Three-Way Aspect Model (TWAPM) and Co-training for Image Retrieval

Anca Doloc-Mihu and Vijay V. Raghavan

The Center of Advanced Computer Studies, University of Louisiana at Lafayette, USA

ABSTRACT

The goal of this work is to investigate the applicability of two probabilistic approaches, namely Three-Way Aspect Model (TWAPM) and Co-training, to retrieve images represented by multiple feature types from large image collections. We test these approaches in a learning context via relevance feedback from user. Our experiments show that Co-training may be a better choice than TWAPM for a Web-based Image Retrieval System.

Keywords: image retrieval, co-training, aspect model, learning

1. INTRODUCTION

The effectiveness of a CBIR system depends on the choice of the visual features and of the similarity metric that models the user perception of similarity. Since the latter is very difficult to model, the current tendency in the Image Retrieval community is to use both content-based image retrieval and text-based image retrieval to enhance the performance of the Image Retrieval System. Learning and result merging in this context have been a concern of both the Information Retrieval community (see[1] for a survey) and the multimedia database community.[2,3]

In a CBIR system, there is a question of how to combine or to use the information derived about an image based on different feature types (e.g., annotation, color, shape, etc.). In this work, we assume that an image is described by multiple feature types, both textual (annotation) and visual (color). In this context, searching for images involves the mixture of these different feature types. The problem is that, in real web based applications, it is not known a priori the importance (weight) of each feature to the user request. We are interested in effective and efficient modalities of dynamically learning the weights of the feature types necessary to form a final best ranking of the image collection.[4–6] Our final goal is building an effective and efficient Web-based Adaptive Image Retrieval System. Our solution is to employ learning methods which can be adapted to use multiple feature types. The reason is that this may be the best way known so far to combine different feature types without explicitly knowing or calculating these weights.

The goal of this work is to investigate the applicability of two probabilistic approaches, namely Three-Way Aspect Model[7,8] (TWAPM) and Co-training,[9,10] to our image retrieval problem, when we employ learning via feedback from user to automatically deal with the weights learning problem. To our knowledge, similar type of study of TWAPM was done for text retrieval, with different results on real collections (e.g.[10]), but not for images. Whereas Co-training can handle only two feature types, TWAPM can be easily adapted to work with multiple feature types representing the images. In this work, user gives feedback on image(s), i.e. query-by-example approach, as relevant and not relevant (binary relevance).

The Three-Way Aspect Model[7] (TWAPM) was proposed in order to unify collaborative and content-based approaches for recommender systems. This probabilistic includes three-way co-occurrence data among users, documents and document content. In our view, images from our collections can fall under several topics, and are represented by two feature types, annotations and color histograms. Conceptually, a user chooses a topic, which generates images and their content according to topic specific distributions. Similar to,[7,8] model parameters are learned by using EM algorithm to find a local maximum of the log-likelihood of the training data, and then, not seen images are ranked according to the learned information and their content. A characteristic of the Three-Way Aspect Model is that it can handle such model,[8] which motivates us to investigate its applicability to our image retrieval task.

The Co-Training algorithm[9,10] uses two distinct, assumed independent, representations of the entire image collection to learn from seen (by the user) and not seen images. In our case, the two representations correspond to annotations and color histograms in RGB. The algorithm uses the feedback information to incrementally build two classifiers, one for each image representation set. At each iteration, each classifier labels two not seen images, one as relevant and one as not relevant, for which it has the highest confidence.

Researchers[7–10] tested these two approaches for document retrieval on different collections. The applicability of either method seems to depend on the collection characteristics. However, to our knowledge, in case of images a such study is missing. Moreover, one can notice that both algorithms assume that the image representations are independent, an assumption which is not realistic for images. Therefore, to study their applicability to real image collections, experiments are necessary.

We implement and test these approaches on three image collections of sizes 5000 and 10000. Based on our experimental results, we discuss the applicability of either approach to a web-based adaptive image retrieval system in Section 3.1.

The paper is organized as follows. In Section 2, we describe the two approaches as introduced for text retrieval. In Section 3 we present the framework used to apply these approaches to our image retrieval problem, followed by the experimental results and a discussion of the results and their applicability. Finally, Section 4 concludes the paper.

2. PROBABILISTIC LEARNING METHODS

In this section, we describe the two probabilistic approaches that we apply to our merging problem based on learning. The first one, the *Three-Way Aspect Model* (TWAPM), combines collaborative and content-based data in order to make recommendations of the respective documents. The second one, the *Co-training,* uses not seen documents to help the retrieval.

2.1. Three-Way Aspect Model (TWAPM)

In this section, we describe a recommender model, which further we adapt to our Image Retrieval System in order to automatically deal with the weights learning problem.

