Homework 3 – Due April 8

1. Write a C function to provide the largest eigenvalue and eigenvector of a nonnegative definite matrix using the power method. The function should take in a function pointer which defines the multiplication of a matrix in appropriate storage format and a vector. [*15 points*]

   (a) Use calls to the above function in another C function which provides the first m eigenvalues of a positive definite matrix. [*5 points*]

   (b) Demonstrate the above in an example program. [*5 points*]

2. Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ be a sample. Let $s_{ij} = s(\boldsymbol{X}_i, \boldsymbol{X}_j)$ be a similarity measure between $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$. For example, $s_{ij} = \text{Corr}(\boldsymbol{X}_i, \boldsymbol{X}_j)$. Let n_i^k be the set of $\boldsymbol{X}_j$s which are the k-nearest neighbors to $\boldsymbol{X}_i$. That is, for each i, let $s_{ij_1} \geq s_{ij_2} \geq \ldots \geq s_{ij_{n-1}}$ be the ordered similarities (in decreasing order) among $\{s_{ij} : j \in (1, 2, \ldots, i-1, i+1, i+2, \ldots, n)\}$. Then $\boldsymbol{X}_{j_1}, \boldsymbol{X}_{j_2}, \ldots, \boldsymbol{X}_{j_k}$ are the k-nearest neighbors of $\boldsymbol{X}_i$.

   (a) Write a function `n_k(i, j,...)` in C (of appropriate arguments) which takes in a dataset and for any two observation pairs (i, j) and a user-supplied similarity measure, returns 0 if i and j are not among the k-nearest neighbors of each other and 1 if they both are among the k-nearest neighbors of the other. Let $\mathcal{N}_k$ be the set of (i, j) for which the above function returns 1. [*15 points*]

   (b) Use the above to write a function in C which takes in a dataset and a user-supplied similarity measure and filters out those observations that are not a K-nearest neighbor to another observation. In order to make this function easy to use in big data problems, make sure that you do not unnecessarily store the distance matrix. Test the function. [*5 points*]

3. For any $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ (assumed to be filtered as with a call to the previous function), calculate the similarity matrix $\boldsymbol{W}$ with non-zero elements $W_{ij} = s_{ij}$ when $(i, j) \in \mathcal{N}_k$ and $s_{ij} = \text{Corr}(\boldsymbol{X}_i, \boldsymbol{X}_j)$ when $\text{Corr}(\boldsymbol{X}_i, \boldsymbol{X}_j) > \rho$. Clearly, $\boldsymbol{W}$ is a sparse symmetric matrix that can be stored in the sparse packed format. (Actually, it can be stored even more efficiently, since the diagonals are all unity but we will ignore this for now.)

   Let $\boldsymbol{G}$ be the diagonal matrix with $\boldsymbol{W}\mathbf{1}$ in the diagonals. Here $\mathbf{1} = (1, 1, \ldots, 1)'$.

   It is common to think of the points $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ as nodes on a graph, with edges between nodes weighted by similarities W_{ij} and g_i's as the so-called node degrees, that is, the sum of the weights of the edges connected to node ii.

   Consider the standardized (with respect to the node degrees) graph Laplacian matrix as $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{G}^{-1}\boldsymbol{W}$. Then $\boldsymbol{L}$ is sparse and further the smallest eigenvalues of $\boldsymbol{L}$ are the same (in reverse order) as the largest eigenvalues of $\boldsymbol{G}^{-1}\boldsymbol{W}$, and they share the same respective eigenvectors. Obtain the first m eigenvectors. How does one

determine m? One may do so on the basis of the eigenvalues of $\boldsymbol{L}$ that are close to zero (equivalently, on the basis of those eigenvalues of $\boldsymbol{G}^{-1}\boldsymbol{W}$ that are close to unity). The above framework provides the background for spectral clustering (which additionally involves clustering these eigenvectors).

Write a function in C which does the above in a general framework. Test the function (if it is too cumbersome, you may test the function using a subset of the observations below for testing purposes.) [*25 points*]

4. *Microarray gene expression data.* The file, `diurnaldata.csv` contains gene expression data on 22,810 genes from Arabidopsis plants exposed to equal periods of light and darkness in the diurnal cycle. Leaves were harvested at eleven time-points, at the start of the experiment (end of the light period) and subsequently after 1, 2, 4, 8 and 12 hours of darkness and light each. Note that there are 23 columns, with the first column representing the gene probeset. Columns 2–12 represent measurements on gene abundance taken at 1, 2, 4, 8 and 12 hours of darkness and light each, while columns 13-23 represent the same for a second replication. Note that the file has a header and also that the first column, in character, is not particularly of value.

   Use the functions written in the problems above to obtain the eigenvectors, using only the first replication, and provide plots of the eigenvectors for different values of k, ρ and m. Comment. [*15 points*]

5. In this problem, you will generate and print out all possible strings $c_1c_2\cdots c_8$ where $c_i \in \{A, C, G, T\}$. Write a program that uses eight nested loops to output the strings. Can you rewrite the program to take advantage of the bitwise operators? Can you adapt the second program to output, with one minor change, all possible strings $c_1c_2\cdots c_{16}$ of length 16? [*15 points*]