Diverse Active Ranking for Multimedia Search

Shyamsundar Rajaram
Hewlett-Packard Labs
1501 Page Mill Rd.
Palo Alto, CA 94304
shyamsundar.rajaram@hp.com

Charlie K. Dagli
Beckman Institute
University of Illinois
Urbana, IL 61801
dagli@ifp.uiuc.edu

Nemanja Petrovic
Google, Inc.
76 9th Avenue
New York, NY 10011
nemanja@ifp.uiuc.edu

Thomas S. Huang
Beckman Institute
University of Illinois
Urbana, IL 61801
huang@ifp.uiuc.edu

Abstract

Interactively learning from a small sample of unlabeled examples is an enormously challenging task, one that often arises in vision applications. Relevance feedback and more recently active learning are two standard techniques that have received much attention towards solving this interactive learning problem. How to best utilize the user's effort for labeling, however, remains unanswered. It has been shown in the past that labeling a diverse set of points is helpful, however, the notion of diversity has either been dependent on the learner used, or computationally expensive. In this paper, we intend to address these issues in the bipartite ranking setting. First, we introduce a scheme for picking the query set which will be labeled by an oracle so that it will aid us in learning the ranker in as few active learning rounds as possible. Secondly, we propose a fundamentally motivated, information theoretic view of diversity and its use in a fast, non-degenerate active learning-based relevance feedback setting. Finally, we report comparative testing and results in a real-time image retrieval setting.

1. Introduction

Consider the following scenario. Harry is looking for images that are similar to a certain target concept in a huge image dataset. Harry's time is expensive and therefore he cannot go over the entire set of images. Instead, he labels a small subset of the data as being either relevant or irrelevant to the target concept. He wonders if, based on this labeled set, someone could come up with an ordering of all the images in such a way that images at the top of the list

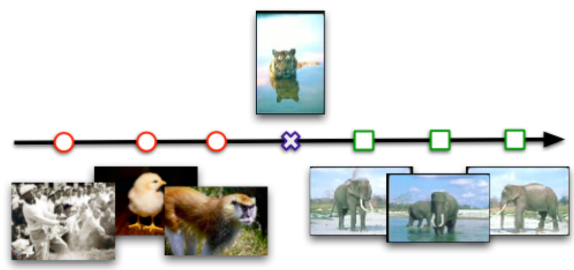

Figure 1. In this illustration, the queried concept is an elephant in a blue background. The plot shows the ranker output on a real line in which square marks indicate relevant instances and the circles indicate irrelevant instances. The cross mark refers to an unlabeled instance which falls between the current labeled set of relevant and irrelevant instances leading to confusion about its label.

are the ones that are relevant and the ones at the bottom are irrelevant.

The above problem can be posed as a learning problem. Assuming a feature based representation of the data, there exists an instance space $\mathcal{X}$ (which in our case is the space of all images) and there is a label space $\mathcal{Y}$ which is given by $\{0, 1\}$ where label 1 indicates the corresponding image is relevant to the target concept. Further, there exists an oracle (who in our case is Harry) who can label a subset of images. Several interesting questions arise at this point. What is our learning strategy? What form should the output of our learning algorithm take? What is our learning algorithm? What is the criterion for picking the most informative pool query that will be presented to the oracle for labeling? In this paper, we attempt to address the above questions.

Learning algorithm: The input to the learning algorithm is a set of training examples $S =$

$((x_1, y_1), \ldots, (x_M, y_M)) \in (\mathcal{X} \times \mathcal{Y})^M$. The learning algorithm could predict if the new instance is relevant or irrelevant and the goal of the algorithm would then be to learn from S a function $h : \mathcal{X} \rightarrow \mathcal{Y}$; given a new instance $x \in \mathcal{X}$, the algorithm would predict the label as $h(x)$. In the case where $\mathcal{Y}$ is $\{0, 1\}$, the learning algorithm is a standard binary classification algorithm. However, in our case, we do not need a labeling of the instances as relevant or irrelevant as in traditional content-based retrieval techniques. Instead, we desire an ordering of the instances in such a way that relevant instances are on the top of the list and the irrelevant ones are in the bottom. Such an ordering problem corresponds to the bipartite feedback ranking formulation introduced in [7] and analyzed further in [1, 5]. The goal of ranking is to obtain an ordered list of entities where order is determined by preference or choice. The preference is either hand crafted, or more desirably learned from the annotated data. Learning a ranker amounts to finding an axis in the feature space that data points are mapped to so that the absolute the axis the relative position reflects the desired preference. Absolute value of the projected example does not have any particular meaning and this feature distinguishes ranking from ordinal regression. Further, absolute rankings do not to have any particular meaning across disjunctive data sets. Ranking is solely based on the relative position of the one dimensional mapping. In this work, we adapt the RankBoost algorithm developed in [7] for minimizing the bipartite loss function.

