

MULTIPLE BOOSTING SVM ACTIVE LEARNING FOR IMAGE RETRIEVAL

Wei Jiang, Guihua Er, Qionghai Dai

Department of Automation, Tsinghua University

ABSTRACT

Content-based image retrieval can be viewed as a classification problem, and the small sample size learning difficulty makes it difficult for most CBIR classifiers to get satisfactory performance. In this paper, using SVM classifier as the component classifier, the method of ensemble of classifiers is incorporated into the relevance feedback process to alleviate this problem from two aspects: 1. Within each feedback round, multiple parallel component classifiers are constructed, one over one feature subspace individually, and then are merged together to get an ensemble classifier. 2. During feedback rounds, boosting method is incorporated to sequentially combine the component classifiers over each feature subspace respectively, which further improves the classification result. Experiments over 5,000 images show that the proposed method can improve the retrieval performance consistently, without loss of efficiency.

1. INTRODUCTION

Content-based image retrieval (CBIR) has been a very important research topic since last decades. In CBIR systems, an image is represented by a low-level visual feature vector and the gap between high-level semantics and low-level features has been the main obstacle hindering further performance improvement. Relevance feedback is introduced to bridge this gap [3, 4], which is recently viewed as a learning and classification problem [1, 5], where the system constructs classifiers by training data from user's feedback, and classifies all images in the database into two categories—the wanted (“relevant” images), and the rest (“irrelevant” ones).

In CBIR context, the training samples are the labeled images from user's feedback, which are usually very few compared with the feature dimensionality and the database size. This small sample size learning difficulty makes most CBIR classifiers weak and unstable [7]. Ensemble of classifiers gives an effective solution to alleviate this problem, which can generate a strong classifier by combining several weak component classifiers [6]. In this paper, the method of ensemble classifiers is incorporated into the relevance feedback process to improve the retrieval performance, and our contribution has two folds (Fig.1).

Thanks to CNSF and BRF of Tsinghua University for funding.

First, the method of combining parallel classifiers is adopted. The input feature space is divided into several subspaces, with each subspace corresponding to one kind of feature (e.g. the 9 dimensional color moment feature). In each feedback round, the system constructs multiple component classifiers, one over one feature subspace individually, and merges them into an ensemble classifier. Since user often emphasizes differently on different features, this parallel ensemble method not only makes it possible for the system to treat features separately, but also improves the retrieval result as many classifier ensemble methods do.

Second, boosting method [9] is introduced into the feedback process. Boosting is a sequential classifier ensemble method to enhance a weak learner. Based on the analysis of characteristic of CBIR classifiers, we follow the basic sample re-weighting idea of boosting method and modify it to be more adaptive to our problem. During the feedback rounds, the component classifiers over each kind of feature are combined sequentially to generate ensemble classifiers, which further improves the retrieval performance.

SVM is used as the component classifier because it performs better than most other classifiers. The mechanism of SVM_{Active} [5] is also adopted in our system. The proposed method is called Multiple Boost SVM Active Learning (MBSVM_{Active}) algorithm. Experiment results over 5,000 images show that MBSVM_{Active} can improve the retrieval performance consistently without losing efficiency.

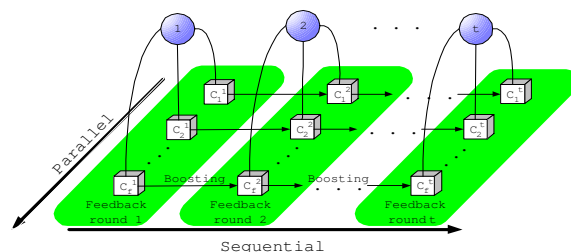


Fig. 1. The enhancing scheme of MBSVM_{Active} algorithm. The 3-D box denotes the component classifier. The ball above each round represents the parallel ensemble classifier. The arrow denotes the sequential combination process.

The rest of this paper is organized as follows. In section 2 we formulate our problem and introduce the algorithm of SVM_{Active}. Section 3 describes our MBSVM_{Active} algo-

rithm in detail. Experiment results are given in section 4. And in section 5 we give our conclusion.

