

TOWARDS OPTIMAL QUERY DESIGN FOR RELEVANCE FEEDBACK IN IMAGE RETRIEVAL

Jingyu Cui, Changshui Zhang

State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Automation, Tsinghua University, Beijing 100084, P.R.China

ABSTRACT

We analyze the sub-optimality of traditional greedy active learning based relevance feedback methods in image retrieval, and propose a novel active learning approach to query labels of multiple images together, which minimize the needed round of feedbacks and achieve satisfactory result in a near optimal manner. Our experiments on real image retrieval demonstrate that our solution can yield comparable precision/recall rate by significantly less relevance feedbacks.

Index Terms— Active learning, relevance feedback

1. INTRODUCTION

1.1. Overview

How to relate low level visual features to high level semantic concepts in multimedia retrieval problems has long been a hot research topic [1]. Most efforts that have been made in multimedia retrieval focus on bridging the “semantic gap” between low-level features and human perceptions [2]. Short term relevance feedback was introduced as a main break through [3, 2], which presents several images to the users and asks them to label whether the images are “relevant” or “irrelevant”. The algorithm improves its performance based on the labeled data obtained.

The performance of relevance feedback highly depends on the training data, especially on its ability to represent the query concept. However, it is unwise to ask the user to label too many images, thus it is crucial for a relevance feedback algorithm to select fewer images for user to label while obtaining more information.

Recently, many active learning methods [4, 5] have been introduced to the image retrieval community [6]. By trying to optimize a certain objective function, these methods suffer from three disadvantages: The objective functions of these methods are always difficult to optimize. Greedy method is commonly unavoidable, but far from satisfactory, since they tend to be trapped into local minima and fail to find the target concept. Moreover, active learning methods need to be generalized to be able to query multiple queries at one time to fit in the commonly accepted relevance feedback scenario, which is difficult for traditional algorithms. Besides, high computational complexity is also one of the barriers which keep most of the algorithms away from being used in real relevance feedback processes.

This paper proposes to improve the relevance feedback by multiple queries active learning, and provides a novel solution to design optimal query mechanisms to leverage user’s effort most efficiently. We can provide user the most informative examples to label in acceptable computational complexity, thereby enhance the user experience by getting better retrieval results in fewer rounds of feedback.

1.2. Previous Work

One of the early applications of active learning algorithms to image retrieval is the one based on support vector machine (SVM), as proposed in [6]. It analyzes the structure of the version space of SVM, and seeks the samples that are closest to the classifier hyperplane and queries their labels. If more than one sample can be queried in one batch, it uses a greedy method called *Simple margin* to query images which are closest to the SVM hyperplane [7].

The main problem of these approaches is sub-optimality. They assume that the SVM solution is at the center of the version space, which is not always true, as discussed in [8]. Also, if all images in a batch are chosen to be the closest to the SVM hyperplane, they can be highly redundant and lack of variety. In the discussion of [9], it is demonstrated that active learners based on greedy optimization tend to be trapped into local minima so badly that they can rarely discover the structure of the data if no sample from that region is presented in the initial training stage.

To solve the sub-optimality issue, there are basically three categories of approaches.

Some researchers try to balance the exploration and exploitation ability of active learners [10], or incorporate diversity when selecting the samples [11] to attack the sub-optimality problem. However, these algorithms are mostly based on heuristic intuition, which adds another term representing “exploration” or “diversity” to the original objective, then get the solution by optimizing this new criterion. The parameter to balance the new term and the original objective function is difficult to choose for different databases. The sub-optimality of the possible solution of the new objective function still remains.

Besides balancing approach, Steven et al. [12] propose the approximation approach to approximate the original cost function with a submodular one, in which case greedy optimization have a guaranteed performance. However, since the objective function is modified, and no theoretical guarantee of the closeness of this approximation is provided, the performance of this approach is also not assured.

