i \neq c_j, \forall i \neq j, \tag{5}$$

where again $|\mathcal{S}| = k$ is given. In other words, the most useful set is the one that has difficult examples that aren't

"too" difficult—they must each come from a different cluster. This balances *exploiting* the margin uncertainty with *exploring* the feature space. See Figure 5(c).

To form the clusters, we use K-means. The number of clusters will affect the selection in a predictable way. Small K values will emphasize diversity, permitting more high margin pairs (e.g., in the extreme, if $K = 1$, no pairs would be diverse enough). Big K values will emphasize uncertainty, permitting examples in the set that are relatively close. We discuss setting K in Sec. 4.

To optimize Eqn. 5, we propose a search strategy that builds on the technique outlined above. The idea is as follows. Only a strictly rank-contiguous set will minimize the total margin; yet there may not be a rank-contiguous set for which diversity holds. Thus, we scan contiguous sets in sequence, always maintaining the current best margin score. If our current best is not diverse, we perturb it using the next nearest sample until it is. The key to efficiency is to exploit the 1D ordering inherent in attribute ranks, even though the clusters are in the high-dimensional descriptor space.

More specifically, we first sort all unlabeled images by their attribute rank values. Then we start with a candidate batch consisting of the k lowest ranked instances in $\mathcal{P}$, and record its summed pairwise margin distances as the current best. Then, we begin shifting the batch's members as described previously. At each shift, if the current batch B does not reduce the current lowest pairwise margin found so far, we disregard it. If it does, *and* is also diverse ($c_i \neq c_j$), then we hold this solution as the current best, along with its pairwise margin value. If B reduces the best margin but is not diverse, we call a subroutine that adjusts the batch members so as to ensure diversity. This subroutine is based on a "worst offender" criterion. Specifically, we identify any instances in B belonging to the same cluster, and of those, pick the one whose total margin distance to the remaining instances is largest. We remove that offender from B, and replace it with the next instance in $\mathcal{P}$ that has a higher rank than the highest ranked item already in B. We iterate between 1) checking for a set that reduces the objective function, and 2) replacing offenders, until we find a batch that improves the best seen so far, or until we exceed the best margin value. See Supp File for pseudo-code.

By exploiting the 1D ordering inherent in attribute ranks, we can incrementally adjust candidate batches, and so the core loop run-time is linear in $|\mathcal{P}|$. If the subroutine is called to improve diversity for the contiguous set located at rank n, we have to (in the worst case) examine the $|\mathcal{P}| - n$ items with higher ranks, making the search bounded by $O(|\mathcal{P}|^2)$ time. In practice, however, the subroutine is rare and/or brief enough, that we observe run-times less than 18% slower than the linear time scan (0.109 s vs. 0.128 s on average, for $|\mathcal{P}| = 14,000$ images and $d = 990$). In comparison, if we attempt exhaustive search (with $|\mathcal{P}| = 50$), it requires 243 s; DSLM requires only 0.015 s yet produces the same selection.

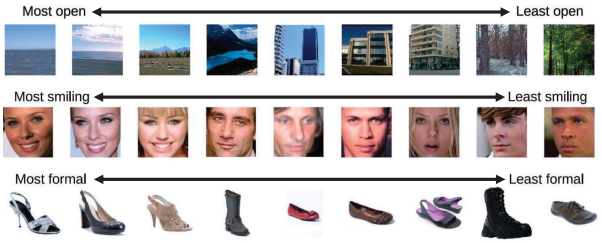


Figure 6. Example attribute spectra from OSR, PubFig, and Shoes

4. Results

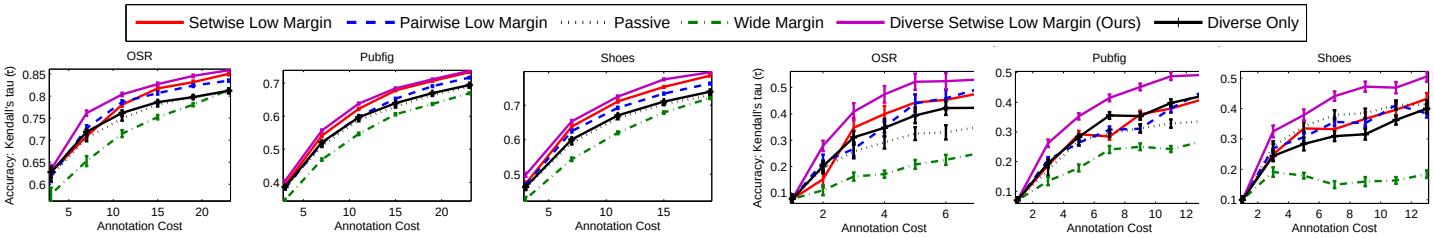
Experimental setup We use 3 public datasets: Outdoor Scene Recognition (OSR) [26], PubFig Faces [19] with attributes from [27], and Shoes [1] with attributes from [17]. They have 2,688, 772, and 14,658 images, respectively, and 6 to 11 attributes each, for a total of 27 attributes. See [1, 26, 19, 17] for more details. For the features $\boldsymbol{x}_i$, we concatenate GIST and LAB color histograms, following prior work [27].