Learning strategy: Active learning is a paradigm that proposes ways to incrementally learn from unlabeled data, provided the system has available to it an oracle, an entity which knows the correct labeling of all examples [4, 8]. Given an initial weak ranker or classifier, the oracle labels a set of points the systems deems to be most informative, the *pool query set*. The information provided from this labeling can then be used to update the system and this process can be repeated indefinitely to improve the accuracy of those points in the returned or *resultant set*.

This paradigm, in different contexts, does well to model the inherent learning scenario in a variety of fields, especially computer vision. The tiger image in Fig. 1, represents the most informative unlabeled instance given the current set of labeled instances, as its ranking appears between the current set of labeled instances.

In Harry's case, the interactive technique of *relevance feedback*([9]) is a special case of active learning for image searching problems. Traditional relevance feedback can be seen as a degenerate case of active learning as the set of top-k returned points serves both as the returned and pool query sets. Using a unique pool query set, however, has been shown to improve performance [13].

Diverse Informative Query Set: In each active learning round, we have to pick a subset of unlabeled instances, the pool query set, to be labeled by the oracle. We develop a scheme for picking the pool query set that is suited for the bipartite ranking framework so that it will aid us in learning the ranker in as few active learning rounds as possible.

In previously developed active learning strategies used for SVM classifiers[13], it was shown that the instance closest to the hyperplane reduces the version space the most in the separable case. In the case when $k : k > 1$ instances can be provided for labeling, the k instances that are closest to the hyperplane are presented to the oracle for labeling. Recently, Brinker [2] introduced the notion of diversity which was incorporated while choosing the subset to be labeled. The key idea in this framework is to pick instances that are diverse with respect to each other while sill being close to the decision hyperplane.

In this work, we extend the angular diversity measure suggested in [2] and further, introduce a parameter-free information theoretic diversity measure based on an empirical measure of entropy. We then use this measure in a unique active ranking algorithm.

The rest of the paper is organized as follows. Sec. 2 elaborates on the bipartite ranking framework and includes a brief review of the RankBoost algorithm. In Sec. 3, we describe our active learning strategy for picking unlabeled instances for labeling. We present two strategies for selecting diverse and representative unlabeled instances in Sec. 3.1 for the ranking framework. We apply our active ranking framework to an image retrieval problem elaborated on in Sec. 4. Finally, we conclude in Sec. 5 with directions for future research.

2. Bipartite Ranking

The input to the learning algorithm is a set of training examples $S = ((x_1, y_1), \ldots, (x_M, y_M)) \in (\mathcal{X} \times \mathcal{Y})^M$ where $\mathcal{X} = \mathbb{R}^n$ and $\mathcal{Y} = \{0, 1\}$. Formally, we require that the learning algorithm learn from S a real-valued ranking function $f : \mathcal{X} \rightarrow \mathbb{R}$ that assigns scores to instances and thereby induces an ordering over $\mathcal{X}$: an instance $x \in \mathcal{X}$ is ranked higher by f than an instance $x' \in \mathcal{X}$ if $f(x) > f(x')$, and lower if $f(x) < f(x')$. In the bipartite ranking setting [7], the quality of a ranking function is measured as follows,

$$\widehat{R}(f; S) = \frac{1}{n_0 n_1} \sum_{\{i:y_i=0\}} \sum_{\{j:y_j=1\}} \mathbf{I}_{\{f(x_j)<=f(x_i)\}} \quad (1)$$

where $n_l = |\{i : y_i = l\}|$ and $\mathbf{I}_{\{.\}}$ denotes the indicator variable whose value is one if its argument is true and zero otherwise. The bipartite ranking error effectively counts an error each time a relevant instance (label 1) is ranked lower by f than an irrelevant instance (label 0), represented by $\mathbf{I}_{\{f(x_j)<=f(x_i)\}}$.

In [7], Freund et al. introduced a boosting style algorithm called RankBoost for solving the ranking problem.

