Online Feature Selection Based on Generalized Feature Contrast Model*

Wei Jiang[1] Mingjing Li[2] Hongjiang Zhang[2] Jinwei Gu[1]

[1]Department of Automation, Tsinghua University, Beijing 100084, China
[2]Microsoft Research Asia, 49 Zhichun Road, Beijing 100080, China

Abstract

*To really bridge the gap between high-level semantics and low-level features in content-based image retrieval (*CBIR*), a problem that must be solved is: which features are suitable for explaining the current query concept. In this paper, we propose a novel feature selection criterion based on a psychological similarity measurement* – generalized feature contrast model, *and implement an online feature selection algorithm in a boosting manner to select the most representative features and do classification during each feedback round. The advantage of the proposed method is: it doesn't require Gaussian assumption for "relevant" images as other online FS methods; it accounts for the intrinsic asymmetry between "relevant" and "irrelevant" image sets in CBIR online learning; it is very fast. Extensive experiments have shown our algorithm's effectiveness.*

1 Introduction

It is well known that the gap between high-level semantics and low-level features limits the development of content-based image retrieval (*CBIR*) systems. *Relevance feedback* is introduced to bridge this gap, which is generally treated as an online supervised learning problem [7, 9]. During each feedback round, user labels some images to be "relevant" or "irrelevant" as training samples to supervise the learning process in subsequent rounds.

To really grasp the query concept from low-level features, a CBIR learner must tackle a fundamental problem – which features are suitable for explaining the current query concept. This refers to the issue of *feature selection* (*FS*). Compared with other machine learning problems, CBIR online learning has to solve three issues: (1) The curse of Dimensionality, the training samples are only the labeled images from user, which are too few compared with the feature dimensionality and the database size. (2) Intrinsic asymmetry, the images labeled to be "relevant" during a query session share some semantic cue, while the "irrelevant" images are different from "relevant" ones in different ways. Thus we need to treat the two image sets unequally. (3) Fast response requirement. The curse of dimensionality makes the system unable to well estimate the sample's distribution, and the distribution based FS methods (e.g. K-LD [2]) are not suitable. Tieu [5] tried to boost small sample learning by AdaBoost [8], but it needs a very large highly selective feature pool to get good result, which is computational infeasible for CBIR online FS. By now the most feasible approach is the *Discriminant Analysis* (*DA*) methods, such as *MDA* [9] and *BiasMap* [7]. They assume Gaussian distribution for 'relevant" images and minimize the covariance of "relevant" set over the between class distance based on this assumption, which avoids distribution estimation. Also they consider the intrinsic asymmetry property, which is neglected by most other FS methods. However, their basic single Gaussian assumption usually doesn't hold, since the few training samples are always scattered in the high dimensional feature space, and their effectiveness will suffer.

In this paper, to address the issue of online feature selection in CBIR context, an feature selection criterion – *Generalized Feature Contrast Model* (*GFCM*) is proposed. GFCM (which is based on the *Feature Contrast Model* [1]) measures the similarity between "relevant" and "irrelevant" sets, which accounts for the asymmetry requirement for CBIR online learning and doesn't require Gaussian assumption. Moreover, a feature selection algorithm is implemented in a boosting manner to select the optimal features one by one from original feature pool by re-weighting the training samples, and combine the incrementally learned classifiers over the selected features. Extensive experiments over 5,000 images show that, compared with DA and AdaBoost FS [5], our method can get good performance while cost less processing time.

2 Feature Selection by GFCM

In CBIR context, the object of feature selection is to find the most discriminative feature subspace where the "relevant" set ($\mathcal{R}$) and "irrelevant" set ($\mathcal{IR}$) are most separable. Here we use GFCM to guide the feature selection in relevance feedback for CBIR online leaning.

