

Clustering and Classification

In this problem set you will consider the data set we have been discussing extensively in class, namely the genechip data from two types of leukemias, ALL and AML. The data sets are posted on the course website and come in four files: ALL_AMLtrain, ALL_AMLtest, trainlabels, and testlabels. Trainlabels and testlabels are vectors of 1's and 0's, the position of each label corresponding to the analogous column in ALL_AMLtrain and ALL_AMLtest respectively (1=ALL,0=AML). Again, the columns of the expression data represent individual leukemia samples and the rows correspond to genes. For more information on the data set you can visit the original website where the data is located: http://www-genome.wi.mit.edu/mpr/data_set_ALL_AML.html.

We will use a pair of complementary techniques to cluster the data as well as two complementary methods for classifying the data. To cluster the data, we will use a “top-down” method, k-means clustering, and a “bottom-up” method, hierarchical clustering. You will be using built in Matlab commands for hierarchical clustering, and you will write your own code for the k-means algorithm. For classification, we will use a generative method, namely mixture models, as well as a discriminative method, logistic regression.

Clustering:

1. Use the built in functions `pdist()` and `linkage()` to perform hierarchical clustering (with a Euclidean metric) of the training data. Use `dendrogram()` to visualize the results. Turn in the class vector you discover (a vector of numbers with the *i*th element marking which cluster the *i*th sample belongs to) by forcing the assignment of only two class labels, as well as a printout of the dendrogram.
2. Write a function to perform k-means clustering. The input to this function is a data set (use the training data). The output is a class vector as defined above. The function should choose *k* random data points to seed the iteration with. With *k*=2, comment on the robustness of the classes discovered by different runs. Perform another run(*k*=2) with the two centroids initialized to the average of the first 19 samples and the last 19 samples of the training data. Turn in the class vector for this initialization as well as the code for k-means.
3. Write a function to compute the significance of the size 2 clusters discovered in problems 1 and 2. The input will be the class vector discovered by clustering as well as the TRUE labels (trainlabels). The output will be a p-value corresponding to the probability that a completely random assignment of labels could have performed *as well or better* than the clustering algorithm at assigning the *correct* labels. This p-value is a kind of performance measure. For example, if you get 3 misassignments, the p-value should reflect the probability that you would have made 1, 2, or 3 misassignments had you been blindfolded. Turn in the p-value associated with the hierarchical class vector (2 cluster case) as well as the p-value associated with the

2-means class vector (with centroids in the fixed initialization described above).
Turn in the code as well.

4. Write a function to perform feature selection. The input will be a data set (training data) and the associated class labels (trainlabels). The output will be a vector of 1's and 0's where a 1 in position i indicates that the i th gene is differentially expressed between ALL and AML. Use the likelihood ratio statistic to determine which genes have this property. We will assume that the true variance of a gene in the ALL samples is the same as its true variance in the AML samples. Don't be careless about how you incorporate this constraint. You will also want to ensure that the significance level you demand is meaningful across the entire set of tests you will perform (a "family wide" p-value of .05). Turn in the indexes of the significant genes (print them) and the code used to discover them. Repeat parts 1-3 of the homework with a new training dataset, one that only keeps data from the significant genes.

Classification:

1. Write a function that performs classification with a mixture model. The inputs are trainingdata, traininglabels, testdata, and testlabels. The output is a class vector. The assumptions of the model distributions are that they be Gaussian with covariance matrices that are equal between the two classes and that are diagonal. Perform the classification with the full data set and then with the feature selected versions of trainingdata and testdata. Turn in the resulting class vectors with each. Turn in the code as well.
2. Write a function that performs discriminative classification with logistic regression (the inputs and outputs will be the same as for the mixture method). You will remember that part of the process involves a nonlinear optimization to discover the optimal parameter vector. The code for this is provided on the website and is called logregparvec.m. Perform the classification with the full data set and then with the feature selected versions of trainingdata and testdata. Turn in the resulting class vectors with each. Turn in the code as well.

ANNOTATE(i.e. use comments) all code very carefully.

The assignment is due April 11th. There will be no extensions.

Notes: The hierarchical clustering functions are part of the statistics toolbox. This is available on Athena Matlab, but not necessarily on a PC version. Explanations of all the functions in the statistics toolbox are available at <http://www.mathworks.com/access/helpdesk/help/toolbox/stats/stats.shtml>. Detailed help is also available from the graphical interface of matlab, accessible by typing "desktop" on Athena Matlab.

