This paper appears in: Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries, 2000

Relevance Feedback Decision Trees in Content-Based Image Retrieval

Sean D. MacArthur, Carla E. Brodley, Chi-Ren Shyu
{macarthu, brodley, chiren}@ecn.purdue.edu
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907

Abstract

Significant time and effort has been devoted to finding feature representations of images in databases in order to enable content-based image retrieval (CBIR). Relevance feedback is a mechanism for improving retrieval precision over time by allowing the user to implicitly communicate to the system which of these features are relevant and which are not. We propose a relevance feedback retrieval system that, for each retrieval iteration, learns a decision tree to uncover a common thread between all images marked as relevant. This tree is then used as a model for inferring which of the unseen images the user would most likely desire. We evaluate our approach within the domain of HRCT images of the lung.

1. Introduction

In CBIR, a query is characterized by a feature vector which is then used by the retrieval mechanism to retrieve images from the database that have similar feature vectors. Similarity to the query is computed using either a default or user-defined similarity metric. The most well-known retrieval procedure is the nearest neighbor retriever which retrieves the K neighbors nearest to the query as measured by the Euclidean distance between feature vectors. In the context of image retrieval, a nearest neighbor retriever has at least one drawback: it assumes that all features are equally relevant [3]. If there is a significant discrepancy between the similarity as calculated by the system and the notion of similarity in the user's mind, the results are destined to be unsatisfactory. Certain features or feature subsets may have varying degrees of importance with respect to the user, the query image, and the particular retrieval goals of the user. This problem has served as the impetus for what's known as *relevance feedback*.

Relevance feedback retrieval systems prompt the user for feedback on retrieval results and then utilize this feedback on subsequent retrievals with the goal of increased retrieval performance. (Precision is the ratio of relevant images to the total number of returned images.) To this end, after a set of images is retrieved, the user is given the ability to mark each image as "relevant" or "irrelevant". These user ratings are relayed back to the system, which then attempts to infer which images in the database would be more pleasing to the user by learning from the ratings. A set of images are retrieved, and the process iterates until the user is satisfied with the results. A relevance feedback retriever should possess the following properties:

- Require limited feedback per iteration, as this alleviates the user's time investment.
- Require only one or two iterations to produce fruitful results.
- Be fast enough for an online implementation.

While exact bounds on what constitutes "reasonable" are domain and user specific, we've made a concerted effort to select ranges of these quanitities in our experiments that would encompass a majority of domain and user preferences.

In this paper, we present a relevance feedback retriever that learns decision trees from feedback information. Based on the learned Relevance Feedback Decision Trees (RFDT's), inferences are made about which images the user would most like to see on a subsequent retrieval iteration. We demonstrate how retrieval precision increases after only one or two iterations, requiring that the user provide feedback on only a handful a few images. We demonstrate that large numbers of images retrieved at a time are not a requirement for this performance. We also show that our retriever is fast enough to use online.

This paper is divided into the following remaining sections. Section 2 describes work related to relevance feedback in general. Section 3 delves into the intricacies of our retrieval system at an algorithmic level. In Section 4, our system's performance is compared to that of another recent relevance feedback retrieval system on medical image data. Finally, Section 5 presents conclusions and our future research directions.

2. Related Work

Many relevance feedback retrieval systems of today employ a weighted version of the K nearest neighbor retriever. The weights placed on features are determined by a function of the feedback information. This has a pleasant interpretation, because a larger weight on a feature intuitively signifies that that feature has greater relevance. In MARS (Multimedia Analysis and Retrieval System), developed by Rui, et al [5], a feature's weight is determined by examining the feature's variance across the set of retrieved images marked as relevant by the user. A low variance indicates that these relevant images are consistent in this feature, such that the feature is given a relatively high weight. A feature whose value across the relevant images varies significantly is, conversely, given a relatively small weight. An interpretation of this is that there is no evidence that the common bond between the relevant images depends on that feature. Thus, a feature's weight is assigned in inverse proportion to the feature's variance across the images marked relevant. Documented results of this technique involve experiments where the number of marked images per iteration is considerable, making this a less than desirable technique.

