An on-line agglomerative clustering method for non-stationary data

I. D. Guedalia M. London M. Werman

February 3, 1998

Abstract

An on-line agglomerative clustering algorithm for non-stationary data is described. Three issues are addressed. The first regards the temporal aspects of the data. The clustering of stationary data by the proposed algorithm is comparable to the other popular algorithms tested (batch and on-line). The second issue addressed is the number of clusters required to represent the data. The algorithm provides an efficient framework to determine the natural number of clusters given the scale of the problem. Finally, the proposed algorithm implicitly minimizes the local distortion – a measure which takes into account clusters with relatively small mass.

In contrast, most existing on-line clustering methods assume stationarity of the data. When used to cluster non-stationary data these methods fail to generate a good representation. Moreover, most current algorithms are computationally intensive when determining the correct number of clusters. These algorithms tend to neglect clusters of small mass due to their minimization of the global distortion (Energy).

1 Introduction

1.1 Scale

Cluster Analysis is the process of finding the intrinsic structure in a data set without relying on *a priori* knowledge. Given a dataset and some measure of distance, or similarity, between data points, the goal in most clustering algorithms is to assign each data point (pattern) to a cluster "such that the patterns in a cluster are more similar to each other than to patterns in different clusters (Jain and Dubes, 1988)." However, the structure determined by the measure of similarity is a function of scale. While, two data points at a high resolution may seem very different, when viewed at a lower resolution they appear similar.

Figure 1 is an example of a data set that has at least two apparent scales. If the data points in the left corner are analyzed in isolation (at high resolution) they appear as three clusters. However, the same data when viewed in the larger picture are part of a single larger cluster. Hence, the answer to the question "How many clusters are there ?", in this data set, is twofold

(either three or nine). The "correct answer" is application dependent. Moreover, finite resources may limit the possible computable answers.

Clustering algorithms which minimize the global distortion [1] using a fixed number of centroids (see (Jain and Dubes, 1988) and (Duda and Hart, 1973)) ignore scale dependent structures. Thus, in the previous example LBG (Linde et al., 1980) using twelve centroids, for example, would find twelve clusters, which does not capture the structure of the data (3 or 9).

Many algorithms address this problem. Sebestyen (Sebestyen, 1962) utilized a threshold based adaptive approach to determine the number of clusters. MacQueen's K-means algorithm (MacQueen, 1967) solves this issue by utilizing two external parameters to define the coarseness and refinement of the clustering. Similarly, ISODATA (Ball and Hall, 1967), a batch algorithm, adjusts the number of clusters with an external threshold. A different approach taken, follows the Minimum Description Length criteria MDL (Rissanen, 1989). This approach tries to minimize the total cost of the representation of the data, when the cost is a parametric function of the distortion and of the model's complexity (Gath and Geva, 1989; Fritzke, 1994; Buhmann and Kuhnel, 1993). However, although these methods find an "optimal" solution, the number of centroids in the final representation depend on an external parameter. This parameter's affect on the outcome of the clustering must be determined experimentally and small perturbations in either the parameter or the data can result in drastically different solutions.

Another approach, which stems from statistical mechanics, uses a pseudo-temperature to escape local minima in the energy (distortion) function (Rose et al., 1990). This approach presents a natural solution to the problem of scale dependent structures. The clustering process, proposed by Rose et. al, consists of a cooling schedule in which the pseudo-temperature is lowered and a solution at each temperature is found. During this process the energy function undergoes something similar to phase transitions. Each such transition reflects a scale dependent solution.

1.2 Stationarity

Clustering algorithms can be divided into two classes, batch and on-line. Batch algorithms process the data off-line, hence, the temporal structure is ignored. Similarly, current on-line algorithms assume the data is produced by a stationary process[2], i.e is randomly drawn. In this situation the data can be sampled and clustered with a batch algorithm.

There exist many real world problems in which the data is produced by a certain type of non-stationary process. If a statistical sample of the data can be stored, then current algorithms can be used to cluster the data utilizing either a batch method or on-line method. However,

[1] See (Linde et al., 1980) for an extensive discussion of different measures of distortion.

[2] The process of clustering involves an exposure to data points, one at a time. This process can be viewed as a Discrete Time Real-valued Stochastic Process. Let $t = 1, 2, 3, \ldots$ be the time steps of points arrival, and let x_t be a d-dimensional point. The sequence x_t is a stochastic process. This process is a stationary process iff the joint distribution function of $(x_{t_1+h}, x_{t_2+h}, \ldots, x_{t_n+h})$ and $(x_{t_1}, x_{t_2}, \ldots, x_{t_n})$ are the same for all $h = 0, 1, 2, \ldots$ and an arbitrary selection of $t_1, t_2, \ldots, t_n$.

this may require computational resources that are not always available.

We address a set of problems which share the following property: on a short time scale it is pseudo-stationary, while on the long time scale the process has a sequential property. For example, in Figure 10, 9 clusters of data were produced sequentially. The points in each cluster were generated in a stationary process. First, all the points from the first cluster arrived randomly. This is followed by the random arrival of all the points in the second cluster etc. In this example, the short time scale is the number of points in each cluster. The long time scale is the whole process.

1.3 Small clusters

Given a set of data which includes a few small distinct clusters, how can the structure of the data be encoded such that the small clusters are represented? Existing algorithms which minimize the global distortion have the following dilemma: Either, the clustering is performed at a high resolution, resulting in an overfitting, or a low resolution clustering misses the small clusters. This is due to one of the following two reasons. If a batch method is used then the effect the small clusters have on the global distortion is diluted by the larger clusters. Similarly, if the data is generated by a stationary process, on-line methods will have the same problem of dilution.

