An Improved Algorithm for Segregating Large Geospatial Data

Kara E. Scott[1]
Research Fellow
Email: skara@siu.edu

Tonny J. Oyana[1]
Assistant Professor and Graduate Program Director,
Email: tjoyana@siu.edu
[1]Department of Geography and Environmental Resources,
Southern Illinois University
1000 Faner Drive, MC 4514,
Carbondale, IL 62901-4514, United States of America
Tel: +1 618-453-3022
Fax: +1 618-453-6465

SUMMARY

This study investigates an improved k-means clustering algorithm for segregating large geospatial data. Although the conventional k-means method is sufficient for datasets with minimal data, it does not perform well and, therefore yields poor accuracy for high-volume datasets. Clustering methods are one of the most important components in data classification, visualization, and mining high-volume datasets. The primary aim of this study is to explore two individual methods that were originally designed to increase the overall performance of k-means clustering: Mashor's updating method and the Davies-Bouldin validity index.

KEYWORDS: discovery geovisualization, *FES-k*-means, large-scale geospatial data, spatial data mining, *k*-means, geography and algorithms

INTRODUCTION

This study investigates an improved *k*-means clustering algorithm for segregating large geospatial data. The significance of improving clustering algorithms stems from an increasing demand for better visual exploration and data mining tools that function efficiently in data-rich and computationally-rich environments. Clustering has been applied in gene expression data (Yano and Kotani 2003; Demiriz et al. 1999), database marketing (Demiriz et al. 1999), georeferencing of biomedical data to support disease informatics research (Oyana et al. 2005; Curtis 1999), and exploratory data analysis, data mining and knowledge discovery (Alahakoon et al. 2000; Kohonen et al. 2000; Bengio et al. 2000), just to name a few applicable areas. Determining the structure of clustered data is a significant factor in classifying, visualizing, and geo-spatial mining of high-volume datasets. Thereby, the importance of strengthening these methods in order to perform such functions is substantial.

The primary aim of this study is to combine two methods that were, individually, designed to increase the overall performance of *k*-means clustering: an updating method explored by Mashor (1998), and the Davies-Bouldin validity index (DBI) (Davies and Bouldin 1979) (an algorithm used to measure partitioning quality following the *k*-means clustering procedure). Intervening during the updating process with Mashor's method is expected to generate equally active cluster centers at a faster convergence rate. Three major advancements that will take place as a result of these improvements are 1) the elimination of cluster number fluctuation; 2) efficient servicing of larger datasets; and 3) adequate analysis of datasets with largely scattered data. The findings of this study are vital for the relatively new and expanding field of geospatial data management.

EXPERIMENTAL DESIGN

The goals in this experiment were to determine if merging the updating method explored by Mashor (1998), and the Davies-Bouldin validity index (1979) would result in more consistent and accurate clusters, which would lead to an overall increase in performance accuracy for k-means.

The plan for this experiment was to run a number of tests, employing the suggested clustering method, on two datasets derived from actual applications in biomedical and disease informatics and on a random computer-generated dataset with 10,000 data points distributed evenly into 10 clusters. The first—obstructive sleep apnea (OSA) identified by ICD-9CM 780.51, .53, and .57 with 3,943 data points in point and polygon vector formats—was used as a test dataset. The second—blood lead levels (BLL) for children living in the City of Chicago—was the working dataset. The second dataset is a large one containing records in excess of 880,000.

These experiments were conducted in SOM Tool box 2.0 for Matlab (SOM Project, HUT, Finland), Matlab 7.0 (The MathWorks, Inc., Natick, Massachusetts), and ESRI ArcGIS 9.0 (ESRI, Inc., Redlands, California). We decided on these computational environments to test suggested improvements because the SOM toolbox and Matlab provide the necessary environment to compute complex equations.

