CSC2515 Assignment 2
Due Oct 30 2007, 11am at START of class

Ruslan Salakhutdinov and Geoffrey Hinton

October 16, 2007

Late assignments will have 25% subtracted from the total out of which they are graded for each day or part of a day that they are late. They will be more than one day late if they are not slipped under my office door (LP 290G) before 11.00am the next day.

1. (1 point) Derive the gradient $\partial \log p(x)/\partial \mu_i$ for the mean of Gaussian i in a mixture of K one-dimensional Gaussians, where x is a single datapoint. Use π_i and σ_i^2 for the mixing proportion and variance of the i^{th} Gaussian. Your derivation should include all of the steps.

2. (1 point) Derive the gradient $\partial \log p(x)/\partial w_i$ for a mixture of one-dimensional Gaussians, where

$$\pi_i = \frac{\exp(w_i)}{\sum_j \exp(w_j)} \tag{1}$$

The reparameterization is necessary to ensure that the constraint $\sum_i \pi_i = 1$ remains satisfied even though we are doing unconstrained optimization. Your derivation should include all of the steps.

3. (1 point) Derive the gradient $\partial \log p(x)/\partial \alpha_i$ for the log variance of Gaussian i in a mixture of one-dimensional Gaussians, where

$$\sigma_i^2 = \exp(\alpha_i) \tag{2}$$

The reparameterization is necessary to ensure that the variances do not become negative even though we are doing unconstrained optimization. It also slows down their approach to zero. Your derivation should include all of the steps.

4. (13 points) In this question you will experiment with 3 different optimization techniques for learning a mixture of two univariate (i.e. one-dimensional) Gaussians (MoG):

1. The Expectation and Maximization algorithm (EM)
2. Steepest Descent using the total gradient on all the training cases
3. The Method of Conjugate Gradients (using the minimize.m matlab software which is provided on the assignments web page).

You will create three datasets (actually you will probably create a lot more than 3, but you must end up picking 3). Each such dataset will be generated from a mixture of two Gaussians where you will choose appropriate parameter values. You are also free to choose the number of datapoints, N_1, N_2, you sample from each of the two Gaussians. To sample from a standard normal distribution use the **randn** matlab command. To sample from a uniform distribution on the interval (0,1) use the **rand** command.

For each of the three artificial datasets you choose, you will learn a MoG model using the three different optimization techniques. Your goal is to discover and illustrate under what conditions each method performs better or worse than the other methods in terms of convergence speed to a local maximum of the likelihood function. So when you create your three artificial datasets, you must choose your parameter values carefully in order to illustrate when you expect each method to be superior or inferior to others. It is possible that one or more of the methods is never superior to one or more of the other methods – that is up to you to discover.

You will also decide on the stopping criteria for all optimization methods. For the EM and Steepest Descent methods you can use the following:
if $(loglikelihood(t+1) - loglikelihood(t))/loglikelihood(t) < tol$ then terminate learning.
For the conjugate gradient code, you will need to specify the number of function evaluations.

For the steepest descent and conjugate gradient methods, do not forget to re-parameterize the mixing proportions and variances (see questions 2 and 3), so that you can do unconstrained optimization. For conjugate gradients you will use the minimize.m optimizer that will be provided to you. You can also use checkgrad.m to check your gradients.

What to hand in:

1. For each dataset:
   - Plot the empirical distribution of the training data (use the hist command with a second argument of 100).
   - Plot the convergence of all 3 optimization algorithms on the same plot with the x-axis showing the number of iterations and y-axis showing log-likelihood of the training data. In this plot, an "iteration" of the conjugate gradient algorithm should correspond to one function evaluation, but only plot the value of the log-likelihood at the end of each line search (note that it may take a few function evaluations to complete one line search).
   - Report the learned mixing proportions, means and variances of each component using each of the three optimization methods and also the values that were used to generate the training data.
2. For optimization with the steepest descent method, you must say how you chose the learning rate. Use a fixed learning rate that does not change during the optimization.
3. Describe, in one page or less, your findings about the relative performance of the three optimization methods and why you chose the particular three datasets that you used.
4. Provide a printout of the matlab code you used for each of the three optimization methods. Do not print out minimize.m.