We compare the 3 active learners defined in Sec. 3 with 3 baselines: 1) **passive**, which selects a k-set at random as in [27]; 2) **diverse only**, which selects samples based on the same diversity constraint as DSLM, but ignores margins; and 3) **wide margin**, which chooses pairs with the widest margins. These baselines help us identify whether differences in learning curves are due to margin, diversity, or both.

The diversity-based methods use $K = 10$ clusters in all results. We chose the value to roughly correspond to the number of object categories present in the datasets, which gives a coarse estimate of the data diversity. Preliminary trials show accuracy is not very sensitive to nearby K values, indicating this is a good prior. See Supp File.

To evaluate the quality of the learned ranking functions on the test set, we use Kendall's τ. At each active learning iteration, all methods select a set of $k = 4$ images for annotation. For the pairwise methods, we take the top 2 pairs according to their selection criterion. Recall that giving a partial order on 4 images requires the same time as comparing 2 independent pairs, meaning the pairwise and setwise methods incur equal human effort per iteration (cf. Sec. 3.2.2). We stress, *the costs per iteration are equalized for all methods*, so the learning curves below reflect accuracy vs. cost (i.e., human annotation time = iterations).

Offline results First we perform experiments in a "sandbox" offline, meaning we already have the true comparative labels, and we reveal them to the active learners when requested. Using publicly available human-generated pairwise ground truth and confidences [17, 27], we apply our



(a) Offline annotation collection (b) Live annotation collection

Figure 7. Learning curves summarizing results on all 3 datasets and 27 attributes, for (a) offline and (b) live settings. Best viewed in color.

rank aggregation procedure, and train one ranking function per attribute. We apply the learned functions to all dataset images, yielding ranks for each attribute. We set these aside as the target ranks. Figure 6 shows example spectra for one attribute per dataset.

For each dataset, we set aside 40 random images as a test set, 4 seed training images per attribute, and an unlabeled pool of 120 images. We run all 6 selection methods for 25 iterations[3], each time revealing the target comparisons on the selected images to the learner so it can update its r. We conduct the entire process 20 times with random splits.

Figure 7(a) shows the results. We display learning curves for all methods, averaged across all attributes per dataset, with standard error for the 20 trials. (See Supp File for per-attribute plots.) For clarity, all plots start at the 3rd iteration, since the methods are very similar in accuracy when they have only a few seed labeled sets. High and steep curves are best, since that means the concept is accurately learned with less human effort.

The active learners consistently outperform passive learning. This constitutes an advance for the state-of-the-art, since all prior relative attribute work trains the model with passive learning [27, 8, 21, 17, 2]. Diverse-only does as well as or better than passive for most attributes, highlighting the role diversity can play in active learning. All learners outperform the wide margin baseline, showing that naive pairwise diversity is inadequate.

Across the board, the proposed DSLM outperforms all other methods. It is stronger for 17 of the 27 attributes, and similar to some other active variant for all others (see Supp). It outperforms the existing setwise low margin method [37], showing the our proposed diversity formulation is critical for best results. It is also better than the diverse-only baseline, showing the need to balance exploration with margin uncertainty.

Live results Next we run the active learners in a more challenging "live" experiment. In this case, we let the methods select comparisons on data for which we have no annotations. While the offline tests let us run all methods more freely (literally!), this is exactly the real-world scenario. We use the interface in Figure 3 to collect the requested partial orders on Mechanical Turk. We get each request done by 5 workers and take the majority vote for the label more/less. After the jobs come back, the methods update their ranking functions, and then repeat. Due to expense, we run for fewer total iterations than the offline results. Otherwise, the setup is as described above.