DESCRIPTION OF ALGORITHMS

K-means Clustering Method

MacQueen (1967) describes *k*-means as a process for partitioning an *N*-dimensional population into *k* sets on the basis of a sample. According to Kanungo et al. (2002), and Fraley and Raftery (1998), *k*-means is the most widely used and simplest form of clustering. One of the drawbacks of this algorithm is that it is computationally expensive. The *k*-means algorithm is formally defined, for this study, as follows:

1. Let *k* be the number of clusters and the input vectors defined as $X = [x_1, x_2,...,x_n];$
2. Initialize the centers to *k* random locations in the data and calculate the mean center of each cluster, μ_i (where *i* is the i^{th} cluster center);
3. Calculate the distance from the center of each cluster to each input vector, assign each input vector to the cluster where the distance between itself and μ_i is minimal, re-compute μ_i for all clusters that have inherited a new input vector and update each cluster center (if there are no changes within the cluster centers discontinue re-computation);
4. Repeat Step 3 until all the data points are assigned to their optimal cluster centers. This ends the cluster updating procedure with *k* disjoint subsets.

The partitions are based on a within-class variance, which measures the dissimilarity between input vectors and cluster representatives, μ_i, using the squared Euclidean distance in equation 1.

$$\sum_{i=1}^{K}\sum_{n=1}^{N}(\|x_n - \mu_i\|)^2 \qquad \text{Equation (1)}$$

In Equation 1, *N*, and *k* are the number of data and the number of centers, respectively, and x_n is the data sample belonging to center μ_i (Mashor, 1998; MacQueen, 1967; Alsabti et al. 1998; Demiriz et al. 1999; Vesanto and Alhoniemi 2000). Refer to Figure 1 for the pseudo code for the standard *k*-means algorithm.

Standard *k*-means

```
Determine the number and the dimensionality of points and set the number of clusters in the training set
Extract the data points
k = num_clusters;
```

```
[m,n]=size(number_clusters);
Initialize and derive center clusters and clusters
for i=1:k  // Main loop of the algorithm
μ = μ_center(i, :);
center(j)= X(1:num_clusters, :);
for j=1:k
                              dist (i,j) = (X(i,:) – centers(j,:))^2;
        center (j,:) = (sum(μ_center(j)):),1)/num_points;
        if (min_dist (j) < 0 | dist<min_dist);
                min_dist = dist;
                num_clusters (i) = k;
                                        end
Recalculate the centers;
Compute the error function until errors do not change significantly;
Update centers until cluster membership no longer changes;
End
```

Figure 1: Pseudo code for standard *k*-means algorithm

Davies-Bouldin Validity Index (DBI)

The *k*-means clustering algorithm employs the Davies-Bouldin Index (DBI) to evaluate cluster quality because DBI is ideal for indexing spherical clusters (Vesanto and Alhoniemi 2000). DBI computes the ratio of within-cluster scatter to between-cluster separation. Hence, the preferred DBI for optimal clustering strives to minimize the ratio of the average dispersions of two clusters, namely C_i and C_j, to the Euclidean distance between the two clusters, according to Equation 2 (Davies and Bouldin 1979; Demiriz et al. 1999; Vesanto and Alhoniemi 2000).

$$\frac{1}{k}\sum_{k=1}^{K}\max_{i\neq j}\frac{e_i+e_j}{D_{ij}} \qquad \text{Equation (2)}$$

In Equation 2, e_i and e_j are the average dispersion of C_i and C_j, respectively. D_{ij} is the Euclidean distance between C_i and C_j. The average dispersion of each cluster and the Euclidean distance are calculated according Equations 3 and 4, respectively (Demiriz et al. 1999).

$$e_i = \frac{1}{N_i}\sum \left\|x-\mu_i\right\|^2 \qquad \text{Equation (3)}$$

$$D_{ij} = \left\|\mu_i-\mu_j\right\|^2 \qquad \text{Equation (4)}$$

In Equation 4, μ_i is the center of cluster C_i consisting of N_i points and x is the input vector. Refer to Figure 2 for the pseudo code for generating DBI for the *k*-means algorithm.

```
k-means Davies-Bouldin Validity Index (DBI)
Determine the number and the dimensionality of points and set the number of clusters in the
training set
Extract the data points
[m,n]=size(number_clusters);
Initialize and derive clusters as provided by the k-means algorithm
for b=1:n
   g(:,b)=number_clusters(:,b);
Minimize the ratio of average dispersion of two clusters to the Euclidean distance between
```

```
these two clusters
[q,r]=size(num_clusters);
for r=1:q
   for c=1:r
      e=[];
      for j=1:n
      e=[e (num_clusters(r,c)-center(j)).^2];
      end
      [val,Ind]=min(e);
      G(count1,Ind)=num_clusters(r,c);
      count1=count1+1;
   end
end
Compute the error function until errors do not change significantly;
Update centers and select one with smallest DBI;
End
```

