A Novel Data Clustering Algorithm Based on Electrostatic Force Concepts

Masoumeh Kalantari Khandani, Parvaneh Saeedi, Yaser P. Fallah, Mehdi K. Khandani

Abstract—**In this paper a new method, called Force, is presented for finding data clusters centroids. This method is based on the concepts of electrostatic fields in which the centroids are positioned at locations where an electrostatic equilibrium or balance could be achieved. After finding the data points, criteria such as minimum distance to centroid can be used for clustering data points. The performance of the proposed method is compared against the k-means algorithm through simulation experiments. Experimental results show that the Force algorithm does not suffer from problems associated with k-means, such as sensitivity to noise and initial selection of centroids, and tendency to converge to poor local optimum. In fact, we show that the proposed algorithm always converges to global equilibrium points, regardless of the initial guesses, and even in presence of high levels of noise.**

I. Introduction

Data Clustering is one of the important topics in data analysis and it has been the subject of many research efforts. Data clustering is the unsupervised classification of patterns of data items into groups or partitions. It has many applications in data mining, pattern recognition, image processing and analysis and bioinformatic. There is a fine line between clustering and classification. The difference is that in classification the classes are predefined and data points are assigned to each class, whereas in clustering data points are grouped into classes that have to be identified. Data clustering discovers the overall distribution patterns of the dataset, through finding centers and clusters of data. Its algorithms are specified as being deterministic or stochastic (probabilistic), numerical or categorical. Our focus in this paper is on numerical, deterministic partitional algorithms.

In general, data clustering is done based on measuring some similarity between data items; for example, based on the proximity of data points in some space. There are many algorithms that cluster data, the most applicable and famous one is k-means [2]. In k-means a user-defined number of centroids are found. The data points are assigned to these centers according to the minimum distance constraints, forming clusters. The algorithm iterates by modifying the centroids according to some rules until the centroids do not move in two successive iterations. While k-means has been proven to be an effective algorithm, it suffers from shortcomings including sensitivity to initial center values and noise sensitivity.

In this paper we present a new numerical data clustering algorithm that is inspired by the rules of electrostatic fields. This novel approach allows efficient and robust clustering of multi dimensional data sets. The algorithm is especially suitable for larger data sets and produces predictable results, which are not sensitive to the initial guess points. It always converges to the same solution under different conditions. The presented algorithm is of deterministic nature. In this algorithm we assume data points are negative electrical charges scattered in a multi dimensional space. To cluster these charges, a number of positive charges (configurable parameter) will be released in the space; these charges will move due to the electrostatic force so that all electrical charges reach to an electrostatic equilibrium or balance. When the balance is achieved, positive charges will be at true centroids of the found clusters. At the end of this process, the data point assignment to each cluster is achieved using the minimum distance constraint.

II. Summary of Relevant Research

There are many numerical clustering algorithms. A detailed survey of different types of clustering methods could be found in [1]. Perhaps the most popular data clustering algorithm is the k-means algorithm [2]. K-means clustering is a well-known partition-based technique in unsupervised learning. K-means algorithm, in particular, has the following carachteristics: works only on numerical data, is efficient for processing large data sets, often converges to a local optimum, and its generated clusters have convex shapes. Despite moderate complexity, k-means algorithm is sensitive to initial seed selection [1]. To address some of k-means issues including sensitivity to the initial centers, several solutuions have been proposed. For instance, ISODATA [4] finds the optimal initial partitions by merging or splitting arbitrarily chosen initial partitions. Another algorithm, presented in [3], tackles the challenge of selecting a good initial cluster by applying dynamic programming over the principal component direction. A heuristic clustering dissimilarity function

This work was supported in part by Natural Sciences and Engineering Research Council of Canada (NSERC).