RankBoost

1 : Initialize $D_0(x_i, x_j) = c\max\{0, \Phi(x_i, x_j)\},\ \forall(i,j)$

2 : **for** $t = 1, ..., T$

3 : Obtain weak ranker $h_t : \mathcal{X} \to \mathbb{R}$ using D_t

4 : Choose $\alpha_t \in \mathbb{R}$

5 : Set $D_{t+1}(x_i, x_j) =$

$$\frac{D_t(x_i, x_j)\exp(\alpha_t(h_t(x_i) - h_t(x_j)))}{Z_t}$$

6 : Obtain final ranker $H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$

Figure 2. The RankBoost Algorithm. The value Z_t is a normalization constant so that each D_t is a distribution. Additionally, the optimal α_t in each round is chosen with respect to the current ranking error. For a more detailed treatment, we refer the reader to [7].

In this work, we use the efficient version of RankBoost [7] which we briefly review next.

2.1. RankBoost

RankBoost is a powerful algorithm that combines weak ranking functions with respect to some initial feedback information that encodes pair-wise preferences between instances in $\mathcal{X}$. At each round of boosting, the algorithm minimizes the weighted number of instances that are misordered. Pairs on which we've made mistakes (with respect to the weaker ranker chosen for that round) are given a higher importance weight for correct ordering in the next round. The final ranking is a linear combination of weak rankers.

Formally, the initial feedback information is encoded by the function $\Phi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Given any two instances x_i and x_j, $\Phi(x_i, x_j) > 0$ means x_i is preferred to x_j, $\Phi(x_i, x_j) < 0$ implies the opposite, and a value of 0 means there is no preference. Furthermore, $|\Phi(x_i, x_j)|$ represents the confidence of the preference.

From this function, we can define the initial distribution $D_0(x_i, x_j) = c\max\{0, \Phi(x_i, x_j)\}$ where c is a normalizing constant so the sum over all instances is 1. (Since $\Phi(x_i, x_j) = -\Phi(x_j, x_i), \forall(i, j)$, no information is lost due to the max operator). During boosting, weak rankers of the form $h_t : \mathcal{X} \to \mathbb{R}$, where t is the boosting round, are used to successively update distributions $D_t(x_i, x_j)$. Difficult-to-learn pairs are given a higher importance rating for the next round. The RankBoost algorithm is given in Fig. 2.

There exists a more efficient version of this algorithm in the bipartite (and in general k-partite) case, where the ranking is defined only among groups of instances from the instance space. Since the interactive information retrieval task is inherently bi-partite (relevant vs. irrevelant), we use this efficient implementation in this work. For more details, the reader is referred to [7].

3. Active Ranking

A general active learning algorithm chooses both a resultant and pool-query set to present to the user at each step. In this section, we present a heuristic strategy for picking the pool query set.

In the case of SVM active learning, the candidates for the pool query set correspond to the unlabeled points which lie in the version space. In a query refinement algorithm, one can choose from a large number of points in the neighborhood of the query centroid. To the best of our knowledge, choosing a pool-query set for an active ranking scenario remains an unexplored area.

In general, the pool-query set is chosen as those instances that are hardest to handle, or most confusing for the current classifier/ranker. We introduce a quantity called the *clarity index* for each unlabeled instance in order to represent this idea.

Let $T = ((x_1, y_1), \ldots, (x_N, y_N))$ be the complete set of labeled instances obtained from previous active learning rounds and f be the current ranker. For every unlabeled instance x_i^u we define *relevance loss* $RL(x_i^u, f, T)$ as,

$$RL(x_i^u, f, T) := \frac{1}{n_0}|\{j : f(x_i^u) <= f(x_j), y_j = 0\}| \quad (2)$$

and *irrelevance loss* $IL(x_i^u, f, T)$ as,

$$IL(x_i^u, f, T) := \frac{1}{n_1}|\{j : f(x_i^u) > f(x_j), y_j = 1\}|. \quad (3)$$

Relevance loss can be interpreted as the bipartite ranking loss $\widehat{R}(f; S_R)$ (defined in Eqn. 1) where the set S_R is given by $((x_i^u, 1), T_I)$ where T_I represents the irrelevant instances present in the set T. Irrelevance clarity is given by the bipartite ranking loss $\widehat{R}(f; S_I)$ (defined in Eqn. 1) where the set S_I is given by $((x_i^u, 0), T_R)$ where T_R represents the relevant instances present in the set T.