Alternatively, if the data is produced by a non-stationary process, the problem becomes how to recognize that a new process began (arrival of a data from a new cluster) and to allocate a centroid to represent it[3].

The ART1 algorithm presents a solution to this problem (Carpenter and Grossberg, 1990).

Recently, Buhmann an Kuhnel (Buhmann and Kuhnel, 1993) proposed batch and on-line clustering algorithms which minimize a complexity term composed of the global distortion and the scale (complexity) of the model. The complexity term helps to solve the previous dilemma by increasing the effect distant points have on the system and minimizing the overfitting of the larger clusters. Unfortunately, the tuning of the scale parameter is very difficult. Moreover, the on-line algorithm assumes the stationarity of the data.

1.4 Example

As an example of a real world application concerned with the issues mentioned, one can consider the problem of quality control of fruit. The problem is how to classify a fruit into a quality class, based on a series of feature vectors measured from the fruit. One solution, is to use a sample of fruits, cluster their feature vectors, and correlates their features with the pre-defined quality classes. Then use the relationship between the clusters and the classes to classify the fruits. Due to the huge amount of data needed, an on-line method should be used, but stationarity of the data cannot be assumed. Some features of the fruits (for example, weather damages) tend to occur in bursts, for example, fruit that is damaged by a cold spell will appear at intervals

[3] In such situations, once the centroid is placed it will continue to represent the cluster even though it is relatively small. This is due to its relatively distant location.

determined by the weather. These features – which correlate with damages – are very meaningful for classifying the fruit and although very distinct, they are infrequent. Thus, the problem of quality control encapsulates the three issues raised: the data is non-stationary, there exist small meaningful clusters and the structure is scale dependent.

The proposed algorithm uses a novel approach towards such cases. The basic idea is that each point of data can belong to a new cluster. Thus, a new centroid is placed on **each and every** new point. Due to the limitation of finite memory this implies that a centroid must be allocated at the cost of the existing representation (centroids). This is done by merging the two closest centroids into one, at every step, minimizing the necessary loss of information.

The resulting algorithm, does not neglect small clusters, regardless if the data is produced by a stationary process or not. Furthermore, if a small cluster is distinct enough, it will not be lost by being merged into an existing cluster. Finally, if the data point was distinct but no other points were close to it enough to be merged with it (e.g, distant noise), the centroid can be removed at the end of the process (revealed by a very small weight).

2 Proposed on-line algorithm

The proposed algorithm is simple and fast. The algorithm can be summarized in the following three steps: For each data point arriving;

1. Move the closest centroid towards the point.
2. Merge the two closest centroids. This results in the creation of a redundant centroid.
3. Set the redundant centroid equal to the data point.

The algorithm can be understood as follows: Three criteria are addressed at each time step, minimization of the within cluster variance, maximization of the distances between the centroids and adaptation to temporal changes in the distribution of the data. In the first step, the within cluster variance is minimized by updating the representation in a manner similar to the K-means algorithm (MacQueen, 1967). The second step maximizes the distances between the centroids by merging the two centroids with the minimum distance (not considering their weight). The merging is similar to most agglomerative methods (see (Sneath and Sokal, 1973) for a review and (Wong, 1993) for a recent paper). Finally, temporal changes in the distribution of the data are anticipated by treating each new point as an indication to a potential new cluster.

The detailed description of the proposed algorithm for on-line clustering follows below (note, we follow the notation used by (Buhmann and Kuhnel, 1993)). For each centroid α, let $\mathbf{y}_\alpha$ be the location and c_α the counter (the number of points this centroid represents) of the centroid. The scale of the desired solution is specified by the maximum number of centroids available (i.e., size of memory). We denote this parameter as K_{max}. The number of centroids participating in the final solution may be less than K_{max} due to the post processing described below. Thus, the true structure of the data is revealed by the remaining centroids.

1. Initialize the system with zero centroids: $K = 0$.

2. Get data point $\mathbf{x}$.

3. The centroid which is closest to the data point is defined as the winner.

$$winner = \alpha \;\; s.t. \;\; \|\mathbf{y}_\alpha - \mathbf{x}\| \;\; is \;\; minimal$$

4. Update the location of the closest centroid and its counter, i.e. compute the running average.

$$\mathbf{y}_{winner} \leftarrow \mathbf{y}_{winner} + \frac{\mathbf{x} - \mathbf{y}_{winner}}{c_{winner} + 1}$$
$$c_{winner} \leftarrow c_{winner} + 1$$

5. If there remains free memory allocate a new centroid, i.e., if $K < K_{max}$ then $K \leftarrow K + 1$, set $\delta \leftarrow K$. Goto step 8.

6. Find the redundant pair of centroids, i.e., the two centroids whose representation of the data is most similar (closest to each other).

$$\{\gamma, \delta\} = \underset{\gamma,\delta,\gamma\neq\delta}{\operatorname{argmin}} \|\mathbf{y}_\gamma - \mathbf{y}_\delta\|$$

7. Merge the two redundant centroids by computing their weighted average location and cumulative number of points (counter).

$$y_\gamma \leftarrow \frac{y_\gamma c_\gamma + y_\delta c_\delta}{c_\gamma + c_\delta}$$

$$c_\gamma \leftarrow c_\gamma + c_\delta$$

8. Initialize the new centroid with the last data point, it may indicate the start of a new process (the arrival of a new cluster of data).

$$\mathbf{y}_\delta = \mathbf{x}; c_\delta = 0$$

9. While there remains data to be clustered, Goto step 2.

10. Post process: remove all clusters with a negligible weight.

$$\forall\alpha \quad if \quad c_\alpha < \epsilon \quad perform \quad steps \quad 6 \quad and \quad 7 \quad (K_{max} \leftarrow K_{max} - 1)$$

This algorithm can cluster the data in a single pass, with performance (minimization of the global distortion) comparable to existing clustering algorithms run in batch mode. Moreover, the proposed algorithm follows new data while preserving the existing structure, i.e., even small clusters are represented.