M. Kalantari Khandani is with the Engineering Science Dept. of Simon Fraser University, BC, Canada. (School of Engineering Science, 8888 University Dr., Burnaby, BC, Canada, 778-782-4371; e-mail: mka47@sfu.ca, psaeedi@sfu.ca)

Y. P. Fallah is with Departments of Electrical Eng. & Computer Sciences, and Civil and Environmental Eng., University of California Berkeley (email: yaserpf@berkeley.edu)

M. K. Khandani is with Dept. of Electrical and computer Eng., University of Maryland, College Park (email: mehkalan@umd.edu)

is incorporated into the sub-optimal k-means algorithm. The approach proposed in [5] uses a combination of GA and k-means; the genetic algoritm is used to find good initial cluster centers and k-means algorithm is then applied to find final partitions.

There are other clustering methods, beside k-means based algorithms that are suitable for large data sets. Two of them are the CLARANS (Clustering Large Applications based on Random Search) [6], and the BIRCH (Balanced Iterative Reducing and Clustering) algorithms [7]. CLARANS is a cluster analysis technique applied to spatial attributes of data points. It analyzes several random samples to find cluster centroids from the original dataset. In general it is stated that cluster analysis is not suitable for large data sets; however, CLARANS is claimed to be an efficient algorithm. The work in [7] (BIRCH) suggests that CLARANS is not efficient in all situations and it may fail to find real local minima. The BIRCH algorithm keeps brief information about candidate clusters by applying a dynamic tree structure, with leaf nodes representing clusters. It claims that is capable of handling the noise.

In this paper a new clustering algorithm is presented that relies on the laws of electrostatic. It simulates electrostatic fields in order to position cluster centroids in the appropriate places. This algorithm is suitable for large data sets and can handle noise as is explained in details in the following sections.

III. "FORCE" CLUSTERING ALGORITHM

The proposed algorithm employs the law of electrostatics, which describes the nature of forces among electric charges. Direction of the electrical force is derived from electric fields around these charges. The main assumptions made here are:

1. Cluster centroids have large, positive, variable and dynamic (in position) charges.
2. Data points have single, negative, fixed and static (in position) charges.

When positive centers are randomly dropped amongst negative data points, an electric field is formed which forces the centers to move to places where forces are balanced. Under the balance condition the centers do not move anymore. We call this configuration of points the electrostatic equilibrium. The force between centers is repulsive, while the force between centers and data points are attractive. These forces are calculated from: $F = c.\frac{q_1 q_2}{r^2}$

Here c is a constant, r is the distance between two charges and q_1, q_2 are two charges that F is computed for. If the distance r tends to zero, it will be replaced with a constant small distance R_0. The direction of the force between two charges can be identified by the unit vector $(\vec{r}_1 - \vec{r}_2)/(\| \vec{r}_1 - \vec{r}_2 \|)$, where r_1 and r_2 are charges' coordinate vectors.

The centers will naturally move toward areas where data points are located. Meanwhile they repulse each other and therefore they will not land in the same cluster of data points. This means that regardless of initial random positions of the centers, they always move toward the center of clusters (if clusters exist). The data points are always associated with the center closest to them. Thus, the clusters are formed based on the minimum distance constraint.

The charge of each data point is a constant value, but charges of centers are updated dynamically. The charge of each center is set in proportion to the number of points associated with it (the presumed cluster which is coupled with this center). For example, if N_j points are associated with a center j, the charge Q_j for the this center (assuming the charge of each point is one), is set as follows:

$$Q_j = \alpha . N_j \qquad 0 < \alpha < 1 \qquad (1)$$

The sum of the charges for the centers (positive charges) is therefore always slightly less than the sum of the charges of the datapoints. This means that the centers will definitely be attracted towards the data points and their mutual repulsive force will not be able to overcome the attractive force of the datapoints. If $\alpha >= 1$, the repulsive force will move the centers very far from the ideal point so that the steady state is reached where the centroids are not located at cluster centers. If $\alpha << 1$, both centers may be attracted to the same cluster and will be placed too close to each other. If $0 << \alpha < 1$, centers will be located close to the ideal points. The total force on each center is calculated as follows:

$$F_j = F_j^{\,D} + F_j^{\,C} \qquad (2)$$

The force applied to each center by other centers (set C) and data points (set D) is calculated as:

$$\vec{F}_j = \sum_{i \neq j, i \in D \cup C} \frac{Q_j Q_i}{R_{ij}^{\,2}} \frac{(c_j - p_i)}{\|c_j - p_i\|}, \qquad R_{ij} = \begin{cases} \|c_j - p_i\| & \|c_j - p_i\| > R_0 \\ R_0 & \|c_j - p_i\| \leq R_0 \end{cases} \qquad (3)$$

Here c_j and p_i are vectors, describing the positions of centers and data points. Note that Q_i for each data point i is -1. R_0 is the minimum distance as explained before.

