Content-Based Image Retrieval Using Multiple-Instance Learning

Qi Zhang QZ@CS.WUSTL.EDU
Sally A. Goldman SG@CS.WUSTL.EDU
Wei Yu WEIYU@CS.WUSTL.EDU
Jason E. Fritts JEFRITTS@CS.WUSTL.EDU
Department of Computer Science, Washington University, St. Louis, MO 63130 USA

Abstract

We explore the application of machine learning techniques to the problem of content-based image retrieval (CBIR). Unlike most existing CBIR systems in which only global information is used or in which a user must explicitly indicate what part of the image is of interest, we apply the multiple-instance (MI) learning model to use a small number of training images to learn what images from the database are of interest to the user.

1. Introduction

In this paper we explore the application of machine learning techniques to the problem of *content-based image retrieval* (CBIR), which is the problem of retrieving semantically relevant images from a large image database. In a typical CBIR application, there is a query language used by the human user to specify a set of images that are desired. Unlike most existing CBIR systems in which only global information is used or in which a user must explicitly indicate what part of the image is of interest, our goal is to take a small set of positive and negative examples of what kinds of images the user wants and from them learn what portion of the image is of interest to the user. Our goal is to integrate the learning component with the information retrieval system versus just using the similarity between a single image and those in the database.

As originally done by Maron and Ratan (1998), we propose applying multiple-instance learning since it is naturally designed for settings where there is ambiguity as to what portion of each example (i.e. image) is important. In the *multiple-instance* (MI) learning model, each training example is a set (or *bag*) of instances (points) along with a single label for the bag. Each bag corresponds to an image and each point in the bag corresponds to a sub-region of the image. The assumption made by the MI learning model is that the label given to the bag is a function of a single sub-region in the bag. However, the training data does not provide any information as to which sub-region is the important one. The hypothesis constructed by the learner can then be used to retrieve the desired images from the database.

Our work differs from that of Maron and Ratan (1998) in several ways. We compare the performance of the diverse density (DD) algorithm (Maron & Lozano-Pérez, 1998) and the EM-DD[1] algorithm (Zhang & Goldman, 2001) which we recently developed for the application area of drug discovery. For both of these algorithms, we explore a variety of methods to choose the final hypothesis among those found via multiple searches. In addition, we report on the differences seen empirically when using a variety of image processing techniques to convert the image to a bag.

2. Image Processing Techniques

We now describe the image processing methods that we use in our work. There are two aspects that we vary: the color representation and the feature extractor (or bag generator). Two systems are commonly used to represent color images: RGB (Red-Green-Blue) and YCrCb (Luminance-Chrominance). RGB is the most widely used color system since the eye's retina samples color using only three broad bands roughly corresponding to red, green and blue light. YCrCb is widely used in image and video compression because it reduces the information redundancy and there is no dependence among these three components.

We now describe the methods we use to extract features (which form the instances in the bag) from the raw images. These methods can be partitioned into two categories: (1) smoothing the images to lower resolution and extracting features using a fixed segmentation scheme, and (2) segmenting the images at their original resolution and including an instance (i.e. point

[1]EM-DD combines the DD algorithm with the Expectation-Maximization (EM) algorithm.

in the bag) for each segment. Both methods can be applied using either color representations.

The first method of feature extraction we use is based on the *SBN* (single-blob with neighbors) method of Maron and Lozano-Perez (1997). This method smoothes and subsamples an image to reduce the resolution of the original image to an 8 x 8 image. Each 2 x 2 set of pixels within the 8 x 8 image is called a *blob*. A SBN is defined as the combination of a single blob with its four neighboring blobs (up, down, left, and right). The sub-image is described as a 15-dimensional vector $(x_1, x_2, \ldots, x_{15})$, where x_1, x_2, x_3 are the mean RGB or YCrCb values of the central blob, x_4, x_5, x_6 are the differences in mean color values between the central blob and the four neighboring blobs. Each bag is therefore a collection of nine 15-dimensional points (obtained by using each of the 9 blobs not along the border as the center blob).

For the smoothing and subsampling process in the SBN feature extraction method, we consider both a wavelet transform (Daubechies, 1988) and a Gaussian filter (Gonzalez & Woods, 1992). The wavelet transform is equivalent to passing the image through low-pass and high-pass filters. The Gaussian filter blurs the image removing both detail and noise. We shall compare results using both filter styles.

