Active Learning and its Scalability for Image Retrieval

Edward Y. Chang and Wei-Cheng Lai
University of California, Santa Barbara & VIMA Technologies

Abstract

Active learning has been shown to be a viable tool for learning complex, subjective query concepts with a small number of training instances. In this work, we compare four active-learning algorithms and study the best sample-selection strategies. We also discuss two scalability issues of active learning: scalability in dataset size, and scalability in concept complexity. To address these challenges, we suggest future directions that research might take.

1. Introduction

Employing active learning together with kernel methods to perform relevance feedback [7] significantly improves image-retrieval accuracy over the traditional IR methods. This breakthrough performance is achieved by two complementary forces. First, active learning finds the most informative examples to solicit user feedback, and hence a target query-concept can be quickly learned. Second, kernel methods (working with e.g., SVMs and BPMs) can effectively model a non-linear query concept, which traditional distance functions like the weighted Minkowski function simply cannot [6].

In this paper, we first present four active learning strategies and compare their performance. We then discuss scalability of active learning in concept complexity. For concept-complexity analysis, we examine the effect of concept scarcity, diversity, and isolation on the learning performance. We conclude this paper with a discussion on future directions that active-learning research might take.

2. Active Learning Strategies

The essential idea of active learning, working together with SVMs, is to find the most uncertain unlabeled instances between the positive and negative support vectors to query the user to refine the hyperplane [7]. In what follows, we present four active learning strategies for selecting unlabeled instances in the margin: speculative, simple, angle-diversity, and error-reduction.

2.1. Speculative Algorithm

We first propose a speculative procedure, which recursively generates samples by speculating about user feedback. The speculative procedure is computationally intensive. We use it as a yardstick to measure how well the other active-learning strategies perform. The algorithm starts by finding one most informative sample (the closest unlabeled instance to the hyperplane). It then speculates upon the two possible labels of the sample, and generates two more samples, one based on the positive speculation and one on the negative speculation. The algorithm speculates recursively, generating a binary tree of samples. Figure 1 presents the speculative algorithm. Steps 6 and 8 of the algorithm speculate the pool-query to be positive and negative, respectively, and recursively call the speculative procedure to select the next samples. The speculative procedure terminates after at least \( h \) samples have been generated.

2.2. Simple Active Algorithm

We previously proposed algorithm simple [7], which chooses \( h \) unlabeled instances closest to the separating hyperplane (between the relevant and the irrelevant instances in the feature space) to solicit user feedback. Based on the labeled pool \( L \), the algorithm first trains a classifier \( f \). The binary classifier \( f \) is then applied to the unlabeled pool \( U \) to compute each unlabeled instance’s distance to the separating hyperplane. The \( h \) unlabeled instances closest to the hyperplane are chosen as the next batch of samples for conducting pool-queries. The main idea of simple is that the \( h \) instances closest to the hyperplane are the most ambiguous ones with respect to \( f \) trained on \( L \). Algorithm simple attempts to achieve
2.3. Angle-Diversity Algorithm

Previously, we have pointed out that the selected samples need to be diversified [7]. One can incorporate a diversity metric into sample selection. The main idea (described in Step 2 of Figure 2) is to select a collection of samples close to the classification hyperplane, and at the same time, maintain their diversity. The diversity of samples is measured by the angles between the samples. Given an example $x_i$, its normal vector is equal to $\Phi(x_i)$. The angle between two hyperplanes $h_i$ and $h_j$, corresponding to instances $x_i$ and $x_j$, can be written in terms of the kernel operator $K$:

$$\cos(\angle(h_i, h_j)) = \frac{|\Phi(x_i) \cdot \Phi(x_j)|}{||\Phi(x_i)|| ||\Phi(x_j)||} = \frac{|K(x_i, x_j)|}{\sqrt{K(x_i, x_i)K(x_j, x_j)}}$$

The angle-diversity algorithm [2] starts with an initial hyperplane $h_1$ trained by the given labeled set $L$. Then, for each unlabeled instance $x$, it computes its distance to the classification hyperplane $h_1$. The angle between the unlabeled instance $x_i$ and the current sample set $S$ is defined as the maximal angle from instance $x_j$ to any instance $x_s$ in set $S$. This angle measures how diverse the resulting sample set $S$ would be, if instance $x_j$ were to be chosen as a sample.

Algorithm angle-diversity introduces parameter $\lambda$ to balance two components: the distance to the classification hyperplane and the diversity of angles among samples. Incorporating the trade-off factor, the final score for the unlabeled instance $x_i$ can be written as

$$\lambda \cdot |f(x_i)| + (1 - \lambda) \cdot \max_{x_j \in S} \frac{|K(x_i, x_j)|}{\sqrt{K(x_i, x_i)K(x_j, x_j)}}$$

where function $f$ computes the distance to the hyperplane, function $K$ is the kernel operator, and $S$ the training set. After that, the algorithm selects the unlabeled instance that enjoys the smallest score in $U$ as the sample. The algorithm repeats the above steps $h$ times to select $h$ samples. In practice, with trade-off parameter $\lambda$ set at 0.5, the work of [2] shows that the algorithm achieves good performance.