The Internet offers tremendous opportunities for Web businesses. One of these opportunities is the *recommender systems,* which are automated techniques that suggest products of interest to consumers based on their preferences and based on the product features. Traditionally, there are two types of recommender systems: *collaborative filtering* methods, where the recommendations are made based on ratings from many users, but ignore user and data attributes, and *content-based* (or *information filtering)* methods, which try to match the information about the product to the user's query, and ignore the information from the other users. Lately, there are some tentatives in the literature to combine these two approaches in order to improve the recommendations, but all of them present different drawbacks (for details refer to[7]).

An interesting model which combines the above approaches is the *three-way aspect model* proposed in.[7] This is a generative probabilistic model that incorporates three-way co-occurrence data into the two-way co-occurrence model previously developed by Hofmann and Puzicha.[11] The two way aspect model is based on the *aspect model* proposed by Hofmann.[12] Hoffmann uses mixture models based on latent classes to represent documents and queries. The latent classes can be thought of as language models for important topics of the collection domain. From the retrieval point of view, to incorporate a new representation into the language model involves the estimation of the language model (probability distributions) for the features of that representation, followed by adding the new model into the overall mixture model.[12] The probability distribution builds a global knowledge about the document.

Similarly, the three-way aspect model assumes that users are interested in a set of latent (hidden) topics from which the items and their content information can be deduced.[7] The system learns the parameters of the model by using the expectation maximization (EM) algorithm briefly described below.

Expectation Maximization. The Expectation-Maximization (EM) algorithm is an iterative optimization technique for learning the parameters of a probabilistic model where some of the variables are observed, and some are hidden.[13] The hidden variables might represent quantities that could be the underlying causes of the observed variables. For example, a model designed to retrieve images by using their color and annotation attributes might use their topics as hidden (latent) variables. In this case, the hidden variables are discrete variables, which represent class labels.

The EM algorithm consists of two steps: an expectation step (E-step), which estimates the parameters (in fact, computes the expected value) of the hidden underlying variables based on the observed data, followed by a maximization step (M-step), which provides a new (better) estimate of the model parameters by using the values estimated at the E-step. For short, the first step computes an expectation value for the model parameters, whereas the second step maximizes the expectation obtained in first step. The two steps are alternated until the estimation of the model parameter reaches a local maximum (i.e. the value of the likelihood function does not increase).[13,14]

Note that to compute the likelihood one needs to iterate over all possible data associations of each individual sample, which might be computationally expensive.[13] However, the EM algorithm is guaranteed to be stable and to converge to a local maximum value of the estimated likelihood, which depends on the initial data, but there is no guarantee that this value is a global maximum.[13]

In the following, we present the three-way aspect model in the context of a document recommender system, and then we will show how it can be applied to our color and annotation combination problem in our Image Retrieval System.

Let us assume that users $u \in U = \{u_1, \ldots, u_M\}$ access documents $d \in D = \{d_1, \ldots, d_n\}$, which are described by a set of words $w \in W = \{w_1, \ldots, w_N\}$. That is, an observation is a triple (u, d, w) corresponding to an event of a user u accessing a document d which contains word w. These observations are associated with one of the latent variables $z \in Z = \{z_1, \ldots, z_T\}$. Conceptually, a user choses a topic z, which in turn generates both documents and their content words.[7] A topic represents the (three-way) co-occurrence data among users, documents, and document content. The algorithms is described below:

E step:

$$P(z|u, d, w) = \frac{P(z)P(u|z)P(d|z)P(w|z)}{\sum_{z'} P(z')P(u|z')P(d|z')P(w|z')}. \tag{1}$$

M step:

$$\begin{aligned}
P(u|z) &\propto \sum_{d,w} n(u, d, w)P(z|u, d, w) \\
P(d|z) &\propto \sum_{u,w} n(u, d, w)P(z|u, d, w) \\
P(w|z) &\propto \sum_{u,d} n(u, d, w)P(z|u, d, w) \\
P(z) &\propto \sum_{u,d,w} n(u, d, w)P(z|u, d, w),
\end{aligned}$$

where $n(u, d, w) = n(u, d) \times n(d, w)$ is the number of times user u saw word w in document d, $n(u, d)$ is the number of times user u accessed document d, and $n(d, w)$ is the number of times word w occurs in document d.

The E and M steps are repeated alternately until a local maximum of the log-likelihood

$$L = \sum_{u,d,w} n(u, d, w) \log P(u, d, w), \tag{2}$$

where

$$P(u, d, w) = \sum_{z} P(z)P(u|z)Pr(d|z)P(w|z), \tag{3}$$

is reached.

To recommend documents to users, the system ranks the documents according to

$$P(d|u) \propto \sum_{w} P(u,d,w). \tag{4}$$

If used as is, this model presents the overfitting problem. In the following, we briefly depict some solutions.