By definition of the bipartite ranking loss, a good ranking function is expected to have low relevance loss for relevant instances and low irrelevance loss for irrelevant instances. We can now define the *clarity index* of an unlabeled instance x_i^u with respect to a ranking function f and labeled set T as,

$$CI(x_i^u, f, T) := |RL(x_i^u, f, T) - IL(x_i^u, f, T)|. \quad (4)$$

Clearly, the clarity index orders the instances in terms of their difficulty for the ranking function. The higher the clarity index, the easier it is to classify an instance. A simple illustration is presented in Fig. 3.

Figure 3. Green squares and red circles represent the ranking function evaluated at the relevant and irrelevant instances respectively. The blue cross indicates the ranking function evaluated on an unlabeled instance. It is fairly clear that the top most case is the easiest, followed by the bottom most case and the middle one is the hardest. Relevance loss = 1; Irrelevance loss = 0; Clarity index = 1 (left). Relevance loss = 0; Irrelevance loss = 0; Clarity index = 0 (center). Relevance loss = 0; Irrelevance loss = 2/3; Clarity index = 2/3 (right). The difficulty in ranking is captured by the clarity index values.

We evaluate the clarity index for every unlabeled instance and the instances with the L smallest clarity index values form the candidate set C for the pool query set.

3.1. Diversity

When using an active learning-based system, the user is typically asked to label examples which are quite similar to one another, often as a result of examples clustering in the same neighborhood of the feature space. In a small-sample setting, especially when both the user's and system's effort is at a premium, it makes more sense for the user to label a *diverse* set of points for each pool-query rather than many similar points which are, in comparison, much less informative. This idea has been exploited to great effect in a broad range of fields. However, there have been only a handful of techniques which have addressed this issue in the active learning domain.

In the traditional relevance feedback scenario, NEC's PicHunter [6] cast diversity for image retrieval as matter of *exploration* versus *explsoitation*. Utilizing Bayesian relevance feedback techniques, they ask users to label images with low-posterior probability in addition to those with high probability. This notion of diversity relies on probabilistic modeling for its calculation, however, which tends to limit its general application due mainly to computational constraints. The CLUE system in [3] also realizes that database images tend to be semantically clustered in the vicinity of query images. To this end, they propose a local-neighborhood based clustering approach to more efficiently present diverse information for labeling by the user. This clustering must occur for every round of feedback, however, and the computational load of doing so may in some cases outweigh the improved retrieval results.

3.1.1 Angular Diversity

One of the first works to incorporate diversity sampling was [2] where the notion of angular diversity was investigated for support vector machines (SVMs). In choosing an angularly diverse set of unlabeled instances from the SVM version space for oracle labeling, the active learning system is able to reduce the size of the version space faster and with much less effort from the user. In the SVM setting, the instance that is closest to the center of the version space has been shown to reduce the size of the version space the most (Version space volume reduction in SVMs corresponds to exponential reduction in generalization error [10]). Thus in [2], Brinker optimized a convex combination of the angular diversity and the distance to the center of the version space. We will adapt the same idea for the ranking framework below.

Given the set of unlabeled instances in the candidate set C, $\{\mathbf{x}_i\}_{i=1}^{L}$ ($\mathcal{X} = \mathbb{R}^n$), the angular diversity between any two instances $\mathbf{x}_i$ and $\mathbf{x}_j$ can be defined as

$$\cos(\angle(\mathbf{x}_i, \mathbf{x}_j)) = \left| \frac{\langle (\mathbf{x}_i - \mathbf{x}_c), (\mathbf{x}_j - \mathbf{x}_c) \rangle}{\|\mathbf{x}_i - \mathbf{x}_c\| \|\mathbf{x}_j - \mathbf{x}_c\|} \right|. \quad (5)$$

where $\mathbf{x}_c$ is the mean of the relevant instances[1]. A diverse set is then constructed in a greedy fashion by minimizing,

$$\max_{l \in QS} \cos(\angle(\mathbf{x}_l, \mathbf{x}_j)) \quad (6)$$

where QS constitute the already chosen query set of manually ranked instances and $\mathbf{x}_j$ is the new diverse instance chosen from the candidate set C.