The third approach is the direct approach, represented by the work in [13], which tries to estimate the reduction of error rate of Naive Bayesian classifier after acquisition of the label of a sample, and select the sample whose label will most reduce the error rate. For many other popular and effective classifiers, such as Support Vector Machine (SVM), whose error rate is difficult to predict, this algorithm is not applicable. Query by committee (QBC) [14] is another approach to directly reduce the size of the version space. It samples two classifiers in the version space, and queries the label of the sample if the two classifiers disagree. This method is made applicable by sampling methods in convex hull [15]. Since it still selects one example each time, it cannot be easily generalized to batch mode which is needed in image retrieval.

1.3. Our Approach

We formulate relevance feedback problem in a general framework which enables batch querying of labels for the need of relevance feedback in image retrieval. Compared to balancing approaches [11, 10], our approach solve the problem systematically rather than heuristically combining two terms. Compared to greedy [6], approximation [12], and direct approaches [14, 13] above, we naturally solve the problem in batch mode with solution nearer to the optimal.

The proposed framework can be summarized as: Given image set D including labeled images D_L and unlabeled images D_U , denote k as the number of queries, find an optimal set Q^* in all possible sets of unlabeled images $\{Q|Q \subseteq D_U, |Q| = k\}$ so that after querying the labels of each image in Q and train classifier f on new labeled set $D'_L = D_L \cup Q$ and unlabeled set $D'_U = D_U \setminus Q$, a certain objective function L is minimized:

$$Q^* = \arg \min_{Q \subseteq D_U, |Q|=k} L [D, D'_L, D'_U, f(D'_L, D'_U)] \quad (1)$$

Previous greedy optimization based algorithms [6, 7, 14] are special cases of this framework when k is constrained to 1. The solution space when $k = 2, 3, 4, \dots$ contains the whole solution space when $k = 1$. With less constraint and larger solution space, this framework is guaranteed to provide result no worse than that of the greedy algorithm, and is more likely to get the global minima.

2. OPTIMAL RELEVANCE FEEDBACK DESIGN

We use SVM as our classifier f in Equation (1). f is a linear classifier in the feature space generated by kernel Φ and best separates the mapped data $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ in feature space \mathcal{F} in a largest margin manner. f can be equivalently represented by its coefficient ω . We neglect the bias weight b for simplicity, and assume the optimal hyperplane passes through the origin in \mathcal{F} . In cases when b is needed, it is easy to alter the kernel or input space to accommodate [6]. A SVM can be denoted as: $f(x) = \omega \cdot \Phi(x)$ with the weight $\omega = \sum_{i \in D_L} \alpha_i \Phi(x_i)$. The summation is over all training samples, with α_i of the non-support-vector samples equal to zero.

We use the size of the SVM version space [16] as objective function L in Equation (1). Since the size of the version space is proportional to the uncertainty of the classifier [6], by minimizing L , we are actually trying to narrow down the classifier to the optimal one.

The version space of SVM is the set of classifiers f or equivalently their coefficients ω that are consistent with all the training samples in D_L , and can be defined as:

$$\mathcal{V}(D_L) = \{ \omega | y_i f(x_i) > 0, \forall x_i \in D_L, \|\omega\|_2 = 1 \} \quad (2)$$

where y_i is the label for a labeled sample x_i .

We propose to directly optimize this objective, which results in a minimum expected size of version space, thus the hypothesis can converge to the target concept in an optimal, or near optimal speed.

2.1. Expected Reduction of Version Space

Assume that the target hypothesis ω^* is in the hypothesis space. At a certain step, the version space is \mathcal{V} with size $|\mathcal{V}|$. After labeling k samples in Q^* , the version space becomes \mathcal{V}' . These k samples corresponds to k hyperplanes in the version space, and may cut \mathcal{V} into J pieces V_i , with $\bigcup_i V_i = \mathcal{V}$, $V_i \cap V_j = \emptyset$.

After labels of these k samples are given, one and only one of the sub-pieces of version space will be consistent with all the labeled

data. Suppose this sub-piece is \mathcal{V}_j , the size of the version space will be reduced to $|\mathcal{V}_j|$.

