Image Retrieval for Highly Similar Objects

Anthony Bianchi, Bir Bhanu and Yu Sun
Center for Research in Intelligent Systems, University of California, Riverside, California, USA
abianchi@vislab.ee.ucr.edu, bhanu@cris.ucr.edu, ysun@vislab.ucr.edu

Abstract

In content-based image retrieval, precision is usually regarded as the top metric used for performance measurement. With image databases reaching hundreds of millions of records, it is apparent that many retrieval strategies will not scale. Data representation and organization has to be better understood. This paper focuses on: (a) feature selection and optimal representation of features and (b) multidimensional tree indexing structure. The paper proposes the use of a forward and conditional backward searching feature selection algorithm. The data is then put through a minimum description length based optimal non-uniform bit allocation algorithm to reduce the size of the stored data, while preserving the structure of the data. The results of our experiments show that the proposed feature selection process with a minimum description length based non-uniform bit allocation method gives a system that improves retrieval time and precision.

1. Introduction

Content based image retrieval (CBIR) has made significant progress in the last twenty years. A field that has expressed great interest in this area is taxonomy. Species classification is a significant problem that can reach fields like education, medicine, and environmental assessment. That being said species classification is a daunting task with millions of species existing in the world and many of them can be very similar. For Example, butterflies encompass twenty thousand known species with and estimated half a million species expected to exist. In [5] DNA data is used to find ten new species which were thought to be only one species. The DNA was need since the adult butterflies did not exhibit many differences in morphology. Databases of animals, using DNA as ground truth, are being created and they can potential be used to correctly recognize new and currently known species. From the images in these databases query-by-examples can retrieve similar species to the query and, thus, help in classification. The main aspects of CBIR for highly similar images have been identified as representation, similarity measure, organization, learning, and performance evaluation. In this paper optimal representation is examined.

2. Related Work and Contributions

2.1. Related Work

Indexing structures help in storing data so that it can be efficiently retrieved. Tree Indexing structures like R*-tree [1] and SR-tree [2] are used to store multidimensional data, but they suffer when the number of dimensions becomes large. These structures support quick K-Nearest Neighbor Searches (K-NN), and they are dynamic structures that can have data inserted and deleted efficiently.

A way to deal with the high dimensional problems of the indexing structures is to use feature selection to get rid of non-optimal features. Feature selection helps by finding a feature subset that has less feature correlation. In this paper the effect of feature selection in a tree structure is examined, and it is shown that the reduction in features leads to much more efficient structures while at the same time increasing the precision.

Another approach for efficient retrieval in multi-dimensional indexing structures is reducing the number of bits used for representation of the data. Bit reduction allows for more samples to be put on a page file, but this is not the only benefit it can give. Less overlapping in indexing structures also occurs when bit reduction is used. When the number of bits used to describe the data is too low more overlapping occurs, so a method is needed to find the optimal number of bits. In this paper a minimum description length (MDL) bit allocation approach is used to optimize the representation of the data. Bit allocation is examined for improvement in classification and tree structure representation. The bit allocation chosen by our approach gives higher bit priority to the feature dimensions with higher variance. This allows the data to be more accurately represented even with a low number of bits. In [3], a low bit representation of the images is also looked at, but they used a small number of classes and the classes were not highly similar.

2.2. Contributions

The contributions of this paper are: 1) A feature selection algorithm that performs a forward and

condition backwards search. 2) MDL based non-uniform bit allocation to better represent the data, which automatically selects the optimal number of bits to represent the features. 3) Experimental results on a database of highly similar objects. Evaluation of the effects of the proposed method on a tree indexing structure is examined.