In addition to pool query points being angularly diverse, they must also have low clarity index in the ranking case. These two factors must be balanced in order to ensure that the pool query set is diverse and hard for the current ranker. To this end, a convex combination of these terms is incorporated into the cost function F:

$$F(i) = \alpha CI(\mathbf{x}_i, f, T)) + (1-\alpha) \max_{l \in QS} (\cos\angle(\mathbf{x}_l, \mathbf{x}_i)) \quad (7)$$

where $\mathbf{x}_i \in C$ and α denotes the mixing parameter. As mentioned earlier, the diverse set is chosen in a greedy sequential manner as in [2].

This notion of diversity was originally motivated by the learning algorithm used, namely SVMs, and may not be a general measure of diversity in an active learning scenario. From an algorithmic viewpoint, the mixing parameter α requires empirical adjustment and is application specific. In the next section, we improve on these initial notions of diversity by motivating and presenting a more general, parameter-free definition of diversity in the active learning setting.

[1] In image retrieval, relevant instances are surrounded by the irrelevant instances and hence the angles are computed with respect to the mean of the relevant instances.

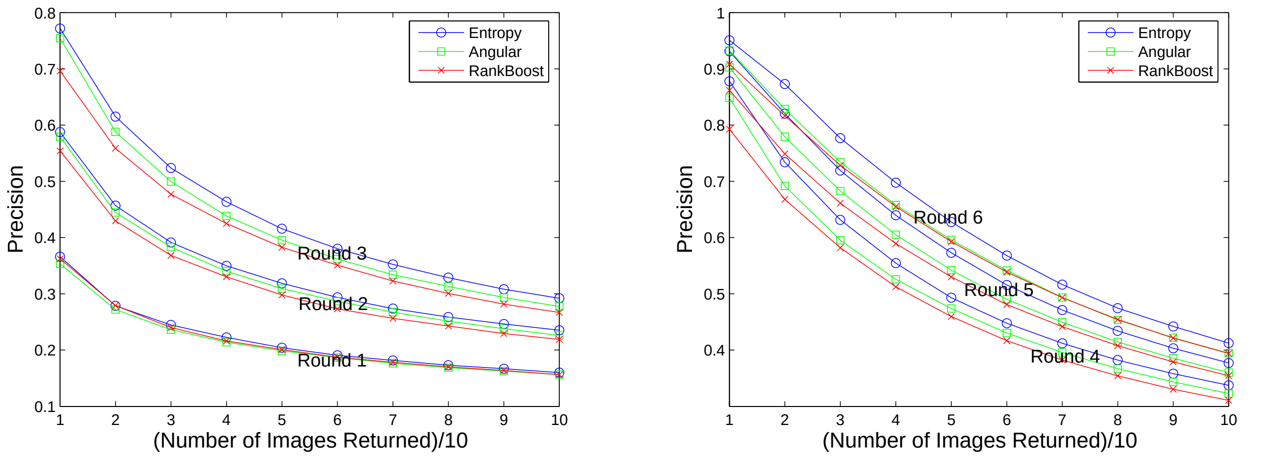

Figure 4. Comparison of the Number of returned images vs. Precision in the three cases namely, RankBoost without diversity, RankBoost with angular diversity($\alpha = 0.5$) and RankBoost with entropy diversity over 6 rounds.

3.1.2 Entropy based Diversity Measures

We first define a basic diversity measure, based on Shannon's entropy [11]. Associating high entropy with diversity is intuitively attractive as entropy is essentially a measure of randomness in a variable. In our problem, we desire to pick a diverse pool query set that is representative of the candidate set C. For any continuous random variable, $\mathbf{x}$, entropy is defined as

$$h(\mathbf{x}) = -\int p(\mathbf{x}) \log\left(p(\mathbf{x})\right) = -E\left[\log\left(p(\mathbf{x})\right)\right] \quad (8)$$

where $E[\cdot]$ is the expectation operator. Calculating entropy in practice involves density estimation, as the underlying distribution is not known. We adopt a non-parametric scheme using Parzen windows for estimating the density as shown below,

$$p(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{W}(\mathbf{x}, \mathbf{x}_i) \quad (9)$$

where $\mathcal{W}$ is the Parzen window function. For all testing and experimentation, we use isotropic Gaussian as the Parzen window function and the variance is estimated using leave one out cross validation [12]. However, evaluating the entropy integral is very difficult and hence we use a sample approximation of the integral as follows,

$$h(\mathbf{x}) \approx -\frac{1}{N}\sum_{j=1}^{N} \log\left(p(\mathbf{x}_j)\right) \quad (10)$$

which by the Law of Large Numbers approaches the true mean in the limit. Using more sophisticated numerical integration techniques, of course, would lead to better estimates of the empirical expectation.

