PROBABILISTIC RELEVANCE FEEDBACK WITH BINARY SEMANTIC FEATURE VECTORS

David Liu , Tsuhan Chen
dliu@cmu.edu, tsuhan@cmu.edu
Department of Electrical and Computer Engineering, Carnegie Mellon University, U.S.A.

ABSTRACT

For information retrieval, relevance feedback is an important technique. This paper proposes a relevance feedback technique which is based on a probabilistic framework. The binary feature vectors in our experiment are high-level semantic features of trademark logo images, each feature representing the presence or absence of a certain shape or object. The images were labeled by human experts of the trademark office. We compared our probabilistic method with several existing methods such as MARS, MindReader, and one-class SVM. Our method outperformed the others.

1. INTRODUCTION

Image retrieval systems aim at providing the user the images in database that are similar to the query the user has in mind. Relevance feedback is a technique to let the user interact with the system by giving examples so that the system has more information of what the user needs. The key is how to make the best out of the feedback information. More formally, we define our problem as follows: Given the database images that the user considers similar to a query, rank all database images from most to least similar to the query. It is worth mentioning that the query does not have to physically exist, but can be a concept the user has in mind. All the system needs are the images that the user identified as similar or relevant to the query.

In MARS [1] and MindReader [2], the positive examples were used to infer the query vector as well as the parameters of a distance measure. Later the authors of [3] proposed a way to integrate MARS and MindReader. All of these methods either calculate the weighted Euclidean distance, or the more general ellipsoid distance. Another approach to relevance feedback with positive examples is the one-class Support Vector Machine (one-class SVM) [4], which extended the two-class SVM to

handle the case where only positive examples are available.

This paper is organized as follows. Section 2 describes the features used for relevance feedback. Section 3 details the relevance feedback method we propose. Section 4 describes several other existing methods. Based on the methods described in Section 3 and 4, we show the experimental results in Section 5.

2. IMAGE REPRESENTATION

At the trademark office, the human examiners are judging whether or not a new incoming query image is conflicting with the existing images. While human judgment is subjective and not always consistent from one human to another, the task of judgment is made more objective by specifying design search codes [5][6] to each image.

The design search code is an index which codifies trademark logo images into several categories in a tree-based structure. Each image is coded into at least one leaf of a tree with 3 levels of depth. The first depth is specified by a two-digit number, the second depth by alphabets A to Z, and the third depth by a two-digit number. For example, in Fig.1, the hexagon is codified as 12-D-01, where code 12 refers to geometrical shapes at the first level, code D refers to shapes with more than five sides at the second level, and code 01 refers to shapes with straight edges at the third level. The arrow and the heart in Fig. 1 are coded as 06-T-01 and 12-D-01. For readers interested in the details of coding, please refer to the trademark office documentations [5][6]. The fact that a logo image can be composed of multiple codes means that when one logo image infringes another one, there might be multiple reasons of infringement.

In our experiment, all the logo images in the database have been labeled by the trademark office with design search codes. Based on the design search code, we build feature vectors, which we call high-level semantic feature vectors. They are binary vectors, each dimension being a feature representing the presence (feature value equal to 1) or absence (equal to 0) of a specific object, concept, or shape. The dimension of these vectors is 476,

which corresponds to the number of different design search codes.

3. PROBABILISTIC RELEVANCE FEEDBACK

As mentioned in [7], relevance feedback is useful when the user finds it difficult to formulate pictorial queries. It is also useful for gradually refining the retrieval. As the user points out that the query image is similar to one or several database images, we expect that after the relevance feedback step, the ranked result should change according to the user's concept in mind. Here, we develop a probabilistic framework for relevance feedback.

3.1. Semantic feature relevance feedback

We introduce the following notations:
(a) $\bar{Q}$: the feature vector of the query image.
(b) $\bar{X}_i$: the feature vector of database image i. $\bar{Q}$ and $\bar{X}_i$ are binary vectors, with elements equal to zero meaning the feature is absent, one meaning the feature is present. The dimensions of vectors $\bar{Q}$ and $\bar{X}_i$ are all N.
(c) ~ denotes "relevant", which is defined in the following way: $\bar{X}_i$ is relevant to $\bar{Q}$, denoted by $\bar{Q} \sim \bar{X}_i$, if and only if

$$\sum_k \left(\bar{Q}^k \wedge \bar{X}_i^k \right) \geq 1$$

where superscript k denotes the kth dimension, and $\wedge$ denotes logical AND. In other words, if two feature vectors are relevant, then they share at least one present (non-absent) feature. Given $\bar{X}_i$ and given that $\bar{Q} \sim \bar{X}_i$, then $\bar{Q}$ belongs to a set. For example, assume that $\bar{X}_i = [1 \ 0]$ and that $\bar{Q} \sim \bar{X}_i$, then $\bar{Q} \in \{[1 \ 0], [1 \ 1]\}$.

