MP1: Metric Learning

Group: Joey Byrnes, Ojus Deshmukh, Akhil Alapaty

I. Introduction

In this MP, we used the mean image-feature vector of the positively marked images (q_c) along with a weighted relevance feedback matrix to compute the Mahalanobis distance between the selected query images and the database images. A feature vector is a vector that contains quantitative information about an image's appearance. The mean image-feature vector is more useful to us than the raw pixels of the image, because it allows us to quantitatively analyze the properties of the image so that we can compare it to the feature vectors of other images. The Mahalanobis distance computes the distance between a point and a distribution. The distance function takes into account the mean image-feature vector, database vector, and the weighting matrix as arguments in order to calculate the distance.

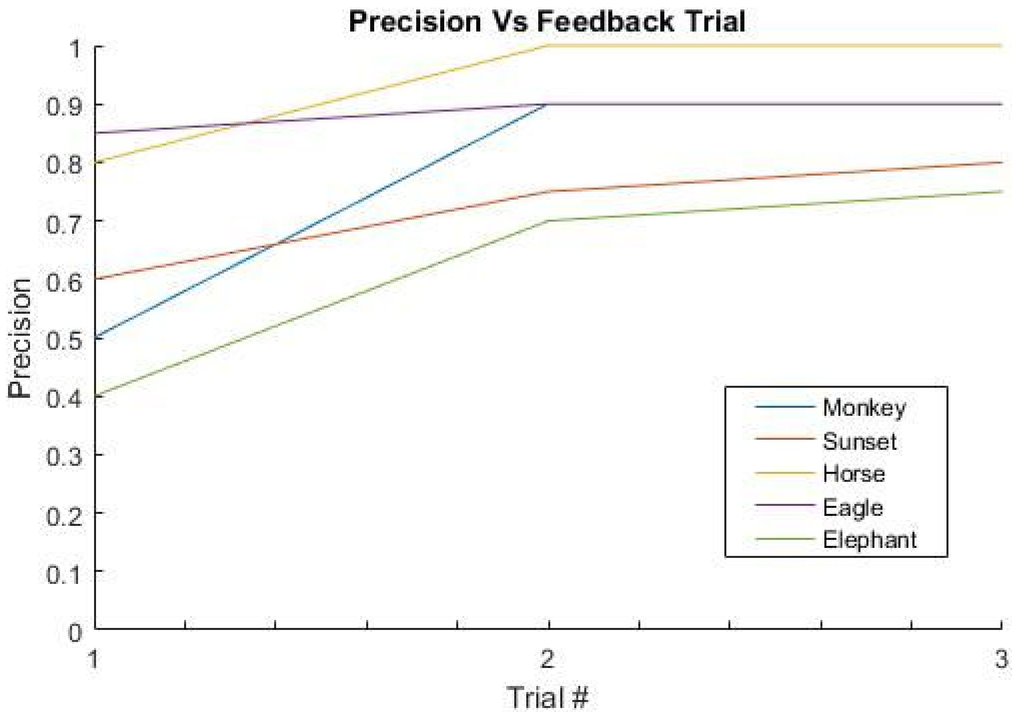
The program works by first taking the feature vector(q_c) of the chosen, or random, seed image. After the user then clicks "QUERY", it provides 20 images for the user to select from. The user then marks similar images as positive. Once the user clicks "QUERY" again, the program takes the mean of all the positively marked images' feature vectors to create a new average feature vector. This selection happens three times, and the algorithm continuously improves with each iteration, as is visible from the included graphs in our results section. The user can choose to specify the image however they want - for example, one of our seed images was a picture of a sunset. However, to better test our algorithm, we only chose images of the sunset in which the sun was clearly visible.

II. Methods

The Mahalanobis Distance takes into account the combination of a query centroid and weighted relevance matrix. The Query Centroid is calculated as the mean of the feature vectors of the positively selected images. We calculate this value in our code on lines 149 to 152 of cbirMP.m. First, we declare a new matrix and load in all of the feature vectors of the previously marked positive images as the columns. Then, we take the mean of every row in the matrix such that the resulting vector is the mean feature vector q_c of the positively marked images. Please refer below to figure 1 for a mathematical representation of the q_c vector. The second major transformation we use in recognizing the images is the weighted relevance matrix. The weighted relevance matrix is a diagonal matrix calculated from the standard deviation of the positive selected images This weighted matrix is calculated in our code throughout lines 222 to 235 in cbirMP.m. Please see figure 2 below for a mathematical representation of this transformation. Finally, putting these two classifiers together and calculating the Mahalanobis distance between the query images and the database images is the core of this MP. Our calculation of the Mahalanobis distance is on line 163 of cbirMP.m. Please see figure 3 below for a mathematical representation of this distance.

Figure 1: Calculating the Query Centroid	Let X be the Query Centroid (q_C) Let M be the Matrix of feature vectors (Meta_data) Let N be the number of positively labelled images. $X_j = \left(\sum_{i=1}^{N} M_{i,j} \right) / N$
Figure 2: Calculating the Weighted Matrix	Let σ be the variance of each row vector of M (See M above) $W_{i,i} = \frac{1}{\sigma_i^2 + 0.0222}$
Figure 3: The Mahalanobis Distance	Let q_C be the query centroid. Let X_j be the feature vector of the database image. $D(j) = (q_c - x_j)^T W (q_c - x_j)$

III. Results



IV. Discussion

Let us take the numbers from the graph for the Monkey Plot. Trial #1 had a precision of 0.5, and Trial #2 had a precision of 0.9. In Trial #1, the feature vector was only based off of the seed image, so it was unable to identify pictures that were significantly different from the seed. Additionally, the weighted matrix was only an identity matrix, since we have not yet collected enough data to compute the variance of positively marked feature vectors. In Trial #2, we have now positively marked ten new images, which allowed us to calculate a relevant weighted matrix for more accurate recognition.

1. What is the general trend of precision versus interaction round? (Or, what should it be?)

Generally, our code generated a successful increasing rate of precision for the 5 images. As expected, the algorithm showed a stronger correlation in terms of pictures resembling the seed picture as our trials progressed. The initial trial depending only on the feature vector of the query image generally started at a precision close to 0.5. As the weighted matrix and the query centroid accumulated positive data, the precision rate consistently increased and in some cases the model displayed 90% accuracy. Sometimes, the code we wrote worked perfectly because the query image would return a 100% accuracy halfway through the trials.

2. Why do you think certain plots are different than others? (from an image representation perspective)

Some precision plots differ from others because of multiple variables. For example, the way the algorithm works is highly dependent on what the user deems “similar” to the seed picture. Some might select a picture because its general RGB values resemble the seed picture, while some might choose a picture because the subject is the same as the original image. These differing reasonings could be a factor in different plots for the respective images. In addition, we also noticed that in some cases, there were less instances of certain query images compared to others. This lead to a higher difficulty in those similar images being queried, which reflected in a slower increasing precision rate. In addition, some types of images are easier for the detection algorithm to recognize because they all have similar objects with clear edges such as the bird pictures, while other types such as the elephant pictures do not have nearly as much contrast or similarity between pictures.