CS 284A Scribe Notes - k-Means and Hierarchical Clustering

Matt Kayala

February 11, 2008

1 Microarray Experiments

Very interesting biological questions involve gene expression:

- How active various genes are under different cell types and conditions?
- How genes interact with each other?

A method to attempt to answer these questions is to measure at different times:

- the amount of mRNA in a cell.
- the amount of expressed protein in a cell.

mRNA microarrays allow researchers to measure mRNA for almost all genes simultaneously. There is a caveat however, mRNA is an indirect measure of the amount of protein expression because of post-transcriptional regulation.

2 Motivation for clustering methods

Given a large number of data points, what are some methods to "group" or cluster data points together? The example given in class entails gene expression data from a time-series microarray experiment. The experiment measures gene expression at different times in the cell cycle.

We are given n data points, $x_1, x_2, \ldots, x_n$, where a data point x_i represents a single gene. The data point is made up of m measurements of expression at each time point. Conceptually, one can imagine that each data point x_i represents a point in m-dimensional space.

Being able to cluster the genes given this time-series microarray experiment will give us groups of genes with similar profiles over time. For example, a potential cluster could include genes that are active (highly expressed) during and around Mitosis, but are not active at other stages in the cell cycle.

3 k-Means Clustering

One method of clustering is called k-Means. Given that there are k clusters to find, what are the centers of these clusters and which cluster does each gene belong to?

3.1 Algorithm

1. Randomly pick k centers: $\{\mu_1, \mu_2, \ldots, \mu_k\}$.

2. Repeat until centers do not change:

- **a. (E-Step)** Assign each point x_i to nearest cluster S_j where $j \in \{1, \ldots, k\}$ for all $i \in \{1, \ldots, n\}$:

$$x_i \rightarrow S_j : \min_j ||x_i - \mu_j||^2$$

- **b. (M-Step)** Update the centers $\mu_1, \mu_2, \ldots, \mu_k$ given the assignments of the points. For all $j \in \{1, \ldots, k\}$:

$$\mu_j = \frac{1}{|S_j|} \sum_{x \in S_j} x$$

3.2 Issues

There are several issues with the algorithm:

1. How do we know the algorithm will converge?
2. How do we choose k?
3. Metrics - how do we measure distance between points (and centers)?
4. How do we choose the initial centers?

1. For convergence, it is easy to see that the two update steps never *increase* the sum of distances from each point to its nearest center. The **2 a.** step assigns each point to a cluster with the center that is closest, i.e., any other choice of cluster assignment would increase the sum of distances. The **2 b.** step, by recalculating the centers, minimizes the sum of distances from center within each cluster, i.e., any other choice of center for a cluster would be larger than the center calculated here. So, while this sum of distances to nearest center criterion is never *increasing* that does not imply always *decreasing*, therefore the algorithm will converge to a locally optimal, but not necessarily globally optimal, set of centers and point labeling.
2. It is not well-known how to choose k for the algorithm. One could try several different values of k and examine the results. The choice of k is constrained by these two facts: choosing $k = 1$ would yield the vacuous result of all data points belonging to a single

cluster with a center that is the center of the data set. Choosing $k = n$ yields the similarly vacuous result that each point is assigned to their own cluster.

3. For metrics, it is very common to use euclidean distance (as used in the algorithm above). One could use another metric if there is a compelling biological motivation to do so.

4. For choosing the initial centers, there are several options. Choosing centers based on some heuristic method such as distributing close to a large amount of points. One could also run the algorithm many times with different random initial centers and examine the results.

3.3 Probabilistic Motivation

One could view k-Means as a method to minimize the following objective function:

$$C(\mu_1, \mu_2, \ldots, \mu_k) = \sum_{j=1}^{k} \sum_{x \in S_j} \mathrm{d}(x, \mu_j)$$

where $\mathrm{d}(x_i, x_j)$ is the distance between two points, i.e., for Euclidean distance $\mathrm{d}(x_i, x_j) = ||x_i - x_j||^2$. How can we state this within a probabilistic framework?