In the segmentation-based feature extraction method, we start by dividing the image into 4×4 pixel regions called *blobs*. Each blob is represented with 6 values. The first 3 dimensions come from the average color of the segment and the remaining three come from applying a Daubechies-4 wavelet transform (Daubechies, 1988) on the Luminance (Y) component of each blob in the segment to obtain orientation information for the vertical, horizontal, and oblique directions. The wavelet coefficients also help retain texture information. So each blob is represented by either ⟨R,G,B,HL(Y),LH(Y),HH(Y)⟩ or ⟨Y,Cr,Cb,HL(Y),LH(Y),HH(Y)⟩ depending on the color representation selected. Next, the k-means segmentation algorithm (Hartigan & Wong, 1979; Wang, Li, & Wiederhold, 2001) is used to segment the image. To apply MI learning we introduce one bag for each image with one 6-dimensional point in the bag for each segment that is obtained by averaging the 6 values among all blobs in the segment.

The approaches based on subsampling along with SBN feature extraction maintains the spatial relationship between features. For example, if the user likes images that contain mountains, since most mountains have a region in which the mountain top meets the sky, using the subsampling approach with the SBN feature selection tends to work quite well. However, because mountains vary so much in their texture and color (depending on whether there is snow, trees, ...), even when very good segmentation is performed the retrieval results may not be as strong as those obtained when using the spatial relationships. Conversely, the segmentation technique is expected to work well in cases where the user is interested in an object that is unique from other objects based on its color, texture, and orientation. However, it has the drawback that the spatial relationship between segments is not maintained. In our experiments, we explore when each of these approaches works best.

3. Multiple-Instance Learning

In multiple-instance (MI) learning the training data $D = \{\langle B_1, \ell_1 \rangle, \ldots, \langle B_m, \ell_m \rangle\}$ consists of a set of m bags where bag B_i has label ℓ_i. Let bag $B_i = \{B_{i1}, \ldots, B_{ij}, \ldots B_{in}\}$ where B_{ij} is the j^{th} instance in bag i. Let ℓ_{ij} be the label for point B_{ij}. The MI model assumes the label of the bag is determined by the instance in the bag with the highest label. Hence, for Boolean labels, $\ell_i = \ell_{i1} \vee \ell_{i2} \vee \ldots \vee \ell_{in}$, and for real-value labels, $\ell_i = max\{\ell_{i1}, \ell_{i2}, \ldots, \ell_{in}\}$.

The MI learning model was introduced by Dietterich et al. (1997). We now briefly overview the most recent MI algorithms. Maron and Lozano-Pérez (1998) present the diverse density (DD) algorithm. The diverse density at a point p in the feature space is a probabilistic measure of both how many *different* positive bags have an instance near p, and how far the negative instances are from p. Intuitively, the diversity density of a hypothesis h is just the likelihood (with respect to the data) that h is the target. A high diverse density indicates a good candidate for a "true" concept. They then add a scale factor for each dimension, and use a gradient search (with multiple starting values) to find the point that maximizes the diverse density. Other approaches have been proposed. For example, Wang and Zucker (2000) proposed citation-kNN and Bayesian k-NN which are both variants of the k nearest neighbor algorithm.

While most MI learning algorithms assume Boolean labels, recent work studies extensions of the DD and citation k-NN algorithms for data with real-value labels (Amar, et al., 2001) and multiple-instance regression (Ray & Page, 2001). The multiple-instance regression work assumes an underlying linear model for the hypothesis and thus has a different inductive bias. Most recently, we developed EM-DD which combines the DD algorithm with the Expectation-Maximization (EM) algorithm (Zhang & Goldman, 2001).

4. Content-Based Image Retrieval

In this section, we give a brief overview of CBIR systems which is based heavily on an excellent survey by Wang, Li and Wiederhold (2001). Most CBIR systems extract a signature for every image based on its pixel values and then define some distance metric between signatures. Existing general-purpose CBIR systems can be roughly divided into three basic categories with some hybrid systems combining characteristics of each. *Color-histogramming* characterizes each image by measuring the overall distribution of colors in the image. While histograms are relatively insensitive to position and orientation changes, a drawback is the use of only global information about color which ignores information about object locations, shape and texture. Another approach, *color layout*, partitions the image into blocks storing the average color of each block. In some cases wavelet coefficients are used instead of averaging. At proper resolutions, the color layout representation retains shape, location and texture information. This approach is sensitive to shifting, cropping, scaling and rotation because images are described by local properties. The third class of systems, *region-based*, attempt to represent the images at the object level by using some form of image segmentation. However, for region-based systems a human user must then indicate which region is of interest and also typically what aspect (e.g. shape, texture, or color) is of interest. Our work can be viewed as a hybrid method combining color layout with region-based methods but where the learning algorithm replaces the need for the user to specify a region of interest.