Based on the force on each centroid, the algorithm estimates a direction in which the centroid should move. The speed of the movement, or the steps taken in each iteration, is subject to many factors such as the weight of the centroids, the charge masses, etc. However, at this time a fixed step size η is utilized. Thus the direction of the force is the only parameter

that is required for estimating the centroid's new position.

$$c_j^{(\tau+1)} = c_j^{(\tau)} + \eta_1 \frac{F_j}{\|F_j\|} \quad (4)$$

Here $c_j^{(\tau+1)}$ is the new position of the cluster centroid, $c_j^{(\tau)}$ is the previous position, and $F_j / \|F_j\|$ is a unit vector of force which provides the center's heading direction. Knowing the direction, there are several ways to control the speed of the center's movement. In this paper, first a fixed step size is utilized. Adaptive adjustment of the step size is also explored in Section IV-A

After each iteration, new centroids' positions are updated and new clusters based on the minimum distance constraint are formed. With new clusters, the charge of each center is recalculated according to (1), and the forces are computed. The algorithm will stop when the position of each cluster center moves less than a predefined threshold in two consecutive iterations. One of the most significant benefits of this algorithm is that the found cluster centers after different runs of the algorithm with different initial centroids are at most different by 2η (the step size). Another advantage of Force over k-means is that it performs a globalized search, while k-means based algorithms perform localized searches.

IV. Enhancing the Algorithm

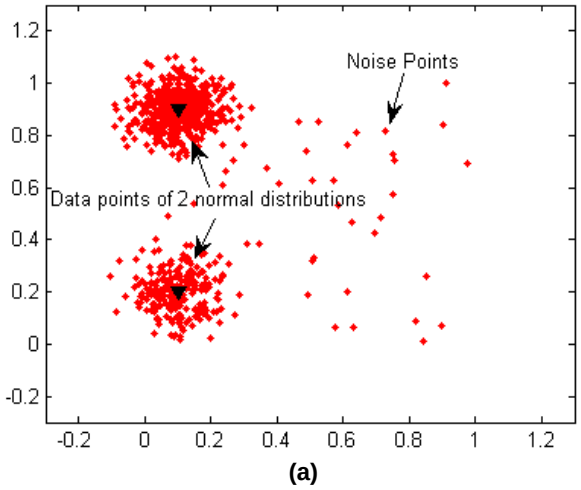
To verify the proposed algorithm, several experiments are conducted. In the first experiment two sets of normally distributed data sets are considered; some additive noise points are also added to the data points. Figure 1-a shows the distribution of the data points; in this figure, the triangles represent the actual cluster centers around which the data points were normally distributed. Figure 1-b shows resulting centroids after running the Force algorithm. The final results (marked by diamonds) are located at a very close distance to the actual cluster centers (error controlled by step size η set to 2η, e.g. less than 0.01 in this case). As displayed in Figure 1-b, the centers repel each other and move toward the clusters (data masses). In the example depicted in Figure 2-a, in the earlier iterations, both positive charges are attracted to the bigger mass of negative data points; however, when they become close to the mass, their mutual repulsive force will repel them and only one of them is attracted to that mass and the other moves toward the other mass.

