6.867: Homework 3

In this assignment we will look at the EM approach for fitting Gaussian Mixtures to data, We will be following the same process that we did for Homeworks 1 and 2 (and repeated in the last section of this handout).

- This assignment may be done by pairs of students.
- You submit your paper via the Easy Chair site by 11:59 PM on Thursday November 15.
- Each student who was an author or co-author on a submission will be assigned several papers to review.
- Reviews must be entered on the Easy Chair site by 11:59PM on Thursday, November 22.
- All reviews will be made visible shortly thereafter, and students will have a 2–3 day period in which to enter rebuttals of their reviews into EasyChair if they wish.

We have given you several data sets (in the `data` folder); some are "small" and some are "large". The corresponding small and large data sets are drawn from the same distribution. There are also some "mystery" data sets to be used only in the last question.

In this assignment, we will once again be comparing performance for a number of algorithms and conditions. You cannot possibly show every possible plot produced by running every possible combination of conditions. Instead, summarize the key trends using tables or graphs of log-likelihoods and some selected plots.

The following questions are the points that your paper should cover in order to receive full credit. Your presentation should roughly follow the order of these questions so that your reviewers can see what you're doing.

1 EM Algorithm

1. Implement EM for Gaussian Mixtures as described in Bishop (section 9.2.2). Your program will need an input data set, initial mixture parameters and a convergence threshold (pick your favorite way of deciding convergence). The file `em_test` has a simple skeleton for reading

data and plotting results. Note that the plots show mixture components as ellipses at two standard deviations.

2. To avoid numerical underflow problems, you will want to compute log likelihood. Note, however, that for a mixture distribution you will need to compute the log of a sum (Bishop equation 9.28). Look up the "logsumexp trick" to see how to deal with this. You can find `logsumexp` in Numpy and there are also Matlab implementations available on-line. In any case, the idea is simple (once you see it).

3. Describe the behavior of your algorithm on the (non-mystery) training data sets provided as you vary (a) the number of components in the mixture, (b) the initial mixture parameters, (c) the convergence parameter and (d) the choice of small/large. Report the log-likelihoods and show selected plots of good and bad performance.

2 Variations

1. Modify your implementation of the EM algorithm so that it can build two types of models: ones with general covariance matrices and ones with diagonal matrices. Note that in each case different components have different means. Explain (in math, not code) the difference in the EM algorithm for these two variants.

2. Implemeent the K-means algorithm as a way to get an initial estimate of the mixture components.

3. Explore the performance of these algorithm variants using similar tests to what you did on the original algorithm.

3 Model Selection

In general, we can pick among candidate models on the basis of our estimate of how likely they make unseen data. We can use average (per data point, instead of sum) log likelihood to rank the models so that we can compare likelihood estimates based on different number of data points. Estimating average log-likelihood of a model on the training data is a very biased estimate of the likelihood on unseen data. One common technique for getting a better estimate is cross-validation, in which multiple training and validation sets are constructed from the original training data.

1. Construct a candidate set of models for each of the data sets that differ on (a) the choices of covariance matrices (diagonal vs. general) and (b) the number of mixture components (1 – 5). Ran the models for each **small** (non-mystery) training set based on average log likelihood from applying EM to the training set. You will need to decide exactly how to use EM (how to

initialize, whether to run multiple times, etc); document your choices. Compare your results to the ranking of the models on the **large** "test" sets. Explain your findings.

2. Implement a cross-validation procedure that can be used to rank models. Your procedure should have a parameter K that determines how many folds of cross-validation are used; when K equals N − 1, where N is the nummber of training data points, this corresponds to "leave one out cross-validation".

3. Test your cross-validation procedure (for different values of K). Compare the results on the small and large data sets. Compare also with the results you got by ranking on training set log likelihood. Explain your findings.

4 Predictions

1. We have given you some `mystery` training data sets. Make your best prediction for mixture parameters for these data sets. Explain how you made the predictions.

Grading process ("same as it ever was")

You will find a zip file with some useful code and data in the Resources section of the Piazza course page. You can do these assignments in any computational system that you are used to. We recommend Matlab or the Pylab/Numpy/Scipy/Matplotlib cluster of packages and we'll try to provide help for those two systems. If you use anything else, you're on your own...

You will be turning in a single, readable "paper" (a single PDF file) with your solutions. We will be emulating the process of submitting papers for publication to a conference. We will be using an actual conference review system (Easy Chair) to have these papers peer reviewed (by the other students). This means that your answers have to be readable and understandable to your peers and, where possible, interesting. Note that when explanations are called for, you will need to convince the reviewers that you understand what you're talking about. The course staff will serve as the Program Committee for the conference and make all final decisions. The details of this process are posted on Piazza.

Grading rubric

Your paper must be anonymous (no identifying information should appear in the PDF file). If it is not, it will automatically receive a 20% deduction, and will be graded by a grumpy staff member.

The paper must be no more than 6 pages long in a font no smaller than 10 point. It should include whatever tables, graphs, plots, etc., are necessary to demonstrate your work and conclusions. *It should not include code.*

Each of the four parts of the assignment will be graded on a scale from 0 to 5 (where 0 is failing and 5 is an A) on two aspects:

- **Content:** Did the solution answer the questions posed? Were the answers correct? Were the experiments well-designed or examples well chosen?
- **Clarity:** Were the results written up clearly? Were the plots labeled appropriate and described well? Did the plots support the points in the paper? Did the discussion in the paper illuminate the plots?

As a reviewer, you will be asked to provide a score for each section, and at at least two paragraphs of feedback, per review, explaining things that were done well and things that could have been improved upon.

Your overall score for this assignment will be:

- **80%**: The average of all 8 scores on your assignment given by all three reviewers.
- **20%**: A score for the quality of your reviews. This will be full credit, by default. But we will skim reviews and examine some carefully and may reduce this grade for review commentary that is sloppy or wrong.

The course staff will spot-check submissions and reviews, paying careful attention to cases where there were rebuttals. The staff will act as the program committee and determine a final score. Our overall goals in this process are:

- To motivate you to work seriously on the problems and learn something about the machine learning material in the process
- To engage you in thinking critically and learning from other students' solutions to the problems

We will arrange to give full credit to anyone who submits a serious and careful solution to the problems and who gives evidence of having read carefully the solutions they were assigned and who writes thoughtful reviews of them.