Within the CBIR literature, our work is closest to SIMPLIcity (Wang, Li, & Wiederhold, 2001). In fact, the k-means segmentation algorithm we use comes directly from SIMPLIcity. However, the other aspects of our work are quite different. First, the k-means segmentation algorithm is just one of the image processing techniques we consider. Even when using this technique, there are fundamental differences between how SIMPLIcity and our system use the output from the segmentation algorithm. In SIMPLIcity, they define an *integrated region matching similarity metric* that is robust against inaccurate segmentation (see also Li et. al (2000)). They then search the image database to find images that are closest to an image provided by the user using this similarity metric. So SIMPLIcity picks images based on *all* segments in the image. Our approach is designed to discover and then use the *single* segment found to best distinguish between the images. Thus the situations when SIMPLIcity and our work are best suited are quite different.

Our work is most closely related to that of Maron and Ratan (1998). They smooth all images using a Gaussian filter and subsample to 8×8 using the RGB color space. They introduced the SBN feature extraction technique, along with a number of other fixed segmentation techniques. They used the DD algorithm to perform the learning. In their experiments they first trained on somewhere between 10 and 20 initial training examples followed by training with an additional 5 most egregious mislabeled examples as selected by a user. They compared the performance of their system with that obtained using color histogramming and one created using hand-crafted rules for the particular task they studied. Very closely related to the work of Maron and Ratan is the work of Cheng and Lozano-Pérez (2000) in which they studied some variations of the segmentation technique, the similarity metric used in defining the distance between points, and some aspects of the DD algorithm.

The key differences between the earlier work and ours is that we compare the performance obtained when using a wide variety of image processing techniques and all perform tests using a broader range of images. Also, we present results for both the EM-DD and DD algorithms with a variety of methods to pick the final hypothesis from those returned by the multiple starts of the gradient search.

5. The DD and EM-DD Algorithms

We now describe the DD and EM-DD algorithms. The diverse density of hypothesized target point h is defined as $DD(h) = \Pr(B, L \mid h)\Pr(h)/\Pr(B, L)$. Assuming a uniform prior on the hypothesis space, independence of the $\langle B_i, \ell_i \rangle$ pairs given h, the maximum likelihood hypothesis is:

$$h_{DD} = \arg\min_{h \in H} \sum_{i=1}^{n} (-\log \Pr(\ell_i \mid h, B_i))$$

We estimate $\Pr(\ell_i \mid h, B_i)$ using $1 - |\ell_i - Label(B_i \mid h)|$ where $Label(B_i \mid h)$ is the label that would be given to B_i if h were the correct hypothesis. For most applications the influence each feature has on the label varies greatly. Hence the target concept really consists of two values per dimension, the ideal attribute value and the scale value. Maron and Lozano-Pérez (1997) introduced the following generative model for estimating the label of bag B_i for hypothesis $h = \{h_1, \ldots, h_n, s_1, \ldots, s_n\}$:

$$Label(B_i \mid h) = \max_j \left\{ exp[-\sum_{d=1}^{n} (s_d(B_{ijd} - h_d))^2] \right\} \quad (1)$$

where s_d is a scale factor indicating the importance of feature d, h_d is the feature value for dimension d, and B_{ijd} is the feature value of instance B_{ij} on dimension d. Let

$$\mathrm{NLDD}(h, D) = \sum_{i=1}^{n}(-\log \Pr(\ell_i \mid h, B_i))$$

where NLDD denote the negative logarithm of DD. The DD algorithm uses a two-step gradient descent search with multiple starting points to find a value of h that minimizes NLDD. For each positive point in each positive bag, DD obtains a starting point by using the feature values for $h_1, \ldots, h_d$ and sets $s_1, \ldots, s_d$ to 1.0. Among the searches, the hypothesis found with the minimum NLDD value is returned.

We now describe, EM-DD, which views the knowledge of which instance corresponds to the label of the bag as a missing attribute and applies the Expectation-Maximization (EM) algorithm of Dempster, Laird, and Rubin (1977) to convert the multiple-instance learning problem to a standard supervised learning problem. EM-DD starts with some initial guess of a target point h and then repeatedly performs the following two steps. In the first step (E-step), the current hypothesis h is used to pick one instance from each bag which is most likely (given the generative model) to be the one responsible for the label. In the second step (M-step), we use a two-step gradient ascent search to find a new h that maximizes $DD(h)$. We then repeat these two steps until the algorithm converges (based on the NLDD value) and return the point with the minimum NLDD values among those found. We begin EM-DD at every positive point from five randomly selected positive bags (or from all positive bags if there are less than five) with all scale factors set to 0.1. For the application area of drug discovery, EM-DD both improves the accuracy and greatly reduces the computation time as compared to the DD algorithm (Zhang & Goldman, 2001).