The next section presents results of two different sets of simulations. The first set of simulations demonstrates the robustness of the algorithm and quantitatively compares the proposed clustering algorithm to a popular batch algorithm (Deterministic Annealing) and two on-line methods (K-means and EquiDistortion). The results indicate that the new algorithm's performance in minimizing the global distortion (Energy) is comparable to the other methods. This

is true even though the proposed algorithm clusters on-line non-stationary data (K-means fails completely to cluster the non-stationary processes). Furthermore, we introduce a new measure of performance, the local distortion. Results from these experiments demonstrate the superior performance of the new algorithm in minimizing the local distortion, i.e., representing the smaller clusters.

The second set of experiments, is an example of how the proposed algorithm determines the solution given an indication of the desired scale.

3 Results of simulations

3.1 Quantitative analysis. Random generated clusters

To quantitatively analyze the performance of the proposed algorithm a series of randomly generated Gaussian mixtures were generated. Four different methods were compared: K-means (MacQueen, 1967) (an on-line method), EquiDistortion (Ueda and Nakano, 1994) (modified to be on-line) Deterministic Annealing (Rose et al., 1990) (a batch method) and the proposed algorithm (AddC).

The K-means and Deterministic Annealing method were chosen to represent baseline performance of an on-line and batch method. These algorithms determine their representation of the data by moving their K centroids, no merging or splitting is performed. The EquiDistortion method merges and splits centroids as a function of their relative variance, i.e., centroids with a relatively large variance are split and those with a relatively small variance are merged. The EquiDistortion method was modified to run in an on-line mode (Guedalia et al., 1995).

The data was presented to the on-line algorithms either in a stationary process or a non-stationary fashion. The non-stationary process has the following feature: on a short time scale it is random, while on the long time scale the process has a sequential property. For example, the data in Figure 10 has 9 small clusters which were produced sequentially. The points in each cluster were generated in a stationary process. Thus, all the points from the first cluster arrived randomly followed by the points in the second cluster etc. The number of centroids was equal to number of clusters.

Deterministic Annealing ran with $\beta = 1$ through $\beta = 11357.8$ incremented by 10%. At each β step the system ran until convergence (maximum 30 Epochs). Experimentally, it was noted that at most β steps convergence occurred relatively early. It is worth noting that $\beta = 11357.8$ was not large enough to be considered infinity (we stopped at this value due to lack of computing resources).

The number of Gaussian mixtures generated was systematically varied from five through twenty four. Ten sets of data were generated for each of the different cases. The data was divided into a training set and test (generalization) set. All results were averaged over ten runs. Each of the Gaussian mixtures had a randomly generated number of points and shape. After the training data was clustered by the different methods the global and local distortion was

measured on the test set. The global distortion was calculated as follows:

$$\frac{1}{S}\sum_{i=1}^{S}\min_{\alpha}\|\mathbf{y}_\alpha - \mathbf{x}_i\|$$

Where S is the size of the data set and the "distance" is computed as the sum of squares.

3.1.1 Global distortion

Figures 4 and 5 presents the global distortion as a function of the number of Gaussian mixtures generated. The Deterministic Annealing energy would probably approach the K-means given more time ($\beta = \infty$). An example of the results can be seen in Figure 3.

The proposed method succeeds in approaching batch results – in minimization of the global distortion – even though it clustered the data in a single sequential pass. Moreover, it better preserved the representation of the data by allocating centroids for the small distant clusters.

Figure 8 and 9 present the global distortion as a function of the dimension of the data with non-stationary and stationary data respectively.

In this situation as well the proposed method succeeds in approaching batch results – in minimization of the global distortion.

3.1.2 Local distortion

While, the global distortion provides a measure of the average performance, it is not a good measure of the quality of the representation of each individual cluster. Hence, the local distortion is determined as follows:

$$\sum_{n=1}^{N}\frac{1}{S_n}\sum_{x\in C_n}^{S_n}\min_{\alpha}\|\mathbf{y}_\alpha - \mathbf{x}_i\|$$

Where N is the number of clusters generated, C_n is the n'th cluster, S_n the number of points in C_n and the distance $\|\mathbf{y}_\alpha - \mathbf{x}_i\|$ is the sum of squares. The distortion of each point is the distance between the point and its most representative centroid, normalized by the size of its originating cluster. This ensures that the affect each cluster has on the performance measure is relatively equal, even small clusters influence the final result.

Figures 6 and 7 graphs the local distortion (averaged over ten runs) as a function of the number of clusters. The K-means and Deterministic Annealing methods which minimize the global distortion ($\beta = \infty$), perform relatively poorly. This is because they ignore small clusters even if they are quite distinct. As the number of clusters increase the affect of missing a single cluster is diminished. By preserving the small distant clusters, the proposed method, also minimizes the local distortion.

3.1.3 Stationarity

The on-line methods were tested on data which was presented once in a pseudo-stationary (random) mode and once in a non-stationary (sequential) mode. While, the K-means method

successfully clustered the stationary data, it failed to capture the structure of the non-stationary data. The reason for its poor performance is demonstrated in Figure 3. The K-means method follows the arrival of the latest set of data. Hence, most of the centroids are located within the central cluster. This is in contrast to the performance of the proposed method in clustering both the stationary and non-stationary data.

