Complex Systems 535/Physics 508: Homework 7

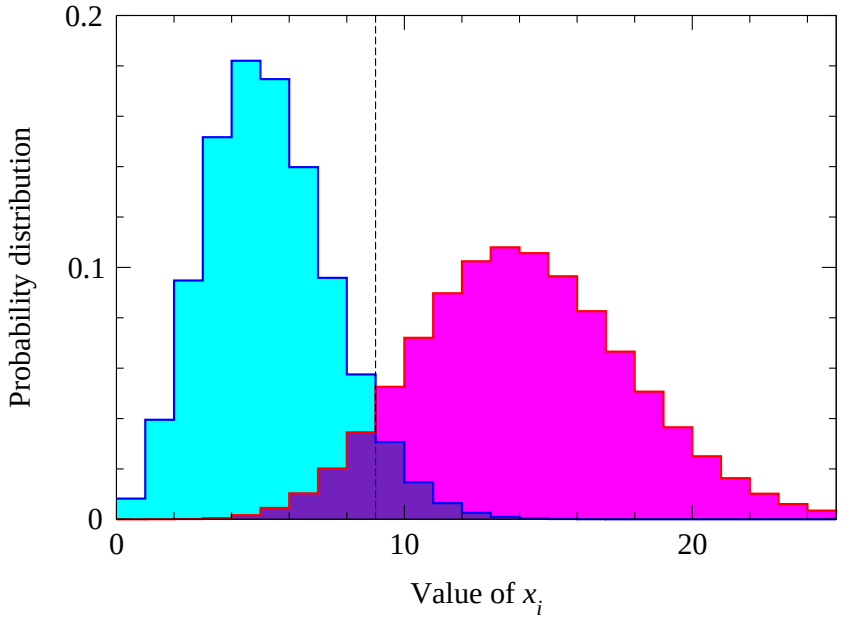
Because of the Thanksgiving Break, you have longer than usual to do this homework. It is due in class on ***December 1****.*

1. When we introduced the maximum likelihood method, we illustrated it with an application to ordinary scalar data drawn from a Gaussian distribution. Let us do a similar thing with the expectation–maximization (EM) algorithm and illustrate it with Poisson data.

   Suppose we have a set of n measurements x_i which are integer numbers drawn independently from Poisson distributions. But here's the catch: each number is drawn from one of two different Poisson distributions with different means μ_1 and μ_2 and we're not told which distribution each number is drawn from, nor are we told the means. All we know is that the probability distributions have the Poisson form

   $$P(x_i|\mu_1) = \frac{\mu_1^{x_i}}{x_i!}\mathrm{e}^{-\mu_1}, \qquad P(x_i|\mu_2) = \frac{\mu_2^{x_i}}{x_i!}\mathrm{e}^{-\mu_2}.$$

   So they might look something like this:



   For example, the numbers might be the degrees of vertices in a network, where there are two different types of nodes with different average degrees. The goal is to work the type of each vertex by looking only at the degrees. If we knew the means, this would be easy, but we don't.

   (i) Let $c_i \in \{1,2\}$ denote the type for x_i, i.e., the distribution from which x_i was drawn. Write down an expression for the total likelihood $P(x|\mu,c)$ of the entire data set x, given the values of the c_i and the two μ parameters. Take the logarithm to get an expression for the log-likelihood.

(ii) The best values of the μ parameters are given by maximizing the likelihood of the whole data set x given only the two parameters μ:

$$P(x|\mu) = \sum_c P(x, c|\mu) = \sum_c P(x|\mu, c)P(c),$$

where the sum is over all sets of values of c and $P(c)$ is the prior probability of the set c, which we assume to be uniform (i.e., constant) over all sets and hence can be ignored (since we don't care about constants when we are maximizing).

Equivalently, we can, if we prefer, maximize the log-likelihood $\log P(x|\mu)$. Recall Jensen's inequality, which says that for any set of positive quantities z_i and any set of probabilities q_i such that $\sum_i q_i = 1$, we have $\log \sum_i z_i \geq \sum_i q_i \log(z_i/q_i)$. Apply Jensen's inequality to the log-likelihood to show that (ignoring constants again)

$$\log P(x|\mu) \geq \sum_c q(c) \log P(x|\mu, c) - \sum_c q(c) \log q(c),$$

where $q(c)$ is any probability distribution over sets of types c such that $\sum_c q(c) = 1$. Also show that the exact equality—i.e., the maximum of the right-hand side over all choices of $q(c)$—is achieved when

$$q(c) = \frac{P(x|\mu, c)}{\sum_c P(x|\mu, c)}.$$

(iii) Using your expression from part (i), show that this choice of $q(c)$ factors as $q(c) = \prod_i q_i(c_i)$, where

$$q_i(r) = \frac{P(x_i|\mu_r)}{P(x_i|\mu_1) + P(x_i|\mu_2)} = \frac{\mathrm{e}^{-\mu_r}\mu_r^{x_i}}{\mathrm{e}^{-\mu_1}\mu_1^{x_i} + \mathrm{e}^{-\mu_2}\mu_2^{x_i}}.$$

(iv) Thus, if the right-hand side of our inequality is maximized over $q(c)$ by making this choice, it becomes equal to the left-hand size, and if we maximize the left-hand side we get the answer to our question, "What is the best value of μ?" The EM algorithm consists of doing these steps, but in the opposite order (since order doesn't matter anyway). We maximize with respect to μ first.