Since DD and EM-DD both use gradient-search with multiple starting points, an important component is how to combine the hypotheses generated. We consider several approaches here. The goal of both DD and EM-DD is to find a point in the feature space that minimizes NLDD (and hence maximizes the diverse density). Thus one way to construct the final hypothesis is to directly use the hypothesis obtained with the lowest NLDD value. Assuming all of the assumptions in deriving the NLDD equation held, then this method would pick the best hypothesis. However, we have found that in practice it does not yield the best results. We also considered directly using the hypothesis that gave the best performance on the training data but this method performed comparably to using NLDD. The method we found to work best predicts according to the average prediction among the hypotheses returned by each start of the gradient search. (We refer to this method as the *average* method and denote it by "avg" in our plots.) As a point of comparison, in a few cases we also show the performance of the single hypothesis among those found that gives the best performance on the test data.

6. Empirical Results

In this section we describe our empirical results. Because there are no benchmarks or publicly available labeled data sets for CBIR it is hard to make direct comparisons with other systems such as SIMPLIcity. Furthermore, with the exception of the work of Maron and Ratan (1998) we are not aware of any other work that can retrieve images based on a specific region of interest without having a user specify which region is of interest. Thus, we use Maron and Ratan's algorithm as a baseline to compare the variations we study. (In our terminology their method uses the SBN feature extraction scheme with RGB color and a Gaussian filter, and uses the NLDD to select the final hypothesis.) Maron and Ratan (1998) demonstrated that their technique gave superior results to those obtained when using color histogramming and was comparable in performance to that obtained with hand-crafted rules. We evaluate our work using images from the Corel image suite, images from "Webshots" (www.webshots.com), and the test database from SIMPLIcity (wang1.ist.psu.edu/docs/related/).

There are 5 image groups (mountain, sunset, waterfall, field and flower) in our data sets. We refer to the task of learning to distinguish mountain from non-mountain images as the *mountain task* and likewise for the other image groups. The results we report are all based on a test data set of 720 examples in which there are 120 positive examples (i.e. from the desired image group) and 600 negative examples (150 from each of the remaining four image groups). We have done our preliminary testing for the mountain, sunset, and waterfall tasks since they have the most variety in terms of image content and feature values. Throughout this section, we use "rgb" versus "yrb" (for YCrCb) to indicate the color scheme used. We use "Gau" (and, "Wav") to denote the Gaussian (and, wavelet) filters with SBN feature extraction. Finally, "Seg" is used to indicate the segmentation technique. As done by Maron and Ratan, the training data consists of an equal number of positive and negative examples.

6.1 Comparisons between DD and EM-DD

We begin by comparing the performance of DD with that of EM-DD when using different methods to select the final hypothesis. The plots shown in Figure 1 are averages across 30 runs with a different random selection for the training data from a set of 130 potential

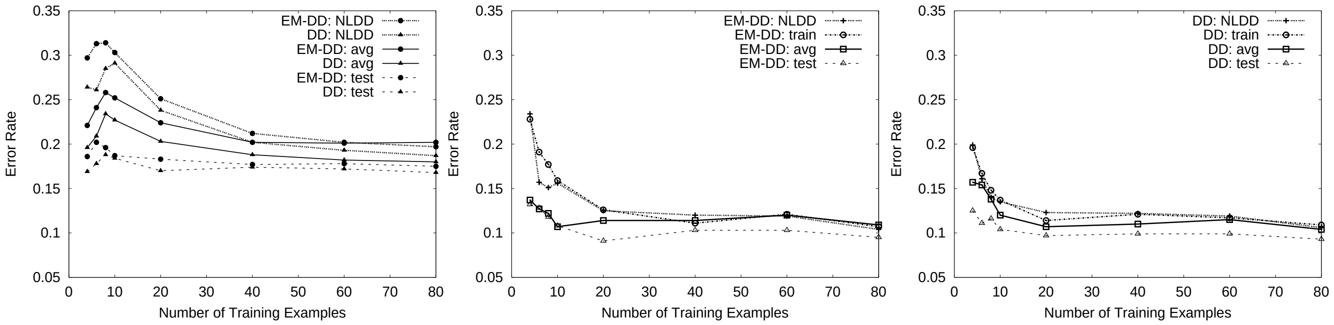

Figure 1. Comparison of the performance of EM-DD and DD algorithms on sunset task (averaged over 30 runs). For the plot on the left the rgbGauSBN image processing method is used. For the second two plots the yrbSeg image processing method is used.

training examples. For these plots, we just use a cutoff of 0.5 (without any tuning) to distinguish between positive and negative. The left plot considers the sunset task when using the rgbGauSBN image processing method (so "DD:NLDD" is the method of Maron and Ratan). Observe that the method used to select the final hypothesis is more important than the choice between DD and EM-DD. While DD generally performs slightly better than EM-DD these differences are well within a standard deviation. We have omitted error bars to reduce the clutter. With 10 training examples, the mean and standard deviation values are $.291 \pm .080$ (DD:NLDD), $.303 \pm .086$ (EM-DD:NLDD), $.227 \pm .057$ (DD:avg), and $.252 \pm .069$ (EM-DD:avg).