To help avoid overfitting and improve generalization, Hofmann[12] proposes the tempered EM, which is a generalization of maximum likelihood for mixture models based on entropic regularization (see[12] for details).

Another approach to overcome the overfitting problem with sparse data is proposed in,[7] where the similarity between items is used to smooth the co-occurrence similarity matrix. Intuitively, in this model if a user didn't see a document d_i but she or he saw a similar (in content) document d_j, then the probability for document d_i is 0, which is not correct because the two documents are similar in content. To overcome this problem, the zero entries of the user-document co-occurrence data matrix are replaced by average similarities (between the respective document and all documents accessed by user) above some threshold. Therefore, this model can make recommendations for new items based on the knowledge inferred from the observed items by using the similarities between these items. These similarities are computed by using the *tf* $\times$ *idf* approach used in text retrieval.[15]

Schein et al.[8] extend the previous work on the three-way aspect model by proposing a set of generative probabilistic models, similar to the three-way aspect model. By analogy to their work, we adapt the three-way aspect model to fit to the characteristics of our similarity based Image Retrieval System (see Section 3.1).

The Three-Way Aspect Model[7] includes three-way co-occurrence data among users, documents and document content. In our view, images from our collections can fall under several topics. In our context, a user gives feedback on a set of images $\mathcal{X} = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_N\}$, which together with the annotations $\mathcal{A} = \{\mathrm{a}_1, \mathrm{a}_2, \ldots, \mathrm{a}_M\}$ and colors $\mathcal{C} = \{\mathrm{c}_1, \mathrm{c}_2, \ldots, \mathrm{c}_S\}$ they contain, form observations $(\mathrm{x}, \mathrm{a}, \mathrm{c})$, which are associated with one of the latent variables $z \in \mathcal{Z} = \{z_1, z_2, \ldots, z_K\}$. We think of the latent variables as they are representing topics. An image might pertain to several topics, but an observation belongs to only one topic (latent variable z). Images, annotations, and colors are assumed independent, given the topic.

Conceptually, a user chooses a topic, which generates images and their content according to topic specific distributions. Similar to,[7,8] model parameters are learned by using EM algorithm to find a local maximum of the log-likelihood of the training data, and then, not seen images are ranked according to the learned information and their content.

2.2. Co-training

Proposed by Blum and Mitchell,[9] the Co-Training method is used successfully for classifying purposes, for example of text documents and web-sites. Its success motivates us to raise the question of whether or not the method can be used (and further, improved) for retrieving images from large image collections when multiple feature types are involved?

The co-training algorithm employs two views of the training (labeled) data, which, in fact, are two separate feature representations of the training data and are assumed conditionally independent given the label. Each view might learn its own classifier over each feature set, and then at each round, they can combine their knowledge to label a small set of unlabeled data (the pool*). In this way, the algorithm learns repeatedly from both labeled and unlabeled data. Co-training stops when there is no unlabeled data left.

Naive Bayes. Most of the co-training algorithms rely on Naive Bayes for classification. A simple, but effective, algorithm used for text classification, Naive Bayes algorithm learns only from labeled data and assumes that each word in a document is generated independently of the other words of the document given the class.[10,16]

Nigam and Ghani[10] use a Laplace smoothing for Naive Bayes to avoid probabilities of zero:

$$P(w_t|c_j) = \frac{1 + \Sigma_{i=1}^{|D|} N(w_t, d_i) P(c_j|d_i)}{|W| + \Sigma_{k=1}^{|W|} \Sigma_{i=1}^{|D|} N(w_k, d_i) P(c_j|d_i)}, \tag{5}$$

*Unless specified, we assume that our method uses a pool of one relevant image and one non-relevant image.

where $N(w_t, d_i)$ is the frequency of feature w_t of document d_i, D is the training set of documents, W is the set of the features, c_j is the class or label given as relevant or non-relevant.

The prior probabilities of each class c_j are calculated as in [10]:

$$P(c_j) = \frac{1 + \Sigma_{i=1}^{|D|} P(c_j|d_i)}{|C| + |D|}. \tag{6}$$

These estimated parameters are used together with the Naive Bayes independence assumption to calculate the probability of each class, for each document. Then, for each document, the most probable class is chosen as being predicted by the classifier.