Figure 2: Pseudo code for generating Davies-Bouldin Validity Index (DBI) for the k-means algorithm

The *K-D* Tree Data Structure

According to Bentley (1975), Bentley (1979), and Gaede and Guenther (1997), the *k-d* tree is one of the most prominent *d*-dimensional data structures. The structure of the *k-d* tree is based on a binary search tree that represents a recursive subdivision of the universe into spaces by means of (*d*-1)-dimensional hyperplanes Gaede and Guenther (1997) and Pelleg and Moore (1999). The two main properties of the *k-d* tree are (1) each splitting hyperplane has to contain at least one data point; and (2) interior nodes must have one or two descendants. These properties make the *k-d* tree data structure an attractive candidate for reducing the computationally expensive nature of *k*-means algorithm. In fact, Alsabti et al. (1998), Pelleg and Moore (1999), and Kanungo et al. (2002) have investigated the use and efficiency of *k-d* tree in *k*-means environment and they concluded that presenting clustering data using this data structure provides enormous computational advantages. Alsabti et al.'s (1998) main principle was based on organizing vector patterns so that all closest patterns to a given prototype can be found efficiently. To realize this principle, the authors employed two main strategies as follows:

1. Considered that all the prototypes were potential candidates for the closest prototype at the root level.

2. Obtained good pruning methods based on simple geometrical constraints. Alsabti et al.'s (1998) pruning method was based on computing the minimum and maximum distances to each cell. For each candidate $\mu_{i,}$ they first obtained the minimum and maximum distances to any point in the subspace; then found the minimum of maximum distances (*MinMax*); and later pruned out all candidates with minimum distance greater than *MinMax*. Other pruning methods have been investigated by Pelleg and Moore (1999) and Kanungo et al. (2002). Pelleg and Moore (1999) used the bisecting hyperplane that assigns the input vector based on the minimal distance to the winning cell. Whereas, Kanungo et al. (2002), used the same approach, but they assigned the input vector to a cell based on minimal distance to the midpoint of the winning cell candidate. In this study, we have adopted Kanungo's pruning method due to greater efficiency than Alsabti et al. (1998) and Pelleg and Moore (1999).

The basic structure of Alsabti et al.'s (1998) algorithm is given below.

1. Let the number of dimensions be *d* and the depth of the *k-d* tree be *D*.

2. Construct a *k-d* tree for the data points in reference by bisecting the dimension with the longest length through the midpoint. Note that the root of such a tree represents all the patterns, while the children of the root represent subsets of the patterns completely contained in subspaces. The nodes at the lower levels represent smaller subspaces.
3. Randomly generate initial prototypes or draw randomly from the dataset.
4. Perform a number of iterations until a termination is met.

Mashor's Updating Method

A method intended to resolve the *k*-means problem has been described by Mashor (1998), who suggested a multi-level approach. Recall that most clustering algorithms employ a similarity measure with a traditional Euclidean distance that calculates the cluster center by finding the minimum distance calculated per Equation 1. In *k*-means clustering, as the data sample is presented, the Euclidean distances between the data sample and all the centers are calculated and the nearest center is updated according to Equation 5.

$$\Delta\mu_i(t) = \eta(t)[x(t) - \mu_i(t-1)] \qquad \text{Equation (5)}$$

In Equation 5, i indicates the nearest center to the data sample $x(t)$. The centers and the data are written in terms of time (t), where $\mu_i(t-1)$ represents the center's location during the previous clustering step, and $\eta(t)$ is the adaptation rate. The adaptation rate, $\eta(t)$, can be selected in a number of ways. Conventional equations for $\eta(t)$ are an adaptive method introduced by MacQueen (1967), or a constant adaptation rate and a square root method introduced by Darken and Moody (1990). These methods adjust the cluster centers at every instant by taking the cluster center at the previous step into consideration. Some of the problems associated with such adjustments are reviewed in Mashor (1998) who suggested a better clustering performance based on a more suitable adaptation rate, $\eta(t)$. According to Mashor, a good updating method is one that has a large clustering rate at the beginning and a small steady state value of the adaptation rate, η(t), at the end of training time. We integrated Mashor's suggested adaptation rate in Equation 5 with the one in Equation 6 to derive the most appropriate cluster centers.

$$\eta(t) = \eta(t-1)/e^{[1/r]} \qquad \text{Equation (6)}$$

In Equation 6, $r = k + t$. At each step of the learning, the adaptation rate should decrease so that the weights of the training data can converge properly.