A. Adaptive Step Size

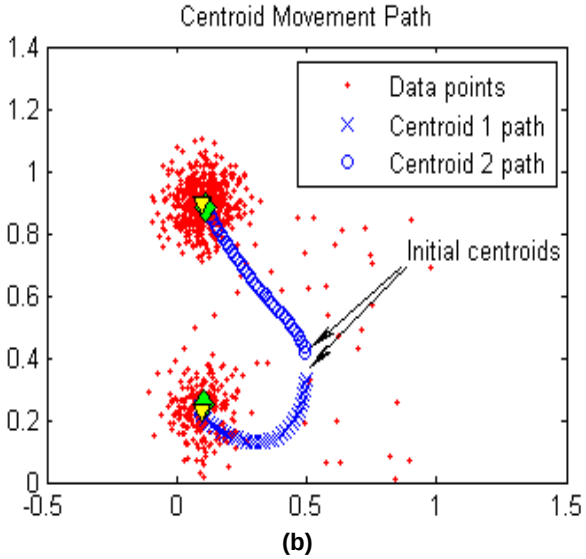
From Figure 2-a, it can be seen that the algorithm has to make many steps for the second centroid, which is located at a far distance from the cluster's center, before reaching to tis final location. This initiated an adaptive step adjustment that makes the algorithm to choose longer steps when the centroid is far from the actual cluster center. Note that for centers located at a far distance from the clusters' data points, the computed force is small. Therefore, the step size is modified to be inverseley proportional to the computed force. The modification of equation (4) is presented by:

$$c_j^{(\tau+1)} = c_j^{(\tau)} + \eta_1 \frac{F_j}{\|F_j\|} + \eta_2 \frac{F_j}{\|F_j\|^2} \quad (5)$$

The result of this enhancement can be seen in Figure 2-b**,** in which a fraction of the number of steps is reduced. Therefore, for the remaining work presented in this paper, the adaptive step adjustment with η_1=0.04, η_2=1000 is incorporated. Also in all presented figures, the final centroids positions found by the Force algorithm are marked with diamonds, while k-means results are marked with squares.



(a)



(b)

Figure 1 a) Data points and noise points, triangles are the actual distribution centers b) movement path of centroids for Force algorithm. Diamonds are the found centroids by the Force algorithm.

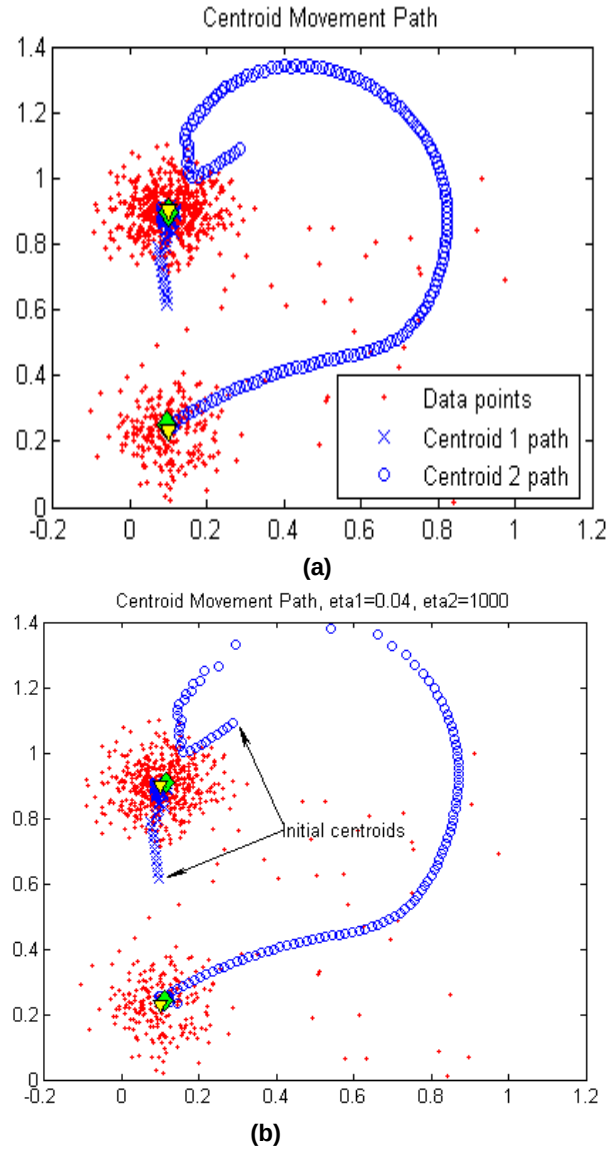


Figure 2 centroid movement path: a) Original algorithm using fixed step size b) enhanced algorithm using adaptive step size.