Suppose hypotheses in version space \mathcal{V} follow uniform distribution p_H , the probability that ω^* lies in \mathcal{V}_j will be $p_j = \int_{\mathcal{V}_j} p_H d\omega$.

Since $|\mathcal{V}_j| = p_j |\mathcal{V}|$, the expected size of the version space will be:

$$E [|\mathcal{V}'|] = E_j [|\mathcal{V}_j|] = \sum_{j=1}^J p_j |\mathcal{V}_j| = \sum_{j=1}^J p_j^2 |\mathcal{V}|.$$

The proportion of expected size of the version space to the original size after labeling k samples is $R = \frac{E[|\mathcal{V}'|]}{|\mathcal{V}|} = \sum_{j=1}^J p_j^2$.

Generally speaking, since $\sum_{j=1}^J p_j = 1$, minimizing R is equivalent to making all p_j as equal as possible, and making J as large as possible. Similar conclusion has been made in [7], but since explicitly calculating p_j and J is infeasible, the author did not figure out an efficient way of minimizing R directly.

We propose to minimize R by directly estimating the size of the version space. Suppose we have a query set $Q = \{x_1^q, \dots, x_k^q\}$ and newly labeled all the k samples in it. m classifiers $C = \{C_1, \dots, C_m\}$ are also uniformly sampled from version space $\mathcal{V}(D'_L)$. $C_i(x_j^q)$, $i = 1, \dots, m$, $j = 1, \dots, k$ takes value from $\{0, 1\}$.

We represent $C_i = [C_i(x_1^q), \dots, C_i(x_k^q)]^T$ with $c_i(Q)$, or simply $c_i = \sum_{j=1}^k C_i(x_j^q) 2^{j-1}$. It is clear that there is a bijective mapping

between $C_i(Q)$ and c_i , and they can be regarded equivalently. C is further divided into non-overlapping sets by $S_j = \{C_i | c_i = j\}$, $j = 0, \dots, 2^k - 1$ and $\bigcup_i S_i = C$, $S_i \cap S_j = \emptyset$. Each S_j corresponds to one possible piece of version space divided by samples in Q , and $\hat{p}_j = \frac{|S_j|}{m}$ is an unbiased estimator of p_j , with its variance $\text{var}(\hat{p}_j) = \frac{1}{m} p_j (1 - p_j)$ decreases inverse proportionally to increasing m . After getting \hat{p}_j , we can estimate R with:

$$\hat{R}(Q) = \sum_{j=1}^J \hat{p}_j^2 \quad (3)$$

2.2. Sampling in Version Space

It is impracticable to directly sample in the high dimensional feature space \mathcal{F} . However, we make use of the representor theory and Kernel PCA [17] to reduce the computational burden. Suppose we have labeled data D_L and unlabeled data D_U . A subset $Q_c \in D_U$, namely candidate set, is generated as candidates for labeling. A certain $Q \in Q_c$ will be added to D_L to form new labeled data $D'_L = D_L \cup Q$, from which the new SVM can be trained.

The version space is convex because it is the intersection of $|D'_L|$ hyperplanes in feature space \mathcal{F} . According to representor theory, the hypothesis of SVM, ω , lies in the space spanned by samples in D'_L : $\omega = \sum_{i=1}^{|D'_L|} \alpha_i \Phi(x_i)$. As a result, we can actually sample hypotheses in a space with much lower dimensionality ($|D'_L|$) than the dimension of \mathcal{F} .

Moreover, we do PCA in this $|D'_L|$ -d space expanded by $\Phi(x_i)$, which is actually KPCA, to capture most of the variance in the version space. Since the principal components approximately expand the version space, the kernel matrix K can be approximated as:

$$K = UU^T \quad (4)$$

where U is a $|D'_L| \times d_k$ matrix with $U^T U = I$, d_k is the dimension that preserves sufficient variance of \mathcal{V} .

With U as orthogonal coordinates of \mathcal{V} trained on D'_L , sampling in this convex hull with constraints in Equation 2 can be easily carried out by common sampling methods, including Gibbs sampling, Hit-and-run or Billiard Playing[18]. Note that results in [15] is a special case of our method when size of Q is constrained to 1.