3.2 Scale dependence

To demonstrate the algorithms ability to follow the structure as a function of scale the data from Figure 1 was clustered with the new algorithm. Figure 10 depicts the clustering of the data while constraining the memory to four centroids. Four stages in the process are presented, after the presentation of the first 1000, 3000, 6000 and 10000 data points. In the first stage all the centroids are placed on the existing data. Next, the centroids represent the 3 clusters that exist in the bottom right corner. The introduction of data at a relatively large distance from the previous data, modifies the perspective. Hence, the previously subdivided clusters are merged into a single large cluster. The final representation of the data with 4 centroids utilizes three of the centroids, placing them in the center of mass of each group of data. The fourth centroid represents the last data point and should be merged into the system.

In comparison, Figure 11 presents the results when using 10 centroids. Similar to the previous example, the first stage places all the centroids on the existing data. After 3000 data points arrive, the local structure is revealed, the data is properly represented by three centroids with the other 7 appearing as satellites around the extremities. These centroids are allocated in the following stages. In the final stage (after the arrival of all 10,000 points) the local structure is preserved due to the relatively large number of centroids. Here again the extra centroid is needed to follow the last data point to arrive. Note that the non-stationarity in the final example is not a necessary condition for the final solution.

Perhaps the most important aspect of the algorithm is its relative insensitivity to the exact choice of K_{max}. In other words one should only specify the order of magnitude of K_{max}. This is demonstrated in Figure 12. A single Gaussian centroid (stationary) was clustered with K_{max} equal 2 through 16. After the clustering process all centroids which represented less then 0.5% of the number of points were merged. Figure 13 graphs the Energy (global distortion) as a function K_{max}. The affect of increasing K_{max} is negligible until a "phase transition" occurs and a split.

The reasoning behind this is as follows: assume a single Gaussian cluster of data which arrives in a stationary process. Let us assume K_{max} is equal to 3. Assume that it has been correctly clustered and we will label the centroids μ, ν and ξ, where ξ is the actual center (mean). When a new data point arrives it forces the merging of the two closest centroids. In order for the centroids in the periphery to accumulate points they must merge with each other. However, since the probability that the distance between μ and ν is smaller than the distance between ξ and either μ or ν is small it is more likely that they will merge with the ξ. Hence, strengthening the center and weakening the periphery. For the centroids on the periphery to have a large mass

they must be closer to each other than to the center (an unlikely event) and this must occur for many time steps consecutively (a very unlikely event). Figure 16 presents a measure of order which quantifies this process.

This process of "phase transitions" is similar to the one described by Rose et. al (Rose et al., 1990). Figures 14 and 15 present the results of clustering the same data using the Deterministic Annealing algorithm. Note the similarity of the behavior of the Energy function in graphs 13 and 15.

4 Summary and Conclusions

Yet another clustering classifier (yacc)? The proposed algorithm is the first to explicitly address the issue of on-line clustering non-stationary data.

The method can be seen as an extension of the work presented by Buhmann and Kuhnel (Buhmann and Kuhnel, 1993) or an on-line version of the clustering by melting algorithm presented by Wong (Wong, 1993) in which each data point is assigned a centroid.

Quantitative analysis of the new algorithm performance in clustering simulated data, demonstrated its superior performance in minimizing the local distortion, and comparable performance in minimizing the global distortion to existing clustering algorithms. This is even more pronounced when clustering non-stationary data.

Unfortunately, the new algorithm is sensitive to data which includes drastically different scales. For example, if the data seen in Figure 1 is corrupted with noise (a very wide Gaussian placed in the center of the data) the performance drops (see Figure 17). The proposed method attaches equal importance to every point. Each new point is potentially a beginning of a new cluster. The solution to this is to assume knowledge of the time scale of the smallest process and further assume that the smallest process is larger than a certain threshold. Then after each time step merge all centroids whose counter is below the threshold.

Currently the algorithm is being tested on the difficult problem of quality control of agricultural produce. Preliminary results indicate that the algorithm shows significantly better results than other on-line clustering algorithms.

5 Acknowledgments

The authors would like to thank Prof. Haim Sompolinksy and Dr. Yael Edan for their help in preparation of this manuscript.

This research was supported by BARD, Binational Agricultural and Research Development Fund No. US-1992-91 and partially supported by the Paul Ivanier Center for Robotics Research and Production Management.

Please address correspondance to either idavidg@cs.huji.ac.il or mikilon@lobster.ls.huji.ac.il.
A demo program of the AddC algorithm is available at:

`ftp://lobster.ls.huji.ac.il/pub/mikilon/Cluster/addcdemo.zip`

References

Ball, G. and Hall, D. (1967). A clustering technique for summarizing multivariate data. *Behavioral Science*, 12:153–155.

Buhmann, J. and Kuhnel, H. (1993). Complexity optimized data clustering by competitive neural networks. *Neural Computation*, 5:75–88.

Carpenter, G. A. and Grossberg, S. (1990). Adaptive resonance theory: Neural network architectures for self-organizing pattern recognition. In Eckmiller, R., Hartmann, G., and Hauske, G., editors, *Parallel Processing in Neural Systems and Computers*, pages 383–389. North-Holland, Amsterdam.

Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. Wiley, New York.

Fritzke, B. (1994). Growing cell structures - a self-organizing network for unsupervised and supervised learning. *Neural Networks*, 7:1441–1460.

Gath, I. and Geva, A. B. (1989). Unsupervised optimal fuzzy clustering. *IEEE trans. on Pattern Anlysis and Machine Intelligence*, 11:773–781.

Guedalia, I. D., Werman, M., and Edan, Y. (1995). A new method for on-line clustering of sparse data. *ASAE Paper No. 95-3606*.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs New Jersey 07632.