In the the PFRL (Probabilistic Feature Relevance Learning) retriever [3], weighted K nearest neighbor is utilized as well, but the technique used to map user feedback to feature weights is quite different. A feature's weight is computed by examining the C marked images closest to the query with respect to only that feature. The higher the frequency of images marked relevant within this set of images, the higher the weight that is assigned to that feature. Retrieval precision for this technique has also only been documented for a large number of images marked per iteration.

The research above has endeavored to integrate relevance feedback within a weighted K nearest neighbor framework. The work reported upon in this paper represents a departure from this path in terms of both the measurement of similarity and the technique used for encoding the feedback history. Our system doesn't depend solely on the nearest neighbor similarity measure, and our system's memory of feedback history is stored verbatim instead of encoded in feature weights. The precise workings of our system are given in the following section.

3. Retrieval with Trees

A relevance feedback image retriever is a device that takes as its input a query image and a list of K images that have each been marked as either relevant or irrelevant by the user. The cycle defined by the marking of images by the user and the retrieval of images continues as long as the user wishes. (Note that on the first iteration, no feedback information exists.)

We view the task of relevance feedback as a machine learning problem. The solution to the problem lies in defining training data (the query and the marked images), inferring a concept from this data, and producing other instances from a database that are consistent with this concept (returning a set of images). We view this machine learning problem as a two class classification problem which was first suggested by van Rijsbergen [10]. In our development, we employ the following two classes: relevant and irrelevant. The query image is automatically labeled relevant, as it is the standard of relevance against which other images in the database will be compared. The K images are classified according to their relevancy markings made by the user. Thus, a pool of $K + 1$ training instances is established (K marked images, plus the query). We then employ a technique for learning from this training data and producing a set of retrieved images to the user using a learned model.

The algorithm behind our Relevance Feedback Decision Tree (RFDT) retriever operates as follows. On the first iteration, no feedback information exists, so the retriever performs an unweighted K nearest neighbor retrieval. The user then marks the retrieved images as relevant or irrelevant as he or she sees fit. This feedback is relayed back to the system and the second iteration begins. On the second iteration, the algorithm is presented with the $K + 1$ labeled images. Our algorithm doesn't operate on the images themselves but rather the associated feature vectors. From these $K+1$ training instances, we induce a decision tree via C4.5 [4]. A decision tree is a method for recursively partitioning a feature space such that each partition is labeled by a single class value. The criteria for making sequential "cuts" in the space is a product of information theory called "entropy" [7]. The algorithm continues to make cuts until all instances within a partition are of the same class; the partition is then labeled with that class value. The C4.5 routine is executed with the default options, except that we (1) allow leaf nodes to have a minimum of one training instance in each of them and (2) turn off pruning. The next step is to classify the entire database of feature vectors via the learned tree. That is, the tree is used to route each image's feature vector down to a leaf node that has class relevant or irrelevant. When an image is routed to a particular leaf, the unique index of the image in the database is stored in that leaf. Thus, a leaf has a record of all of the images that have been routed to it. (We'll say that images that have been routed to a node are "in" that node.) When all instances in the database have filtered through the decision tree, all the instances that filtered down into a leaf with class "relevant" are assembled into a list. From this list, the K images closest to the query are retrieved by executing an unweighted K nearest neighbor retrieval on the list.) On the next iteration, the retriever's operation is identical to that on the second iteration, except that now there are a total of $2K$ instances of user feedback from which the system will induce a decision tree. (The feedback relevancy markings are retained from one iteration to the next. This retention can be turned off if it better suits the needs of the user.) Similarly, every subsequent iteration allows the retriever to learn from K more images than the previous iteration. This process continues until the user becomes satisfied with the result or until the user's patience is expended.