Probability of a document to be relevant is calculated as:

$$P(c_j|d_i) \quad \propto \quad P(c_j) * P(d_i|c_j) = P(c_j) * \Pi_{k=1}^{|d_i|} P(w_{d_{i,k}}|c_j). \tag{7}$$

Blum and Mitchell[9] provide theoretical learning guarantees for the co-training algorithm under certain assumptions. The first assumption is that the data distribution is compatible with the target function.[10] The second assumption is that the two views of the instance are conditionally independent, given the class of the instance.[10] These assumptions are somewhat unrealistic in practice because the two views describing a document could be somehow related (eg. sky and blue). However, the experiments performed by Nigam and Ghani[10] for text classification showed that the independence assumption influences the performance of the co-training, but the algorithm is still more effective than other algorithms that use unlabeled data. Thus, as[10] noted, "it is an important empirical question to ask how sensitive are co-training algorithms to the correctness of these assumptions".

The Co-Training algorithm[9,10] uses two distinct, assumed independent, representations of the entire image collection to learn from seen (by the user) and not seen images. In our image retrieval setting, the two views are the color and annotation representations of the images from our collections. The algorithm is initialized with the seen images, i.e to which the user gave feedback, and, based on this information, builds incrementally two classifiers, one for each image representation set. At each iteration, each classifier labels two not seen images, one as relevant and one as not relevant. For classification, each classifier chooses the image for which it has the highest confidence. Then, each classifier learns from the new set of seen images. The process is repeated until all not seen images are classified in either relevant or not relevant class. Next, we experimentally test these algorithms on our image collections.

3. EXPERIMENTS AND RESULTS

Since we are dealing with real images, and therefore, with their correlated features' representations, it is expected that the above algorithms might not be capable to achieve good retrieval results. Therefore, in this section, we experimentally study their behavior.

3.1. Experimental Setup

For our experiments, we use three image test collections, two of size 5000, which include 10 and 100 relevant images, respectively, for each query image, and one of size 10000, which includes 100 relevant images for each query image. For convenience, we name these sets as 5000_10, 5000_100, and 10000_100. We use the same set of 10 images as queries (Q_1,Q_2, ..., Q_{10}) for each experiment.

For evaluation purposes, we use the Test and Control method. The images in the three training sets and the testing set are randomly distributed. The number of relevant images within the training and the testing sets for each query is given in Table 1. To evaluate the quality of retrieval, we use the R_{norm} measure, which characterizes the ability of a retrieval method to retrieve the relevant documents before the non-relevant documents[17]:

$$R_{norm} = \frac{1}{2}\left(1 + \frac{S^+ - S^-}{S^+_{max}}\right), \tag{8}$$

5000_10	Train	Test
Q_1	0	4
Q_2	1	4
Q_3	0	7
Q_4	0	7
Q_5	1	7
Q_6	0	4
Q_7	0	5
Q_8	0	4
Q_9	0	2
Q_{10}	0	4

5000_100	Train	Test
Q_1	5	53
Q_2	6	50
Q_3	3	51
Q_4	8	56
Q_5	4	43
Q_6	5	56
Q_7	2	56
Q_8	4	48
Q_9	4	48
Q_{10}	6	50

10000_100	Train	Test
Q_1	0	62
Q_2	0	49
Q_3	4	52
Q_4	3	46
Q_5	2	47
Q_6	1	46
Q_7	3	44
Q_8	4	46
Q_9	1	49
Q_{10}	2	48

Table 1. Number of the relevant images within the training and the testing sets.

where $S^+ = |\{(d_i, d_j)\,|d_i \in \mathcal{R}, d_j \in \mathcal{NR}\}|$, $S^- = |\{(d_j, d_i)\,|d_i \in \mathcal{R}, d_j \in \mathcal{NR}\}|$, and S^+_{max} is the maximum possible number of S^+. $\mathcal{R}$ and $\mathcal{NR}$ are the sets of relevant and non-relevant data, respectively.

The experiments were performed on a Sunn v60x cluster running RedHat Enterprise Linux 4, with 2.8 GHz Intel Xeon Processor and 1G memory per node.

Feature representations. Our image test collections are described by the image color histograms and image annotations. All images are quantized to 256 colors in RGB by using the median-cut algorithm included in the Unix package. The image annotations are obtained by using the procedure described below, which is similar to the one proposed in.[18] Following the observation that the search based on specific keywords is more effective in retrieving relevant documents than a search based on common terms, which appear in many documents, we assume that annotation words are only keywords.

Semi-automatic extraction of annotations. We consider our image collection as being previously organized into folders by some users. In our approach, the hierarchical arrangement of images in directories and sub-directories is used for annotation extraction process. We try to make it easy for the user to give annotations. In this sense, an user can just insert into a directory the image, if he or she considers that the path to the image describes its content. In other words, our annotation of an image contains nothing else but the whole set of words which constitute the folder names of its whole path.

