Clustering With EM and K-Means

Neil Alldrin
Department of Computer Science
University of California, San Diego
La Jolla, CA 92037
nalldrin@cs.ucsd.edu

Andrew Smith
Department of Computer Science
University of California, San Diego
La Jolla, CA 92037
atsmith@cs.ucsd.edu

Doug Turnbull
Department of Computer Science
University of California, San Diego
La Jolla, CA 92037
dturnbul@cs.ucsd.edu

Abstract

Two standard algorithms for data clustering are expectation maximization (EM) and K-means. We run these algorithms on various data sets to evaluate how well they work. For high dimensional data we use random projection and principal components analysis (PCA) to reduce the dimensionality.

1 Introduction

The K-Means algorithm finds k clusters by choosing k data points at random as initial cluster centers. Each data point is then assigned to the cluster with center that is closest to that point. Each cluster center is then replaced by the mean of all the data points that have been assigned to that cluster. This process is iterated until no data point is reassigned to a different cluster.

EM finds clusters by determining a mixture of Gaussians that fit a given data set. Each Gaussian has an associated mean and covariance matrix. However, since we use shperical Gaussians, a variance scalar is used in place of the covariance matrix. The prior probability for each Gaussian is the fraction of points in the cluster defined by that Gaussian. These parameters can be initialized by randomly selecting means of the Gaussians, or by using the output of K-means for initial centers. The algorithm converges on a locally optimal solution by iteratively updating values for means and variances.

2 Low Dimensional Data Clustering

For the first part of our project, we implemented the EM and K-Means algorithms. Our implementations were tested on two sets of two-dimensional data: a distribution generated by two Gaussians and an annulus-shaped distribution.

2.1 K-Means on Two-Dimensional, Two Gaussian Data

The K-Means algorithm works very well on this data set, effectively converging in three or four iterations (see figure 1).

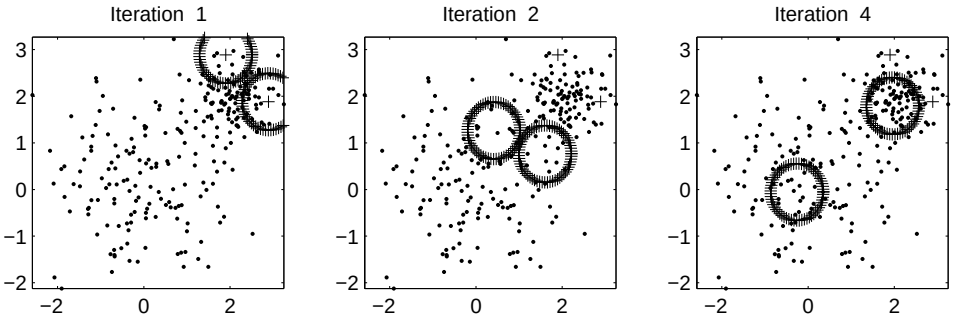

Figure 1: The progress of the K-Means algorithm with $k = 2$ and random initialization on the two-Gaussian data set (note: some data points omitted for clarity).

2.2 EM on Two-Dimensional, Two Gaussian Data

The EM algorithm also performs well, typically converging within 5 iterations (see figure 2).

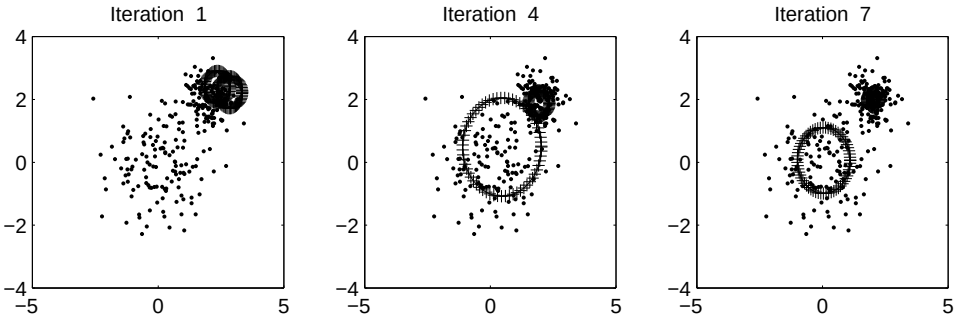

Figure 2: The progress of the EM algorithm with $k = 2$ and random initialization on the two-Gaussian data set (note: some data points omitted for clarity). The radius of the circle around each Gaussian is set to its variance.

2.3 K-Means on Two-Dimensional, Annulus Data

On the annulus data, K-Means also works well, with the centers converging to points evenly distributed around the annulus in four or five iterations (see figure 3).

2.4 EM on Two Dimensional, Annulus Data

On the annulus data set, the EM algorithm also performs well, converging within 10 iterations (see figure 4).