2.4. Error-Reduction Algorithm

Arriving from another perspective, Roy and McCallum [5] proposed an active learning algorithm that attempts to reduce the expected error on future test examples. In other words, their approach aims to reduce future generalization error. Since the true error on future examples cannot be known in advance, Roy and McCallum proposed a method estimating future error. Suppose we are given a labeled set $L$ and an unlabeled set $U = \{x_1 \ldots x_n\}$ where each $x_i$ is a vector. The distribution $P(x)$ of the image vectors is assumed to be i.i.d. In addition, each image $x_i$ is associated with a label $y_i \in \{-1, 1\}$ according to some unknown conditional distribution $P(y|x)$. The classifier trained by the labeled set $L$ can estimate an output distribution $P_L(y|x)$ for a given input $x$. Then the expected error of the classifier can be written as

$$E[Error] = \int \text{Loss}(P(y|x), P_L(y|x))P(x)dx$$

The function Loss is some loss function employed to measure the difference between the true distribution, $P(y|x)$, and its estimation, $P_L(y|x)$. One popular loss function is the log loss function which is defined as follows:

$$Loss(P(y|x), P_L(y|x)) = \sum_{y \in \{-1, 1\}} P(y|x) \log(P_L(y|x))$$

The algorithm proposed by Roy and McCallum selects a query, $x^*$, that causes minimal error. In other words, the algorithm includes $x^*$ in its sample set if $E[Error]_{P_{L \cup \{x^*\}}}$ is smaller than $E[Error]_{P_{L \cup \{x\}}}$ for any other instance $x$. Figure 3 summarizes the algorithm. Given the training pool $L$, the algorithm first computes the current classifier’s posterior, $P_L(y|x)$ in the loop of step 2. For each unlabeled image $x$ with each possible label $y$, the algorithm then adds the pair $(x, y)$ to the training set, re-trains the classifier with the enlarged training set, and computes the expected log loss. The algorithm then computes the expected log loss caused by adding $(x, y)$ to the training data. For the details of the algorithm, please refer to [5].

3. Scalability Issues

When dealing with a large dataset consisting of a large number of potential query concepts, active learning faces two scalability challenges. scalability in dataset size and scalability in concept complexity. The first scalability issue is an obvious one. When the size of the unlabeled pool $U$ is large, it is computationally prohibitive to scan the entire unlabeled pool to select samples. We need an effective indexing method to select samples without involving the entire $U$. Due to the space limitations, we do not present our indexing method in this paper. Please refer to [3] for details.

The second scalability issue that active learning confronts is concept complexity. For measuring concept complexity, we use three measures: diversity, scarcity, and isolation.

- **Diversity.** Diversity characterizes the distribution of a concept in the input space. A diversified concept tends to spread out in the input space (that formed by the perceptual features), and can be challenging to learn.
- **Scarcity.** We use hit-rate to characterize scarcity, defined as the percentage of images in the dataset matching a query concept.
3.2. Concept Scarcity

The hit-rate is a dataset-dependent measure. A very general concept may be scarce (having a low hit-rate) simply because few matching instances exist in the dataset, and a very specific concept may have abundant matching instances. Figure 5 reports the top-20 precision given hit-rate on the 50K dataset, after eight iterations of active learning. The figure shows that for concepts that have more than 2% matching images, their precision is good in general (above 45%). But when a concept is

---

1 We use the L1 distance function to measure distance in a 144-dimensional space where each the value of each feature ranges between zero and one.
scarce, its precision can be poor, though sometimes the performance can be surprisingly good. The results reveal one important design hint: that a learning algorithm needs to boost the hit-rate intelligently.

3.3. Concept Isolation

Isolation measures the degree of separation of a concept from the others. If a concept mingles with some other concepts in the input space, its isolation is low, or poor. We estimate isolation using the average difference between intra-concept distance and inter-concept distance, over inter-concept distance. The isolation measure of a concept can be between zero and one, where one means a concept is well isolated, and zero means a concept overlaps spatially with others.

![Figure 6. Top-20 Precision vs. Isolation.](image)

Figure 6 shows the top-20 precision given concept isolation for all 107 concepts of the 50k dataset. The correlation between the query performance and isolation is a moderate 0.6514. Isolation is a good metric to characterize concept complexity.

4. Experiments and Discussion

We conducted experiments to compare the performance of the four active learning strategies. For empirical evaluation of our learning methods, we used a 107-category, 50k-image dataset collected from the Corel Image CDs. In the following experiments, we report the precision of the top-20 retrievals to measure performance. We ran each experiment twenty times and report the average precision.

The first strategy selects random images from the dataset as samples for user feedback. The rest of the sample algorithms have been described in Section 2. We conduct experiments to compare these five sampling strategies in terms of three factors: (1) the top-k retrieval accuracy, and (2) execution times. In the experiments, we fixed the number of sample images at sixteen.

Figure 7 reports the results for the large, 107-category dataset. For the 50k dataset, it is too expensive to scan the entire dataset to perform sample selection and retrieval. Therefore, we conducted sampling and retrieval approximately through our indexing structure. The figure shows that the angle diversity algorithm performs the best among all active-learning algorithms. The angle diversity algorithm performs as well, and even better in some interactions, as the speculative algorithm, which is supposed to achieve nearly optimal performance. This result confirms that the samples should be diverse, as well as semantically uncertain (near the hyperplane). Our experiments also revealed that the fastest sampling algorithm is the random method, followed by angle diversity, simple active, speculative, and then error-reduction. Thus, angle diversity algorithm is the ideal choice in terms of both effectiveness and efficiency.

5. Conclusions

We have shown in this work the effectiveness of using active learning in learning complex, subjective query concepts with minimal information. We showed that the best sample-selection strategy is to select ones that are most uncertain and diversified with respect to the current concept boundary. We also discussed two important scalability issues of active learning: scalability in concept complexity and scalability in dataset size. When a concept has rare matches in the dataset and is not well isolated from other concepts, learnability of the concept can suffer. We deem that these scalability problems must be addressed so that active learning can be deployed in practical scenarios. Please refer to [4] for our detailed approaches to address these problems.

References