*This work was performed at Microsoft Research Asia

0-7803-8603-5/04/$20.00 ©2004 IEEE

2.1 Generalized Feature Contrast Model

The *Feature Contrast Model (FCM)* is firstly proposed as a psychological similarity measurement between two objects, which is based on set theory. Let a, b be two stimuli, FCM represents each of them by a set of binary features they possess, denoted by A, B, and defines their similarity as:

$$S(A,B) = f(A \otimes B) - \alpha f(A \ominus B) - \beta f(B \ominus A) \quad (1)$$

$A \otimes B$ is the *common feature* contained by a and b, $A \ominus B$ ($B \ominus A$) is the *distinctive feature* contained by a but not b (b but not a). $f(\cdot)$ is a salient function, whose value increases monotonically when the variable in bracket increase. If $\alpha \neq \beta$, we emphasize the features in different stimuli unequally. The asymmetry direction is determined by the relative "salience" of stimuli: if a is more salient, $\alpha \geq \beta$ [6].

FCM can be generalized to measure the similarity between $\mathcal{R}$ and $\mathcal{IR}$ as:

$$S(\mathcal{R},\mathcal{IR}) = f(\mathcal{R} \otimes \mathcal{IR}) - \alpha f(\mathcal{R} \ominus \mathcal{IR}) - \beta f(\mathcal{IR} \ominus \mathcal{R}) \quad (2)$$

Since the features shared by images in $\mathcal{R}$ but not contained by images in $\mathcal{IR}$ are more important than the features shared by images in $\mathcal{IR}$ but not contained by images in $\mathcal{R}$ to grasp the query concept, $\mathcal{R}$ is more salient than $\mathcal{IR}$. We can use $\alpha \geq \beta$ to describe this asymmetry. This is the generalized feature contrast model.

2.2 GFCM similarity measurement

To discard the Gaussian assumption for $\mathcal{R}$ or $\mathcal{IR}$, instead of covariance and between class scatter matrix, the pairwise relationship between each image in $\mathcal{R}$ and each in $\mathcal{IR}$ are directly evaluated, based on which $S(\mathcal{R},\mathcal{IR})$ is calculated.

Let $\vec{\mathbf{x}}$ denote an image in the d-dimensional original feature space, and we have an m size $\mathcal{R}$ and n size $\mathcal{IR}$. $\mathcal{R}_k = \{x_{ik}^{(\mathcal{R})}, i = 1,\ldots,m\}$ and $\mathcal{IR}_k = \{x_{jk}^{(\mathcal{IR})}, j = 1,\ldots, n\}$ represent the projected $\mathcal{R}$ and $\mathcal{IR}$ into the k-th feature axis respectively. Weights $\mathcal{W}^{(\mathcal{R})} = \{w(\vec{\mathbf{x}}_i^{(\mathcal{R})}),\ i = 1,\ldots,m\}$ and $\mathcal{W}^{(\mathcal{IR})} = \{w(\vec{\mathbf{x}}_j^{(\mathcal{IR})}),\ j = 1,\ldots,n\}$ are the sample weights. The more tightly $\mathcal{R}$ or $\mathcal{IR}$ clusters along this feature, the more salient this feature axis is for $\mathcal{R}$ or $\mathcal{IR}$. Value $\mu_k(\vec{\mathbf{x}}_i, \vec{\mathbf{x}}_j) = |x_{ik} - x_{jk}|$ denotes the "farness" degree between $\vec{\mathbf{x}}_i$ and $\vec{\mathbf{x}}_j$ along this feature axis. $\min_{j=1}^{n}\left\{\mu_k(\vec{\mathbf{x}}_i^{(\mathcal{R})}, \vec{\mathbf{x}}_j^{(\mathcal{IR})})\right\}$ denotes the strongest relationship between each image $\vec{\mathbf{x}}_i^{(\mathcal{R})}$ in $\mathcal{R}$ and set $\mathcal{IR}$, and $\min_{i=1}^{m}\left\{\mu_k(\vec{\mathbf{x}}_j^{(\mathcal{IR})}, \vec{\mathbf{x}}_i^{(\mathcal{R})})\right\}$ denotes the strongest relationship between each $\vec{\mathbf{x}}_j^{(\mathcal{IR})}$ and set $\mathcal{R}$. Thus the common feature part in Eqn(2) is given by:

$$f(\mathcal{R}_k \otimes \mathcal{IR}_k) = -\sum_{i=1}^{m} w(\vec{\mathbf{x}}_i^{(\mathcal{R})}) \min_{j=1}^{n}\left\{\mu_k(\vec{\mathbf{x}}_i^{(\mathcal{R})}, \vec{\mathbf{x}}_j^{(\mathcal{IR})})\right\} - \sum_{j=1}^{n} w(\vec{\mathbf{x}}_j^{(\mathcal{IR})}) \min_{i=1}^{m}\left\{\mu_k(\vec{\mathbf{x}}_j^{(\mathcal{IR})}, \vec{\mathbf{x}}_i^{(\mathcal{R})})\right\} \quad (3)$$