As pointed out in [14], two samples are enough to estimate the entropy of a density. The first sample is used to estimate the density and the second sample is used to estimate the entropy. To reiterate, in our case, we have to choose K instances which are diverse and representative of the L instances in candidate set C. In other words, the problem reduces to identifying K points which are used to estimate the density in such a way that the entropy estimated over the remaining $L - K$ points is maximized. We note that picking the optimal query set is computationally infeasible and hence, we perform the optimization in a greedy fashion by starting off with the unlabeled instance that has the least clarity index. Subsequently, we add instances (one at a time) to the pool query set such that their addition to the query set maximizes the entropy computed with respect to the instances in the $C \setminus QS$ where QS represents the current pool query set. This information theoretic diversity measure is intuitively well motivated and, more importantly is parameter free and independent of the learning algorithm.

4. Image Retrieval

4.1. Data Set and Image Features

To explore the practical performance of our diverse active ranking system, extensive experimentation was performed using a 5000 image subset of the COREL image database. To appropriately model the small sample learning scenario, only 1400 images were used for target sets. The remaining points were unlabeled. The target set consisted of 13 unique query concepts from images of both natural and man-made scenes. In addition, the classes were also chosen to make discrimination challenging. For example, images of tigers, elephants and apes are in a similar background: the jungle. An example set of these images can be

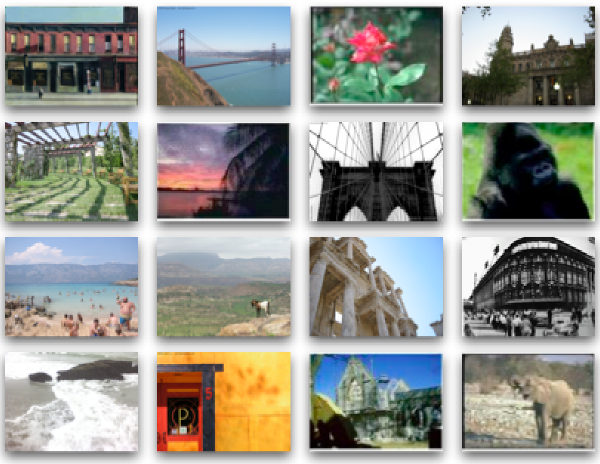

Figure 5. A diverse sample of images used in experimentation.

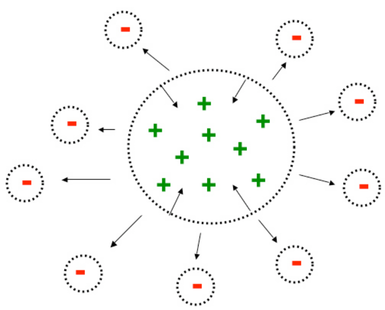

Figure 6. The BDA Problem

seen in Fig. 5.

The first, second and third moments in each channel of the HSV color space, first and second wavelet sub-band moments at three levels of decomposition and a Waterfilling algorithm were used for color, texture and shape features respectively. In total, a 47-dimensional feature vector was extracted from each image.

4.2. Biased Discriminant Analysis

Biased Discriminant Analysis (BDA) [15] was developed to address the inherent problem in retrieval systems which try to cluster all negative examples together. This does not make intuitive sense because negative examples can come from many different classes and many different parts of the feature space. BDA points out that relevance feedback is a one-to-many class problem. The idea is that positive examples are derived from one class only while negative examples may come from multiple classes. The goal, as illustrated in Fig. 6, is to closely cluster the positive examples while pushing away the negative ones.

BDA can be characterized by the following objective function,

$$\underset{\mathbf{W}}{\operatorname{argmax}} \left| \frac{\mathbf{W}^T \mathbf{S}_{PN} \mathbf{W}}{\mathbf{W}^T \mathbf{S}_P \mathbf{W}} \right| \tag{11}$$

where $\mathbf{S}_P$ is the intra-class-scatter matrix for positive examples and $\mathbf{S}_{PN}$ is the inter-class scatter matrix between positive and negative examples.