B. Run Time Reduction by Informed Initial Guesses

In order to improve the speed of the algorithm, a simple yet effective adjustment is suggested in this section. This adjustment includes placing the initial location of the centers in the middle of the space where data points are scattered. For this purpose, the center of gravity for the entire data point collection is found first. The initial centers are then placed at that location with some distance from each other. To see the effect o this adjustment, we plotted the path of the centroids for when the initial centroids were selected at the edge of the data range (Figure 3-a and Figure 4-a), and when they were selected through the adjustment (Figure 3-b and Figure 4-b). The difference between Figure 3 and Figure 4 is the level of additive noise, respectively 5%, 20%). The results clearly demonstrate that this enhancement significantly reduces the number of iterations required by the algorithm. It must be noted that the accuracy of the final solution is not jeopardized by this enhancement (maximum difference is still within the step size). Calculating these initial centroids is also a rather simple and computationally inexpensive task.

Faster convergence may also be achieved, if the initial centers are placed where a mass of data are detected. For example, a rough histogram of the data points could assist in placing the centers at local optima of the histogram. This will be investigated in future and therefore is not discussed further here

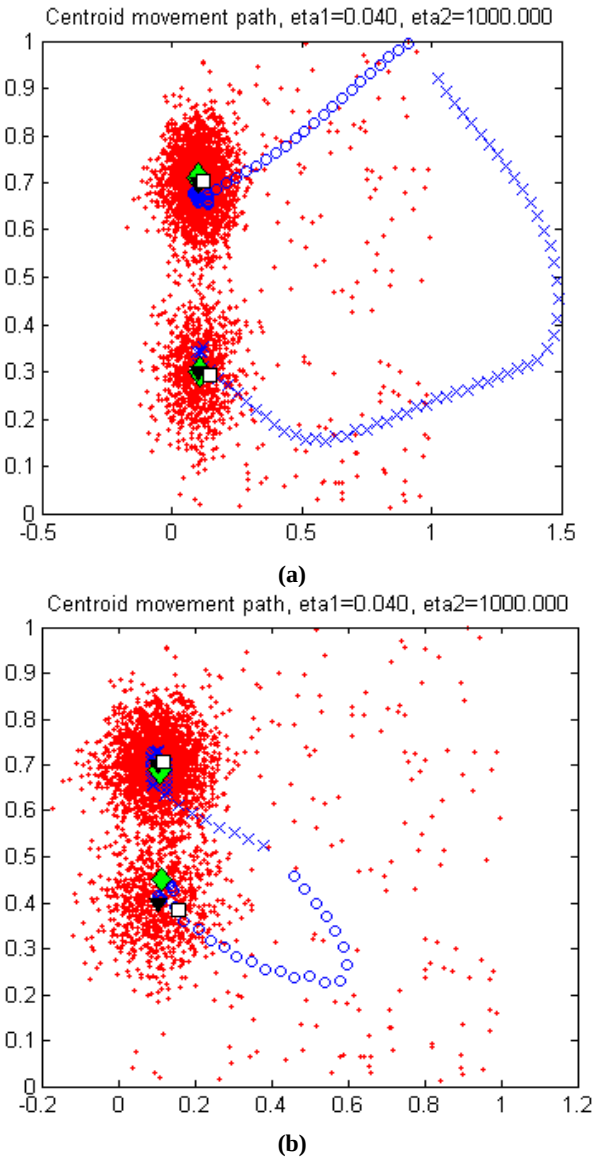
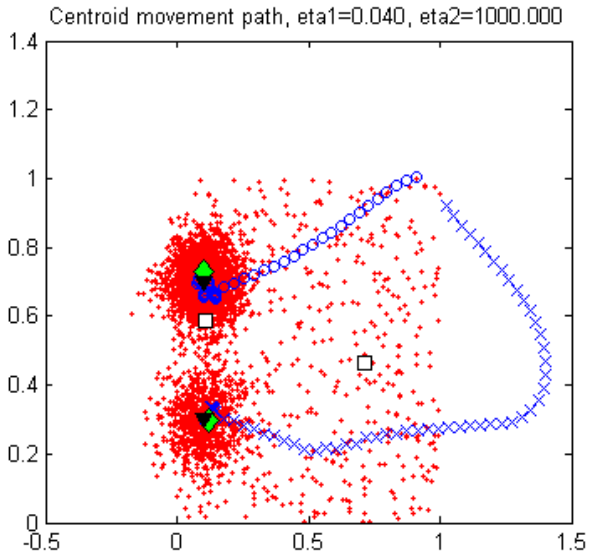
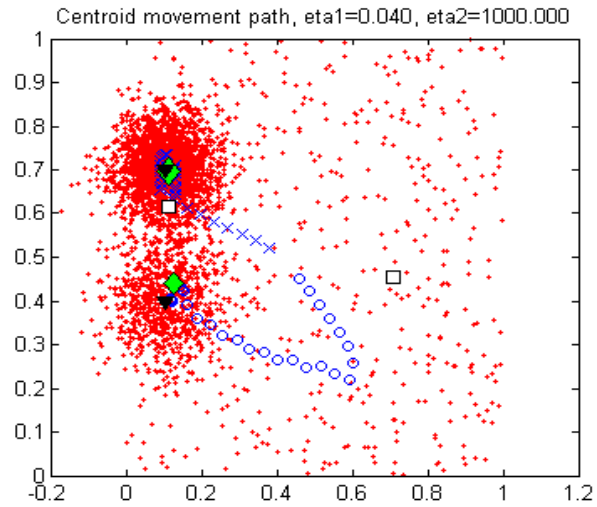