2. PROBLEM FORMULATION

Assume that there are M images in the database, denoted by $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_M]$. Each image \mathbf{x}_i is a d -dimensional feature vector, which can be viewed as a point in the d -dimensional feature space, $\mathbf{x}_i \in \mathcal{F}^d$. The relevance feedback process of SVM_{Active} [5] can be described as follows. In t^{th} feedback round, we have a training set \mathcal{S}^t in feature space \mathcal{F}^d . The system constructs a SVM classifier over \mathcal{S}^t and classifies all \mathbf{X} in the database. The output of this classifier is a set of distance $\mathcal{D}^t = \{D^t(\mathbf{x}_i), i=1, \dots, M\}$, which denotes the distance of each \mathbf{x}_i to the decision boundary (positive distance represents “relevant”, and negative distance represents “irrelevant”). In rest of this paper, we represent this classifier constructing and classifying process by $\mathcal{D}^t = C(\mathcal{S}^t, \mathbf{X})$. Then a set of images (with largest $D^t(\mathbf{x}_i) > 0$) as the retrieval result is given out, called *return set*, denoted by \mathcal{R}^t . If user doesn’t satisfy with \mathcal{R}^t , the system provides a set of images (with smallest $|D^t(\mathbf{x}_i)|$) for user to label, called *label set*, denoted by \mathcal{L}^{t+1} . Let y_i denote the label result of \mathbf{x}_i , where $y_i = 1$ if \mathbf{x}_i is “relevant”; $y_i = -1$ if \mathbf{x}_i is “irrelevant”. then \mathcal{L}^{t+1} is added into \mathcal{S}^t to be $\mathcal{S}^{t+1} = \mathcal{S}^t \cup \mathcal{L}^{t+1}$, and go to the next feedback round.

3. ENSEMBLE OF CLASSIFIERS

In this section, we will discuss the MBSVM_{Active} algorithm in detail from the parallel and sequential aspects respectively.

3.1. Parallel Combination

The feature space \mathcal{F}^d is divided into f subspaces $\mathcal{F}^{d_1}, \dots, \mathcal{F}^{d_f}$, where each \mathcal{F}^{d_i} corresponds to one kind of feature, such as the first three color moments whose $d_i = 9$. For the t^{th} feedback round, let \mathcal{S}_i^t and \mathbf{X}_i denote the projection of \mathcal{S}^t and \mathbf{X} into the feature subspace \mathcal{F}^{d_i} respectively. Assume that $\mathcal{D}_i^t = C(\mathcal{S}_i^t, \mathbf{X}_i)$ represents the i^{th} SVM component classifier, and has corresponding $D_i^t(\mathbf{x})$ for image \mathbf{x} in the database. Define the ensemble distance set as \mathcal{D}^t , whose element is given by:

$$D^t(\mathbf{x}) = \sum_{i=1}^f w_i^t D_i^t(\mathbf{x}) \quad (1)$$

According to \mathcal{D}^t , \mathcal{R}^t and \mathcal{L}^{t+1} are selected following the SVM_{Active} mechanism.

We calculate the weights $w_i^t, i = 1, \dots, f$ according to the training error of the component classifier. For a “relevant”/“irrelevant” sample, the distance of it to the decision

boundary in the “relevant”/“irrelevant” side reflects the degree of how correctly it is classified. This information is utilized to calculate a soft error rate here instead of the error rate calculated by decision hypothesis. Define the error degree of the i^{th} component classifier as:

$$\epsilon_i^t = \frac{1}{\left(\sum_{\substack{\mathbf{x} \in \mathcal{S}_i^t \\ y=1}} D_i^t(\mathbf{x}) - \sum_{\substack{\mathbf{x} \in \mathcal{S}_i^t \\ y=-1}} D_i^t(\mathbf{x}) \right)} \quad (2)$$

The weight is given by:

$$w_i^t = \frac{1}{Z_1^t} \log \left(\frac{1 - \epsilon_i^t}{\epsilon_i^t} \right) \quad (3)$$

where Z_1^t is normalization constant to make $\sum_{i=1}^f w_i^t = 1$.