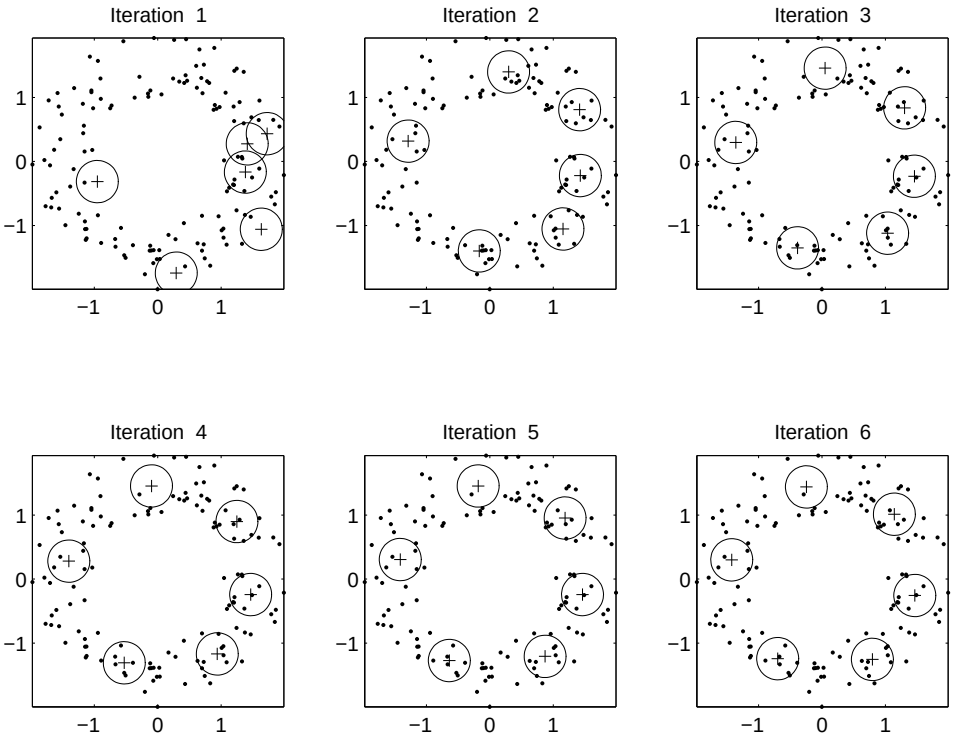

Figure 3: The progress of the K-Means algorithm with $k = 6$ and random initialization on the annulus data set (note: some data points omitted for clarity).

We verify that our code for EM is progressively finding a better fit for the data by checking that the negative log likelihood after each iteration never increases. As can be seen in Figure 5, this value decreases after each iteration.

3 High Dimensional Data Clustering

Most real-world data sets are very high-dimensional. However, the performance of clustering algorithms tends to scale poorly as the dimension of the data grows. For this reason the dimensionality of data sets is often reduced by various techniques before it is clustered.

Our data set is very high-dimensional, since each data point is a 240 x 292 image with 256 shades of gray. Treating each pixel as a dimension yields a 70080-dimensional data set, which makes clustering difficult given our computing resources. To reduce the dimensionality of our data set, we experimented with random projections and principal component analysis (PCA). Random projections have the desirable property that highly eccentric, high-dimensional Gaussians become more spherical when projected down to a small random basis.

Our data set is a collecton of images of the faces of 14 different people expressing different emotions. Each person was instructed to make a happy, sad, surprised, afraid, disgusted, and angry face. Our primary goal is to classify the facial expression of a given image by clustering our data set into six clusters, one for each emotion, and then calculating which cluster is most likely to contain that image. We are also interested in clustering our data set