As for the distinctive feature parts, if $\mathcal{R}$ is salient and $\mathcal{IR}$ is not salient along this feature, $f(\mathcal{R}_k \ominus \mathcal{IR}_k)$ should be large. Value $\max_{j=1}^{n}\left\{\mu_k(\vec{\mathbf{x}}_i^{(\mathcal{R})}, \vec{\mathbf{x}}_j^{(\mathcal{IR})})\right\} - \min_{j=1}^{n}\left\{\mu_k(\vec{\mathbf{x}}_i^{(\mathcal{R})}, \vec{\mathbf{x}}_j^{(\mathcal{IR})})\right\}$ denotes the scatter degree of set $\mathcal{IR}$ respect to image $\vec{\mathbf{x}}_i^{(\mathcal{R})}$ along this feature axis, and $f(\mathcal{R}_k \ominus \mathcal{IR}_k)$ is given by:

$$f(\mathcal{R}_k \ominus \mathcal{IR}_k) = \sum_{i=1}^{m} w(\vec{\mathbf{x}}_i^{(\mathcal{R})}) \left[\max_{j=1}^{n}\left\{\mu_k(\vec{\mathbf{x}}_i^{(\mathcal{R})}, \vec{\mathbf{x}}_j^{(\mathcal{IR})})\right\} - \min_{j=1}^{n}\left\{\mu_k(\vec{\mathbf{x}}_i^{(\mathcal{R})}, \vec{\mathbf{x}}_j^{(\mathcal{IR})})\right\}\right] \quad (4)$$

With similar analysis, $f(\mathcal{IR}_k \ominus \mathcal{R}_k)$ is given by:

$$f(\mathcal{IR}_k \ominus \mathcal{R}_k) = \sum_{j=1}^{n} w(\vec{\mathbf{x}}_j^{(\mathcal{IR})}) \left[\max_{i=1}^{m}\left\{\mu_k(\vec{\mathbf{x}}_j^{(\mathcal{IR})}, \vec{\mathbf{x}}_i^{(\mathcal{R})})\right\} - \min_{i=1}^{m}\left\{\mu_k(\vec{\mathbf{x}}_j^{(\mathcal{IR})}, \vec{\mathbf{x}}_i^{(\mathcal{R})})\right\}\right] \quad (5)$$

With Eqn(3-5) and Eqn(2), we get the similarity measurement function $S_k(\mathcal{R}_k, \mathcal{IR}_k)$ as follows:

$$\begin{aligned} S_k(\mathcal{R}_k,\mathcal{IR}_k) = {} & (\alpha-1)\sum_{i=1}^{m} w(\vec{\mathbf{x}}_i^{(\mathcal{R})}) \min_{j=1}^{n}\left\{\mu(x_{ik}^{(R)}, x_{jk}^{(IR)})\right\} \\ & + (\beta-1)\sum_{j=1}^{n} w(\vec{\mathbf{x}}_j^{(\mathcal{IR})}) \min_{i=1}^{m}\left\{\mu(x_{ik}^{(R)}, x_{jk}^{(IR)})\right\} \\ & - \alpha\sum_{i=1}^{m} w(\vec{\mathbf{x}}_i^{(\mathcal{R})}) \max_{j=1}^{n}\left\{\mu(x_{jk}^{(IR)}, x_{ik}^{(R)})\right\} \\ & - \beta\sum_{j=1}^{n} w(\vec{\mathbf{x}}_j^{(\mathcal{IR})}) \max_{i=1}^{m}\left\{\mu(x_{jk}^{(IR)}, x_{ik}^{(R)})\right\} \end{aligned} \quad (6)$$

Coefficients α and β adjust the relative importance of features shared by $\mathcal{R}$ and $\mathcal{IR}$, whose effect will be investigated *in later experiments.*