3.2. Semantic feature inference from one positive example

Given one positive example as feedback, i.e., $\bar{Q} \sim \bar{X}_i$, we will derive the probability of $\bar{Q} \sim \bar{X}_j$ for all feature vectors $\bar{X}_j$, i.e., calculate the conditional probability $P(\bar{Q} \sim \bar{X}_j | \bar{Q} \sim \bar{X}_i)$. Following the definition of conditional probability, we compute the terms $P(\bar{Q} \sim \bar{X}_j , \bar{Q} \sim \bar{X}_i)$ and $P(\bar{Q} \sim \bar{X}_i)$. The computation is explained by this example:

Suppose the feature vectors are four dimensional, such as $\bar{X}_i = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ and $\bar{X}_j = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$.

Then,

$$P\left(\bar{Q} \sim \bar{X}_i , \bar{Q} \sim \bar{X}_j \right)$$

$$= P\left(\bar{Q} = \begin{bmatrix} \times \\ \times \\ \times \\ \times \end{bmatrix} \right) - P\left(\bar{Q} = \begin{bmatrix} 0 \\ 0 \\ \times \\ \times \end{bmatrix} \right) - P\left(\bar{Q} = \begin{bmatrix} 0 \\ \times \\ 0 \\ \times \end{bmatrix} \right) + P\left(\bar{Q} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \times \end{bmatrix} \right)$$

$$= 1 - q_1 q_2 - q_1 q_3 + q_1 q_2 q_3 ,$$

where $\times$ denotes "don't care", $q_k \equiv 1 - p_k, \forall k$, and $p_k \equiv P(\bar{Q}^k = 1)$ is the prior probability for the kth feature of the query being present, which is estimated from the positive example(s). Since the number of positive example(s) is usually smaller than 5, we use the *m-estimate of probability* [8] to provide more robust estimates of the prior probability p_k . This finishes the semantic feature inference from one feedback.

Based on the same reasoning, we have

$$P\left(\bar{Q} \sim \bar{X}_i \right) = 1 - P\left(\bar{Q} = \begin{bmatrix} 0 \\ 0 \\ \times \\ \times \end{bmatrix} \right) = 1 - q_1 q_2$$

which will later be used in Sect. 3.3.

3.3. Semantic feature inference from multiple positive examples

Given n positive examples as feedback, we want to calculate the following:

$$P\left(\bar{Q} \sim \bar{X}_j \middle| \bar{Q} \sim \bar{X}_{i_1}, ..., \bar{Q} \sim \bar{X}_{i_n} \right), \quad \forall \bar{X}_j \in database \qquad (1)$$

and then rank all objects based on the calculated probability. Starting with

$$P\left(\bar{Q} \sim \bar{X}_j \middle| \bar{Q} \sim \bar{X}_{i_1}, ..., \bar{Q} \sim \bar{X}_{i_n} \right)$$

$$= \frac{P\left(\bar{Q} \sim \bar{X}_{i_1}, ..., \bar{Q} \sim \bar{X}_{i_n} \middle| \bar{Q} \sim \bar{X}_j \right) P\left(\bar{Q} \sim \bar{X}_j \right)}{P\left(\bar{Q} \sim \bar{X}_{i_1}, ..., \bar{Q} \sim \bar{X}_{i_n} \right)}$$

we assume the following conditional independence assumption holds:

$$P\left(\bar{Q} \sim \bar{X}_{i_1}, ..., \bar{Q} \sim \bar{X}_{i_n} \middle| \bar{Q} \sim \bar{X}_j\right)$$
$$= P\left(\bar{Q} \sim \bar{X}_{i_1} \middle| \bar{Q} \sim \bar{X}_j\right) \cdot .... \cdot P\left(\bar{Q} \sim \bar{X}_{i_n} \middle| \bar{Q} \sim \bar{X}_j\right)$$

where

$$P\left(\bar{Q} \sim \bar{X}_i \middle| \bar{Q} \sim \bar{X}_j\right)$$
$$= \frac{P\left(\bar{Q} \sim \bar{X}_j \middle| \bar{Q} \sim \bar{X}_i\right) P\left(\bar{Q} \sim \bar{X}_i\right)}{P\left(\bar{Q} \sim \bar{X}_j\right)}$$