Due to user subjectivity it might happen that at different moments an user or maybe different users might include an image into separate folders. In this case, the respective annotation will include the whole set of folder names, i.e. an image annotation is the union set of all directory names coming from all paths that include the respective image. Two folders with the same name are considered similar, and the annotation will include their name only once, but it will include completely their different paths. Then, the annotations are generated from the names of the hierarchical directories in which the image resides. Note that we consider that the names of the folders are meaningful (there is no folder with the name "it", for example), i.e. they are keywords, and they do not require any further processing (as stemming).

The annotation vocabulary is very conveniently build by collecting all unique annotation words for all images from the collection. The annotation vocabulary acts exactly as the color-map used when working with the color feature. By using the vocabulary we can easily build vector representations for images. For this, we consider that each annotation word from vocabulary represents a term or feature of an annotation vector. To each term occurring in an image we associate a weight equal with the inverse of the number of terms within the annotation. The weight is 0 if the respective term does not occur in the image. The weight quantifies the importance of the term in describing the image semantic contents. These weights are used to compute the similarities between each image from the collection and the user query. By using these weights we can build an annotation histogram like vector for each image.

By creating annotations in this way, we give the annotator more flexibility because she or he can choose more than one subject as relevant for the image. In this case, the annotation of an image will be complete, in the sense that it will capture all semantics given by the same or by different users.

3.2. Discussion

We perform experiments for several feedback steps and we assume that at each feedback step there are 10 images seen by user. For each image test collection and each query, we record the R_{norm} values after each feedback step. Figure 1 shows these values for six queries for the 10000_100 image test collection. The results obtained for the other collections are similar.

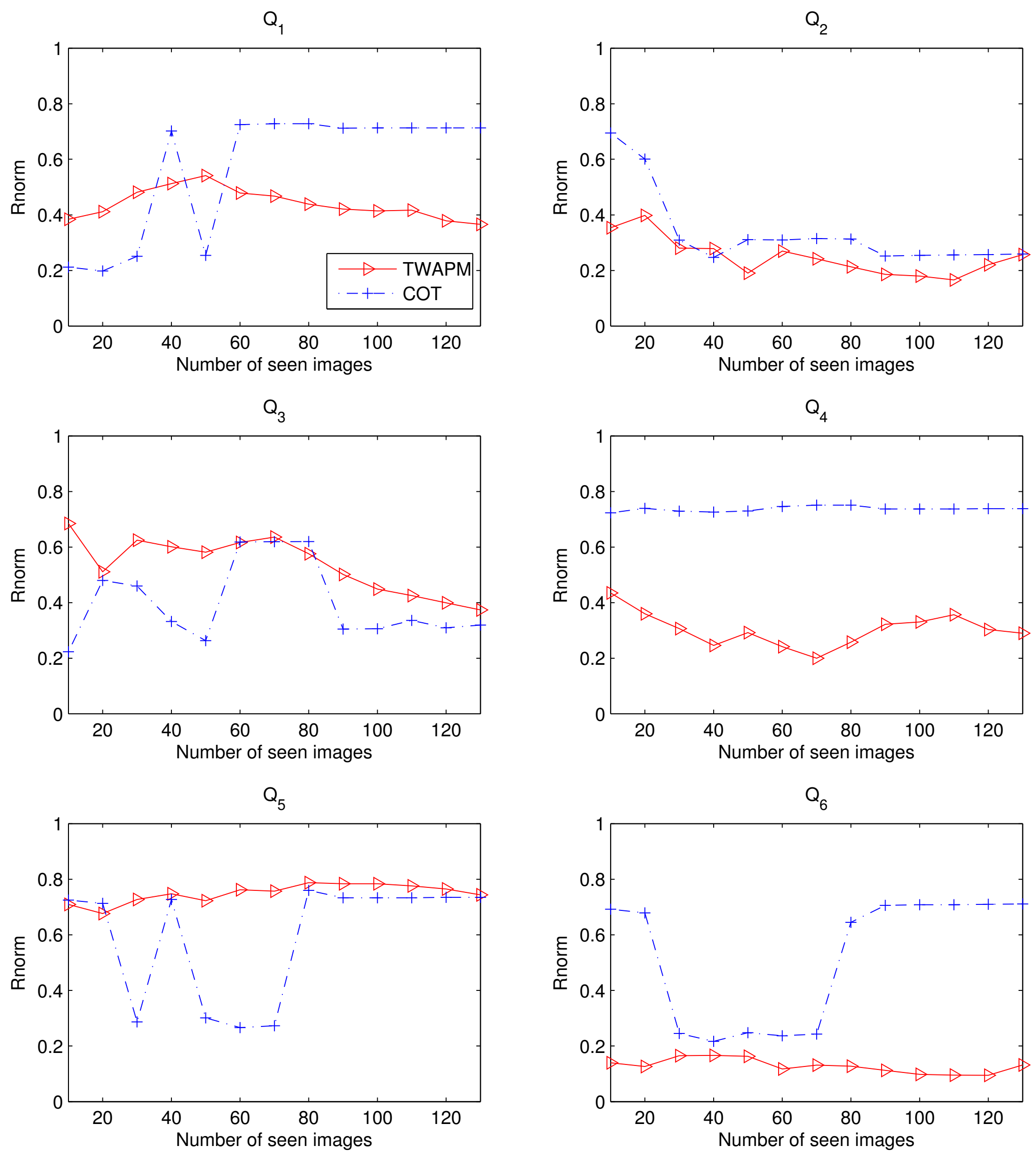

Figure 1. TWAPM and Co-training results for 10000_100 image test collection.