The solution reduces to solving the generalized eigenvalue problem for the Rayleigh quotient in Eqn. 11. The columns of the W matrix correspond to the generalized eigenvectors corresponding to the largest eigenvalues.

$$\mathbf{S}_{PN}\mathbf{w}_i = \lambda_i \mathbf{S}_W \mathbf{w}_i \tag{12}$$

Once these are found, the discriminating transformation matrix is computed as

$$\mathbf{A} = \mathbf{\Phi}\Lambda^{1/2} \tag{13}$$

where $\mathbf{\Lambda}$ is the diagonal eigenvalue matrix and $\mathbf{\Phi}$ is the corresponding eigenvector matrix. The distance between two image features then becomes:

$$\mathrm{d}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{A} (\mathbf{x}_i - \mathbf{x}_j) \tag{14}$$

In the image retrieval framework, we use the ranking obtained from the Euclidean distance computed with respect to the mean of the relevant instances in the BDA-transformed feature space as the weak ranking. As mentioned above, there are three different sets of image features (shape, texture and color). In every RankBoost iteration, the optimal BDA matrix (with respect to the current distribution over the labeled instances) is estimated in the three feature sets and the one which yields the least bipartite loss is chosen for that RankBoost iteration.

4.3. Experimental Details

In the experiments, the user (oracle) is initially presented with 20 randomly chosen images which they label as being either relevant or irrelevant to their query concept. With this initial labeling, the system derives its initial ranking function via RankBoost with BDA as the weak learner. It returns to the user both the K similar images and the pool-query set of images to label for active ranking based on the entropic diversity criterion. For every subsequent round of interaction, the labeling of the pool-query set in the previous round is added to the labeled data set for RankBoost to train on in the current round. For each target set, we conducted 25 random feedback sessions of 6 rounds each. In total, there were 325 user-guided sessions, with 1950 total rounds of feedback for each method tested. Twenty images were chosen as the size of the pool-query set to reflect a reasonable number of images a human operator could label at one time. Results were averaged across all 325 tests.

4.4. Results

To evaluate the performance of our diverse active ranking framework, comparative testing with other ranking based algorithms were conducted. We present comparison between our entropic diversity active ranking model, angular diversity active ranking model (with $\alpha = (0.0, 0.25, 0.5, 0.75)$, random active ranking model (pool query set is chosen at random) and normal active ranking model (pool query set includes the lowest K clarity index instances). The RankBoost algorithm developed in [7] was used with BDA as the weak learner system as elaborated earlier in Sec. 4.2. Experiments were conducted across six feedback rounds for returned image set sizes of between 10 and 100 images. The same batch of 325 user-guided tests using pool-query size of 20 were conducted. The results of these tests are summarized below.

We performed experiments with different α values as mentioned above, but for clarity we present the results for the best parameter setting. From Fig. 4, we observe that our algorithm for diverse entropy ranking consistently performs better than other techniques. One can observe that utilizing an *active* ranking paradigm improves performance, as both the entropic and angular diversity results indicate. The curves for random sampling were incomparably poor, and are not presented for figure clarity.

From Fig. 7, we clearly note that the entropic diversity framework achieves results approximately $40 - 120$ %, $8 - 10$ % and $4 - 6\%$ over random active ranking, normal active ranking and active ranking with angular diversity respectively. More interestingly, these results point toward an interesting empirical property of this entropic diversity active ranking algorithm, namely it's tendency towards improvement at lower rounds and lower numbers of returned images. This has practical significance in a information retrieval scenario. Users tend not to spend much time giving feedback or exploring many pages of returns to find what they are looking for. The low-end performance bias of this diverse active ranking system helps users find what they are looking for quickly and without scanning multiple pages of results.