3. Technical approach

3.1. System Overview

The proposed approach (shown in Figure 1) integrates ideas from several areas to produce a data set that follows the MDL principle. This approach first uses feature selection to get rid of features that hinder classification results. After this the new data set with fewer features is sent to a MDL bit allocation algorithm to find the optimal number of bits that describe the system. This is a sequential process starting from 2 bits per feature and iterating until the bits per feature reaches the size of the original features. For each iteration of this process the data set from the feature selection process is sent to a non-uniform bit allocation algorithm. This algorithm gives higher bit priority to features with higher variance. Then the data is put through a K-NN classifier and the precision results are sent back to be plugged into the criterion function. Once all of the bit sizes have been tested the bit size with the highest criteria value is chosen. This modified data set is then put into an indexing structure to be queried at a later time. The indexing structure allows for quick KNN queries and the system proposed here aids in this process.

Figure 1: Overall System Diagram

3.2. Feature Selection Algorithm

Feature selection chooses a subset of the features to use for classification that have better classification results than classification with the whole feature set. The feature selection process [6-9] is shown in Figure2. The first step in the feature selection is Forward Selection. This is where each of the original features is added to the candidate feature set. These are put through a wrapper evaluation, comprised of Bayesian classifier to evaluate performance. The set that has the highest classification results is added to the candidate selection to build the new candidate feature set. Then a conditional backwards search is performed. This process searches in the selected subset and deletes a feature if it is shown to bring down the classification rate. If a feature is deleted, another feature is chosen to take its place.

The above process has a number of iterations equal to the number of features. The features are ranked by the iteration that they joined the feature set. The ranked feature set and corresponding classification rates, at each iteration, are obtained to find the feature subset with the highest classification.

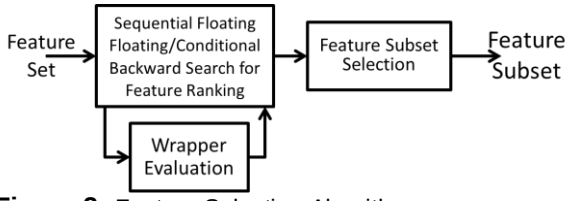

Figure 2: Feature Selection Algorithm

3.3. Optimal Bit Allocation

Originally, each feature was given the same number of bits. After examining the features that were used it was found that some features had more information than others. In [4], a bit allocation method was used to more accurately describe the feature space in a VA-file (a Hash-like multidimensional indexing approach). In quantization theory it has been found that if $\sigma^2_g \geq 4k\sigma^2_p$, then $b_g \geq b_p + k$ (where σ^2 is the variance and b is the number of bits). This allows features with higher variance to have higher resolution. The algorithm for non-uniform bit allocation sorts the variance of each feature and allocates a bit to the highest one. Then the variance of that feature is divided by four and the variance vector is sorted. This process is iterated until the total number of allotted bits is reached.

3.3.1 MDL based Optimal bit allocation. MDL is the notion that a simpler model is better than a complex one [10]. We formulate a function given by equation (1) to find the optimal number of bits to describe the features. Equation (1) automatically chooses the total number of bits to describe the dataset without losing much information.

$$Argmin(k \log B + m \log n) \quad (1)$$

Where *k* is the total number of bits selected, *B* is the total number of bits of the feature sub set given by the feature selection process, *m* is the number of incorrectly classified samples, and *n* is the total number of samples. This function can be described by an example of data transmission. Only the sender knows the correct labels of the transmitted data, and if n is large then the communication costs will be high. So, the sender can tell the receiver the number of bits each feature is described by. There are a total of *B* bits

and log(B) is needed to describe each feature. If only k bits are selected then $k$log(B) is needed to inform the receiver the number of bits per feature. There are a total of n items and log(n) is needed to describe them. There are m incorrectly classified samples, which means $m$log(n) is needed to tell the receiver these incorrectly classified samples. By using these two terms the classification rate term and the description length term balance to find an optimal point. The search space for this process is small so an exhaustive search is used to find the global minimum.