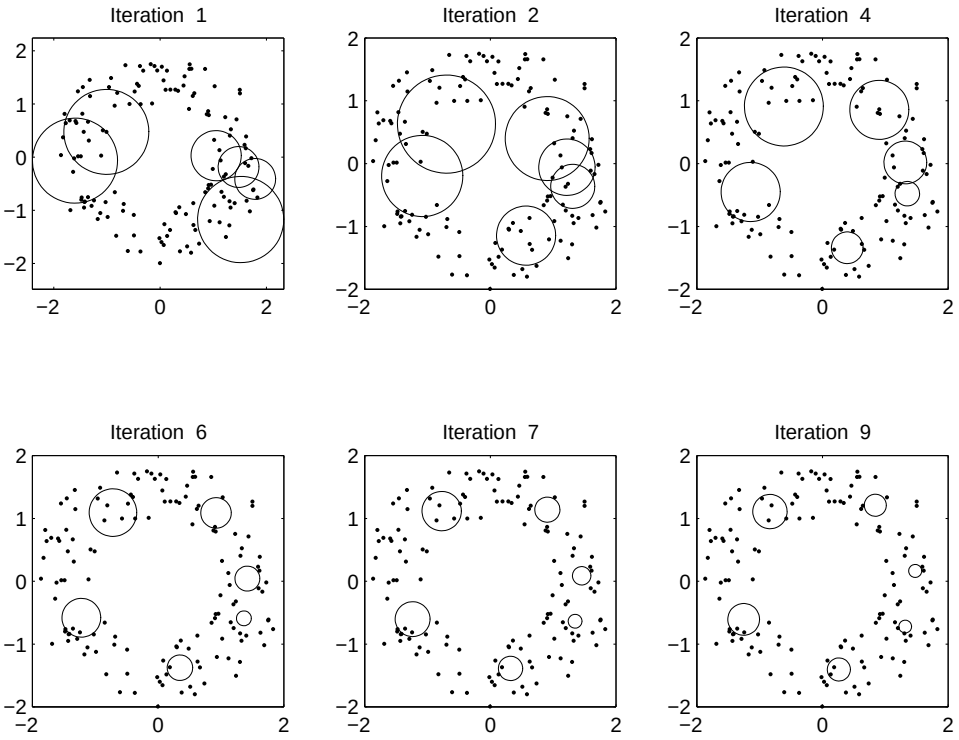

Figure 4: The progress of the EM algorithm with $k = 6$ and random initialization on the annulus data set (note: some data points omitted for clarity). The radius of the circle around each Gaussian is set to its variance.

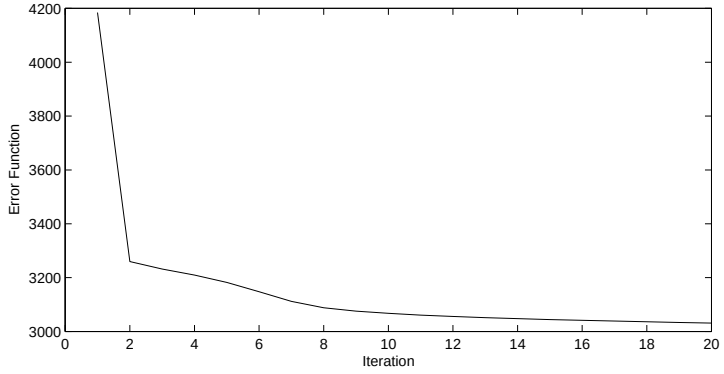

Figure 5: The negative log likelihood of the EM algorithm on the annulus data set with $k = 6$ and random initialization. This error function is $E = -\sum_{n=1}^{N}\sum_{j=1}^{M} p^{old}(j|\boldsymbol{x}^n)\ln\{P^{new}(j)p^{new}(\boldsymbol{x}^n|j)\}$.

into 14 clusters, one for each person, and conducting the analogous experiment of classifying images by person. We were concerned that clustering images to distinguish between emotions would find clusters of different people, rather than different facial expressions.

To avoid this, we make use of an image of each person making a “neutral” face. We add a”difference image,” defined as the difference between each image of a person expressing an emotion and that person’s neutral face to the data set. This set is used to cluster based on facial expressions, whereas we use the raw images to classify particular people. Our intuition was that clustering by person would be more successful than clustering by facial expression.

We downsample all images by a factor of 64 (8 in the x and y dimensions) to reduce the effects of noise. Intuitively, the downsampling does not remove information crucial to clustering since a human can still identify people and their facial expressions at this resolution.

3.1 Classifying People

Supervised Clustering

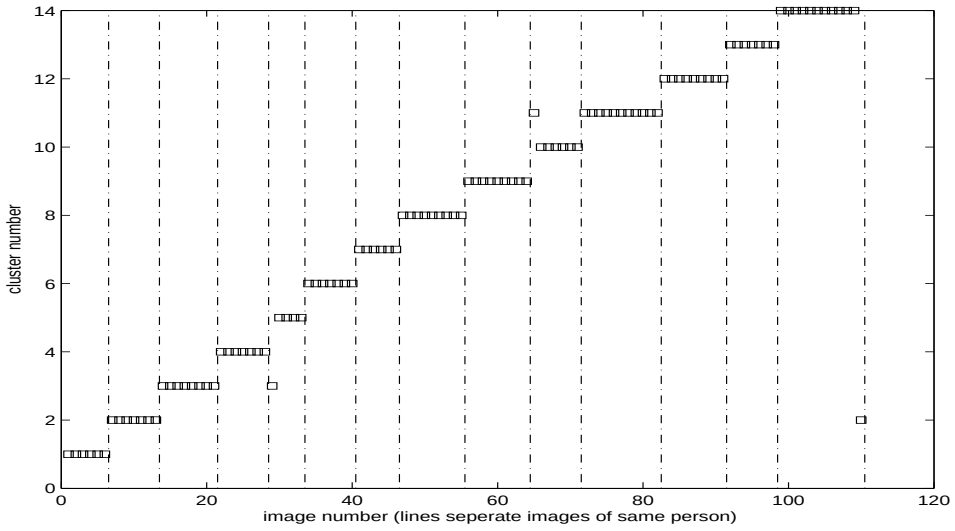

Figure 6: Supervised clustering, 1080-dimensional data (no dimension reduction).