Figure 3 initial guesses: squares are found k-means centers and diamonds are Force centers a) initial centroids are far from the data masses, b) initial centroids are in the middle space of the data point distributions. The level of noise is low in both a and b cases.



(a)



(b)

Figure 4 initial guesses: squares are found k-means centers and diamonds are Force centers a) initial centroids are far from the data masses, b) initial centroids are in the middle space of the data masses. The level of noise is high in both cases.

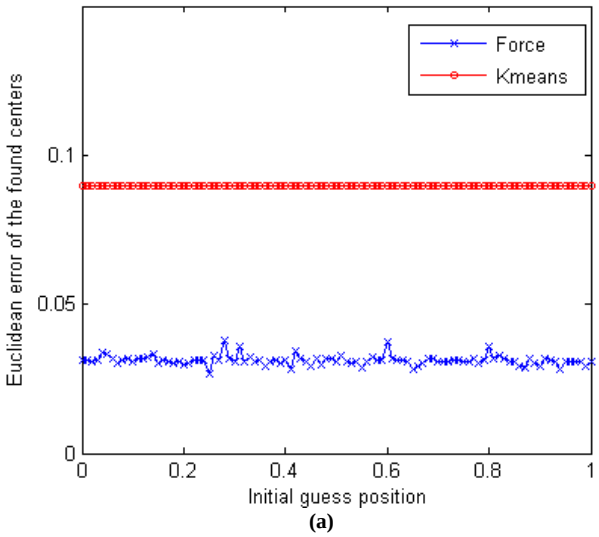
V. Performance Evaluation

The performance of the proposed method is examined for various initial centroid locations and different additive noise levels and distributions. The performance is also compared against k-means method. The code utilized in this paper for k-means is from Matlab's Statistics Tool Box.

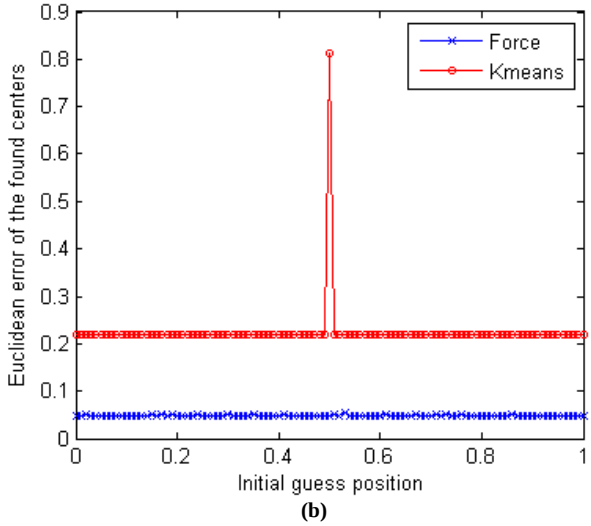
To evaluate the performance of the algorithm for various initial centroid locations, we measured the Euclidean distance of the found centroids and actual cluster centers. The two initial centers were placed at positions (x,y) and (x,1-y), where x and y changed from 0 to 1 in 100 steps (data and noise points are distributed in this range as well, as shown in Figure 3). The level of noise in Figure 5-a is 5%, and in Figure 5-b 20%. The results depicted in Figure 5 show that the Force algorithm always converges to the equilibrium point with error limited to $2\eta_1$ (as in equation (5)).