There are contingencies for certain scenarios that may, and do, arise. If the list of all the images that are in relevant nodes of the tree has fewer than K items, the tree is pruned in the following manner. The images in the deepest relevant node in the tree are merged with that node's sibling's images, and the merged image set is placed in their parent node. The parent node is then labeled as a relevant node and it becomes a leaf. The tree may be pruned as much as needed until at least K images exist in the assembled list of images in relevant nodes. Another possibility is that the tree consists of only one node, and it is irrelevant. This poses a problem, because then the retriever has no pool of relevant images from which to select. Due to the time constraints of this project, we have essentially deferred on this issue for the time being. At this time, we simply relabel the single node as relevant and continue the algorithm. This, of course, will lead to a simple K nearest neighbor retrieval over all subsequent iterations and provide no improvement in retrieval precision over time. This occurred roughly 10% of the time. A more appropriate course of action will be explored in future work.

4. Empirical Results

4.1. HRCT Lung Image Characterization

The image database upon which our experiments are based consists of 1004 high resolution computed-tomography (HRCT) greyscale images of human lungs, each containing at least one form of lung disease such as centrilobular emphysema or bronchiectasis. The database contains images having pathologies in 27 disease classes, but almost one third of these are diagnosed as centrilobular emphysema. Each image has had the subregion or subregions containing disease delineated by a radiologist and each subregion has been labeled with a disease class that uniquely defines the pathology. On average, each image contains about two of these pathology-bearing regions, or PBR's. Image feature extraction is performed local to each PBR as well as on each image as a whole. These two vectors are concatenated and the resultant vector is then is labeled with the disease class. Our database consists of 1916 labeled vectors of this nature. Our feature extraction software extracts 202 features that relate to properties such as greyscale histogram, texture, and geometry of the PBR [8]. We have found that this volume of features hinders retrieval precision, so we've used SFS (sequential forward selection) wrapped around K nearest neighbor to to determine a subset of features that are best for discriminating different disease classes. SFS selected sixteen features. It is this feature subset that is used by the retrievers in the experiments in this paper. Because these features have values on scales from the very large to the very small, all feature vectors are normalized to have zero mean and unity standard deviation before being entered into the database. All queries not originating from the database are necessarily normalized in the same way. This normalization is crucial in preventing certain features from having an initial bias towards being more relevant than other features in the K nearest neighbor computation.

4.2. Experimental Method

An ideal experiment to evaluate the performance of a relevance feedback retriever in this domain would involve a domain expert, a radiologist, as the user. The radiologist would submit a series of queries to the system, provide expert relevancy feedback to the system after each retrieval, and iterate the feedback process a prespecified number of times. The retrieval results could then be logged and evaluated with respect to precision. However, at this point in our early exploration of our retrieval mechanism we have employed an automated approach to evaluation. We first make the observation that we have labeled instances in the database; their disease class is known. If we select queries from the database itself, then the query's class is known as well. Conveniently, this allows us to define the traditional measure of the retrieval precision after any given retrieval as:

$$Precision = \frac{N_{matching}}{K}$$

The numerator is the number of retrieved images whose class matches that of the query and the denominator is the total number of retrieved images. Note that because this is calculable without the need of a domain expert, any number of relevance feedback iterations may be performed on any of the possible queries. Thus, the entire process is automated. It should be noted that this is only valid if we make the assumption that a typical user would be marking retrieved results in this way, that is, marking images whose disease class matches the query as relevant and those whose class is different as irrelevant. One potential use of our CBIR system is for retrieving visually similar images that have pathologies that match the query's disease. Nonetheless, we recognize that there are niche applications of such a system where this assumption may not hold. Future research will involve evaluation on a random sample of the database by domain experts.

Our database contains images from scores of patients, but in numerous cases there are multiple images per patient. (HRCT scans of different cross-sections of the lung are called *slices*. The database often contains multiple slices from a given patient.) It's important to note that our retrieval system is intended for retrieving images that have some similarity to the query *but are not from the same patient.* In our experimental setup, we have modified our retriever to prevent this from happening.