Equation 5 is re-written by substituting $\eta(t)$ from Equation 6 to obtain the final equation (Equation 7) as follows:

$$\Delta\mu_i(t) = ((\eta(t-1)/e^{[1/r]})([x(t) - \mu_i(t-1)])) \qquad \text{Equation (7)}$$

Refer to Figure 3 for the pseudo code for Mashor's updating procedure for the improved *k*-means algorithm.

The results from these experiments reflect the mathematical improvements highlighted below.

***k*-means Mashor using a *k-d*-tree data structure**

```
        Determine the number and the dimensionality of points and set the number of clusters in the
        training set
        Extract the data points
        Construct a k-d-tree for the data points in reference
        Find closest points using a k-d tree
        Initialize centers
        [cp, dist] = kdtree( ... );
        [m,n]=size(number_clusters);
```

```
Initialize and derive clusters as provided by the k-means algorithm
for b=1:n
   g(:,b)=number_clusters(:,b);
Define a suitable adaptation rate by applying Mashor's update procedure
[q,r]=size(num_clusters);
       for r=1:q
          for c=1:r
             e=[];
             a=10; // adaptive constant
Adjust cluster centers at every instant by considering the previous one
         for j=1:num_clusters
                nc=num_clusters; // nc is the number of clusters
                t = num_clusters-j;
                if j= =1
                   a(j)=a(j)*(exp(1/(nc+t)));
                    e=[e (num_clusters(r,c)-center(j)).^2];
                else
                   a(j)=a(j-1)*(exp(1/(nc+t)));
                   e=a(j)*[e (num_cluster(r,c)-center(j-1)).^2];
                end
             end
Compute the error function until errors do not change significantly;
Update centers until cluster membership no longer changes;
End
```

Figure 3: Pseudo code for Mashor's update procedure for the k-means algorithm

Mathematical Improvements to the *K*-means Clustering Method

The three specific issues that were addressed by implementing the proposed improvements of the *k*-means algorithm in this study are as follows:

1. Cluster stabilization.
2. Provide efficient service to a population of data with large numbers of clusters.
3. Ability to service extensively scattered data.

Figure 4 illustrates the properties associated with the standard *k*-means algorithm and the resulting improvements of the new algorithm. These improvements were achieved via the implementation of an adaptation factor selected that could suitably adjust itself at each learning step in order to find the winning cluster for each data point efficiently. The suggested improvements updated the cluster centers by taking time into consideration and also analyzing the cluster center during the previous clustering steps while generating new cluster centers. To handle large geospatial data, we implemented the *k-d* tree data structure.

RESULTS AND DISCUSSIONS

Experimental results support our early assumption that by employing the Davies-Bouldin validity Index in conjunction with Mashor's proposed updating method, we have successfully resolved