3.2. Sequential Combination

As discussed above, the component classifiers over each feature subspace are combined sequentially during feedback rounds. That is, each component classifier for parallel combination in the previous subsection is actually a boosted classifier. AdaBoost is an effective sequential combination algorithm, and can scale up the SVM classifier [2]. Following its central sample re-weighting idea, we modify it to be adaptive to our problem, and combine the component classifiers $\mathcal{D}_i^\tau = C(\mathcal{S}_i^\tau, \mathbf{X}_i), \tau = 1, \dots, t$ to form an ensemble classifier for each feature subspace $i = 1, \dots, f$.

3.2.1. Re-Weighting by Sampling Method

The central idea of original AdaBoost is to re-weight the distribution of training samples by emphasizing on the “hard” ones (samples misclassified by previous classifiers). We propose to realize the sample re-weighting idea by sampling method. When a training example is important, we generate more samples around it. This can be viewed as another way to re-weight the distribution.

Suppose in the t^{th} feedback round, the training set for the i^{th} component classifier in previous $t-1$ round is \mathcal{S}_i^{t-1} , with corresponding \mathcal{D}_i^{t-1} . Define the important set for this classifier in round t as:

$$\mathcal{T}_i^t = \left\{ \mathbf{x} : \mathbf{x} \in \mathcal{S}_i^{t-1}, y D_i^{t-1}(\mathbf{x}) < 0 \right\} \cup \left\{ \mathbf{x} : \mathbf{x} \in \mathcal{L}^t, y = 1 \right\} \quad (4)$$

Then the updated training set is given by:

$$\mathcal{S}_i^t = \mathcal{S}_i^{t-1} \cup \mathcal{T}_i^t \cup \mathcal{L}^t \quad (5)$$

The \mathcal{T}_i^t in Eqn(4) contains two parts: the training samples misclassified by the previous classifier, and the new labeled “relevant” samples. This is different from boosting

method since we have a training set whose size is increasing during the feedback rounds, and new added samples should also be weighted. In the CBIR context, a usually assumed fact is, the “relevant” images share some semantic cues which reflect the query concept, while the “irrelevant” ones come from different semantic categories and have little correlation. Thus the “relevant” images are important to grasp the query concept, and are also set to be important in our system.

3.2.2. Weighted Voting Method

Similar to the parallel ensemble method in section 3.1, $D_i^t(\mathbf{x})$ denotes the “relevant” degree of image \mathbf{x} judged by the i^{th} component classifier. We adopt a soft combination strategy which combines the judgements of the component classifiers into a set of ensemble distance $D_{i\text{en}}^t$, instead of the hypothesis voting in AdaBoost. The element of $D_{i\text{en}}^t$ is:

$$D_{i\text{en}}^t(\mathbf{x}) = \sum_{\tau=1}^t W_i^\tau D_i^\tau(\mathbf{x}) \quad (6)$$

where W_i^k is the classifier’s weight.

Note that, the size of training set is increasing in our problem, and for small sample learning problem, the training set size influences the classifier’s performance greatly, more training samples can result in more accurate classification result. Thus, besides training error, the importance of each classifier should relate to the training set size too. Thus the weight W_i^τ is given as:

$$W_i^\tau = \frac{1}{Z_2^t} \left(\log \left(\frac{1 - \epsilon_i^\tau}{\epsilon_i^\tau} \right) + \beta |\mathcal{S}_i^\tau| \right) \quad (7)$$

where $|\mathcal{S}_i^\tau|$ is the cardinality of \mathcal{S}_i^τ . ϵ_i^τ is the training error defined in Eqn(2), and Z_2^t is the normalization constant which makes $\sum_{\tau=1}^t W_i^\tau = 1$. β is a parameter which determines the relative importance of $|\mathcal{S}_i^\tau|$ and ϵ_i^τ .

By Eqn(6) we get the boosted classifier, $D_{i\text{en}}^t$, which is the i^{th} component classifier for parallel combination in Eqn(1). The entire MBSVM_{Active} algorithm is summarized in Fig.2.