and hence yielding

$$P\left(\bar{Q} \sim \bar{X}_j \middle| \bar{Q} \sim \bar{X}_{i_1}, ..., \bar{Q} \sim \bar{X}_{i_n}\right)$$
$$= c \cdot \frac{P\left(\bar{Q} \sim \bar{X}_j \middle| \bar{Q} \sim \bar{X}_{i_1}\right) \cdot .... \cdot P\left(\bar{Q} \sim \bar{X}_j \middle| \bar{Q} \sim \bar{X}_{i_n}\right)}{\left(P\left(\bar{Q} \sim \bar{X}_j\right)\right)^{n-1}} \quad (2)$$

Since

$$P\left(\bar{Q} \sim \bar{X}_j \middle| \bar{Q} \sim \bar{X}_{i_1}, ..., \bar{Q} \sim \bar{X}_{i_n}\right)$$
$$+ P\left(\bar{Q}! \sim \bar{X}_j \middle| \bar{Q} \sim \bar{X}_{i_1}, ..., \bar{Q} \sim \bar{X}_{i_n}\right) = 1, \quad (3)$$

where the symbol "!~" denotes the negation of "~", we can solve c in Eq. (2) by using Eq. (3) and the results from Section 3.2 and finally evaluate Eq. (1).

4. OTHER METHODS

4.1. MindReader, MARS, and related

The MindReader [2] assumes the distance of a relevant database image feature vector $\bar{X}_i$ from the unknown query vector $\bar{Q}$ as follows:

$$dist(\bar{X}_i, \bar{Q}) = (\bar{X}_i - \bar{Q})^T \mathbf{M}(\bar{X}_i - \bar{Q}), \text{ with } \det(\mathbf{M}) = 1.$$

The goal is to find out the optimal $\mathbf{M}$ and $\bar{Q}$ such that the sum of distances from $\bar{Q}$ to all relevant $\bar{X}_i$'s is minimized. It turns out that the optimal $\bar{Q}$ is the centroid of all relevant $\bar{X}_i$, and the weighting matrix $\mathbf{M}$ is the inverse of the covariance matrix of the relevant vectors $\bar{X}_i$.

In the case where the feature dimension is higher than the number of feedbacks, the covariance matrix is singular. The MindReader proposed to use Moore-Penrose pseudoinverse in place of matrix inverse to compute $\mathbf{M}$ from the covariance matrix. [3][9] pointed out potential problems of this approach. Instead, [3] suggested using the MARS [1] approach to circumvent singularity, which can be considered as a special case of MindReader: Restrict the matrix $\mathbf{M}$ to be diagonal,

thereby getting diagonal terms equal to the inverse of the variance of each dimension, i.e.,

$$m_{jj} \propto \frac{1}{\sigma_j^2},$$

where σ_j^2 is the variance in the jth dimension of the relevant feature vectors. The distance measure defined in MindReader then reduces from general ellipsoid distance to weighted Euclidean distance. The idea of using the inverse of variance as a weighting was proposed originally as a heuristic in [1], which has an intuitive explanation: If the jth feature captures the concept or shape the user has in mind, then the relevant feedbacks should have small variance in that dimension.

In image retrieval it is often the case that the number of user feedback is smaller than the feature dimension (which is 476 in our application), in this case MARS, MindReader, and the method proposed in [3] are consistent in using the inverse of the variance as the weights for weighted Euclidean distance. Therefore we will refer to them as the *inverse variance method*.

4.2. One-Class SVM

Schölkopf et.al. [4] extended the SVM to handle training data with only one class. The training data is the set of relevant feature vectors. Manevitz et. al. [10] reported a comparison of one-class SVM with neural networks, nearest neighbor, Rocchio [11] based relevance feedback and naïve Bayes on several document classification tasks. The one-class SVM and neural networks were essentially comparable, and always outperformed the others. As in their experiments, we also used the LIBSVM package [12][13], which implemented the one-class SVM based on [4].

5. EXPERIMENTS

Our quantitative results are based on ground truth data which we collected as follows. We provided the human experts in the Intellectual Property Office (IPO) of Taiwan with a ground-truth data collecting system, and collected ground-truth from the human experts. The human experts were trained in judging infringement based on some rules, which make the task of trademark examination as objective as possible. Hence the data collected from the human experts are considered as ground truth.