Note that at the equilibrium state, the force F_j is large and the second tem in (5) tends to zero; whereas the first term always has a magnitude of η_1. Therefore, the expected difference between the found centroids in different runs of the algorithm is at most $2\eta_1$.

In Figure 5 we also show the Euclidean error for the k-means algorithm. It is observed that for low noise situations, k-means always converges to the same point, for this specific experiment; however, increasing the noise cause the algorithm to misidentify the cluster centers for one case. This situation is also depicted in Figure 4, where an additional noise causes k-means algorithm to converge to a wrong position. The Force algorithm remains robust to noise.



(a)



(b)

Figure 5 Euclidean error of the found centers for the k-means and Force algorithm for different initial guesses, starting close to clusters, and moving to the far right of the data range: a) with 5% noise b) with 20% noise.

To further study the effect of noise, we measured the Euclidean error at different levels and various noise distributions in two additional sets of experiments. .

In the first set, the Euclidean distance between the found centroids (by k-means and Force) and the actual clusters' centers at different noise levels are measured. Here, the noise was uniformly distributed in the space, (0,1), along both axes. Figure 6 compares the Euclidean error for both algorithms at different noise levels. The error was computed by averaging the results for 50 repetitions of the experiment using different initial points. The level of noise is specified by ratio of the number of noise points to the total number of data points. It is expected that the error will increase as the noise level increases. It can be seen that the Euclidean error for k-means method increases rather quickly while the error for Force algorithm rises very slowly. As mentioned earlier, this test was performed for a uniformly distributed noise.

The second test set examines the behavior of both methods under the condition in which the mean value of noise distribution varies from far left to the far right side of the data masses. In this test, two levels of noise (30% in Figure 7-a and 40% in Figure 7-b) have been simulated.

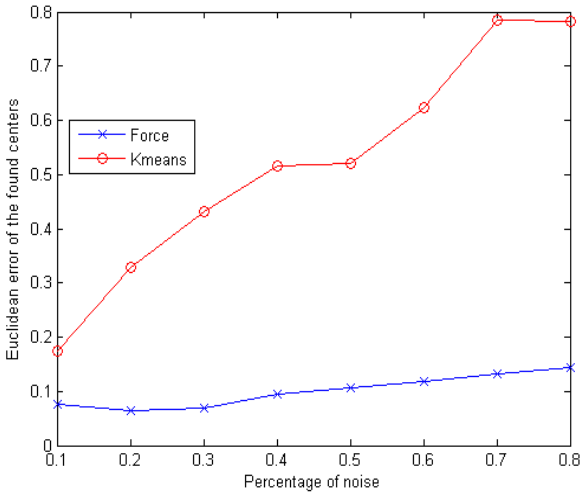
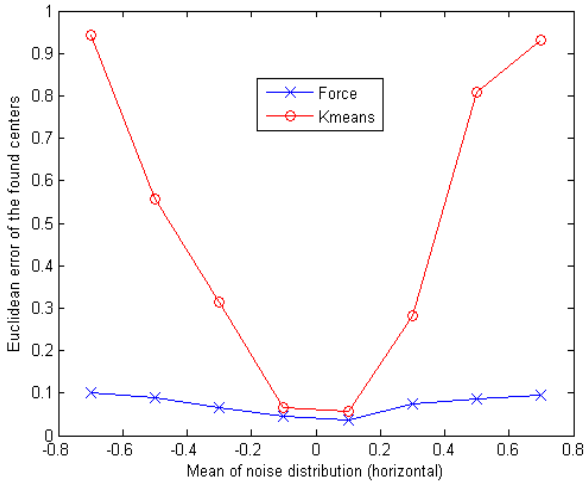
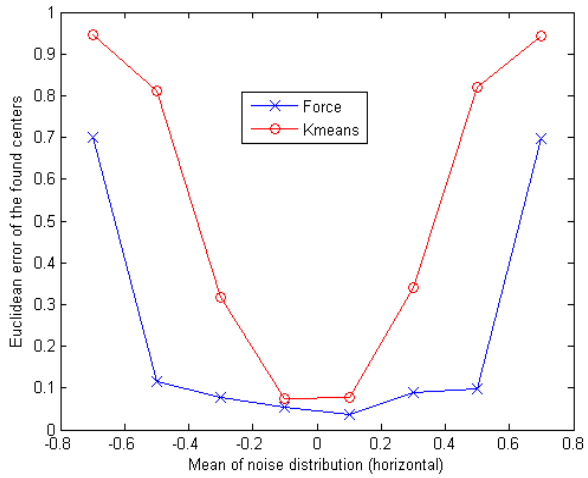