The rightmost two plots in Figure 1 consider the sunset task when using the yrbSeg image processing method. The most striking difference is the significant improvement in performance when using yrbSeg versus rgbGauSBN. As we show later, this difference is statistically significant (i.e. the error bars do not overlap). Only about 10 examples are needed to obtain good performance which is important for CBIR since we expect to have a very small amount of training data. When using the yrbSeg processing method, all options considered performed similarly (with all differences well within a standard deviation). For very small data sizes, EM-DD appears not to perform well when using NLDD to pick the final hypothesis (as compared to using the average method) whereas the DD algorithm had less of a difference.

For CBIR, the hypothesis is used to rank all images in the database and then the highest ranked images are those retrieved. Learning curves do not provide information about how accurately the hypothesis ranks images in the database. As common in CBIR, to evaluate the quality of the ranking, we use recall plots (which plots recall as a function of the number of images retrieved) and recall-precision plots (which plots precision as a function of recall) where *recall* is the ratio of the number of positive images retrieved to the total number of positive images in the test set, and *precision* is the ratio of the number of positive images retrieved to the total number of images retrieved. Since there are 120 positive images, the ideal recall plot would grow linearly from $(0, 0)$ to $(120, 1)$ and then remain at 1. The ideal recall-precision plot would have value 1 for all values of recall.

Figure 2 shows recall and recall-precision plots for the sunset task using the rgbGauSBN processing method with 10 training examples. Here the differences between DD and EM-DD are quite small. The difference between methods to select the final hypothesis are also within the error bars (not shown). Figure 3 shows recall and recall-precision curves for the sunset task when using the yrbSeg processing method. For the leftmost figures only 4 examples are used for training yet the performance obtained is better than that obtained with 10 examples using the rgbGauSBN method. For the DD algorithm the methods for obtaining the final hypothesis make less of a difference than with the EM-DD algorithm. Finally, the rightmost plot shows the recall-precision curve with 60 examples where the improvement in performance can be observed with smaller differences seen between the methods.

An important advantage of EM-DD is that it runs roughly 15 times faster than DD. With optimizations we believe that a CBIR system using DD would take 1-2 minutes to respond to a user's request versus only 5-10 seconds when using EM-DD. The performance (measured both via the error rate and using recall-precision curves) between DD and EM-DD are well within a standard deviation and hence for the remainder of this paper we just use the EM-DD algorithm because of its efficiency. Finally, we found that using the average prediction value among the hypotheses works well across all data sets and processing techniques and hence we use it for the remainder of this section.

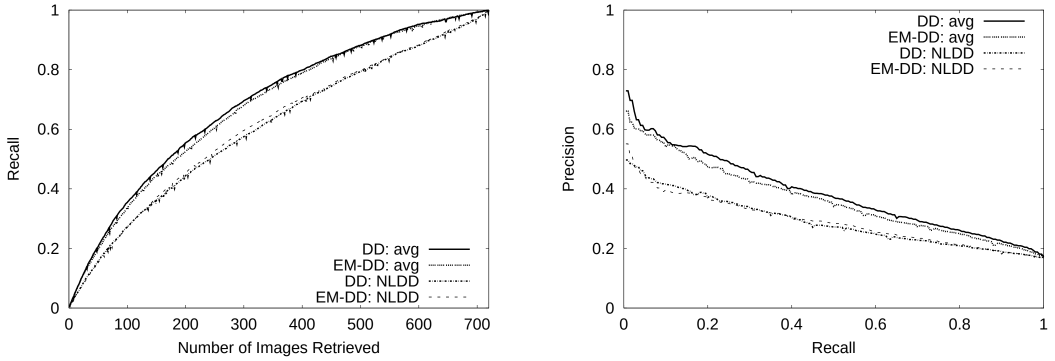

Figure 2. Recall and recall-precision plots for the sunset task using the rgbGauSBN processing method with 10 training examples.