2.3 Feature selection in boosting manner

Given a set of optimal feature axis selected by above GFCM FS criterion, boosting mechanism gives an effective way for selecting a new added feature axis by re-weighting the training samples, and combines the classifiers constructed over the incrementally learned features into an ensemble classifier with a decreased training error. Assume we have selected k feature axis, and constructed a classifier C_i over each feature, $i = 1,\ldots,k$. The classification result of an image $\vec{\mathbf{x}}$ by C_k is $q_k(\vec{\mathbf{x}})$. The weight for $\vec{\mathbf{x}}$ is updated as:

$$w(\vec{\mathbf{x}}) = \frac{1}{Z} w(\vec{\mathbf{x}}) \beta_k^{1 - y q_k(\vec{\mathbf{x}})} \quad (7)$$

where Z is the normalization factor for $W(\vec{\mathbf{x}})$, y is the label of $\vec{\mathbf{x}}$ from user ($y = 1$ if $\vec{\mathbf{x}}$ is "relevant", $y =$

-1 otherwise). The update factor β_k is given by $\beta_k = [(\epsilon_k)/(1-\epsilon_k)]^{(1+c\cdot\gamma^k)}$, where ϵ_k is the training error of C_k. $c \geq 1$ is a parameter to make weight change enough for new feature selection. $\gamma \in (0,1)$ is a parameter which assures large update speed at the beginning of training. This β_k is proved to be more effective than original β in AdaBoost [2]. Practically we prefer to use $c=1$, $\gamma=0.65$.

The *Fuzzy K-Nearest Neighbor (FKNN)* classifier [3] is adopted as the weak learner for each selected feature. For the classifier over the i-th feature, it assigns a membership for each image $\vec{x}$ in the database to represent its "relevant" degree to the current query concept as:

$$\phi_i(\vec{x}) = \frac{\sum_{j=1}^{k} w_j^{(t)} \left(1/\|\vec{x} - \vec{x}_j^{(t)}\|^2\right)}{\left(\sum_{j=1}^{k} w_j^{(t)}\right) \cdot \sum_{j=1}^{k} \left(1/\|\vec{x} - \vec{x}_j^{(t)}\|^2\right)} \quad (8)$$

where $\vec{x}_j^{(t)}, j=1,\ldots,k$ are the k nearest neighbors of $\vec{x}$ in the training set, $w_j^{(t)} = w(\vec{x}_j^{(t)})$ if $\vec{x}_j^{(t)} \in \mathcal{R}$, $w_j^{(t)} = -w(\vec{x}_j^{(t)})$ if $\vec{x}_j^{(t)} \in \mathcal{IR}$. $\vec{x}$ is predicted to be "relevant" if $\phi_i(\vec{x}) > 0.5$; and "irrelevant" otherwise. And for misclassified "relevant" ("irrelevant") training samples, the larger (smaller) the membership, the smaller the mistake is. Thus we calculate ϵ_i softly with $\phi_i(\vec{x})$ by leave one out method [4] as:

$$\epsilon_i = \frac{1}{Z_0}\left[-\sum_{j=1}^{m} \phi_i(\vec{x}_j^{(\mathcal{R})}) + \sum_{j=1}^{n} \phi_i(\vec{x}_j^{(\mathcal{IR})})\right] \quad (9)$$

where Z_0 is the normalization factor for ϵ_i. The ensemble classifier is formed by weighted voting:

$$\phi(\vec{x})_{en} = \sum_{i=1}^{k} \log\left(\frac{1-\epsilon_i}{\epsilon_i}\right) \phi_i(\vec{x}) \quad (10)$$

2.4 Entire algorithm

The entire feature selection algorithm (*GFCM boosting*) is given in Fig.1. User selects one query image to start query, and the first round retrieval is carried by nearest searching. During each subsequent feedback round, assume we have m size $\mathcal{R}$ and n size $\mathcal{IR}$. The sample weights are initialized as $w(\vec{x}_i^{(\mathcal{R})}) = 1/2m$, $w(\vec{x}_j^{(\mathcal{IR})}) = 1/2n$. For selecting the i-th feature, we calculate $S_k(\mathcal{R}_k, \mathcal{IR}_k)$ by Eqn(6) for each $k = 1,\ldots,d$, and select the optimal one with smallest $S_k(\mathcal{R}_k, \mathcal{IR}_k)$. Then an FKNN classifier C_i is constructed over the selected feature, and the training samples are re-weighted by Eqn(7). When ϵ_i is less than a small ϵ_{min}, or when we have already selected Dim_{max} feature axis, the ensemble classifier is calculated by Eqn(10). Images with largest $\phi(\vec{x})_{en}$ is selected as the retrieval result, and images whose $\phi(\vec{x})_{en}$ is closest to 0.5 is selected to be labeled by user in this round. Then new labeled images are added into the training set and go to the next round. In practical, we set $\epsilon_{min} = 0.001$, $Dim_{max} = 30$.