Note about figures 6 and 7: The graphs show how each image was classified, by cluster number on the vertical axis, and the images on the horizontal axis. These images are sorted by known category, where each category is seperated by vertical dash-dot lines. A square indicates a point classified by finding the cluster-center (mean) to which that point is closest. A dot indicates a point classified by finding the Gaussian with the highest probability at that point.

As an initial test, we use all the images of a particular person to calculate a maximum-likelihood mean and variance of that person, and then use these 14 Gaussians to classify each image in the data set. We use two classification methods, one classifies a data point by finding the mean to which that point is closest, the other finds the Gaussian with the highest probability at that point. This supervised clustering test represents an upper bound on how well we can expect unsupervised clustering algorithms to perform.

Figure 6 is the result of fitting 14 Gaussians to the raw image data and then trying to classify each image. Evaluating Gaussians on such high-dimensional data causes numerical precision problems, so we are unable to use the probability method to classify images.

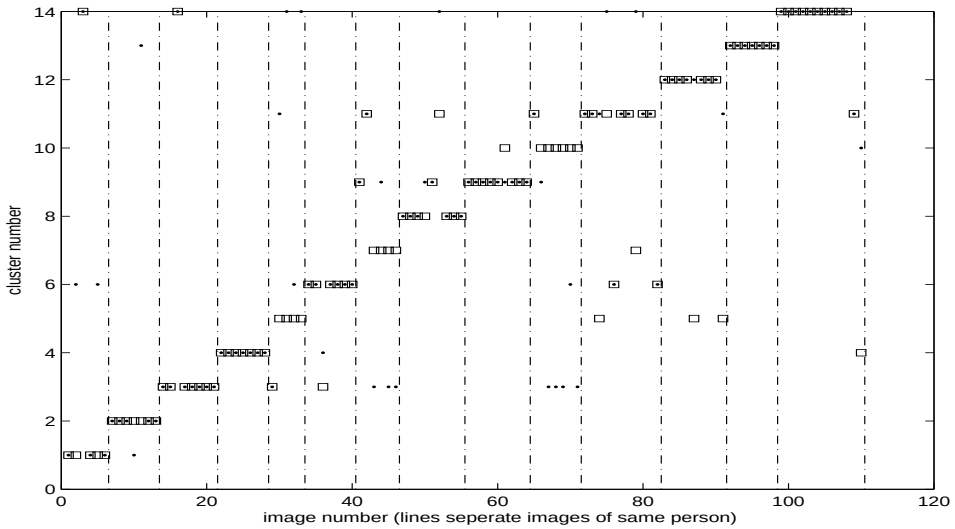

Figure 7: Supervised clustering, 15-dimensional data (random projection).

Notice that only three (of 110) images are misclassified, indicating that the raw data are well seperated.

Figure 7 is the same as figure 6, except the data were pojected down to a random 15-dimensional basis. These low-dimensional data are still well seperated, but there are a few more misclassifications than in the high-dimensional data. Reducing the number of dimensions with PCA yields comperable results.

Unsupervised Clustering

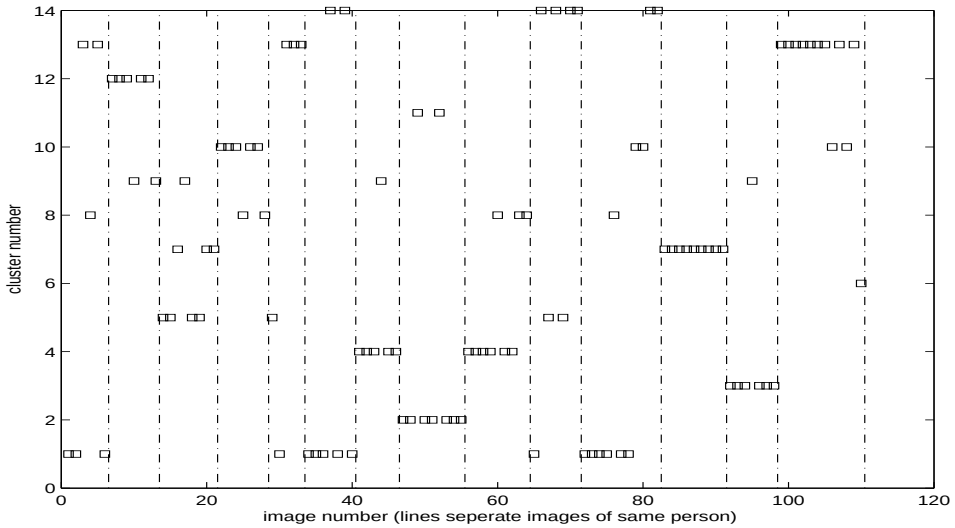

Figure 8: Unsupervised clustering, 1080-dimensional data (no dimension reduction).

