Lab 7
CO902 – Probabilistic and statistical inference – 2012-13 Term 2
Lecturer: Tom Nichols
Clustering with Gaussian Mixture Models – How many ways are there to write a digit?

Use a combination of Principal Components Analysis (PCA) and Gaussian Mixture Modelling (GMM) to discover latent classes in the handwritten digit data. Each file `data1.mat`, `data2.mat`, …, `data0.mat` contains a single variable, `data`, a 256 × 1100 matrix consisting of 1100 instances of the corresponding digit. Previously, I had given you only a tiny fraction of this data; today we will use all the available data.

1. Pick a digit, 1, 2, …, 8, 9, or 0. For *just* this single digit, compute the PCA decomposition (via SVD) to reduce the data dimension and to create orthogonal variables. Remember to use the 0 option with svd to get the "economy size" decomposition (and save time). Use enough components to capture at least 95% of the variance. Call the reduced data `Y`; it should have dimensions d* × 1100, where d* is the reduced data dimension you chose.

2. Use GMM to find the number of latent components in Y using the Matlab GMM tool, e.g. for k classes the appropriate command is

   ```
   fit = gmdistribution.fit(Y',k, ...
     'CovType', 'diagonal', 'Replicates',10)
   ```

   Importantly, note that the function expects Y to have observations in rows and variables in columns; I have also specified that the covariance is diagonal, appropriate since we're working with PCA-derived features; finally, I have asked for multiple-restarts of the algorithm, always a good idea.

   Compare several latent class dimensions k, e.g., k=2, 3, etc. For each k, transform the d*-dimensional means into the full dimension (d=256) space and then use the "display_digit.m" function (or your own) to visualize the means. The class-means are in the "fit" data structure, accessible as (according to the example above) `fit.mu`, but don't forget that variables are in columns, not rows.

   Use AIC (`fit.AIC`) and BIC (`fit.BIC`) to determine the model order, and select your best k. AIC and BIC will likely give different answers for the optimal k; using the visualized mean images, do you prefer the optimal k from AIC or BIC? Or a different one?

3. The command cluster will estimate the most likely cluster for each observation,

   ```
   ClassEst = cluster(fit,Y');
   tabulate(ClassEst)
   ```

   What is the tabulate command doing? We can also compute the Mahalanobis distance between each case and the latent class means, and then find the closest class

   ```
   Dist             = mahal(fit,Y');
   [MinDist,ClassM] = min(Dist,[],2);
   tabulate(ClassM)
   ```

   Why do these two approaches give difference answers. Which is preferred? Do this for the optimal k from AIC and from BIC; does it make you favor one more than the other?

4. *Time permitting*. Do it all again, but now do a PCA decomposition that uses *half* as many components. Do you get roughly the same answers? Dramatically different answers? What about using *all* components? Now try a different digit (e.g. 1 or 8 if not tried already).