Linde, Y., Buzo, A., and Gray, R. M. (1980). An algorithm for vector quantizer design. *IEEE Trans. on Communications*, 28(1):84–95.

MacQueen, J. (1967). Some methods for classification and analysis of multi-variate observations. *Proc. 5th Brekeley Symp. Mathematical Statist. and Probability*, pages 281–297.

Rissanen, J. (1989). *Stochastic Complexity in Statistical Inquiry*. World Scientific.

Rose, K., Gurewitz, E., and Fox, G. (1990). A deterministic annealing appraoch to clustering. *Patt. Rec. Letters*, 11(4):589–594.

Sebestyen, G. S. (1962). Pattern recognition by an adaptive process of sample set construction. *IRE Trans. on Info. Theory*, 8:S82–S91.

Sneath, P. H. A. and Sokal, R. R. (1973). *Numerical Taxonomy*. W. H. Freeman and Company.

Ueda, N. and Nakano, R. (1994). A new competitive learning approach based on an equidistortion princple for designing optimal vector quantizers. *Neural Networks*, 7(8):1211–1227.

Wong, Y. (1993). Clustering data by melting. *Neural Computation*, 5:89–104.

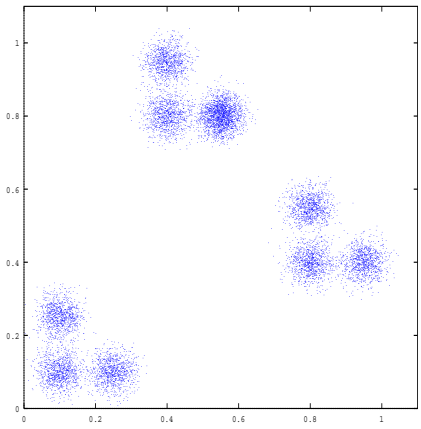

Figure 1: Example of scale dependent intrinsic structure. The data in this figures is composed of 9 gaussian clusters. Each cluster contains 1000 points, except the top right cluster which contains 2000 points. Note, how the data can be grouped either into 3 or 9 clusters.

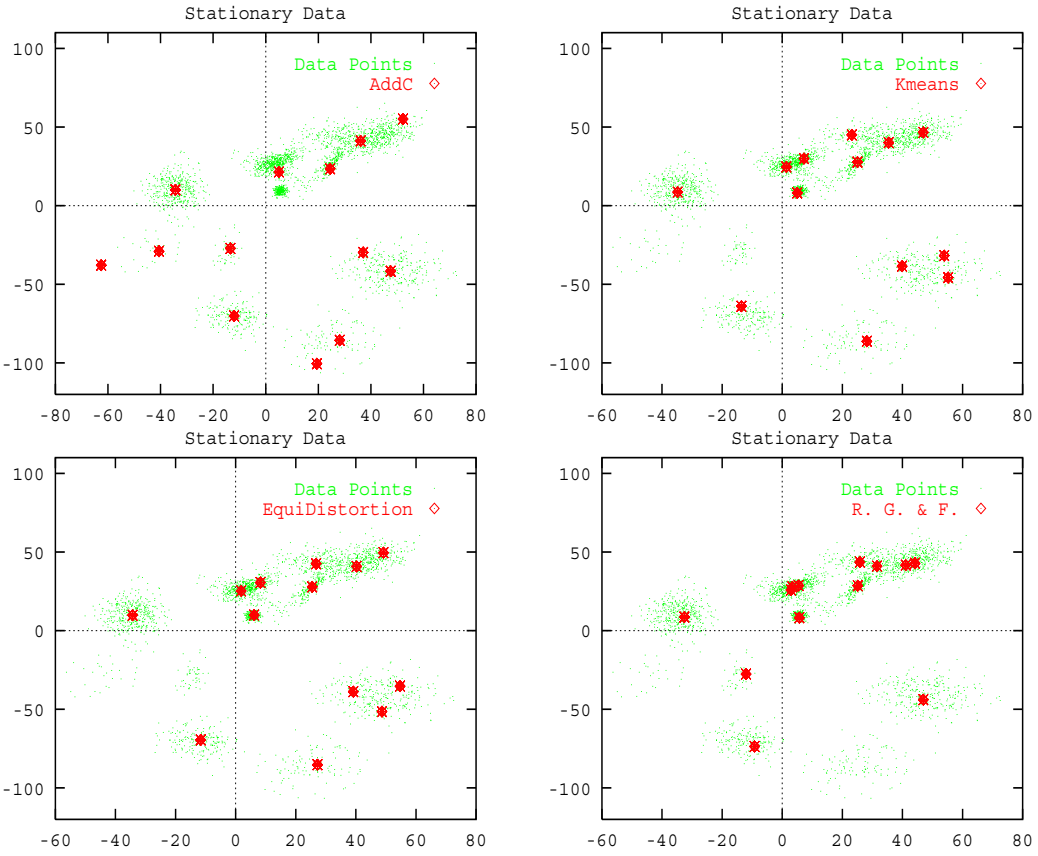

Figure 2: Clustering of randomly generated stationary data by four different methods, Proposed method [Add constantly], K-means, EquiDistortion and Deterministic Annealing. Note, how the K-means and EquiDistortion methods missed two small clusters in the center left section (the Deterministic Annealing missed one). This contributes to the relatively high local distortion of these methods as compared with the proposed method.