4.3. Experimental Results on Real Data

Our experiments compare our Feedback Decision Trees retriever to the Probabilistic Feature Relevance Learning (PFRL) retriever developed by Peng, et al [3]. PFRL is a weighted K nearest neighbor retriever that adjusts feature weights based on user feedback, and it has been shown to be one of the better retrievers in existence. PFRL is controlled by two parameters: C, a bias/variance tradeoff, and T, which controls the learning rate. The C parameter may take integer values between 1 and $K-1$, while the T parameter may be set to any positive real number. Experimentation proved that a value of $T = 4$ optimized the retriever's precision with respect to our database. The precision is insensitive to the choice of C; it was set to the integer closest to $K/2$ with good results. Our retriever doesn't use any parameters, so no such optimization steps were necessary. Each retriever was run 1916 times, using each instance in the database as a query exactly once. We evaluate two values of K: four and ten. Based on our experience with the two radiologists with whom our group works, we conclude that requiring more than ten images to be marked after each retrieval is overly burdensome. We also conclude that performing more than ten retrieval iterations would be unrealistic. We argue that less than four marked images per retrieval will degrade the performance of either of the two retrievers due to insufficient feedback. While substantial research [3, 6] has demonstrated the efficacy of systems that routinely use $K > 10$, we feel that marking such a large number of images per retrieval is an excessive burden for the user, particularly in our domain. Marking HRCT images with respect to relevance to a query is expensive in time and effort.

The results of the four experiments are displayed in Figures 1 and 2. Figure 1 shows the average precision of each retriever after each iteration with $K = 4$. (These averages are taken across all 1916 potential queries in the database.) Figure 2 shows the same information for $K = 10$. Note that, for a given value of K, the retrieval precision on the first iteration is always the same for both retrievers because both default to a simple unweighted K nearest neighbor on the first iteration.

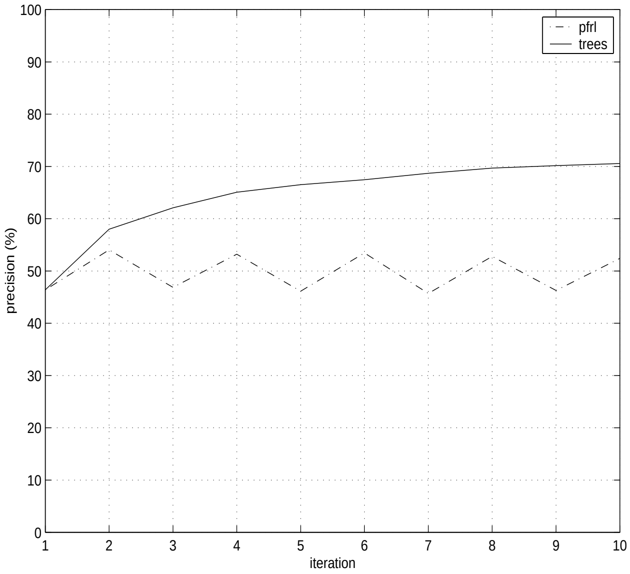

Figure 1. Average retrieval precision ($K = 4$)

The oscillatory behavior of the average precision of the PFRL retriever with respect to iteration indicates that the weights that it is storing are oscillating as well. This could have been dampened somewhat by decreasing the learning rate T. We chose T to be the value that produced the best mean precision results on this domain. The accumulation of knowledge by the RFDT retriever is evidenced by smooth asymptotic behavior of the RFDT precision curve.

Due to the fact that the disease class distribution of our database is very skewed, retrieval precision with respect to each of the classes is not uniform. In fact, in ten of the classes which constitute approximately 10% of the database's images, neither of the two retrievers were ever able to return even one relevant (same class as query) image. This indicates that we lack the necessary features at this time to differentiate these minority classes from the majority classes. This result may depend on the feature subset that was selected with SFS, which attempts to select features with respect to overall classification accuracy, not mean class accuracy. For this reason, SFS tends to be biased toward selecting features that best distinguish the majority class. With K=10, the base K nearest neighbor mean retrieval precision for the majority class is 73.7%. PFRL has a mean precision of 82.4% on this class after ten iterations and FDT has a mean precision of 97.3% on this class after ten iterations. Notice that the difference in retrieval precision with the different K values is slight.