As seen in Figure 1, both methods can learn from both non-relevant as well as from relevant images, but the way they learn is different. As a general observation, the curve displayed by the TWAPM has a more smooth shape than the one displayed by the Co-training. If we compare the results of the two algorithms, we can group the six queries into three main groups:

- Q_2, Q_4, Q_6 - for which Co-training outperforms TWAPM
- Q_3, Q_5 - for which TWAMP outperforms Co-training
- Q_1 - for which one method outperforms the other one for the first several feedback steps, after which they switch.

Further, for each image test collection, we average the results over the ten queries and we present these results in Figure 2. However, if one compares the average values of the two methods over the feedback steps, the differences are significant. As displayed in Figure 2, for all test collections, Co-training outperforms TWAPM with significant differences (of 0.23,0.21, and 0.149 for 5000_10,5000_100, and 10000_100, respectively). Moreover, Co-training gives a retrieval time approximately 10 times faster than the TWAPM.

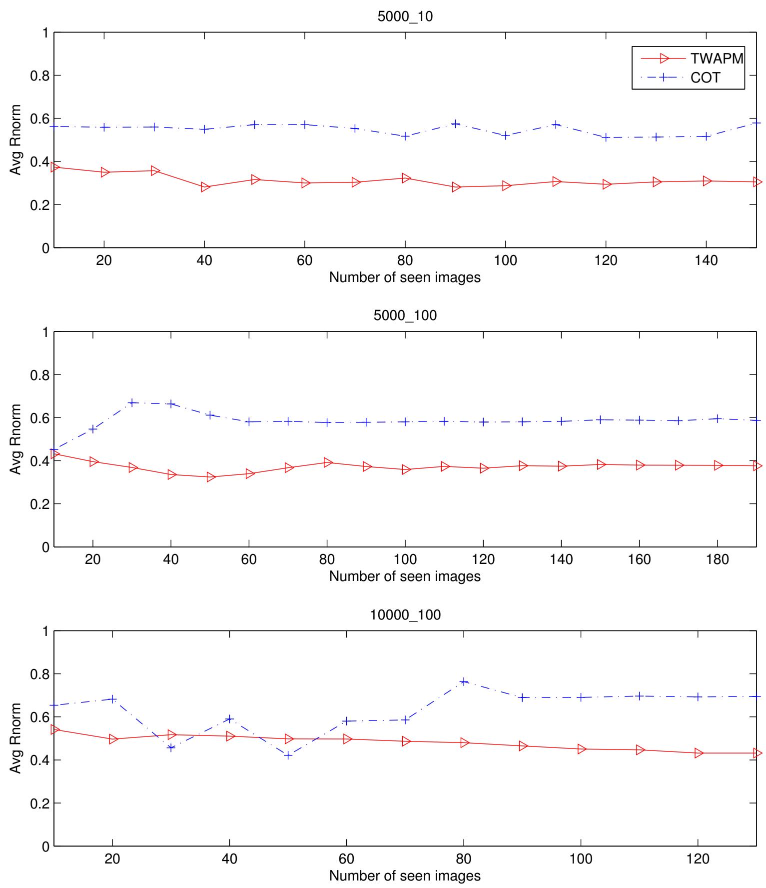

Figure 2. TWAPM and Co-training average results over the ten queries.

These results show us that maybe the Co-training is a better choice for a Web-based Image Retrieval application than TWAPM. The problem with Co-training is that it can only† be used for two feature types

† To our best knowledge.

(corresponding to the two views). Instead, TWAPM can be easily extended and/or adapted to more than two feature types by using the models presented in .[8] However, as given by our experiments, the method shows a low performance if used as it is. Therefore, as future work we will experiment with the improvements proposed by different researchers (see Section 2.1).

For Co-training, these results were obtained when the algorithm labels one relevant image and one non-relevant image at each round. This approach is similar to the one used by Nigam and Ghani .[9,10] However, a such augmentation of the labeled set of images will conduct to a slow, but effective, algorithm which is not suitable for web-based image retrieval applications. A solution could be to have the algorithm label more images (relevant and non-relevant) at each round (a bigger pool of unlabeled data). In this case, the question is what is the best size for this pool, such that the algorithm is both effective and efficient? Next we present our experimental study on this issue.