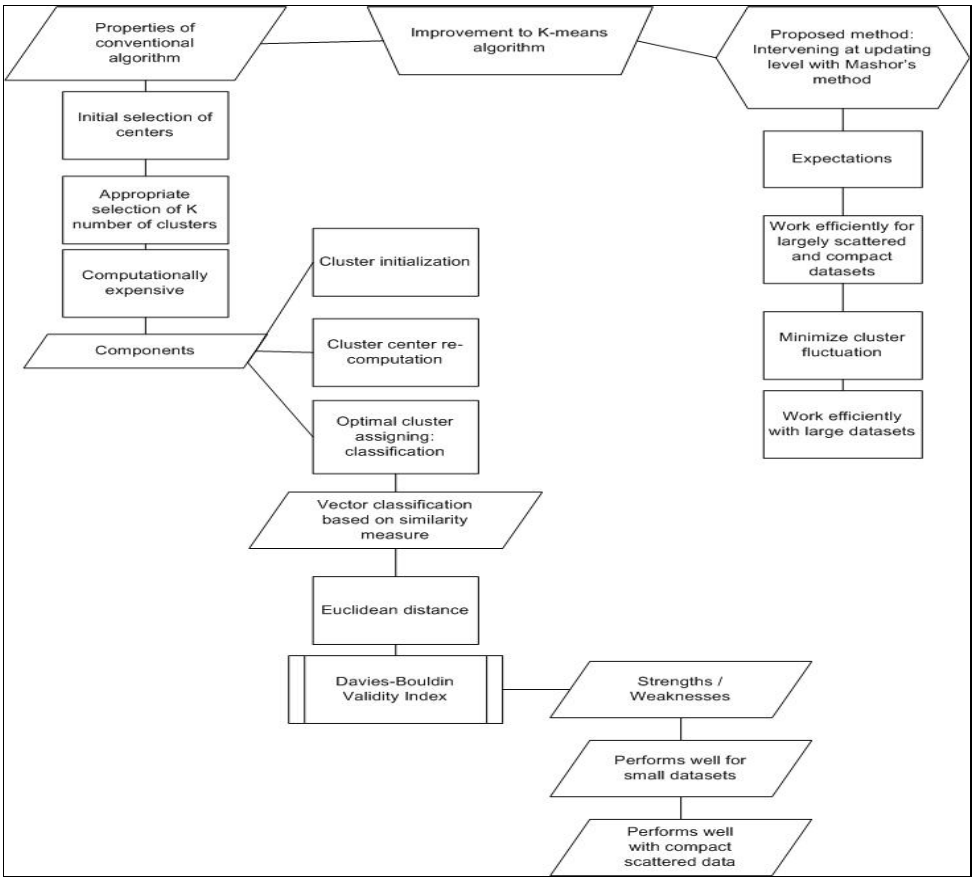
Properties of conventional algorithm
Improvement to K-means algorithm
Proposed method: Intervening at updating level with Mashor's method
Initial selection of centers
Appropriate selection of K number of clusters
Computationally expensive
Components
Cluster initialization
Cluster center re-computation
Optimal cluster assigning: classification
Vector classification based on similarity measure
Euclidean distance
Davies-Bouldin Validity Index
Strengths / Weaknesses
Performs well for small datasets
Performs well with compact scattered data
Expectations
Work efficiently for largely scattered and compact datasets
Minimize cluster fluctuation
Work efficiently with large datasets

Figure 4: Flowchart illustrates the properties associated with K-means algorithm and suggested improvements

efficiency concerns associated with MacQueen's *k*-means clustering method (MacQueen 1967; Mashor 1998). These experiments resulted in an improvement in the cost equations involved in the standard *k*-means clustering procedure in order to maintain consistency and accuracy in clustering results. The improvements have provided a good means of generating approximately the same number and correct clusters every time. We have implemented a more efficient approach to present large scale datasets using the *k-d* tree data structure as suggested by Alsabti et al. (1998), Pelleg and Moore (1999), and Kanungo et al. (2002).

The *k-d* tree data structure, nearest neighbour query, original *k*-means, and Mashor's adaptability rate, when implemented and utilized together, provide a very fast, computationally-efficient, and scalable environment to segregate a very large geospatial dataset. Preliminary results lend credibility to this original research idea; nonetheless, further research is required to investigate how to effectively integrate existing methods and the montage of these methods, which have been found to be highly successful.

From the results, we find that the suggested compilation of the conventional clustering and pruning methods have resolved many problems that exist within each of the individual methods. The two new clever methods, referenced as mathematically improved learning-SOM (MIL-SOM*) (Oyana et al. 2006) followed by fast, efficient and scalable *k*-means (*FES-k*-means*) have shown ground-breaking improvements in the following areas: segregation of very large geospatial data; efficient transformation of high-dimensional datasets to low dimensions, while preserving topological relationships; minimization of cluster fluctuation; and adequate analysis of largely scattered datasets. Lastly, a very prominent benefit is its ability to perform the aforementioned tasks at an efficient speed.