4. Experimental Results

4.1 Data

The dataset that used in these experiments is a subset of the Janzen Butterfly Database (http://janzen.sas.upenn.edu/). This is a database with images of top side views of butterflies, and the subset we used has 9200 individual encompassing 118 species (only species with over 20 individuals were chosen). This data set has species that are very similar, which makes them difficult to classify. Also, this dataset has a high number of features which makes the training a difficult task for many classification algorithms. For example, the Astraptes fulgerator species was thought to be a single species, but after DNA analysis there were 10 species found [5] (adult morphology analysis did not reveal all the species). Many of the species in the database had their species determined using DNA. There were a total of 21 features obtained from the images, which are: 1) RGB mean and standard deviation 2) Mean and standard deviation of Gabor filters at 4 orientations in steps of 45 degrees 3) the first 7 central geometric moments.

Figure 3: Sample images of the database, species are: Astraptes SENNOV, Astraptes INGCUP, Automeris ZuganaDHJ01.

4.2 Results of Feature Selection

After the data went through feature selection 9 (out of 21) features was chosen to be optimal. The features that were chosen are 6 of the color features and 2 of the shape features. We see in Figure 4 that precision increased with the selected feature sub-set, while at the same time reducing the storage space to less than 43% of its original size. We also see in Figure 4 that the probability of the correct class being in the top n results increase. The images in the top n results are usually sent to a rigorous recognition algorithm, so this performance increase may be more important that the increase in K-NN precision.

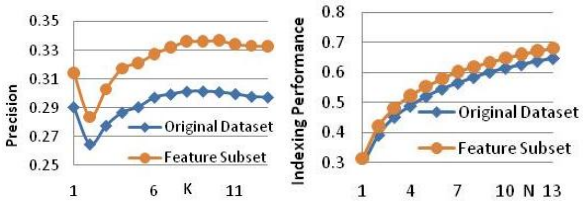

Figure 4: Left: precision for KNN classification averaged over all tests. This shows that classification performance is better with the selected features. Right: Indexing performance (correct result in the top N). This shows indexing performance also increases with feature selection.

4.2.1 Effect of Feature Selection.

An indirect way to show advantage of feature selection is to show its effectiveness in image segmentation. In Figure 5, (a) is the image segmentation results by symmetry-based region growing algorithm [11], using all 21 dimensions of original features to compute the pixel similarity, while image (b) is the segmentation results using the 9 selected features, and it's regions are improved. The improvement of the segmentation performance is 2.1% for this image.

Figure 5: (a) No feature selection; (b) with feature selection. Note the segmented areas are larger using feature selection.

4.3 Results of Optimal Bit Allocation

The feature subset from the Feature Selection was then sent through the MDL based bit allocation. Equation (1) was used to pick the best number of bits to describe the data.The optimal number of bits was found to be a total of 72 bits (8-bits per feature) non-uniformly distributed through the 9 features. The K-NN precision results in Figure 6 show that the chosen bit reduction does not adversely affect classification.

4.4 Indexing Performance Evaluation

4.3.1 Effect of Non-Uniform Bit Allocation on Indexing. To evaluate a more realistic data retrieval process an R*-tree [1] is used as an indexing structure for the data. The R*-tree is a basic multidimensional indexing structure that stores data in Minimum

Bounding Rectangles (MBR). These MBRs are put into a tree structure that allows for KNN queries that do not search the entire data space. The problem with the R*-tree is when the number of dimensions increase the amount of overlapping among MBRs grows, which leads to more search IOs. For the experiments in this paper we verified that finding the optimal number of bits does not increase the number of IO operations in the KNN search of the R*-tree. The R*-tree is a widely used multidimensional indexing structure that has properties of many other tree indexing structures, so the results obtained here should extend to other structures. To look at the structure of the trees created by varying the number of bits, the number of points allowed at each node of the tree is set to be a constant. This is because we know reducing the bit will have lower IOs due to the fact that more data points are in a page file, but we want to look at the actual structural changes of the tree.

