

Project 3: CS621, Fall 2009
Due Date: 3:15PM, November 24, 2009

Parallel Clustering using the k -Means Algorithm

Let a term-by-document matrix be

$$A = \begin{pmatrix} 0.577 & 0 & 0 & 0.408 & 0 & 0.577 \\ 0.577 & 0 & 1.0 & 0.408 & 0.707 & 0 \\ 0.577 & 0 & 0 & 0.408 & 0 & 0.577 \\ 0 & 0 & 0 & 0.408 & 0 & 0 \\ 0 & 1.0 & 0 & 0.408 & 0.707 & 0.577 \\ 0 & 0 & 0 & 0.408 & 0 & 0 \end{pmatrix},$$

where the columns of A are normalized document vectors. The centroid of a dataset is the average of the documents, computed as

$$c = \frac{1}{n} \sum_{i=1}^n d_i.$$

To organize a document collection efficiently, we would like to partition it into a few subcollections with closely related documents. This process is called *clustering*. Formally, let $\{S_1, S_2, \dots, S_k\}$ be a series of nonempty and non-intersect subsets of A . The problem of finding the k clusters of the document set A is to find k centroids $\{c_1, c_2, \dots, c_k\}$ such that the function

$$f(c_1, c_2, \dots, c_k) = \frac{1}{n} \sum_{i=1}^n \left(\min_j \text{dist}^2(d_i, c_j) \right) \quad (1)$$

is minimized. Here the function $\text{dist}(d_i, c_j)$ is the Euclidean distance between the documents d_i and c_j .

The k -means is an approximate solution procedure to the problem (1). It gained its popularity due to its simplicity and intuition. The algorithm is an iteration procedure and requires the number of clusters, k , is given *a priori*. Suppose that a set of guessed centroids $\{c_1^{(0)}, c_2^{(0)}, \dots, c_k^{(0)}\}$ is given, the algorithm iterates as follows.

- Step 1: for each document $d_i, 1 \leq i \leq n$, compute $\text{dist}(d_i, c_j^{(m)})$ with respect to each cluster centroid, $c_j, 1 \leq j \leq k$. Put d_i in the closest cluster;
- Step 2: for each cluster $1 \leq j \leq k$, compute the new centroid pseudo document c_j .
- Step 3: repeat Steps 1 and 2 until the convergence is reached.

The criterion function for convergence can be computed, e.g., as

$$f_m = \frac{1}{n} \sum_{i=1}^n \left(\text{dist}^2(d_i, c_j^{(m)}) \right).$$

Given a convergence tolerance ϵ , the k -means algorithm stops when $|f_{m+1} - f_m| < \epsilon$.

Parallel Computation. Assume you use 3 processors. You should first read the matrix from a file and then distribute each two columns of the matrix A to each processor. Assume you have 3 clusters, each one consists of the documents in a processor to start with. Do about 10 iterations and print out the values of the objective function f_m at each iteration. Plot a graph showing the value of f_m as a function of the number of iterations and show the final clustering solution, i.e., which documents belong to which clusters.

Note that there is no requirement that all clusters have the same number of documents. In addition to the code, you need to write down and *submit your strategies of implementations*. You should avoid moving documents between the processors.

What to Submit. Submit a hard copy of your code and the computed results. Please also e-mail your electronic code to jzhang@cs.uky.edu. Your electronic copy should match your hardcopy turned in.