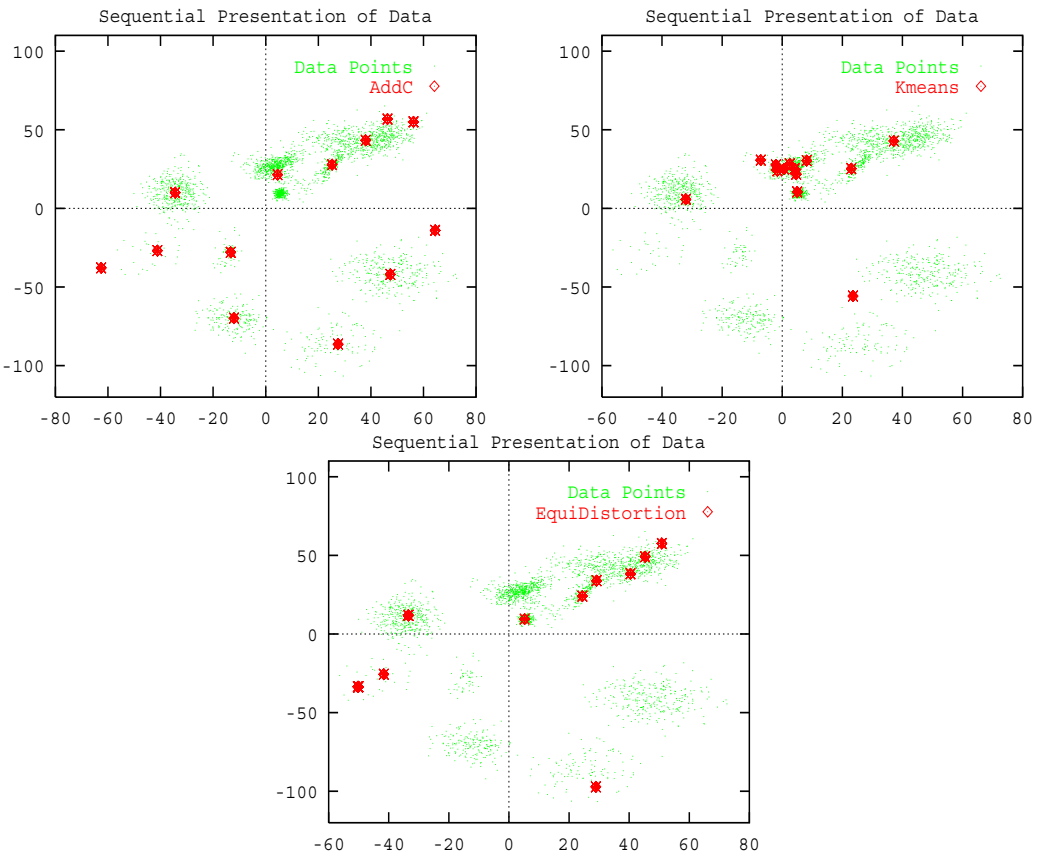

Figure 3: Clustering of randomly generated non-stationary data by three on-line methods, Proposed method [Add constantly], K-means, EquiDistortion. Note, how the proposed method successfully clusters the data even though it is presented in a sequential fashion. Furthermore, the solution found by the proposed method here is virtually identical with the solution obtained when the data is processed randomly (see Figure 2).

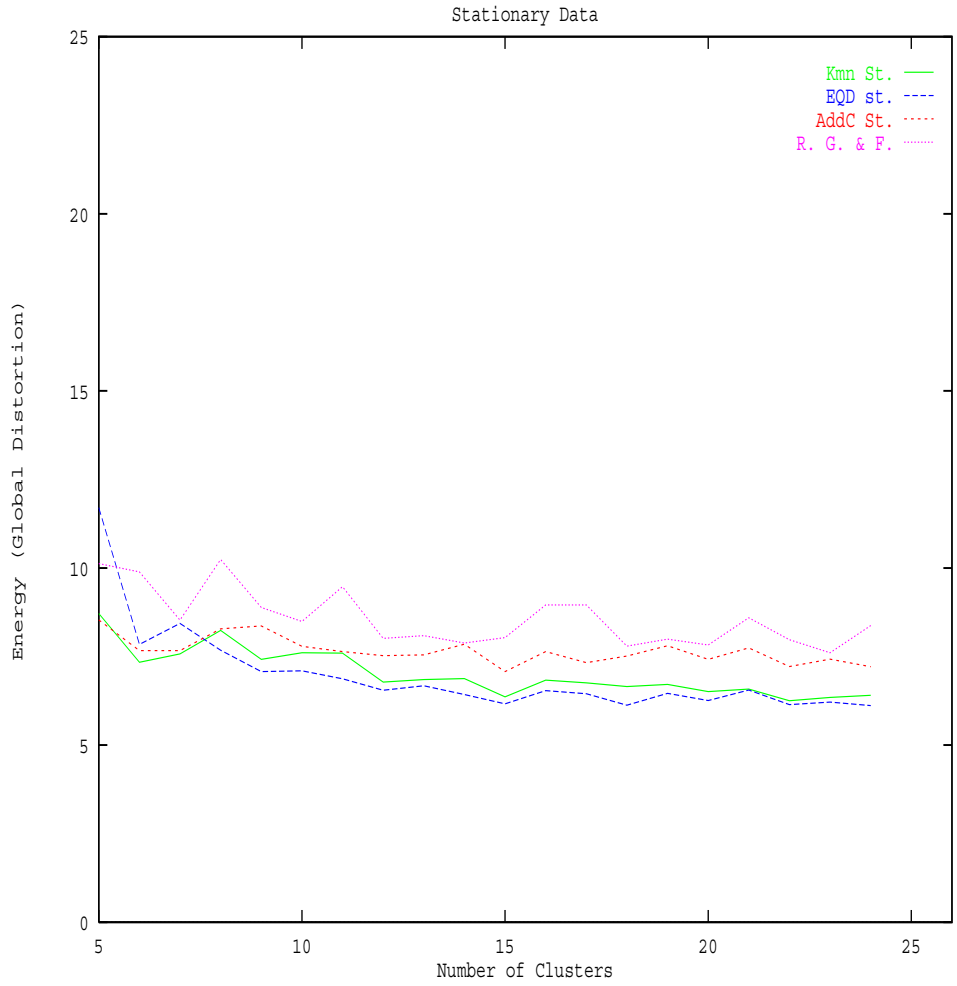

Figure 4: A comparison of the performance of the different stationary methodologies tested. Plot of the energy of the system (averaged over ten runs) as a function of the different data sets, i.e., different numbers of clusters. The solutions found for one instance of sixteen clusters are depicted in Figure 2. The Rose, Gurewitz & Fox method would have reached a lower energy throughout had the process not been stopped early.