In the literature, there are several proposals for choosing the size of the pool. In their experiments, Blum and Mitchell [9] use different number of the positive (p=1) and negative examples (p=3). They propose a method to dynamically select the size of the pool as (2*p + 2*n). However, the experiments done by Nigam and Ghani [10] indicated that "the pool did not provide extra benefit, and its removal reduces the number of tunable parameters".

The above motivate us to experiment Co-training with three different pool sizes, as follows:

1. $pool\_size1 = \#Rel + \#NonRel$
2. $pool\_size2 = (1 + \#Rel) + (1 + \#NonRel)$
3. $pool\_size3 = (2 * \#Rel) + (2 * \#NonRel)$

One can notice that all three pool choices have their advantage and disadvantage. The first and the third choices present the same problem as the one mentioned when the are no relevant images in the training set. On the other side, the second choice always will label at least one relevant image at each round regardless the number of relevant images in the training set. This might not be always good. Then, the bigger the size of the pool, the faster the retrieval. That is, the order of the three choices, from the most efficient to the last efficient one, is: choice 3, choice 2, choice 1, and the standard pool used so far, $pool\_size$ of one relevant and one non-relevant image. Figure 3 displays the results obtained for these pool sizes at each feedback step.

Interesting, for 5000_10 collection, the differences between the results obtained for the four pool sizes seem to be very small (less than 0.006). For 5000_100 collection, the order is: the standard pool size, $pool\_size2$, $pool\_size3$, and $pool\_size1$. For this collection, the differences are bigger (around 0.02) than those shown for the 5000_10 collection. For 10000_100 collection, the differences between the different pool sizes are significant (around 0.05). The order in this case is: $pool\_size1$, $pool\_size2$, $pool\_size3$, and standard pool choice. From the above, it seems that for all collections either $pool\_size1$, either $pool\_size2$ outperforms $pool\_size3$ pool size, at the cost of being approximately 30% slower. We believe that this might be a good compromise between efficiency and effectiveness of Co-training. The choice of which of the two ($pool\_size1$, $pool\_size2$) pool sizes is more appropriate for a particular collection seems to be dependent of the respective collection characteristics, and should be made according to the feedback information. If there is relevant information from user, then Co-training should have a pool of at least one relevant image to label (i.e. chose $pool\_size2$). Otherwise, the algorithm maybe should not label any relevant image (i.e. chose $pool\_size1$).

4. CONCLUSIONS AND FUTURE WORK

In a CBIR system, there is a question of how to combine the information derived about an image represented by different feature types. Our solution is to employ learning methods which can be adapted to use multiple feature types without explicitly calculating their corresponding weights.

In this context, the goal of this paper was to investigate the applicability of two probabilistic approaches, namely Three-Way Aspect Model[7,8] (TWAPM) and Co-training,[9,10] to our image retrieval problem, when we employ learning via feedback from user. To our knowledge, similar type of study of TWAPM was done for text

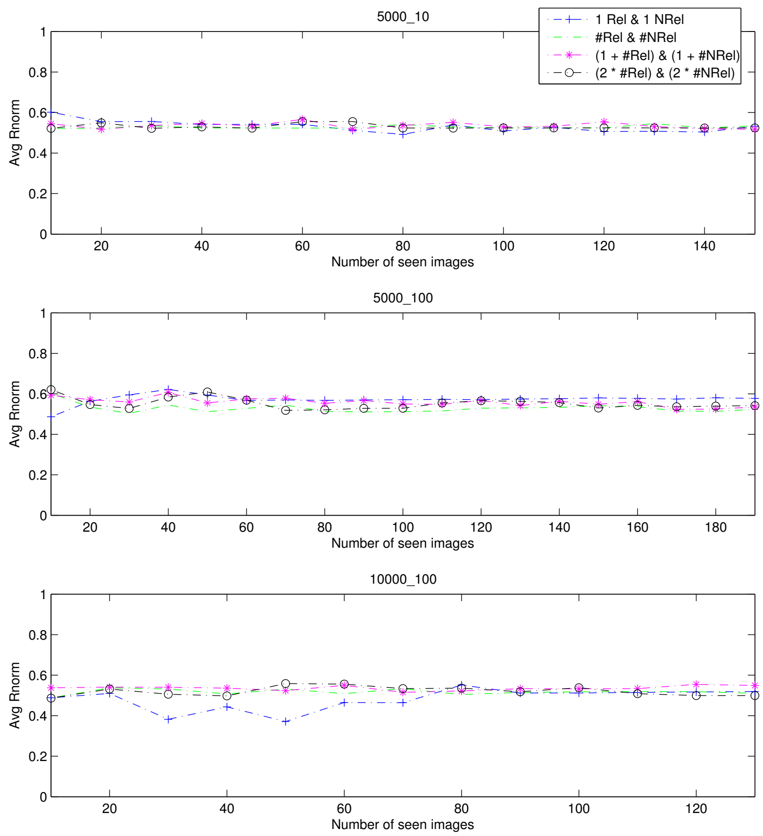

Figure 3. Average results over the 10 queries showing the influence of the pool size for Co-training.