2.3. Multiple Queries Active Learning

Now we can evaluate the expected reduction of version space given a new set of labeled data Q , what is left is to select the optimal Q^* from all possible subsets of D_U to reduce the size of \mathcal{V} most. This combinatorial optimization problem is hard to solve by brute force.

We use two strategies to drive the computational burden down:

1. Sampling. We can trade a little bit optimality for tremendous increase in speed. Despite of enormous amount of combinations, if we only expect the queried Q to be good enough, i.e., is top $\theta\%$ of all the combinations with confidence level $\delta\%$, the only necessity is to sample N combinations, return the best of the them as the final result. N can be calculated as:

$$N = \frac{\ln(1 - \delta\%)}{\ln(1 - \theta\%)} \quad (5)$$

For example, if we want the result to be top 5% with confidence level 99%, we only need to try $\ln(1 - 0.99) / \ln(1 - 0.05) \approx 90$ combinations and return best of them, which is practical in real applications.

2. Candidate Set. Label information of different samples may contribute differently to the classifier. For SVM, it is only the support vectors that make contribution. If we can identify support vectors beforehand, our active learner will achieve comparable performance with SVM having all the data labeled as its training set.

Support vectors are samples with the smallest margin. Our candidate set can be formed according to this prior knowledge, and we reasonably assume that unlabeled samples with smaller margin are more likely to become support vectors, and have higher probability of reducing the size of version space.

In order to balance exploration and exploitation, samples that are not "explored" should also be considered valuable, i.e., if some dense regions of unlabeled samples are far from all the labeled data, we might also need to put them into the candidate set.

Base on the heuristic considerations above, we give all the unlabeled data in D_U a weight score $s(x)$ to evaluate different expected contribution of unlabeled data. $s(x) = \min_{x_i \in D_L} d(x, x_i) - \alpha |m(x)|$ and transform it into a probability measure:

$$p(x) = \frac{e^{s(x)}}{\sum_{i=1}^{|D_U|} e^{s(x)}} \quad (6)$$

where $d(x, x_i)$ is the distance between x and x_i , $m(x)$ represents margin of x with respect to current SVM classifier, α is a parameter to control the contribution of two terms in the score $s(x)$.

Note that by this criterion, we favor the sample points that either have small margin $|m(x)|$ (higher probability of being support vectors) or are far from all the labeled samples with large $\min_{x_i \in D_L} d(x, x_i)$ (possible representative to new unexplored data region).

Here we get Algorithm 1 to form the candidate set Q_c .

With the reduced set Q_c , we consider samples in a smaller set Q_c rather than D_U , the computational burden can be greatly reduced. Our Multiple Queries Active Learning (*MQActive*) algorithm as a general framework is presented in Algorithm 2. Compared to

Algorithm 1 Candidate Query Set Generation

- 1: **Input:** Labeled data D_L , unlabeled data D_U , SVM classifier f , candidate set size k_c , hyper parameter α .
 - 2: Initialize $Q_c = \emptyset$;
 - 3: For each sample x_i in D_U , calculate $p(x_i)$ according to Equation 6;
 - 4: Pose prior distribution over D_U with $p(x_i)$;
 - 5: **while** $|Q_c| < k$ **do**
 - 6: Sample x from D_U according to distribution p ;
 - 7: $Q_c = Q_c \cup \{x\}$;
 - 8: **end while**
 - 9: **Output:** return Q_c .
-

traditional query process, our algorithm provides much more probability that the algorithm converges to the global optimal solution.