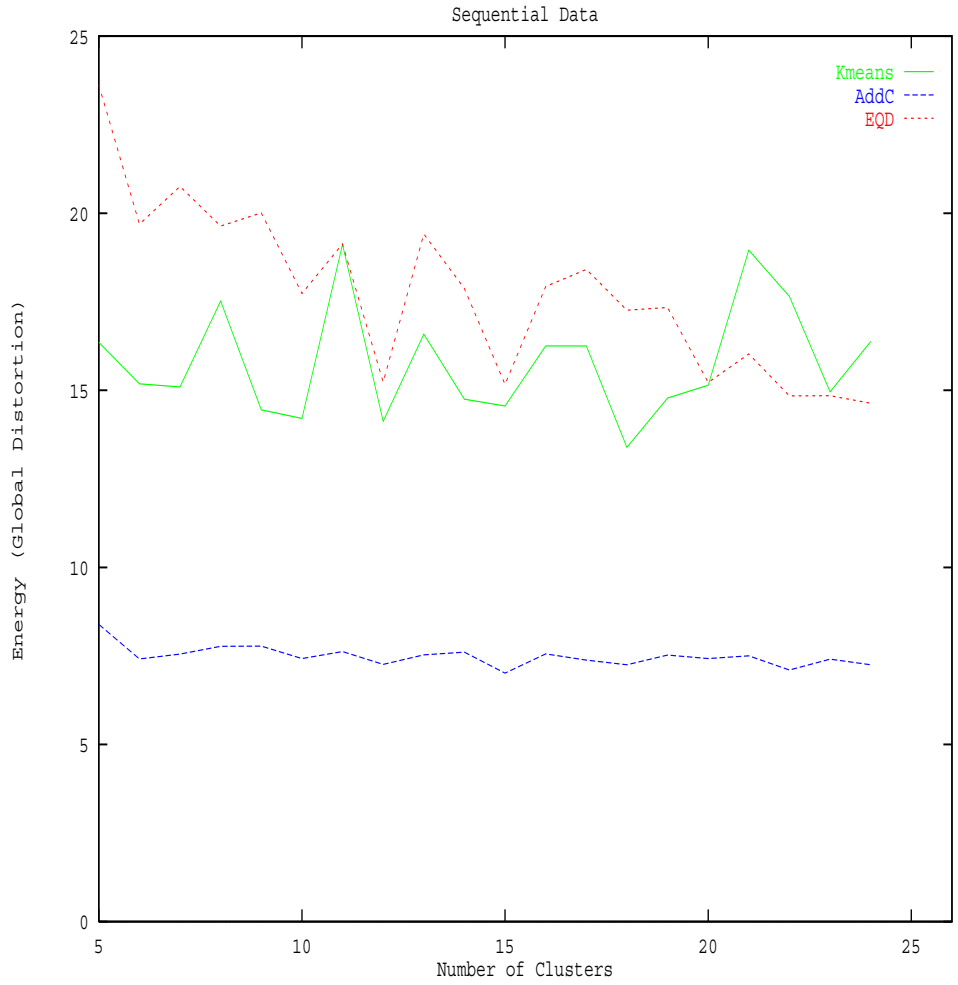

Figure 5: A comparison of the performance of the different sequential methodologies tested. Plot of the energy of the system (averaged over ten runs) as a function of the different data sets, i.e., different numbers of clusters. The solutions found for one instance of sixteen clusters are depicted in Figure 3.