retrieval, with different results on real collections (e.g.[10]), but not for images. Since both methods are based on assumptions that are not realistic for images, to study their applicability to real image collections, experiments are necessary.

We implement and test these approaches on three image collections of sizes 5000 and 10000. Based on our experimental results, we discuss the applicability of either approach to a web-based adaptive image retrieval system. Our results show that maybe the Co-training is a better choice for a Web-based Image Retrieval application than TWAPM. As future work we plan to experiment different improvements proposed by researchers for both approaches.

REFERENCES

1. W. B. Croft, ed., *Advances in Information Retrieval*, ch. Combining Approaches to Information Retrieval, pp. 1–36. Kluwer Academic Publishers, 2000.
2. R. Fagin, R. Kumar, and D. Sivakumar, "Efficient Similarity Search and Classification via Rank Aggregation," in *Proceedings of the 2003 ACM International Conference on Management of Data, SIGMOD*, pp. 301–312, ACM Press, (San Diego, CA), June 9-12 2003.
3. P. Ciaccia, M. Patella, and P. Zezula, "Processing Complex Similarity Queries with Distance-based Access Methods," in *Proceedings of the 6th International Conference on Extending Database Technology, EDBT*, *LNCS* **1377**, pp. 9–23, Springer-Verlag, (Valencia, Spain), March 1998.

4. A. Doloc-Mihu, V. V. Raghavan, and P. Bollmann-Sdorra, "Integration of multiple feature types with adaptive retrieval in vector space model," in *Proceedings of the 27th International ACM SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval MF/IR*, ACM, ed., pp. 1–15, 2004.
5. A. Doloc-Mihu and V. V. Raghavan, "Selecting the Kernel Type for a Web-based Adaptive Image Retrieval System (AIRS)," in *Internet Imaging VII, Proceedings of SPIE-IS&T Electronic Imaging, SPIE*, **6061**, (San Jose, USA), 18-19 January 2006.
6. A. Doloc-Mihu, *Adaptive Image Retrieval System: Retrieval, Learning, Fusion and Visualization.* PhD thesis, University of Louisiana at Lafayette, May 2007. (expected date).
7. A. Popescul, L. H. Ungar, D. M. Pennock, and S. Lawrence, "Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments," in *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence, UAI*, pp. 437–444, (Seattle, WA), August 2-5 2001.
8. A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Generative Models for Cold-Start Recommendations," in *Proceedings of the ACM SIGIR Workshop on Recommender Systems*, (New Orleans, LA), 2001.
9. A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-training," in *Proceedings of the 11th Workshop on Computational Learning Theory,COLT*, pp. 92–100, Morgan Kaufmann, (Madison, WI), July 1998.
10. K. Nigam and R. Ghani, "Analyzing the Effectiveness and Applicability of Co-training," in *Proceedings of 9th International Conference on Information and Knowledge, CIKM2000*, pp. 86–93, 2000.
11. T. Hofmann and J. Puzicha, "Latent Class Models for Collaborative filtering," in *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pp. 688–693, (Stockholm, Sweden), July 31- August 6 1999.
12. T. Hofmann, "Probabilistic Latent Semantic Indexing," in *Proceedings of the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 50–57, ACM Press, (Berkeley, California,), 1999.
13. A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B,* **39**(1), pp. 1–38, 1977.
14. T. M. Mitchell, *Machine Learning*, McGraw-Hill Companies, Inc., 1997.
15. G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, ch. Text analysis and automatic indexing, pp. 59–71, 201–211. McGraw-Hill Book Company, 1983.
16. D. D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval," in *Proceedings of the 10th European Conference on Machine Learning, ECML98*, pp. 4–15, 1998.
17. P. Bollmann-Sdorra, F. Jochum, U. Reiner, V. Weissmann, and H. Zuse, "The LIVE Project - Retrieval Experiments Based on Evaluation Viewpoints," in *Proceedings of the 8th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 213–214, (Montreal, Canada), June 1985.
18. A. B. Benitez and S.-F. Chang, "Perceptual Knowledge Construction from Annotated Image Collections," in *Proceedings of the International Conference on Multimedia & Expo, ICME*, (Lausanne, Switzerland), August 26-29 2002.