Figure 7(b) shows the results averaged over all attributes per dataset. The outcomes are very much in line with the offline results, only our advantage compared to the baselines is noticeably *stronger*. Across all attributes, our method requires 39% fewer annotations to attain the same accuracy reached by the passive learner in the last iteration. This is a very encouraging result that demonstrates the practical impact of our idea. The absolute τ values are lower for live than for offline. This may be due to the MTurkers' disagreement on attributes' precise meaning, causing label inconsistencies. In fact, in the live setting, there may not exist a single function that can accommodate all annotated comparisons, whereas in the offline tests, a consistent global ranker defines the target ground truth.

Figure 8 shows an example illustrating why the margin-based learners are at a disadvantage. Looking at samples selected in the 2nd iteration, we see that images that are too similar-looking may be causing MTurkers difficulty. In comparison, our diversity-based method fares well.

Looking at individual attributes where the margin between active and passive is smallest, we find that an attribute with an ambiguous definition can cause problems. For example, for *bright-in-color*, some people see a shoe as brighter if colors are more vibrant (red, yellow); others see a shoe as brighter if it is shinier and glossy, regardless of the actual color (e.g., ranking a black shiny shoe higher than a red matte shoe). See Figure 9, top. This can impede active learning's impact, since a viable model is needed to do reasonable exploitation.

Another case where the active methods have less advantage is when an attribute is localized. Since our descriptors are global, this makes it difficult to isolate the relevant spatial region with few training examples, leading to weaker active choices at the onset. For example, the PubFig *smil-*

[3]Since Shoes has less ground truth data, we restrict its test set to 30 images and the number of iterations to 22.

Figure 8. Images selected by the two setwise active learners when learning *diagonal plane*. Boxes denote ambiguous pairs. Our diverse setwise method produces sets that are both informative to the system and easy for humans to compare.

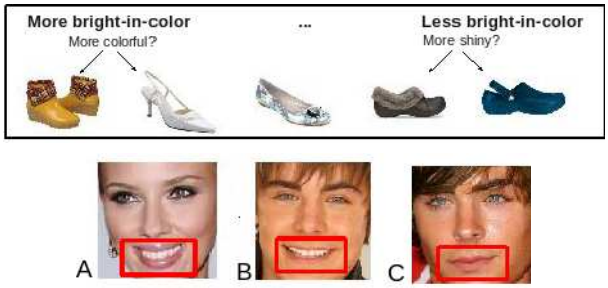


Figure 9. Difficult attributes: **Top**: Images sorted according to the learned attribute *bright in color*. When annotators disagree about an attribute meaning, active learning tends to have less advantage. **Bottom**: Global descriptors cause ambiguity for diversity-based active learners when the attribute is localized. While B and C have the same face, they have different degrees of *smiling*. While A and B have dissimilar faces, they have the same degree of *smiling*. (Red boxes for emphasis, but not available to the methods.)

ing attribute depends largely on the mouth region. Yet GIST and color histograms will not easily expose that region (GIST's spatial bins need not align with the mouth). The diversity-based learners can suffer especially from global descriptors, since they will tend to prefer globally different instances while missing the finer detail. For example, they may choose faces from different individuals, though not necessarily faces with mouths that look different. See Figure 9, bottom. Localized features more tailored to the attribute semantics may be interesting to explore.

5. Conclusion

This work takes a close look at active learning for relative attributes. We introduced a novel diverse-setwise selection strategy to account for the shortcomings of existing methods. Our results show the promise in focusing human attention on comparisons that are useful for discriminatively trained ranking functions. The live online experiments in particular strongly support the proposed method as a means to gather partial orders on images. We improve over not only the status quo for attribute learning (passive), but also prior active learning to rank formulations. In future work, we plan to investigate ways to account for attribute relationships and localized attributes during active learning.

Acknowledgements: We thank Adriana Kovashka and Jaechul Kim for helpful discussions, and the anonymous reviewers for their comments. This research is supported in part by ONR YIP.