4. EXPERIMENTS

The proposed MBSVM_{Active} algorithm is tested and compared with SVM_{Active} over a 5,000 real world images dataset, which has 50 semantic categories, such as “ship”, “car”, etc. Each category has 100 images, and all the images are collected from Corel CDs. The feature space has a dimensionality of 155, which consists of four kinds of features in total. They are color coherence in HSV color space, the first

Initialize:

1. $w_i^1 = 1/f$, get the user’s query image \mathbf{x}_q
2. Get \mathcal{L}^1 by random selection
3. Get $\mathcal{S}_1^1 = \dots = \mathcal{S}_f^1 = \mathcal{L}^1 \cup \{\mathbf{x}_q\}$

Recursion: for each feedback round t

1. if $t = 1$, for each feature i , construct $D_{i\text{en}}^t = \mathcal{D}_i^t = C(\mathcal{S}_i^t, \mathbf{X}_i)$, go to **Parallel Step**
2. **Sequential Step** For each feature i
 - get \mathcal{T}_i^t by Eqn(4)
 - $\mathcal{S}_i^t = \mathcal{S}_i^{t-1} \cup \mathcal{T}_i^t \cup \mathcal{L}^t$
 - Built $\mathcal{D}_i^t = C(\mathcal{S}_i^t, \mathbf{X}_i)$, get $D_{i\text{en}}^t$ by Eqn(6)
3. **Parallel Step**
 - Get \mathcal{D}^t by Eqn(1) (replacing \mathcal{D}_i^t with $D_{i\text{en}}^t$ in Eqn(1))
 - Get \mathcal{R}^t and \mathcal{L}^{t+1} by SVM_{Active} mechanism
4. If user satisfy with \mathcal{R}^t , stop; otherwise, label \mathcal{L}^{t+1} , go to next round

Fig. 2. Pseudo-code for the MBSVM_{Active} algorithm.

three color moments in LUV color space, the directional-ity texture, and the coarseness vector texture. Details about these features can be found in [8].

Assume that user is querying for images in one semantic category in each query session, and have 5 rounds of feedback. In each round $\mathcal{L}^t = 20$. The performance measurement used is the average top- k precision (the percentage of “relevant” images in return set, with $\mathcal{R}^t = k$). Each value listed is the average result of 500 independent search processes. we use RBF kernel for SVM classifier.

Fig.3(a, b) shows the average P_{10} and P_{20} of the MBSVM_{Active} and SVM_{Active} algorithms after each feedback round respectively. The figures indicate that from the second feedback round, each average precision of our MBSVM_{Active} consistently outperforms the corresponding one of SVM_{Active}, and the precision improvements obtained by MBSVM_{Active} for feedback round 2, 3, 4, and 5 are 11.89%, 5.74%, 2.76%, 2.20% and 8.89%, 6.39%, 3.10%, 1.58% for P_{10} and P_{20} respectively. Also, the advantage of the MBSVM_{Active} algorithm is more obvious for the first few rounds. Since actually user has no patience to give feedback for many rounds, this performance improvement in the first few rounds is very appealing. Moreover, the time cost for MBSVM_{Active} has no large increase, since although we constructs 4 component classifiers in each round, each classifier is built over a subspace with smaller dimensionality. It shows MBSVM_{Active} can improve the retrieval performance without any lost of efficiency.

Fig.4 gives a retrieval example querying for images in “ship” category for both MBSVM_{Active} and SVM_{Active} algorithms. We have 3 rounds of feedback. Label sets for these rounds and the retrieval results after these rounds are

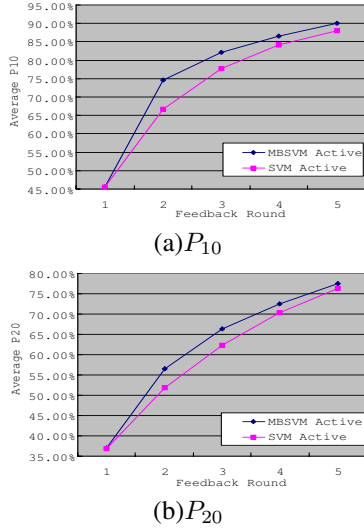


Fig. 3. The average precision of the $MBSVM_{Active}$ and SVM_{Active} after each feedback round. Legend order reflects the order of curves.

listed (\mathcal{L}^1 for the two methods are the same, which is not listed). The figures indicate that, compared with SVM_{Active} , $MBSVM_{Active}$ can find the boundary between “relevant” and “irrelevant” images more quickly, and select more images near the boundary to form the label set, which results in better performance.

