

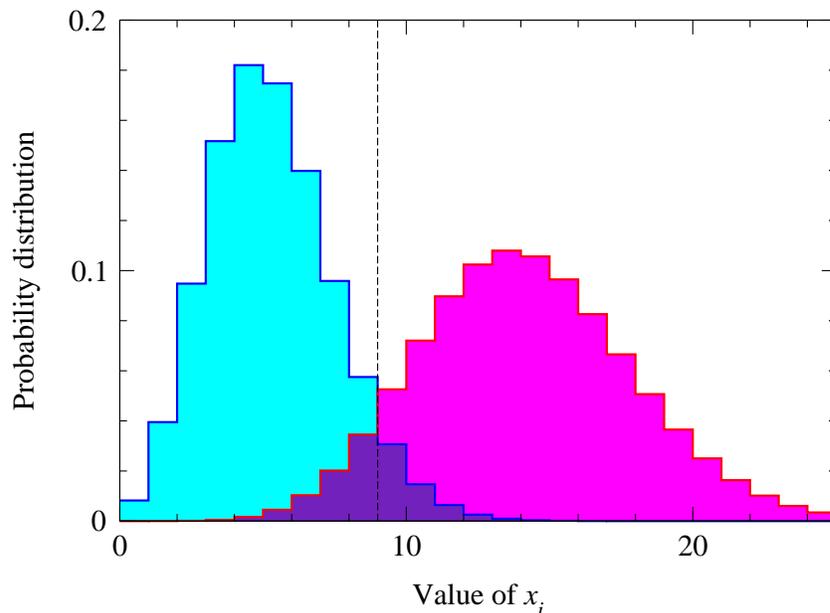
Complex Systems 535/Physics 508: Homework 7

1. When we first introduced the maximum likelihood method, we illustrated it with an application to ordinary scalar data, drawn in that case 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 , 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. The probability distributions for the two Poissons are:

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

and they look something like this:



So, 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 out which type each vertex belongs to by looking only at the degrees.

- (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

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

- (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 side, 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 from the equation above, 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. Recall that the modularity matrix is defined as the adjacency matrix minus the expected value of the adjacency matrix. In the Poisson random graph, the expected value of the adjacency matrix element A_{ij} is just equal to p , the average probability of an edge. Hence the modularity matrix \mathbf{B} for this model has elements $B_{ij} = A_{ij} - p$. Notice that for a random graph this is a *centered* matrix, meaning that all elements have mean zero.

- (i) Write down an expression for the spectral density $\rho(z)$ of this modularity matrix for the Poisson random graph. You do not need to give a derivation—this is a standard result.
- (ii) Recall that the modularity for a division of any network into two communities can be written as

$$Q = \frac{1}{4m} \mathbf{s}^T \mathbf{B} \mathbf{s},$$

where the elements of the vector \mathbf{s} are $s_i = \pm 1$ depending on which group vertex i belongs to. Relaxing the values of s_i to any real value, subject to the spherical constraint $\sum_i s_i^2 = n$, show that the maximum modularity is achieved when \mathbf{s} is equal to the leading eigenvector of \mathbf{B} , and that the maximum value of the modularity is equal to $\beta/2c$ where β is the leading eigenvalue and c is the average degree.

- (iii) Combining the results of (i) and (ii) give a formula for the approximate modularity of the maximum-modularity division of a random graph. Why is this result only approximate?
- (iv) Obviously the random graph has no community structure—it's just a random graph. So why is the modularity positive?

3. Consider a configuration model network that has vertices of degree 1, 2, and 3 only, in fractions p_1, p_2 , and p_3 , respectively.

- (i) Find the value of the critical vertex occupation probability ϕ_c at which site percolation takes place on the network.
- (ii) Show that there is no giant cluster for any value of the occupation probability ϕ if $p_1 > 3p_3$. In terms of the structure of the network, why is this? And why does the result not depend on p_2 ?
- (iii) Find the size of the giant cluster as a function of ϕ . (Hint: You may find it useful to remember that $u = 1$ is always a solution of the equation $u = 1 - \phi + \phi g_1(u)$.)

4. **Extra credit:** On the course web site you'll find a file called `polblogs.gml`, which contains a copy of one of the networks we discussed in class: a network of Internet blogs on the subject of US politics, along with the hyperlinks between them. The data were compiled by Michigan Professor of Information Lada Adamic. Each block 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%.)