Algorithm 2 Multiple Queries Active Learning (*MQActive*)

- 1: **Input:** Labeled data D_L , unlabeled data D_U , SVM classifier f , query batch size k , size of candidate set k_c , hyper parameter δ , θ , m , α .
 - 2: **while** User is not satisfied **do**
 - 3: Get query candidate set $Q_c \in D_U$ using Algorithm 1;
 - 4: Sample m hypotheses $C = \{C_1, \dots, C_m\}$ in \mathcal{V} based on Equation 2 and 4;
 - 5: Calculate $c_i(x_j)$ for all $c_i \in C$ and $x_j \in Q_c$;
 - 6: Calculate N using δ and θ according to Equation 5;
 - 7: Sample N subsets $Q_i \subset Q_c$, $|Q_i| = k$, $i = 1, \dots, N$;
 - 8: Get $Q^* = \arg \max_{Q_i} \hat{R}(Q_i)$ by Equation 3, $|Q^*| = k$;
 - 9: Query labels of k samples in Q^* ;
 - 10: $D_L = D_L \cup Q^*$, $D_U = D_U \setminus Q^*$, retrain f ;
 - 11: **end while**
 - 12: **Output:** Return proper samples according to f .
-

2.4. Computational Complexity

Random and *SVMActive* have smallest computational complexity. Our algorithm is slower than those two, but much faster than *KQBC* and *Batch*. We only need to do sampling once for a batch of query, thus is much faster than the approach used in *KQBC*, where only one sample can be acquired through one sampling process. The time complexity of our proposal is approximately $O(m)$, while *Batch* has complexity of $O(kn^2)$, thus lacks of scalability. Here $n = |D|$, and other denotations are defined above.

3. EXPERIMENT

To evaluate the method Multiple Queries Active learning (*MQActive*) proposed in this paper, we setup experiments on both toy and real data sets, and compare with the results of *Random Sample (Random)*, *SVMActive* [6], *Kernel Query by Committee (KQBC)* [15], and *Batch Mode Active Learning (Batch)* [12].

Firstly, we test the algorithm on typical toy data set *Checkerboard* (Figure 1(a)). We use RBF kernel for SVM, and the parameter is fixed as $\sigma = 0.05$. Batch size is set as $k = 4$. Other parameters are set as $k_c = 200$, $\delta\% = 99\%$, $\theta\% = 5\%$, $m = 500$, $\alpha = 1$. Average accuracy over 100 runs with standard error are shown in Figure 1(b), from which we can see that our method outperforms other four methods in general.

Batch performs better at early stage, but fails to steadily improve its performance since its criterion favors samples that have small margin, thus its performance approaches to the performance of *SVMActive* when labeled samples keeps increasing. It is also worth noticing that *SVMActive* performs almost the worst in this data set. This is due to the greedy mechanism that focus only on the data that near the classification boundary decided by the initial stage, which is quite shortsighted.

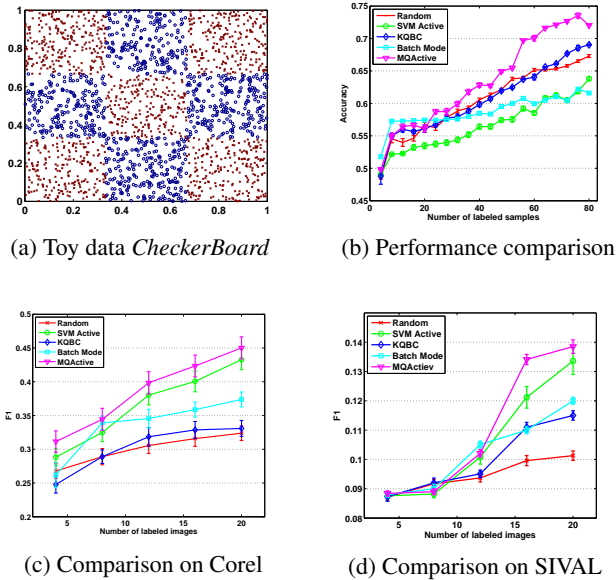


Fig. 1. Comparison on toy *CheckerBoard* and two real data sets.

Secondly, we conduct comparison on two real world image sets: 1. Subset of Corel gallery with 30 categories and 100 images in each category; 2. SIVAL Image Repository with 25 categories and 60 images in each category.