5. CONCLUSION

In this paper, we have proposed a new $MBSVM_{Active}$ algorithm to incorporate classifier ensemble method into the relevance feedback process to improve the performance of CBIR classifiers. The parallel classifier ensemble method constructs multiple component SVM classifiers over different feature subspaces respectively, and merges them together to generate the ensemble classifier within each feedback round. The sequential classifier ensemble method adopts the sample re-weighting idea of boosting algorithm and combines the component classifiers over each feature subspace during feedback rounds to further enhance retrieval result. Since the multiple classifier ensemble method gives a way to treat feature subspaces separately, and enhance the weak classifier’s performance, more work can be done to utilize it in CBIR systems. Also, the feature subspaces are pre-defined here, how to adaptively generate them during different query sessions is another aspect of our future work.

6. REFERENCES

[1] G.D. Guo. et al, “Learning similarity measure for natural image retrieval with relevance feedback,” *IEEE Trans. Neural Networks*, 13(4), pp.811-820, 2002

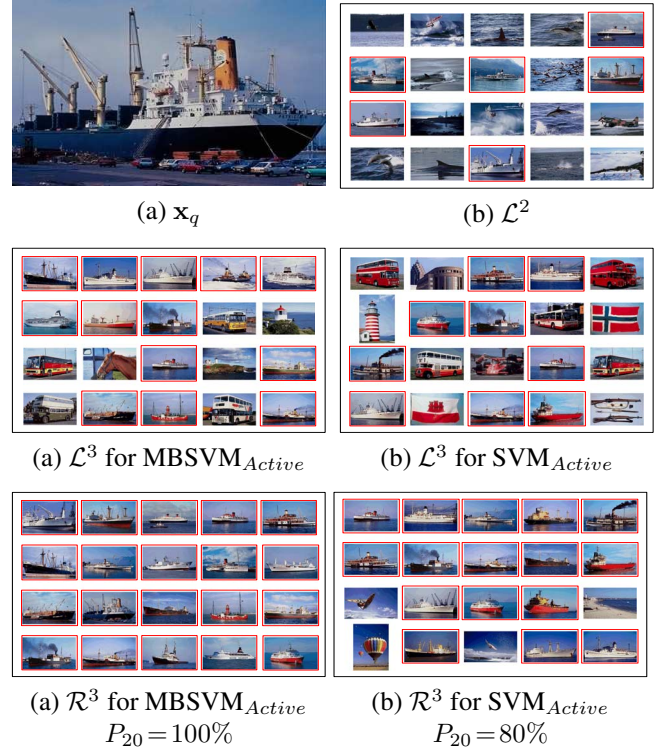


Fig. 4. Example for querying images in “ship” semantic category. Images with red frame is the “relevant” images.

[2] D. Pavlov, and J.C. Mao, “Scaling-up support vector machines using boosting algorithm,” *International Conference on Pattern Recognition*, Barcelona, Spain 2000

[3] I.J. Cox. et al, “The Bayesian image retrieval system, PicHunter: theory, implementation and psychophysical experiments,” *IEEE Trans. Image Processing*, 9(1), pp.20-37, 2000.

[4] K. Porkaew, and K. Chakrabarti, “Query refinement for multimedia similarity retrieval in MARS,” *Proc. of ACM Multimedia*, pp.235-238, Florida, USA, 1999

[5] S. Tong, and E. Chang, “Support vector machine active learning for image retrieval,” *ACM Multimedia*, Ottawa, Canada, 2001

[6] TK. Ho, “Multiple classifier combination: lessons and next steps,” *Hybrid methods in pattern recognition*, pp.171-198, World Scientific, 2002

[7] X.S. Zhou, and T.S. Huang, “Relevance Feedback in Image retrieval: a comprehensive review,” *Multimedia Systems*, 8(6), pp.536-544, 2003

[8] X.Y. Jin. et al, “Feature evaluation and optimizing feature combination for content-based image retrieval,” technique report, http://www.cs.virginia.edu.cn/xj3a/publication/feature_selection.pdf

[9] Y. Freund, and R. Schapire, “Experiments with a new boosting algorithm,” *Proc. of the 13th International Conference on Machine Learning*, pp.148-156, Bari, Italy, 1996