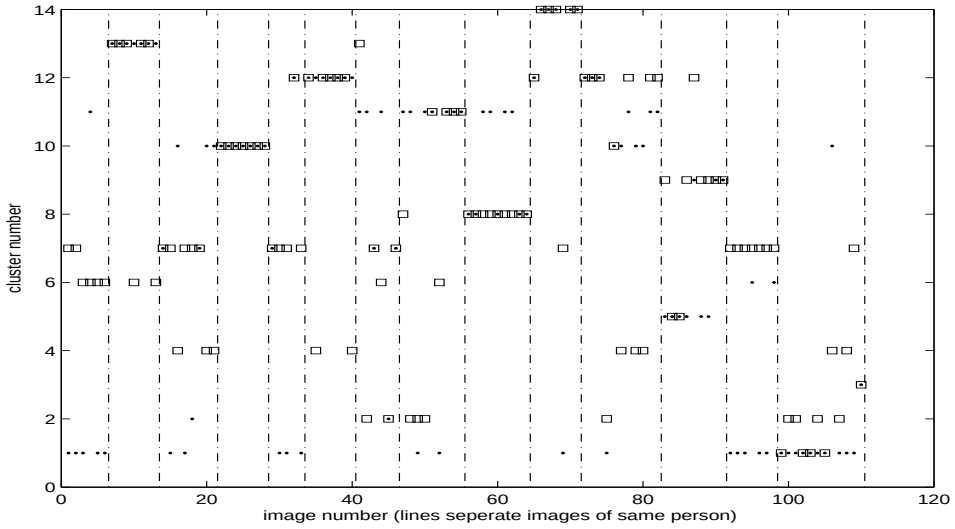

Figure 9: Unsupervised clustering, 15-dimensional data (random projection).

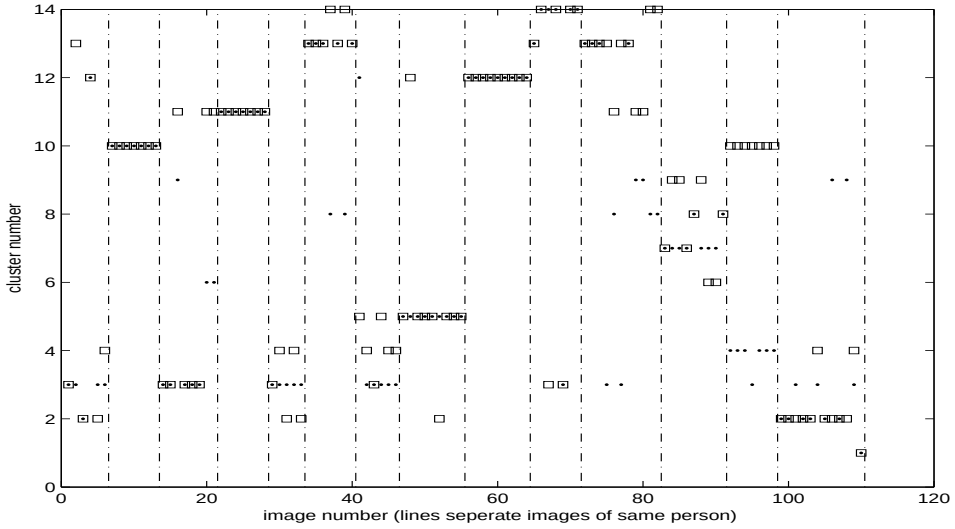

Figure 10: Unsupervised clustering, 15-dimensional data (PCA projection).

Note about figures 8 through 10: The squares indicate the K-Means results and the dots indicate the EM results. Some EM results are not present due to numerical precision problems.

Figure 8 is the result of running K-Means (EM failed due to numerical precision problems) on the entire high-dimensional data set, looking for 14 clusters, and classifying all the data points according to the clusters those algorithms found. About ten clusters were found by K-Means that correlate well to distinct people.

Figure 9 is the same as figure 8 except the data have been projected down to a 20 dimen-

sional random basis and EM results are included. About five clusters were found by both algorithms that correlated well with distinct people.