Low-level image features including 64-d HSV color histogram, 9-d LUV color moment, etc., are extracted from the images. We use Chi-square distance for histogram features, and Euclidian distance for others. By combining different features, we get the kernel K for SVM with parameter $\sigma = 20$, our experiments on all the active learners are based on the this kernel. Batch size k is set as 4. Other parameters are set the same as those in the toy experiment.

For empirical evaluation, we adopt the $F1$ metric which takes into account both the precision and the recall, and defined as $F1 = \frac{2pr}{p+r}$, where p denotes precision and r denotes recall. Top 100 images are returned, and average $F1$ measures over 100 runs with standard error are shown in Figure 1(c)(d).

The results demonstrate that *MQActive* outperforms other active learning methods on both data sets in image retrieval scenario. Our method requires fewer rounds of relevance feedbacks to achieve comparable performance as other methods, both on the easy Corel data set and the much harder SIVAL data set. Take Corel for example, given a required $F1$ of 0.35, our method needs approximately 8 labeled images, while *SVMActive* requires 10, *Batch* requires 14, and the other two methods requires even more.

4. CONCLUSION

We attack the sub-optimality problem of traditional active learning algorithms by directly minimizing the expected size of SVM version

space in a global optimization way. Efficient sampling and estimation techniques are utilized to boost the efficiency of the algorithm for realtime use. Experimental evaluation in image retrieval shows that the round of relevance feedbacks needed to achieve satisfactory retrieval performance is reduced significantly.

5. ACKNOWLEDGMENT

Supported by National 863 project(No. 2006AA01Z121).

6. REFERENCES

- [1] Michael S. Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 2, no. 1, pp. 1–19, 2006.
- [2] Xiang Sean Zhou and Thomas S. Huang, "Relevance feedback in image retrieval: A comprehensive review," *Multimedia Systems*, vol. 8, no. 6, pp. 536–544, 2003.
- [3] Y. Rui, T. Huang, and S. Mehrotra, "Content-Based image retrieval with relevance feedback in MARS," pp. 815–818.
- [4] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of artificial intelligence research*, vol. 4, pp. 129, 1996.
- [5] D. MacKay, "Information-based objective functions for active data selection," *Neural Computation*, vol. 4, 1992.
- [6] Simon Tong and Edward Chang, "Support vector machine active learning for image retrieval," in *MULTIMEDIA '01*, 2001.
- [7] Lei Wang, Kap Luk Chan, and Zhihua Zhang, "Bootstrapping svm active learning by incorporating unlabelled images for image retrieval," in *CVPR '03*, 2003.
- [8] Ralf Herbrich, Thore Graepel, and Colin Campbell, "Bayes point machines," *J. Mach. Learn. Res.*, vol. 1, pp. 245–279, 2001.
- [9] David A. Cohn, Les Atlas, and Richard E. Ladner, "Improving generalization with active learning," *Machine Learning*, vol. 15, no. 2, pp. 201–221, 1994.
- [10] Thomas Osugi, Deng Kun, and Stephen Scott, "Balancing exploration and exploitation: A new algorithm for active machine learning," in *ICDM '05*, 2005.
- [11] Klaus Brinker, "Incorporating diversity in active learning with support vector machines," in *ICML '03*, 2003.
- [12] Steven C. H. Hoi, Rong Jin, Jianke Zhu, and Michael R. Lyu, "Batch mode active learning and its application to medical image classification," in *ICML '06*, 2006.
- [13] Nicholas Roy and Andrew McCallum, "Toward optimal active learning through sampling estimation of error reduction," in *ICML '01*, 2001.
- [14] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *COLT '92*, 1992.
- [15] Ran Gilad-Bachrach, Amir Navot, and Naftali Tishby, "Query by committee made real," in *NIPS*, 2005.
- [16] Tom Mitchell, *Machine Learning*, McGraw Hill, 1997.
- [17] Bernhard Schölkopf and Alexander J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, The MIT Press, 2001.
- [18] Pal Rujan, "Playing billiards in version space," *Neural Comput.*, vol. 9, no. 1, pp. 99–122, 1997.