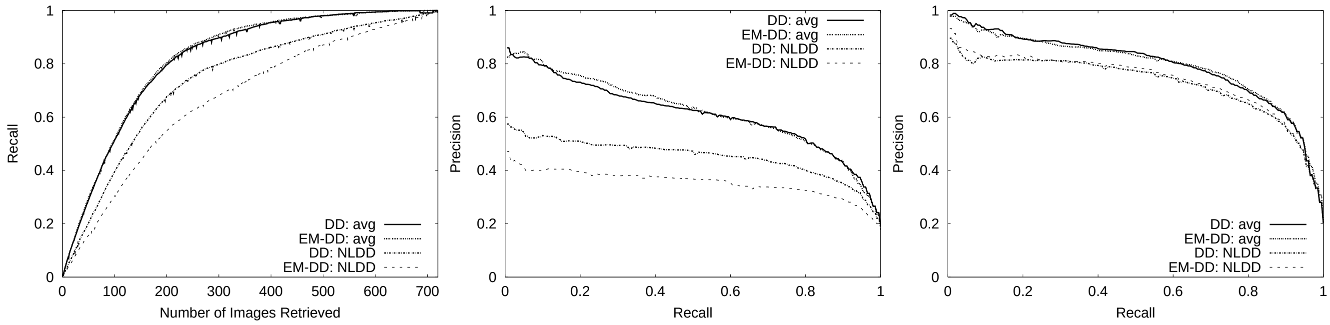

Figure 3. Recall and recall-precision plots for the sunset task using the yrbSeg processing method with 4 examples (left two plots) and 60 examples (right plot).

6.2 Comparison of Image Processing Methods

We now study the performance of EM-DD (with the average method) across different image processing techniques. Table 1 shows our results (averaged 30 runs for the sunset task and 20 runs for the mountain and waterfall tasks). For all columns except "10 opt" we used 0.5 as the threshold between a positive and negative label. For "10 opt" we picked the threshold for which 1/6 of the examples are classified as positive. While one would not expect the learner to know what fraction of the test data is positive, this column demonstrates that improvement in the accuracy can be achieved by tuning the threshold.

For the mountain task, on average, the best performance is obtained using the wavelet filter. The segmentation method is not well-suited since it does not maintain any shape or spatial information which is important in recognizing that an image contains a mountain. Furthermore, the ability for the wavelet filter to represent some texture information is probably valuable. For the sunset task the best result is given using the segmentation approach with significant differences between yrbSeg and rgbGauSBN for 8 or more training examples. Unlike the mountain task, the average color and wavelet information for the segment containing the colors associated with the sunset is unique and hence no shape or spatial information is needed. Finally, for the waterfall task the best results are obtained with either the Gaussian or wavelet filter. (For 40 examples both of these results are significantly better than that obtained using the yrbSeg approach.) As with the mountain task, the segmentation approach was not as effective since the spatial information is important in recognizing which images contain a waterfall. Finally, observe that depending on the task the majority of the performance gains occur after only receiving about 10 images which is very important for the CBIR application.

Figure 4 shows recall and recall-precision plots that illustrate the differences in performance when using the various processing methods for the sunset task when 6 training examples are provided. The rgbGauSBN and yrbSeg methods differ by more than a standard deviation when there are between 100 and 400 images retrieved. The yrbWavSBN and yrbSeg methods differ by more than a standard deviation when there are between 150 and 400 images retrieved. Figure 5 shows the recall-precision curves as a function of the number of training examples and the processing technique. The left plot is for the sunset data set using the rgbGauSBN method and the right plot is for the sunset data using the yrbSeg method. These plots

Table 1. Error rates across the various tasks and processing techniques for the EM-DD algorithm using the average prediction values across all hypotheses. We have shown in bold, results for which the error bars do not intersect.

