CSC346/646 Parallel Computation Spring 2015

K-means Clustering

For our first project, we will develop a parallel program using OpenMP to do K-means clustering. Your main program should accept the name of a data file (data format will be discussed in class) and a positive integer k. E.g. `% cluster datafile 12`

Your program should use the Forgy method to initialize the k-means. See *menehune.opt.wfu.edu/csc346* for a link to download a sample data file. Assume for our first try at this, that the sample data file will be 2-dimensional array of ones and zeros indicating "position occupied" and "position not occupied" respectively. Clustering is performed on the occupied cells; your program should ignore the not-occupied cells. Your output should be a similar-sized two dimensional array where each entry is given a cluster number (i.e., 1, 2, 3, ..., k). Unoccupied cells should be given the number zero. We can use `octave` to visualize our clustering by assigning a different color to each cluster.

Description (from Wikipedia)

Given a set of observations $(x_1, x_2, , x_n)$, where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into $k \leq n$ sets $S = \{S_1, S_2, , S_k\}$ so as to minimize the within-cluster sum of squares (WCSS). In other words, its objective is to find:

$$\mathbf{S} = \arg\min \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} ||\mathbf{x} - \mu_i||_2^2$$

where μ_i is the mean (centroid) of the points in S_i.

History (from Wikipedia)

The term "k-means" was first used by James MacQueen in 1967, though the idea goes back to Hugo Steinhaus in 1957. The standard algorithm was first proposed by Stuart Lloyd in 1957 as a technique for pulse-code modulation, though it wasn't published outside of Bell Labs until 1982. In 1965, E.W.Forgy published essentially the same method, which is why it is sometimes referred to as Lloyd-Forgy. A more efficient version was proposed and published in Fortran by Hartigan and Wong in 1975/1979.

Standard algorithm (from Wikipedia)

The most common algorithm uses an iterative refinement technique. Due to its ubiquity it is often called the k-means algorithm; it is also referred to as Lloyd's algorithm, particularly in the computer science community.

Given an initial set of k means $\{m_1^{(1)}, ..., m_k^{(1)}\}$, the algorithm proceeds by alternating between two steps:

Assignment step: Assign each observation to the cluster whose mean yields the least within-cluster sum of squares (WCSS). Since the sum of squares is the squared Euclidean distance, this is intuitively the "nearest" mean. (Mathematically, this means partitioning the observations according to the Voronoi diagram generated by the means).

$$S_i^{(t)} = \{x_p : ||x_p - m_i^{(t)}||_2^2 \leq ||x_p - m_j^{(t)}||_2^2 \text{ for all } j, 1 \leq j \leq k\},$$

where each x_p is assigned to exactly one $S^{(t)}$, even if it could be assigned to two or more of them.

Update step: Calculate the new means to be the centroids of the observations in the new clusters.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Discussion (from Wikipedia)

Since the arithmetic mean is a least-squares estimator, this also minimizes the within-cluster sum of squares (WCSS) objective.

The algorithm has converged when the assignments no longer change. Since both steps optimize the WCSS objective, and there only exists a finite number of such partitionings, the algorithm must converge to a (local) optimum. There is no guarantee that the global optimum is found using this algorithm.

The algorithm is often presented as assigning objects to the nearest cluster by distance. The standard algorithm aims at minimizing the WCSS objective, and thus assigns by "least sum of squares", which is exactly equivalent to assigning by the smallest Euclidean distance. Using a different distance function other than (squared) Euclidean distance may stop the algorithm from converging. Various modifications of k-means such as spherical k-means and k-medoids have been proposed to allow using other distance measures. Initialization methods

Commonly used initialization methods are Forgy and Random Partition. The Forgy method randomly chooses k observations from the data set and uses these as the initial means. The Random Partition method first randomly assigns a cluster to each observation and then proceeds to the update step, thus computing the initial mean to be the centroid of the cluster's randomly assigned points. The Forgy method tends to spread the initial means out, while Random Partition places all of them close to the center of the data set. According to Hamerly et al., the Random Partition method is generally preferable for algorithms such as the k-harmonic means and fuzzy k-means. For expectation maximization and standard k-means algorithms, the Forgy method of initialization is preferable.