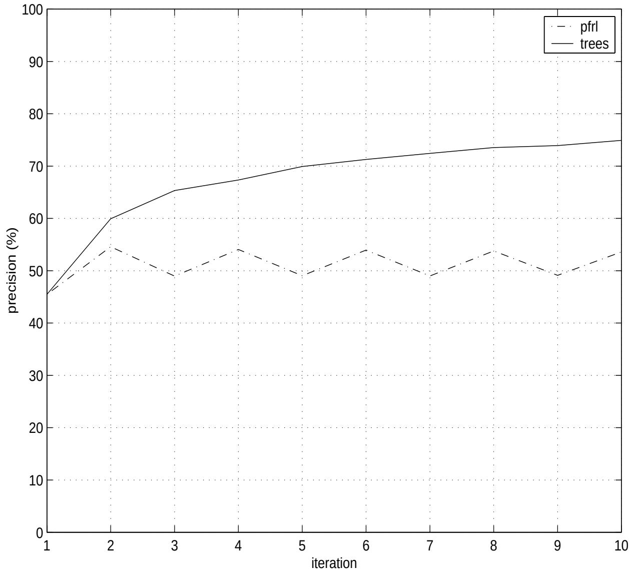

Figure 2. Average retrieval precision ($K = 10$)

Both retrievers are fast enough for online implementation. While the FDT retriever takes approximately 1–2 seconds to perform a retrieval on average, the PFRL retriever is virtually instantaneous by comparison as it takes less than 300 ms to do the same.

5. Discussion and Future Work

The RFDT retriever keeps a complete record of previous relevance feedback and uses this entire history when inducing its decision tree. It stores this information explicitly, rather than encoded in feature weights. The RFDT retriever's model of what the user considers relevant is continually being updated, and, based on this accumulating data, we can infer that the feature subspace defined by the features selected with SFS is being partitioned more and more accurately by the induced decision trees. The FDT retriever has the ability exploit a small number of features in a certain locale of the feature space while ignoring other features entirely. The decision trees induced are only utilizing the features in a particular region *that are useful for differentiating relevant and irrelevant instances.*

Our future work will involve evaluation of the system by a domain expert. We would like to further examine the behavior of the RFDT retriever in a variety of image domains. We will also investigate methods for generalizing the retrieval algorithm to work with relevance feedback that isn't so coarsely quantized. Instead of "relevant" or "irrelevant", we will allow finer gradations of relevancy ratings such as "somewhat relevant" and "somewhat irrelevant".

References

[1] T.P. Minka and R.W. Picard. Interactive learning using a society of models. Technical Report 349, MIT, 1995.

[2] T. M. Mitchell. *Machine learning*. McGraw-Hill, New York, NY, 1997.

[3] Jing Peng, Bir Bhanu, and Shan Qing. Probabilistic feature relevance learning for content-based image retrieval. *Computer Vision and Image Understanding*, 75:150–164, 1999.

[4] J. R. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[5] Y. Rui, T.S. Huang, and S. Mehrotra. Content-based image retrieval with relevance feedback in MARS. In *Proceedings of the IEEE International Conference on Image Processing*, 1997.

[6] Yong Rui, Thomas Huang, Machael Ortega, and Sharad Mehrotra. Relevance reedback: A power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Video Technology*, 1998.

[7] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 1948.

[8] C. Shyu, C. E. Brodley, A. Kak, A. Kosaka, A. Aisen, and L. Broderick. Assert, a physician-in-the-loop content-based image retrieval system for hrct image databases. *Computer Vision and Image Understanding*, 74:111–132, 1999.

[9] Leonid Taycher, Marco La Cascia, and Stan Sclaroff. Image digestion and relevance feedback in the ImageRover WWW search engine. Technical Report BU CS TR97-014, Boston University, 1997.

[10] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.