Task	Processing	Number of Training Examples 4	6	8	10	10 opt	20	40
Mountain	rgbGauSBN	$.23 \pm .05$	$.24 \pm .07$	$.24 \pm .08$	$.23 \pm .07$	$.18 \pm .03$	$.21 \pm .03$	$.21 \pm .03$
	yrbGauSBN	$.24 \pm .07$	$.27 \pm .10$	$.23 \pm .06$	$.25 \pm .08$	$.18 \pm .02$	$.25 \pm .05$	$.23 \pm .04$
	yrbWavSBN	$.19 \pm .04$	$.21 \pm .04$	$.21 \pm .06$	$.22 \pm .05$	$.18 \pm .03$	$.21 \pm .06$	$.18 \pm .04$
	yrbSeg	$.21 \pm .05$	$.25 \pm .07$	$.26 \pm .07$	$.26 \pm .06$	$.20 \pm .02$	$.27 \pm .04$	$.25 \pm .04$
Sunset	rgbGauSBN	$.22 \pm .06$	$.24 \pm .08$	**$.26 \pm .08$**	**$.25 \pm .07$**	**$.21 \pm .03$**	**$.22 \pm .06$**	**$.20 \pm .04$**
	yrbGauSBN	$.19 \pm .05$	$.20 \pm .07$	$.18 \pm .05$	$.19 \pm .06$	$.17 \pm .04$	$.15 \pm .03$	$.15 \pm .02$
	yrbWavSBN	$.17 \pm .05$	$.16 \pm .05$	$.15 \pm .03$	$.15 \pm .03$	**$.13 \pm .02$**	$.15 \pm .03$	$.15 \pm .02$
	yrbSeg	$.14 \pm .03$	$.13 \pm .04$	**$.12 \pm .03$**	**$.11 \pm .03$**	**$.09 \pm .02$**	**$.11 \pm .03$**	**$.11 \pm .04$**
Waterfall	rgbGauSBN	$.23 \pm .12$	$.23 \pm .08$	$.22 \pm .08$	$.25 \pm .11$	$.19 \pm .04$	$.23 \pm .08$	$.19 \pm .03$
	yrbGauSBN	$.27 \pm .13$	$.23 \pm .10$	$.23 \pm .11$	$.20 \pm .05$	$.15 \pm .03$	$.16 \pm .04$	**$.14 \pm .03$**
	yrbWavSBN	$.20 \pm .08$	$.22 \pm .09$	$.21 \pm .10$	$.19 \pm .07$	$.16 \pm .03$	$.17 \pm .04$	**$.15 \pm .03$**
	yrbSeg	$.26 \pm .07$	$.22 \pm .07$	$.21 \pm .06$	$.22 \pm .05$	$.18 \pm .02$	$.22 \pm .04$	**$.21 \pm .02$**

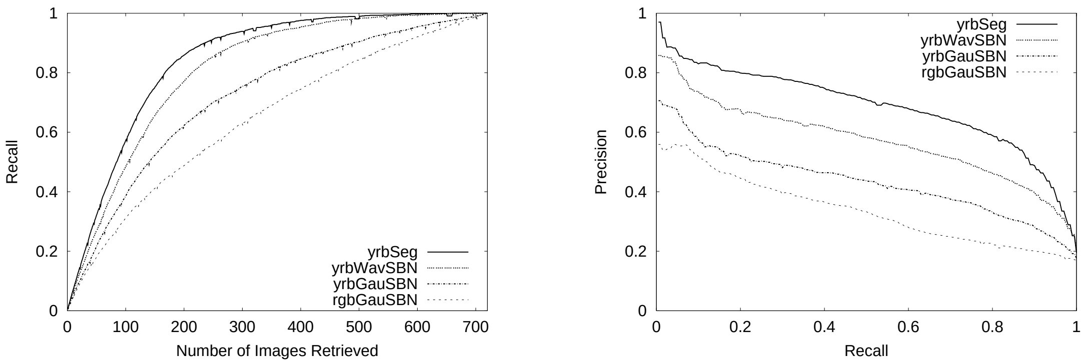

Figure 4. Recall and recall-precision plots for the sunset task when 6 training examples are provided.

demonstrate the importance of selecting the appropriate image processing and segmentation technique. In the plot on the left, significant improvements occur as the number of training images increases. On the other hand when using yrbSeg there are relatively small differences that occur, with the exception of moving from 4 to 10 images for training.

In closing, we want to briefly discuss the differences between DD and EM-DD results found here versus the results we reported on the drug discovery musk benchmark, where EM-DD was not only much faster but learned a hypothesis that had higher accuracy. Recall that the MI model assumes there is a single point in each bag that is responsible for the label. While this assumption is likely to hold for the drug discovery application (with just one shape that is completely responsible for binding), for the CBIR application it is likely that there are several points in the bag which together really explain why the image is desirable. With DD the use of a softmax allows several instances in the bag to influence the DD. However, EM-DD picks just one point from each bag to use for making its prediction. Distinguishing between the five image groups is fairly complex and there are likely to be multiple ways in which one could distinguish between them. When using the single hypothesis that minimizes NLDD then only a single portion of the image can influence the final prediction. In contrast, using the average prediction among all of the hypothesis returned from the multiple searches, allows all important portions of the image (which are those corresponding to the local optimum found during the multiple starts of the search) to influence the final prediction. This theory is also consistent with our findings here that both DD and EM-DD performed better using this averaging combination method. Furthermore, for EM-DD larger improvements were obtained than for DD. Further testing is needed to confirm this theory.