Figure 10 shows the clusters found by K-Means and EM on the data set projected down to 15 principal components. EM clusters marginally better than K-Means in this case. It also appears that the PCA basis worked better than the random basis.

3.2 Classifying Facial Expressions

Supervised Clustering

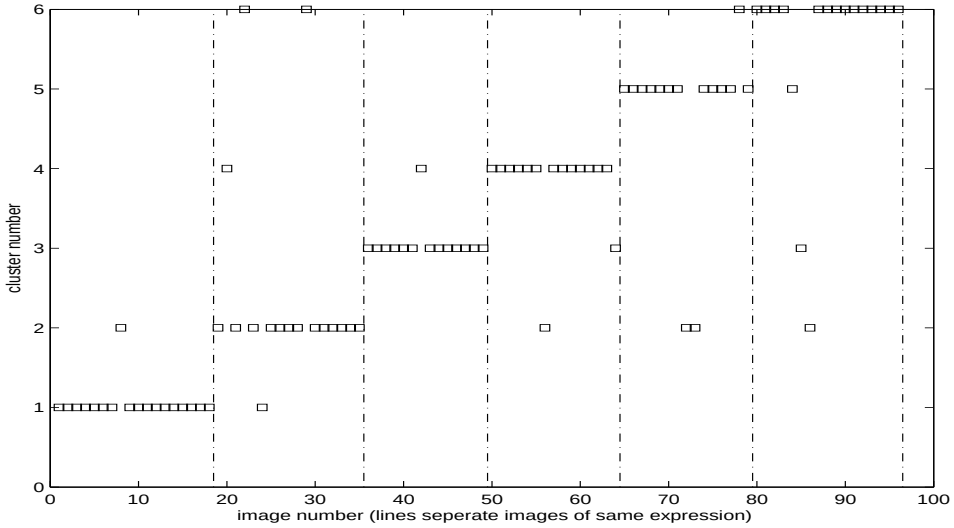

Figure 11: Supervised clustering, 1080-dimensional data (no dimension reduction).

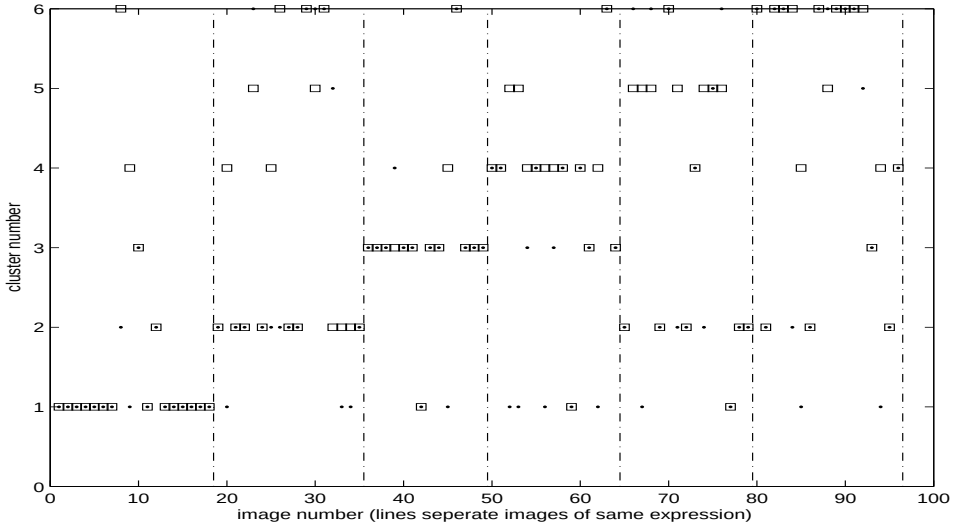

Figure 12: Supervised clustering, 15-dimensional data (random projection).

To see if disjoint clusters of facial expressions exist in our data set, we use all the difference-images of each facial expression to calculate a maximum-likelihood mean and variance of that expression, and then use these six Gaussians to classify the images. We use the same two classification methods as in section 3.1, classifying a data point by finding the mean to which that point is closest and the other finding the Gaussian with the highest probability at that point. This supervised clustering test represents an upper bound on how well we can expect unsupervised clustering algorithms to perform.

Figure 11 shows that most data points can be identified with the correct cluster, indicating the raw data can be partitioned into distinct clusters of facial expressions.

Figure 12 is the same as figure 11 except the data have been projected down to a 15-dimensional random basis. About a quarter of the data are misclassified, indicating the clusters in the projected data are less distinct.