Taking your expression for the log-likelihood from part (i), putting it into the right-hand side of the inequality above and maximizing with fixed $q(c)$, show that the optimal value of μ_r is given by

$$\mu_r = \frac{\sum_i x_i q_i(r)}{\sum_i q_i(r)}.$$

You now have all the elements of the algorithm. Given the data, you would make an initial random guess about the values of the two parameters μ_1 and μ_2 and from them calculate the $2n$ quantities $q_i(r)$ as above. Then you would use those values to calculate a new value of μ_r, and repeat until you reach convergence. The end result would be the optimal values of the means μ_1 and μ_2, plus the probabilities $q_i(r)$ that each data point belongs to each of the two Poisson distributions.

2. Consider a network of n nodes generated using the standard random graph model $G(n, p)$ and let us divide this network at random into two equally sized parts of $\frac{1}{2}n$ nodes each. You can assume that n is large.

   (i) Show that on average half of the edges in the graph will run between the two parts, i.e., the cut size is $R = \frac{1}{2}m$, where m is the total number of edges.

   (ii) Since the edges are independent, the actual number of edges between the two parts will be Poisson distribution with mean $\mu = \frac{1}{2}m$. Since m is a large number this Poisson distribution is well approximated by a normal distribution with the same mean and standard deviation $\sqrt{\mu}$. Hence what is the probability that the actual cut size will satisfy $R \leq am$ for some constant $a < \frac{1}{2}$? Hint: there is no closed-form answer for this question—write your answer in terms of the Gaussian error function, $\operatorname{erf} x$.

   (iii) There are 2^n different ways to divide the network into two equally sized parts. Hence show that among those, the smallest cut size is about am where

$$\operatorname{erf}\bigl[(a - \tfrac{1}{2})\sqrt{m}\bigr] \simeq -1 + 2^{-n+1}.$$

   (iv) For values of $\operatorname{erf} x$ very close to -1, a good approximation for the error function is

$$\operatorname{erf} x \simeq -1 - \frac{\mathrm{e}^{-x^2}}{\sqrt{\pi}x}.$$

   Using this approximation, and neglecting terms $\mathrm{O}(1)$ and $\mathrm{O}(\log m)$ by comparison with terms $\mathrm{O}(m)$ or $\mathrm{O}(n)$, show that

$$\tfrac{1}{2} - a \simeq \sqrt{\frac{2\ln 2}{c}}.$$

   (v) Given that the expected cut size is $\frac{1}{2}m$, show that the modularity corresponding the division with smallest cut size is $Q = \frac{1}{2} - a$, and hence that for a random graph with mean degree $c = 8$ there should exist at least one division with modularity around $Q = 0.42$.

   Thus it is possible for a division of a random graph to have large modularity, even though the random graph obviously doesn't contain any communities. This is a cautionary tale: high modularity may tell you the *best* division of a network, but it doesn't tell you whether it's a *good* division.

3. On the course web site you'll find a file called `polblogs.gml`, which contains a copy of a network in GML format. This is a network of Internet blogs on the subject of US politics, along with the hyperlinks between them. The data were compiled by former Michigan Professor of Information Lada Adamic. Each node is also accompanied by a single scalar value which is either 0, for Democratic (liberal) blogs, or 1 for Republican (conservative) ones.

   You can use any tools you like to do the following operations, or a combination of tools if you prefer. Gephi, Matlab, Mathematica, R, Python, or any general-purpose programming language would be good choices.

(i) Read the network file and create an adjacency matrix representing the edges. Hyperlinks are directed, but you should ignore the directions, treating the edges as undirected, so that the matrix you get is symmetric.

(ii) Calculate the degrees of all the nodes and the total number of edges.

(iii) Hence calculate the modularity matrix for the network.

(iv) Calculate the leading eigenvector of the modularity matrix, and split the nodes into two groups according to the signs of the vector elements.

(v) Compare the two groups you get with the real-life classification of the nodes as Democratic and Republican. What fraction of the nodes does the algorithm classify correctly? (Hint: You should find it's pretty high—over 80%.)