7. Concluding Remarks

There are many interesting directions for future work. We want to explore other methods for feature selection. For example, we are currently performing experiments using the two-blob with neighbors bag generator suggested by Maron and Ratan (1998). We also want to perform more extensive testing including some tests on training data in which the ratio of positive to

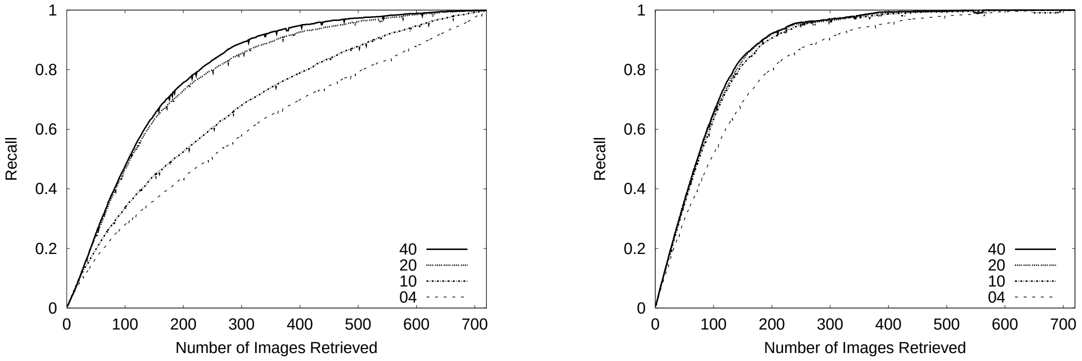

Figure 5. Effect of number of training examples on the recall plots. The leftmost two plots are for the sunset data set with the first using rgbGauSBN and the middle one using yrbSeg. The number given in the key is the number of training examples that were provided. The rightmost plot looks at real vs. boolean for the mountain task for yrbWavSBN.

negative examples are the same as those in the training data. Our eventual goal is to have a system in which the user selects which images from a randomly generated subset of the database are of interest. Then using, these as training data, we can apply our learning algorithm. The resulting hypothesis can then be used to pick the next set of images to show the user and then this process is repeated until the desired images are retrieved. In this setting, we would expect the ratio of positive to negative images to change dramatically over time. Based on some preliminary tests, our algorithms are not sensitive to the ratio of positive and negative examples but much further testing is needed. Finally, as our results have shown, in some cases the segmentation technique works best but in others it does not work well due to the fact that spatial information about the segments is not retained. We have designed a hierarchical segmentation technique that also maintains spatial information and we hope it will perform well across all tasks.

Acknowledgments

We thank Dr. James Z. Wang for making the image database from SIMPLICity available and to Tatdow Pansombut for her help in developing code.

References

Amar, R.A., Dooly, D.R., Goldman, S.A. & Zhang, Q. (2001). Multiple-instance learning of real-valued data. *Proc. 18th Int. Conf. Machine Learning*, (pp. 3–10).

Carson, C. Thomas, M., Belongie, S., Hellerstein, J.M., & Malik, J. (1999). Blobworld: a system for region-based image indexing and retrieval. *Proc. of Visual Information Systems*, (pp. 509–516).

Cheng, Y. & Lozano-Pérez, T. (2000). Multiple-instance learning techniques. *IEEE Int. Conf. on Data Engineering.*

Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Comm. Pure. And Appl. Math.*, **41**, 909–996.

Dempster, A.P., Laird, N.M., & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistics Society*, Series B, **39**(1), 1-38.

Dietterich, T. G., Lathrop, R. H., & Lozano-Pérez, T. (1997). Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, **89**, 31–71.

Gonzalez, R. & Woods, R. (1992) *Digital Image Processing* Addison-Wesley, p. 191.

Hartigan, J.A. & Wong, M.A. (1979). Algorithm AS136: a k-means clustering algorithm. *Applied Statistics*, **28**: 100–108.

Li, J., Wang, J. & Wiederhold, G. (2000). IRM: integrated region matching for image retrieval. *Proc. of the 2000 ACM Multimedia Conf.*, (pp. 147–156).

Ma, W.Y. & Manjunath, B. (1997). NaTra: a toolbox for navigating large image databases. *Proc. of IEEE Int. Conf. Image Processing*, (pp. 568–571).

Maron, O. & Lozano-Pérez, T. (1998). A framework for multiple-instance learning. *Neural Information Processing Systems*, **10**.

Maron, O. & Ratan, A. (1998). Multiple-instance learning for natural scene classification. *Proc. 15th Int. Conf. on Machine Learning* (pp. 341–349).

Ray, S. & Page, D. (2001). Multiple-Instance Regression. *Proc. 18th Int. Conf. on Machine Learning*, (pp. 425–432).

Wang, J., Li, J., & Wiederhold, G. (2001). SIMPLIcity: semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(9), 947–963.

Wang, J. & Zucker, J.-D. (2000). Solving the multiple-instance learning problem: a lazy learning approach. *Proc. 17th Int. Conf. on Machine Learning* (pp. 1119–1125).

Zhang Q. & Goldman, S.A. (2001). EM-DD: an improved multiple-instance learning technique. *Neural Information Processing Systems.*