References

[1] T. Berg, A. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010.
[2] A. Biswas and D. Parikh. Simultaneous active learning of classifirs & attributes via relative feedback. In *CVPR*, 2013.
[3] 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.
[4] K. Brinker. Incorporating Diversity in Active Learning with Support Vector Machines. In *ICML*, 2003.
[5] K. Brinker. Active learning of label ranking functions. In *ICML*, 2004.
[6] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML*, 2005.
[7] Z. Cao, T. Qin, T. Liu, M. Tsai, and H. Li. Learning to rank: From pairwise approach to listwise approach. In *ICML*, 2007.
[8] A. Datta, R. Feris, and D. Vaquero. Hierarchical ranking of facial attributes. In *Automatic Face and Gesture Recognition Workshops*, 2011.
[9] P. Donmez and J. Carbonell. Optimizing estimated loss reduction for active sampling in rank learning. In *ICML*, 2008.
[10] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing Objects by their Attributes. In *CVPR*, 2009.
[11] V. Ferrari and A. Zisserman. Learning Visual Attributes. In *NIPS*, 2007.
[12] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning Globally-Consistent Local Distance Functions for Image Retrieval and Classification. In *ICCV*, 2007.
[13] Y. Hu, M. Li, and N. Yu. Multiple-instance ranking: Learning to rank images for image retrieval. In *CVPR*, 2008.
[14] V. Jain and M. Varma. Learning to re-rank: Query-dependent image re-ranking using click data. In *WWW*, 2011.
[15] T. Joachims. Optimizing search engines with clickthrough data. In *KDD*, 2002.
[16] A. Joshi, F. Porikli, and N. Papanikolopoulos. Breaking the interactive bottleneck in multi-class classification with active selection and binary feedback. In *CVPR*, 2010.
[17] A. Kovashka, D. Parikh, and K. Grauman. WhittleSearch: Image search with relative attribute feedback. In *CVPR*, 2012.
[18] A. Kovashka, S. Vijayanarasimhan, and K. Grauman. Actively selecting annotations among objects and attributes. In *ICCV*, 2011.
[19] N. Kumar, P. Belhumeur, and S. Nayar. Facetracer: A Search Engine for Large Collections of Images with Faces. In *ECCV*, 2008.
[20] C. Lampert, H. Nickisch, and S. Harmeling. Learning to Detect Unseen Object Classes by Between-Class Attribute Transfer. In *CVPR*, 2009.
[21] S. Li, S. Shan, and X. Chen. Relative forest for attribute prediction. In *ACCV*, 2012.
[22] T. Liu. Learning to rank for information retrieval. *Fnd & Trnds in IR*, 2009.
[23] B. Long, O. Chapelle, Y. Zhang, Y. Chang, Z. Zheng, and B. Tseng. Active learning for ranking through expected loss optimization. In *SIGIR*, 2010.
[24] S. Ma, S. Sclaroff, and N. Ikizler-Cinbis. Unsupervised learning of discriminative relative visual attributes. In *ECCV Wksp on Parts and Attributes*, 2012.
[25] S. Maji. Discovering a lexicon of parts and attributes. In *ECCV Workshop on Parts and Attributes*, 2012.
[26] A. Oliva and A. Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *IJCV*, 42:145–175, 2001.
[27] D. Parikh and K. Grauman. Relative Attributes. In *ICCV*, 2011.
[28] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012.
[29] S. Rajaram, C. Dagli, N. Petrovic, and T. Huang. Diverse active ranking for multimedia search. In *CVPR*, 2007.
[30] A. Shrivastava, S. Singh, and A. Gupta. Constrained semi-supervised learning using attributes and comparative attributes. In *ECCV*, 2012.
[31] B. Siddiquie, R. S. Feris, and L. S. Davis. Image ranking and retrieval based on multi-attribute queries. In *CVPR*, 2011.
[32] O. Tamuz, C. Liu, S. Belongie, O. Shamir, and A. Kalai. Adaptively learning the crowd kernel. In *ICML*, 2011.
[33] X. Tian, D. Tao, X. Hua, and X. Wu. Active reranking for web image search. *IEEE Transactions on Image Processing*, 2010.
[34] S. Tong and D. Koller. Support Vector Machine Active Learning with Applications to Text Classification. In *ICML*, 2000.
[35] S. Vijayanarasimhan and K. Grauman. Large-Scale Live Active Learning: Training Object Detectors with Crawled Data and Crowds. In *CVPR*, 2011.
[36] Z. Xu, R. Akella, and Y. Zhang. Incorporating diversity and density in active learning for relevance feedback. In *ECIR*, 2007.
[37] H. Yu. SVM selective sampling for ranking with application to data retrieval. In *KDD*, 2005.
[38] L. Zhao, G. Sukthankar, and R. Sukthankar. Robust active learning using crowdsourced annotations for activity recognition. In *HCOMP*, 2011.