The ground truth data collecting system operates in the following way. The system contains about 1200 real world trademark logo images, and each time it presents two images to the human expert, and asks the expert if they are similar (infringing) or not. The collected information is saved as ground truth of whether two

images are similar or not. This ground truth data provides a reliable way to compare and evaluate different algorithms.

To evaluate each algorithm, we did the following. Within the 1200 images, we picked 492 which have 2 or more relevant images, according to the ground truth data. For each of the 492 images, we randomly picked some of the relevant images as feedback, and the rest were used to evaluate precision-recall. The number of images we pick as feedback is also random. We compute the overall precision-recall for these 492 cases. Finally we repeat this process for 30 times to get the mean and standard deviation of precision for each recall level. The result is shown in Fig. 2. The precision-recall curves for the three methods as well as the error bars are shown. We see that the proposed method gives the best overall performance.

6. CONCLUSIONS

In this paper, we described the theoretical and experimental details of a probabilistic relevance feedback technique applied to image retrieval. It was designed exclusively for binary semantic feature vectors. We compared our probabilistic method with MARS, MindReader and related methods, as well as one-class SVM. Our method outperformed the others in precision-recall.

Extending our method to include negative examples will be a future direction.

7. REFERENCES

[1] Y. Rui, T.S. Huang, and S. Mehrotra, "Content-based Image Retrieval with Relevance Feedback in MARS," *IEEE International Conference on Image Processing*, IEEE Press, pp. 815-818, 1997.

[2] Y. Ishikawa, R. Subramanys, and C. Faloutsos, "MindReader: Querying databases through multiple examples," *Proc. of the 24th VLDB Conference*, pp. 433-438, 1998.

[3] Y. Rui and T.S. Huang, "Optimizing Learning In Image Retrieval," *IEEE Intl. Conf. on Computer Vision and Pattern Recognition,* pp. 236-243, 2000.

[4] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, Vol.13, pp. 1443-1471, 2001.
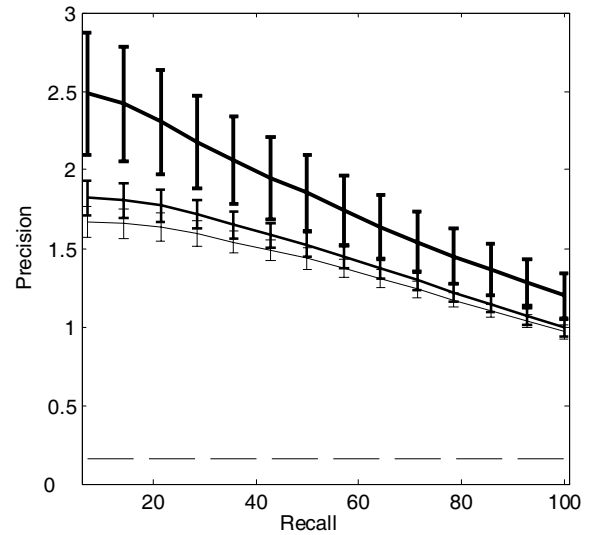
Figure.2: The three curves, from top to bottom, are the precision-recall in percentage of the proposed method, the one class SVM, and the inverse variance method. The bars are error-bars with one standard deviation. The bottom dashed line is the result of random retrieval.

[5] http://tess2.uspto.gov/tmdb/dscm/dsc_01.htm

[6] http://www.tipo.gov.tw/ipotraa/classifiedpathF.html

[7] T. Gevers and A.W.M. Smeulders, *Content-based Image Retrieval: An Overview*, Chap.1, Edited by G. Medioni and S.B. Kang, Prentice Hall, 2004.

[8] T. Mitchell, *Machine Learning*, McGraw Hill, 1997.

[9] X.S. Zhou and T.S. Huang: "Relevance feedback in image retrieval: A comprehensive review," *Multimedia Systems*, Vol.8(6), pp. 536-544, 2003.

[10] L.M. Manevitz and M. Yousef, "One-class SVMs for Document Classification," *Journal of Machine Learning Research*, MIT Press, pp. 139-154, Vol.2, 2002.

[11] J.J. Rocchio, "Relevance feedback in information retrieval," G. Salton (Editor), *The SMART Retrieval System*, Prentice-Hall, Englewood NJ, pp. 313-323, 1971.

[12] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," http://www.csie.ntu.edu.tw/~cjlin/libsvm/, Version 2.33.

[13] J. Ma, Y. Zhao, and S. Ahalt, "OSU SVM Classifier Matlab Toolbox," Version 3.00.

Figure 1. A trademark logo image.