First, let a given x_i be a member of the j-th cluster S_j, and lets assume that

$$p(x_i | x \in S_j) \propto N(\mu_j, \sigma^2).$$

That is, each data point is drawn from a normal distribution centered around its cluster's mean and with a fixed variance σ^2. Furthermore, lets assume that $\mu_1, \mu_2, \ldots, \mu_k$ are all known and that x_i has a membership z_i where $z_i \in \{1, 2, \ldots, k\}$. Then the total likelihood of the data is given by:

$$p(x_1, x_2, \ldots, x_n | \mu, z) \propto \prod_{i=1}^{n} \exp\{-\frac{||x_i - \mu_{z_i}||^2}{2\sigma^2}\}.$$

Now, we do not know z_i, so this yields:

$$p(x_1, x_2, \ldots, x_n | \mu) \propto \prod_{i=1}^{n} \sum_{j=1}^{k} p(z_i = j) \exp\{-\frac{||x_i - \mu_j||^2}{2\sigma^2}\}.$$

Now to find the Maximum Likelihood estimate of the μ's, we try to find the μ^* (a vector of μ's) that maximizes the above expression:

$$\mu^* = \max_{\mu^*} \prod_{i=1}^{n} \sum_{j=1}^{k} p(z_i = j) \exp\{-\frac{||x_i - \mu_j||^2}{2\sigma^2}\}$$

This form of maximum likelihood is similar to the form that motivated the EM algorithm. We can maximize by the following two steps:

E Step $p(z_i = j|x_i) \propto \exp\{-\frac{||x_i - \mu_j||^2}{2\sigma^2}\}$

M Step $\mu_j^* \propto \sum_{i=1}^{n} p(z_i = j)x_i.$

This is very similar to the algorithm stated above. In fact if we let $\sigma \to \infty$ this is equivalent to the k-Means algorithm. This is easy to see because of the normalization ($\propto$ denotes some normalization constant such that the sum over all probabilities = 1) the first term will be one for a single j for each i and zero for the remaining j. This is equivalent to assigning to the closest center. The second term will reduce to recalculating the centers with the normalization. Hence when $\sigma \to \infty$ the probabilistic framework is equivalent to k-Means.

4 Hierarchical Clustering

Hierarchical clustering is a method to cluster data points without specifying how many clusters *a priori*. The end result is a *dendrogram* where the leaves are data points and interior nodes represent a cluster made up of all children of the node. This is termed "*hierarchical*" because one produces a hierarchy of clusters. With the dendrogram, one can choose how many clusters to use by cutting the tree at some particular height.

4.1 Algorithm

Given a data set $D = (x_1, x_2, \ldots, x_n)$ of n points to begin with, calculate a distance matrix M with all of the pairwise distances between points. Process the following recursively until D has only a single data point:

1. Choose the two points x_i, x_j from D such that the distance between the two points is the minimum.

$$i, j = \min_{i,j} d(x_i, x_j), \text{ where } i \neq j.$$

2. Cluster x_i, x_j to form a new point c. The new point could be the mean of the two points or some other distance metric.

3. Remove x_i, x_j from D and insert c. Recalculate the pairwise distance matrix M.

4.2 Issues

The main issue with hierarchical clustering is the metric to use. Not only does one have to consider distance between points, but one needs to consider distance between clusters (combined data points). Most often the actual distance is determined by Euclidean distance (as in k-Means). The issue then becomes what do we consider to be important in combined data points. There are several methods proposed:

1. **Center** Calculate the Euclidean distance from the means of the clusters.

$$d_{ctr}(c_i, c_j) = \left|\left| \frac{1}{|c_i|} \sum_{x \in c_i} x - \frac{1}{|c_j|} \sum_{x \in c_j} x \right|\right|^2$$

2. **Average** Calculate the average Euclidean distance between all pairwise combinations of data points in the two clusters.

$$d_{avg}(c_i, c_j) = \frac{1}{|c_i| \cdot |c_j|} \sum_{x \in c_i} \sum_{y \in c_j} ||x - y||^2$$

3. **Min or Max** Take the max or the minimum distance between any two points in the respective clusters.

$$d_{max}(c_i, c_j) = \max d(x, y) \text{ for } x \in c_i, y \in c_j$$

$$d_{min}(c_i, c_j) = \min d(x, y) \text{ for } x \in c_i, y \in c_j$$