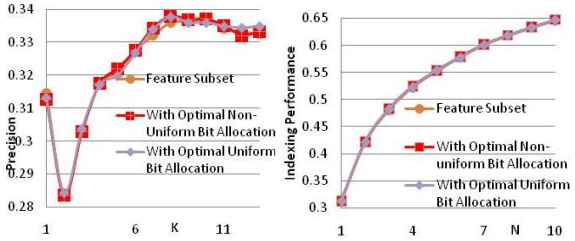

Figure 6: Left: precision for KNN classification averaged over all tests. This shows that classification performance is not affected by bit reduction chosen by the minimization of the MDL criterion function. Right: Indexing performance (correct result in the top N). This shows there the optimal bit allocation does not hurt the performance.

	Original Features	Selected Features	Optimal Uniform Bit Allocation	Optimal Non-Uniform Bit Allocation
Average IO	250.1	79	78.3	**76.5**

Table 1: Average IO cost in the R*-tree. This test only takes into account the structure of the tree. If the size is also taken into account the optimal non-uniform bit allocation would be about 10% of the shown value.

4.3.1 IO Cost Evaluation: K-NN with k=10 was used to in the indexing IO evaluation. From Table 1 we see that the IO cost for 21 features is over 300% higher than for the 9 selected features. Also, in Table 1 it is apparent that allocating 8 bits with feature selection has the least amount of IO cost. Over millions of queries the average savings from the proposed approach will become increasingly significant. Note that what is being measured here is the change in tree structure only since each node is not dependent on the size of the data. Taking into account the size reduction in the features for the optimal bit allocation the average IOs would only be 10% the average IOs given in Table 1.

5. Conclusions

This paper shows that feature selection with optimal non-uniform bit allocation improves precision, IO cost in tree indexing structures, and size of the Database. These improvements allow the databases to scale to large sizes while being efficient, since the indexing structure allows KNN searches to search a fraction of the database and the proposed approach aids this function of the indexing structure without losing precision. This process has been shown to not only save space using bit allocation but to create a better structure for tree indexing. The MDL based criterion function allows for the optimal number of bits to be automatically chosen. The actual size of processed datasets is less than 11% of the original dataset, and if this is taken into account the IO costs will be dramatically better. The importance of knowing the data space has been shown, and by doing so improves the overall performance of the system.

References

[1] N. Beckmann, H.-P. Begel, R. Schneider, B. Seeger, "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles." SIGMOD Conference, 1990: p. 322-331

[2] N. Katayama, S. Satoh., "The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries."ACM SIGMOD International Conf. on Management of Data, 1997.

[3] A. Torralba, R.Fergus, Y. Weiss, "Small codes and large image databases for recognition," IEEE CVPR, 2008

[4] H. Ferhatosmanoglu, E.T., Agrawal, A. El Abbadi, High dimensional nearest neighbor searching. Information Systems, 2006. 31(6): p. 512–540.

[5] P.D.N. Hebert, E.H.P., J.M.Burns, D.H.Janzen, W. Hallwachs, "Ten species in one: DNA barcoding reveals cryptic species in the neotropical skipper butterfly Astraptes fulgerator." Proc Nat. Acad. Sci. USA 101:14812-14, 2004.

[6] H. Wei, S.A Billings.Feature Subset Selection and Ranking for Data Dimensionality Reduction. IEEE PAMI, Jan 2007.

[7]H. Peng, F.Long, C. Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy." IEEE PAMI, Vol. 19, No. 2, Feb 1997

[8] D. Ververidis, C. Kotropoulos. "Fast and Accurate Sequential Floating Forward Feature Selection with the Bayes Classifier Applied to Speech Emotion Recognition." Signal Processing, Vol. 88, No. 12, 2008.

[9] W. H. Hsu. "Genetic wrappers for feature selection in decision tree induction and variable ordering in Bayesian network structure learning." Int. Journal on Information Science, Vol. 163, No. 1-3, June 2004.

[10] J. Rissanen, "A universal prior for integers and estimation by minimum description length," Annals of Statistics,1983

[11] Y. Sun and B. Bhanu. Symmetry-integrated Region Based Image Segmentation. *IEEE CVPR*, 2009