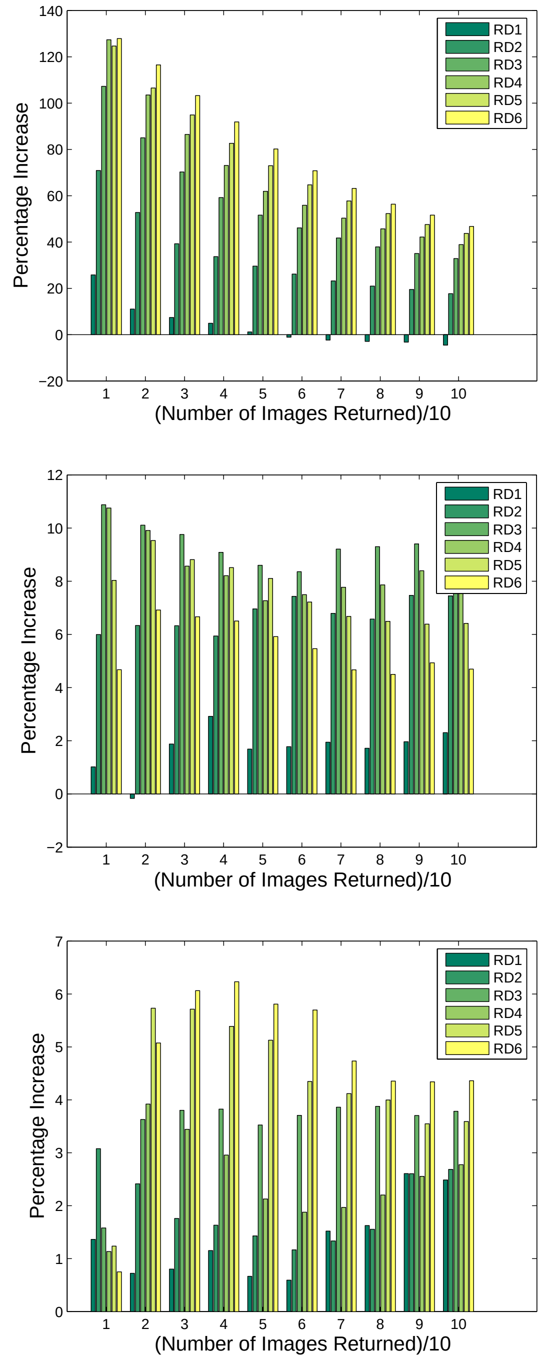

Figure 7. Comparison of the percentage increase in precision obtained when using the entropic diversity active ranking system over (top) random active ranking, (center) normal active ranking and (bottom) active ranking with angular diversity($\alpha = 0.5$) over 6 rounds

5. Conclusion and Future Work

In this work, we have proposed a scheme for performing active learning in the bipartite ranking setting. We have introduced a fundamentally motivated, information theoretic view of diversity and incorporated it into a non-degenerate, query-point refinement scheme for relevance feedback. Our results support the parameter-free entropic diversity's viability as a useful tool for pool-query selection in active learning.

In the future, we plan to investigate the theoretical issues involved in using our active learning strategy in the bipartite ranking framework and analyze the effect of diversity in the active learning setting. We are also working on applying these ideas to text and video retrieval.

References

[1] S. Agarwal, T. Graepel, R. Herbrich, S. Har-Peled, and D. Roth. Generalization bounds for the area under the ROC curve. *Journal of Machine Learning Research*, 6:393–425, 2005.

[2] K. Brinker. Active learning of label ranking functions. In *Proceedings of the 21st International Conference on Machine Learning*, pages 129–136, 2004.

[3] Y. Chen, J. Wang, and R. Krovetz. Clue: Cluster-based retrieval of images by unsupervised learning. *IEEE Transactions on Image Processing*, 14(8):1187–1201, 2005.

[4] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 705–712. The MIT Press, 1995.

[5] C. Cortes and M. Mohri. Auc optimization vs. error rate minimization. In S. Thrun, L. Saul, and B. Scholkopf, editors, *Advances in Neural Information Processing Systems*, volume 16. The MIT Press, 2004.

[6] I. J. Cox, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. Pichunter: Bayesian relevance feedback for image retrieval. In *in International Conference on Pattern Recognition*, 1996.

[7] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. In J. W. Shavlik, editor, *Proceedings of ICML-98, 15th International Conference on Machine Learning*, pages 170–178, Madison, US, 1998. Morgan Kaufmann Publishers, San Francisco, US.

[8] N. Roy and A. McCallum. Toward optimal active learning through monte carlo estimation of error reduction. In *International Conference on Machine Learning*, 2001.

[9] Y. Rui and T. Huang. Relevance feedback techniques in image retrieval. In *Principles of Visual Information Retrieval*, 2001.

[10] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. *Computational Learning Theory*, pages 287–294, 1992.

[11] C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:623–656, 1948.

[12] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. CRC Press, 1986.

[13] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2001.

[14] P. Viola, N. N. Schraudolph, and T. J. Sejnowski. Empirical entropy manipulation for real-world problems. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 851–857. The MIT Press, 1996.

[15] X. Zhou and T. S. Huang. Small sample learning during multimedia retrieval using biasmap. In *IEEE Conference Computer Vision and Pattern Recognition*, 2001.