The computational complexity for GFCM boosting FS is: $O(m \times n)$ to calculate GFCM similarity; $O(d)$ to find the optimal feature; $O\left((m+n)^2\right)$ to construct FKNN classifier; and $O(m+n)$ to re-weight the samples. The total computational cost for GFCM boosting FS is much less than that for eigenvector decomposition process in DA FS (which is $O(d^3)$), especially when m and n are usually small.

Recursion: for each feedback round

1. **Initialize:** $N_r = 0$, for m size $\mathcal{R}$ and n size $\mathcal{IR}$, set $w(\vec{x}_i^{(\mathcal{R})}) = 1/2m$, $w(\vec{x}_j^{(\mathcal{IR})}) = 1/2n$
2. **Iteration:**
   - Calculate $S_k(\mathcal{R}_k, \mathcal{IR}_k)$ by Eqn(6) for each k
   - Select the optimal feature axis
   - Construct an FKNN classifier over this feature, and re-weight the samples by Eqn(7)
   - $N_r = N_r + 1$. If $\epsilon_k < \epsilon_{min}$ or $N_r > Dim_{max}$, break **Iteration**
3. Calculate the ensemble classifier by Eqn(10)

Figure 1: GFCM Boosting feature selection algorithm.

3 Experimental Results

The experiments are carried on 5,000 images from Corel CDs, which come from 50 semantic categories, with 100 images for each category. The low-level features used are the color coherence in HSV color space, the first three color moments in LUV color space, the directionality texture, and the coarseness vector texture, which comprise a 155-dimensional feature space in total. The statistical average top-k precision is adopted as the performance measurement: the percentage of "relevant" images in a k size return set. We use each of the 5,000 image for querying, and calculate the average result. In each query session we have 5 rounds of feedback, and user labels 10 images in each round.

3.1 Comparison with AdaBoost FS

In this experiment, we set $\alpha = 0.4$, $\beta = 0.2$ for GFCM boosting algorithm, and compare it with the original AdaBoost FS method. Fig.2 gives the average P_{20} of the two methods during each of the 5 rounds, which shows that GFCM boosting outperforms AdaBoost FS significantly from the second feedback round, and the largest performance improvement is 48.43% in round 5. Also, GFCM boosting is faster than AdaBoost FS. E.g. the time cost for GFCM boosting in round 5 is 0.77 (s), for AdaBoost FS is 1.94 (s) (The time is calculated from the system gets the training samples until it gives out the retrieval result in each round). This is because AdaBoost constructs 155 weak classifiers for selecting one feature, while our method only constructs one FKNN classifier.

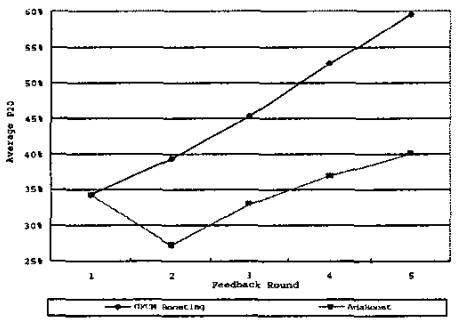


Figure 2: Average P_{20} for GFCM boosting and AdaBoost FS methods.