Unsupervised Clustering

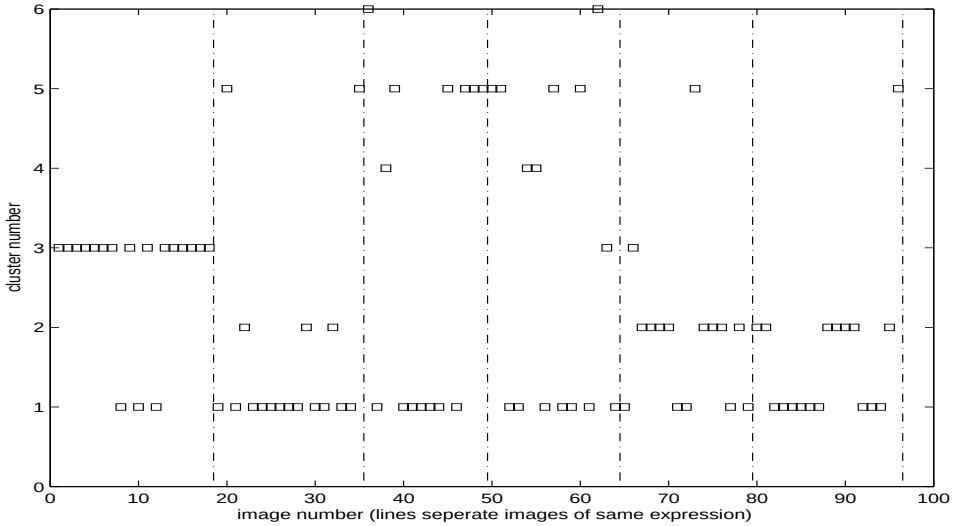

Figure 13: Unsupervised clustering, 1080-dimensional data (no dimension reduction).

Figure 13 is the result of running K-Means (EM failed due to numerical precision problems) on the entire high-dimensional data set, looking for 6 clusters, and classifying all the data points according to the clusters found. The only cluster that correlates well with a particular facial expression is cluster 1 corresponding to the happy expressions.

Figure 14 is the same as figure 13 except the data have been projected down to a 20 dimensional random basis and EM results are included. Again, the only cluster that correlates well with a particular facial expression is the cluster associated with happy expressions, but this cluster is not as disjoint (from other clusters) as in the high-dimensional data set, since there are more false positives and misses.

Figure 15 shows the clusters found by K-Means and EM on the data set projected down to 20 principal components. As in the previous two figures, only happiness correlates well with a particular cluster. There are slightly fewer misses than with the high dimensional data; however, there are many more false positives.

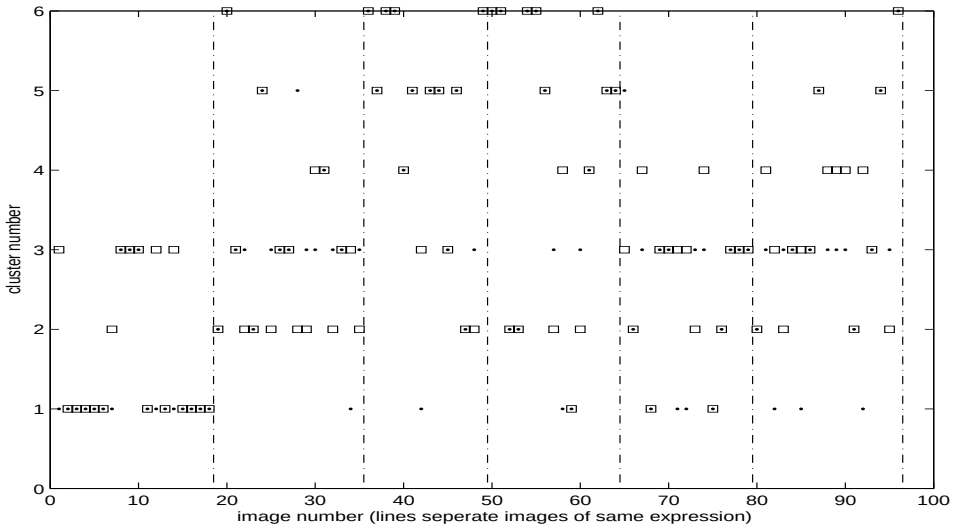

Figure 14: Unsupervised clustering, 20-dimensional data (random projection).

