CSI 5325 Assignment 4

Greg Hamerly

Assigned: 4/16/2014; Due: 4/30/2014

Instructions

For this assignment, you should write your derivations carefully and clearly out by hand, or you should write them up in LaTeX. In either case, please make them concise and easy to follow, with appropriate English descriptions rather than just mathematics (i.e. explain what you are doing as necessary).

Write any programs in Matlab (or Octave), and attach their code with your assignment. Include any relevant figures and graphs (plotted using Matlab / Octave) in your writeup.

Submit your assignment in two ways: hardcopy (in class) and by email (to hamerly@cs.baylor.edu). The email should contain a single attachment as a ZIP file. It should be named "lastname-xx.zip", where lastname is your last name, and xx is the number of the assignment.

Finally, please keep your submitted email attachments small. In particular, make sure you are only submitting things that are necessary (omit datasets I gave you, compiled programs, etc.). Also, try to keep your graphics small by using vector (rather than bitmap) formats (e.g. PDF or EPS rather than JPG or BMP). Vector graphics are generally smaller in size and better quality than bitmap.

1 K-means clustering (20 points)

In this problem you'll implement the K-means clustering algorithm on a synthetic data set. There is code and data in the `student_programs/` directory. Run `load 'X.dat'`; to load the data file for clustering. Implement the `[clusters, centers] = k_means(X, k)` function in this directory. As input, this function takes the $m \times n$ data matrix `X` and the number of clusters `k`. It should output a m element vector, clusters, which indicates which of the clusters each data point belongs to, and a $k \times n$ matrix, `centers`, which contains the centroids of each cluster. Run the algorithm on the data provided, with $k = 3$ and $k = 4$. Plot the cluster assignments and centroids for each iteration of the algorithm using the `draw_clusters(X, clusters, centroids)` function. For each k, be sure to run the algorithm several times using different initial centroids.

What can you learn in these experiments about the k-means algorithm? You should at least consider things like its convergence properties, the role of initialization, and the role of the value of k.

2 The Generalized EM Algorithm (20 points)

When attempting to run the EM algorithm, it may sometimes be difficult to perform the M step exactly – recall that we often need to implement numerical optimization to perform the maximization, which can

be costly. Therefore, instead of finding the global maximum of our lower bound on the log-likelihood, and alternative is to just increase this lower bound a little bit, by taking one step of gradient ascent, for example. This is commonly known as the Generalized EM (GEM) algorithm. Put slightly more formally, recall that the M-step of the standard EM algorithm performs the maximization

$$\theta \leftarrow \arg\max_{\theta} \sum_{i=1}^{m} \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}.$$

The GEM algorithm, in contrast, performs the following update in the M-step:

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \sum_{i=1}^{m} \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}$$

where α is a learning rate which we assume is chosen small enough such that we do not decrease the objective function when taking this gradient step.

(a) Prove that the GEM algorithm described above converges. To do this, you should show that the likelihood is monotonically improving, as it does for the EM algorithm – i.e., show that $\ell(\theta^{(t+1)}) \geq \ell(\theta^{(t)})$.

(b) Instead of using the EM algorithm at all, suppose we just want to apply gradient ascent to maximize the log-likelihood directly. In other words, we are trying to maximize the (non-convex) function

$$\ell(\theta) = \sum_{i=1}^{m} \log \sum_{z^{(i)}} p(x^{(i)}, z^{(i)}; \theta)$$

so we could simply use the update

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \sum_{i=1}^{m} \log \sum_{z^{(i)}} p(x^{(i)}, z^{(i)}; \theta).$$

Show that this procedure in fact gives the same update as the GEM algorithm described above. [Hint: Try taking the partial derivative of $\ell(\theta)$ with respect to θ_j, and simplify. Then show that the same derivative of the lower-bound used in GEM, with specific value of Q_i chosen in the E-step, leads to this result.]