Figure 6 Euclidean error between the actual dataset centroids (ground truth) and found centroids as the noise level increases.

At each level the mean location of the noise distribution is moved from -0.7 to 0.7 in the test space, while the noise samples still maintained a uniform distribution around the mean location within the (0,1) interval. Figure 7 shows that around point zero, the error is very small for both methods. However, as the noise distribution center moves farther, the error increases significantly and rather quickly for k-means. The Force seems to handle the 30% added noise very gracefully. For the 40% added noise, Figure 7-b, only when the noise distribution center moves about +/-0.5 from the real data mass centroid, the Force's error becomes large



(a)



(b)

Figure 7 Mean location of noise distribution varies between -0.7 to 0.7: a) 30% added noise. b) 40% added noise.

VI. Conclusion

In this paper, a new unsupervised learning method for data clustering has been introduced. This method employs the rules of electrostatics, and finds the equilibrium points of a electric field, formed by data points as negative charges, and cluster centers as positive charges. The algorithm controls the path and charges assigned to the centers, to ensure that they converge to the equilibrium points regardless of the initial position.

The performance of the proposed algorithm is evaluated through simulation experiments, and is compared against k-means algorithm. Simulation results show that the proposed method is capable of handling noise better than k-means. Moreover, it does not suffer from instability rising from initial starting centroid guesses. An interesting future enhancement could be to incorporate histograms with large bins for estimating initial centers resulting in faster convergence to the final solutions.

One of the applications of data clustering is in image processing, especially in image segmentation. Investigating the performance of the Force scheme in processing very noisy images is of importance to the future development of this algorithm.

REFERENCES

[1] A.K. Jain, M.N. Murthy, P.J. Flynn, "Data Clustering: A Review", *ACM Computing Surveys*, Vol 31, No. 3, pp. 264-323, Sept. 1999

[2] J.McQueen,"Some methods for classification and analysis of multivariate observations"*In Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281–297, 1967

[3] Mantao Xu; Franti, P, "A heuristic K-means clustering algorithm by kernel PCA". *International Conference on Image Processing*, ICIP2004., Volume 5, pp. 3503 – 3506, 2004.

[4] BALL, G. H. AND HALL, D. J. "ISODATA, a novel method of data analysis and classification". *Tech. Rep.. Stanford University*, Stanford, CA. 1965

[5] BABU, G. P. AND MURTY, M. N. "A near optimal initial seed value selection in *K*-means algorithm using a genetic algorithm". *Pattern Recogn. Lett. 14,* 763–69. 1993

[6] R. Ng and J. Han, "Efficient and Effective Clustering Methods for Spatial Data Mining," Proc. 20th Conf. Very Large Databases, pp. 144–155, 1994

[7] ZHANG, T., RAMAKRISHNAN, R., AND LIVNY, M.. BIRCH: An efficient data clustering method for very large databases. *SIGMOD Rec. 25*, 2, 103–114. 1996