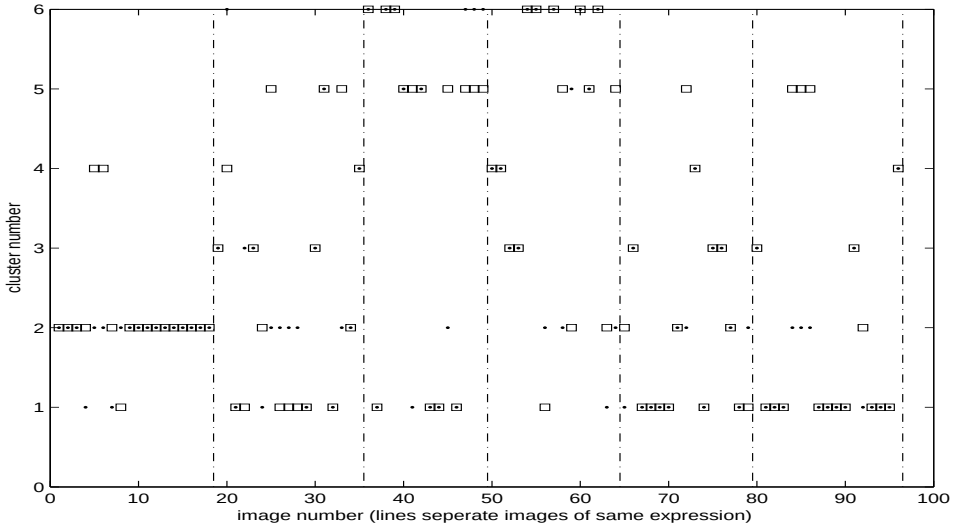

Figure 15: Unsupervised clustering, 20-dimensional data (PCA projection).

4 Conclusion

Down-sampling the images greatly improved clustering in all cases. We suspect this is because most noise was averaged out.

We tried two techniques to reduce the dimensionality of our data set, projecting it down to a low-dimensional random basis, and principal component analysis. Both of these techniques degraded the quality of the clustering, but made the expectation maximization algorithm feasible. Our observation is that PCA was only marginally better, if at all, than a random projection despite its computational intensity.

In general, we noticed that K-Means performs comparably to EM; however, EM fails on high-dimensional data sets due to numerical precision problems. Another problem with EM is that Gaussians often collapsed to delta functions. Our technique to prevent this was to reset the variance of collapsed Gaussians to a more reasonable value, and to set the mean of those Gaussians to random data points.

We would have liked to run our clustering algorithms on our data sets and then validate the results by classifying novel data, however when we reserved a portion of our data for validation, the clusters the algorithms found did not correspond at all to the classes we were trying to find. We strongly suspect this is due to the lack of enough sample points to define accurate Gaussians.

5 Future Work

The technique we use to convert images to feature vectors is simply to list all of the pixels in the image. Clearly, this is the a naive approach because it ignores the correlations between neighboring pixels. Our first attempt was to create feature vectors that were the Gabor wavelet transforms of each image using 4 scales and 6 rotations. This took a long time and produced a feature vector that was far too large for our computational reasources. Future work should explore the potential of this approach by using a more efficient wavelet transform procedure.

In addition to K-Means and EM, K-Harmonic-Means is another clustering algorithm that could be used to classify images. For each cluster center, K-Harmonic-Means computes the harmoic mean of the distance to every data point, and update that cluster center accordingly. This algorithm is less sensitive to initial cluster centers than K-Means, but does not have the problem of collapsing Gaussians exhibited by EM. For these reasons, K-Harmonic-Means might find better clusters in high-dimensional data.

Individual Contributions

Neil Alldrin:
My primary contribution was writing the EM code (most of the em_* files). This included a lot of effort devoted to how to handle collapsing Gaussians and how to prevent divide by zeros caused by lack of numerical precision (which was only a problem on high dimensional data). I also generated the graphs for the 2-dimensional data and helped write the latex document you are now reading. I had lots of fun.

Andrew Smith:
I learned Matlab. I wrote the script to load the images (loadFaces.m). I wrote the function to downsample images. I wrote K-means. I wrote the scripts to generate the high-dimensional data graphs (section 3) in this paper. I wrote a function to generate a Gabor filter bank, using code from [2] to evaluate Gabor functions. I experimented with using Gabor wavelet transforms for feature vectors of images, but it was too slow. I had lots of fun.

Doug Turnbull:
My primary focus in this assignment was to design and implement various experiments for high dimensional data using kmeans and EM. These tests included implementing random projection and PCA precompuation algorithms, creating scripts to run tests for various data sets, and collecting results for analysis. Developing these tests were often a nontrival task due to a large number of parameters (projection matrices, clustering algorithm, data sets, etc.) that greatly affect the quality of the results. I had a little bit of fun.

References

[1] Bishop, C.M. (1995) Neural Networks for Pattern Recognition, *Oxford University Press, New York*.

[2] Manjunath, B.S. & Ma, W.Y. (1996) Texture Features for Browsing and Retrieval of Image Data, *IEEE PAMI, vol 18, no. 8, pp. 837-842*.

[3] Sanjoy, D. (1999) Learning Mixtures of Gaussians, *IEEE Symposium on Foundations of Computer Science (FOCS)*.

[4] Zhang, B. (2000) Generalized K-Harmonic Means, *Hewlett-Packard Laboratoris Technical Report*.