The engaging implementation and performance of the *FES- k*-means* clustering algorithm when applied to three different geospatial datasets undoubtedly reveals great potential. This study is, however, continuing to explore these potentials, along with other properties, while anticipating additional benefits. Further research is also required in four key areas: 1) how to explore and handle outliers; 2) how to evaluate resulting clusters; 3) how to measure reliability of results; 4) and how to effectively display more than three dimensions. Presently, we are only able to classify multidimensional large geospatial datasets; however, the challenge is with the visual display of datasets containing more than three dimensions.

The success of these methods will provide better visual exploration and data mining tools for a range of disciplines that rely of clustering methods for managing, exploring and visualizing large geospatial datasets.

CONCLUSIONS AND RECOMMENDATIONS

Implementation of the suggested algorithm during the *k*-means clustering procedure has proven to be efficient in each of the problem areas mentioned. This improvement in the traditional *k*-means clustering method, allows for an even more efficient tool for visualizing and mining vast and extensive datasets. The improved *k*-means algorithm (*FES-k*-means) has provided a better result than the original *k*-means algorithm, which delineates cluster boundaries based on the best DBI validation. To reduce the computational-expensive nature of the newly-merged *k*-means algorithm *(FES-k*-means), we have implemented the *k-d* tree data structure.

BIBLIOGRAPHY

Alahakoon, D. S.K. Halgamuge. 2000. Dynamic self-organizing maps with controlled growth for knowledge discovery. IEEE Transactions on Neural Networks. 11(3). 601–614.

Alsabti, K., S. Ranka, V. Singh. 1998. An efficient K-means clustering algorithm Proceedings of IN/IPPS/SPDP Workshop on High Performance Data Mining 1998.

Bengio, Y., J. M. Buhmann, M. J. Embrechts, J. M. Zurada. 2000. Introduction to the special issue on neural networks for data mining and knowledge discovery. IEEE. 545–549.

Curtis, A. 1999. Using a spatial filter and a geographic information system to improve rabies surveillance data. Emerging Infectious Diseases. 5(5): 603–606.

Darken, C., and J. Moody. 1990. Fast adaptive k-means clustering: some empirical results. *International joint conference on neural networks*. (2) 233–238.

Davies, D. L. and D. W. Bouldin. 1979. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 1(2): 224–227.

Demiriz, A., Bennet, K. P., and Embrechts, M. J. 1999. Semi-supervised clustering using genetic algorithms. *Artificial Neural Networks in Engineering (ANNIE-99)*: 809–814.

Fraley, C., and A. E. Raftery. 1998. How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. *The Computer Journal* 41(8): 578–588.

Kanungo, T., D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A.Y. Wu. 2002. An efficient k-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 24(7): 881–892.

Kohonen T., S. Kaski, K. Lagus, J. Salojarvi, J. Honkela, V. Paatero, and A. Saarela. 2000. Self organization of a massive document collection. *IEEE Transactions on Neural Networks*. 11(3): 574–585.

MacQueen, J.B. 1967. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability; held at the Statistical Laboratory, University of California.* 282–297.

Mashor, M.Y. 1998. Improving the performance of *k*-means clustering algorithm to position the centers of RBF network. *International Journal of the Computer, The Internet and Management.* 6(2).

Oyana, T.J., Achenie, L.E.K., Cuadros-Vargas, E., Rivers, P.A., and Scott, K.E. (2006). A Mathematical Improvement of the Self-Organizing Map Algorithm. In Proceedings of the International Conference on Advances in Engineering and Technology (AET 2006), July 16–19, 2006.

Oyana, T.J., Boppidi, D., Yan, J., and Lwebuga-Mukasa, J.S. 2005. Exploration of geographic information systems-based medical databases with self-organizing maps: A case study of adult asthma. In Proceedings of the 8th International Conference on GeoComputation, 1st–3rd August 2005, Ann Arbor, University of Michigan.

Vesanto, J. and E. Alhoniemi. 2000. Clustering of the Self-Organizing Map. *IEEE Transactions on Neural Networks.* 11(3): 586–600.

Yano and Kotomi. 2003. Clustering gene expression data using self-organizing maps and *k*-means. SICE Annual Conference in Fukui. Fukui University.3211–3215.