3.2 Comparison with DA FS

To make the comparison more clearly, the DA FS methods are compared with GFCM boosting also in boosting manner: in each boosting step, one optimal feature is selected by MDA or BiasMap instead of GFCM FS criterion in algorithm shown in Fig.1. Fig.3 gives the average P_{20} of GFCM boosting and DA boosting algorithms, where we also set $\alpha = 0.4$, $\beta = 0.2$ for GFCM boosting. The figure shows that GFCM boosting outperforms MDA and BiasMap boosting consistently from the second round, and the advantage of GFCM boosting is more obvious in the first few rounds. The precision improvement in round 2 are 24.77% and 28.44% compared with MDA and BiasMap boosting respectively. Since user usually has no patience to feedback for many rounds, the performance improvement in the first few rounds is very appealing. Moreover, GFCM boosting is much faster than DA boosting. For example, the time costs for MDA boosting and BiasMap boosting in round 5 is 45.97 (s) and 42.04 (s) respectively, which are about 60 times that for GFCM boosting feature selection.

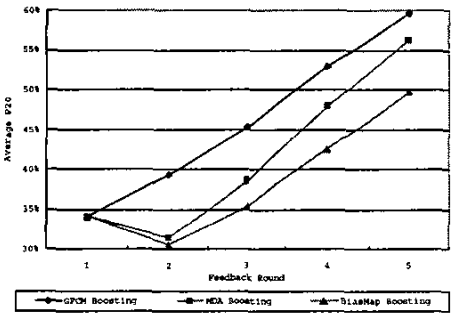
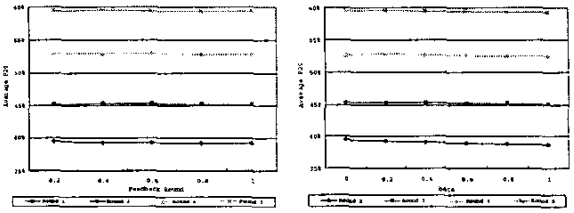


Figure 3: Average P_{20} for GFCM boosting and DA boosting algorithms.

3.3 Influence of parameters

Fig. 4 (a, b) show the average P_{20} of GFCM boosting with $\beta = 0.2$, α from 0 to 1, and $\alpha = 0.4$, β from 0 to 1 respectively. Which show that the general result for $\alpha > \beta$ is better than that for $\alpha \leq \beta$ (e.g. the average P_{20} for $\alpha > \beta$ is better than that for $\alpha \leq \beta$ with 1.23% in round 2). This approves the analysis that GFCM FS criterion accounts for the asymmetry property in CBIR online learning. When $\alpha = 0.4, \beta = 0.2$, the accuracy attains an optimum, but the influence of α is not significant.



(a) (b)

Figure 4: Average P_{20} for GFCM boosting when (a) $\beta = 0.5$, α from 1 to 6 (b) α=0.4, β from 0 to 1.

4 Conclusion

To sum up, in this paper we have proposed a GFCM FS criterion, which doesn't depend on Gaussian distribution assumption for training samples, and accounts for the intrinsic asymmetry requirement of CBIR online learning. A GFCM boosting feature selection algorithm is implemented in boosting manner, which is experimentally proved to be effective for CBIR online feature selection.

References

[1] A. Tvesky, "Feature of similarity," *Psychological review*, 84(4):327-352, 1977.

[2] C. Liu, H.Y. Shum, "Kullback-Leibler Boosting," *IEEE Proc. of CVPR*, vol.1, pp.587-594, Wisconsin, USA, 2003.

[3] J.M. Keller, M.R. Gray, J.A. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE Trans. System, Man, and Cybernetics*, 15(4):580-585, 1985.

[4] K. Fukunaga, "Introduction to statistical pattern recognition (2^{nd} edition)," *Academic Press*, New York, USA, 1990.

[5] K. Tieu, P. Viola, "Boosting image retrieval," *IEEE Proc. of CVPR*, pp.228–235, 2000.

[6] S. Santini, R. Jain, "Similarity measures," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(9):871-883, 1999.

[7] X.S. Zhou, T.S. Huang, "Small sample learning during multimedia retrieval using BiasMap," *IEEE Proc. of CVPR*, vol.1, pp.11-17, Hawaii, USA, 2001.

[8] Y.Freund, R. Schapire, "Experiments with a new boosting algorithm," *Proc. of 13th International conference on Machine Learning*, pp.148-156, Bari, Italy, 1996.

[9] Y. Wu, Q. Tian, T.S. Huang, "Discriminant EM algorithm with application to image retrieval," *IEEE Proc. of CVPR*, vol.1, pp.222-227, Hilton Head Island, SC, USA, 2000.