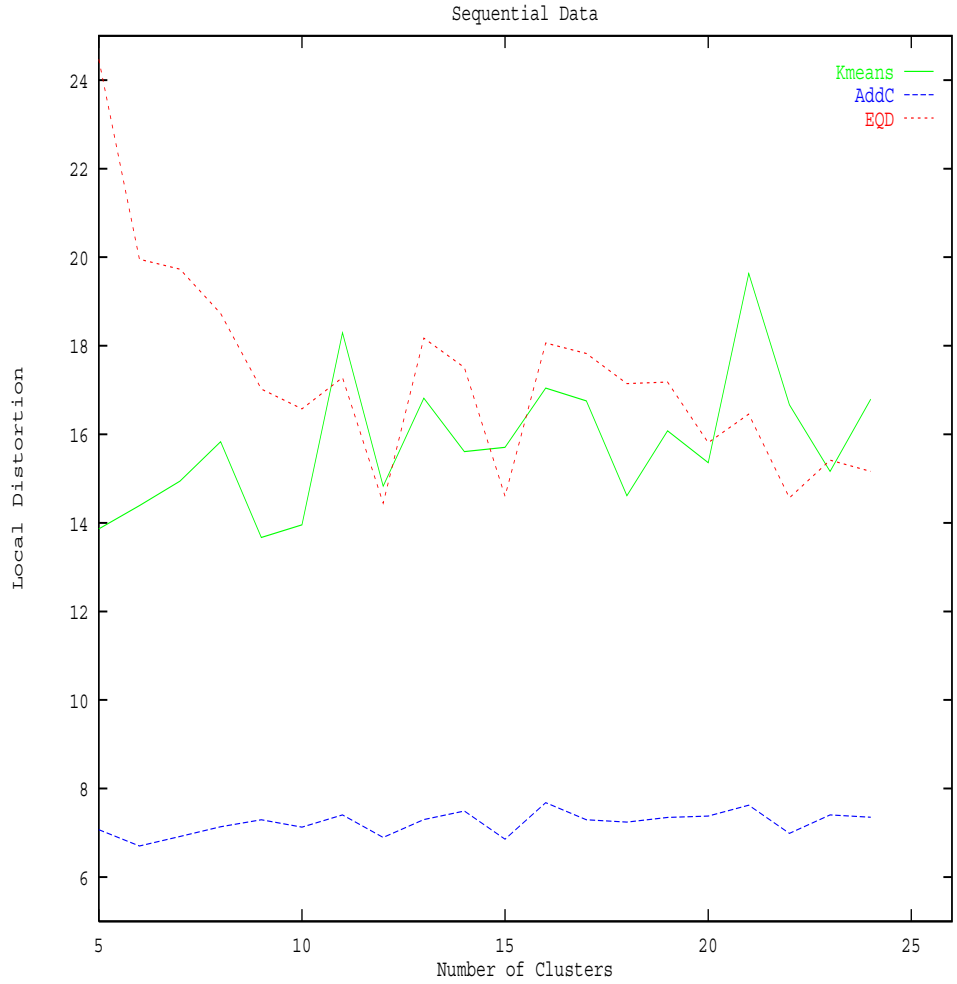

Figure 6: A comparison of the performance of the different sequential methodologies tested with respect to small clusters. Plot of the **local distortion** of the system (averaged over ten runs) as a function of the different data sets, i.e., different numbers of clusters. The solutions found for one instance of sixteen clusters are depicted in Figure 3. The proposed algorithms succeeds in preserving the representation of even the small clusters. This is due to their relatively large distance from other clusters.

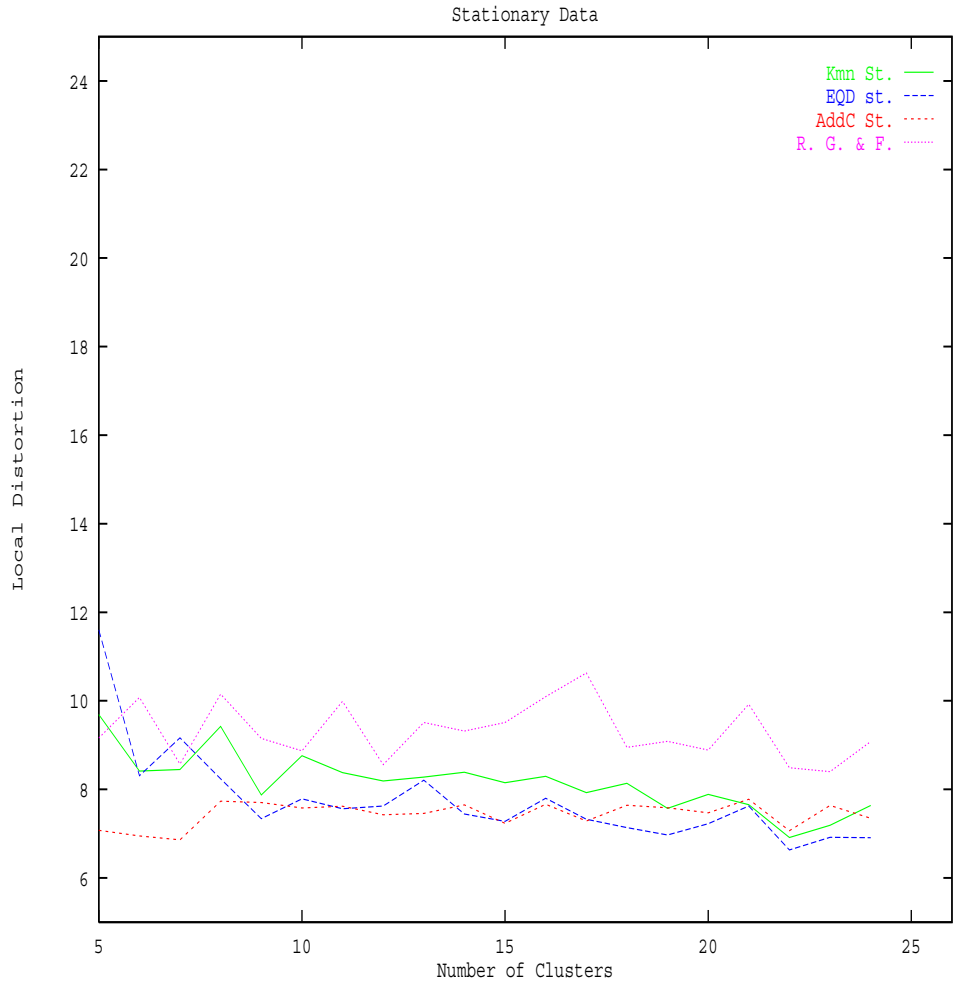

Figure 7: A comparison of the performance of the different stationary methodologies tested. Plot of the local distortion of the system (averaged over ten runs) as a function of the different data sets, i.e., different numbers of clusters. The solutions found for one instance of sixteen clusters are depicted in Figure 2. The proposed algorithms succeeds in preserving the representation of even the small clusters. This is due to their relatively large distance from other clusters.

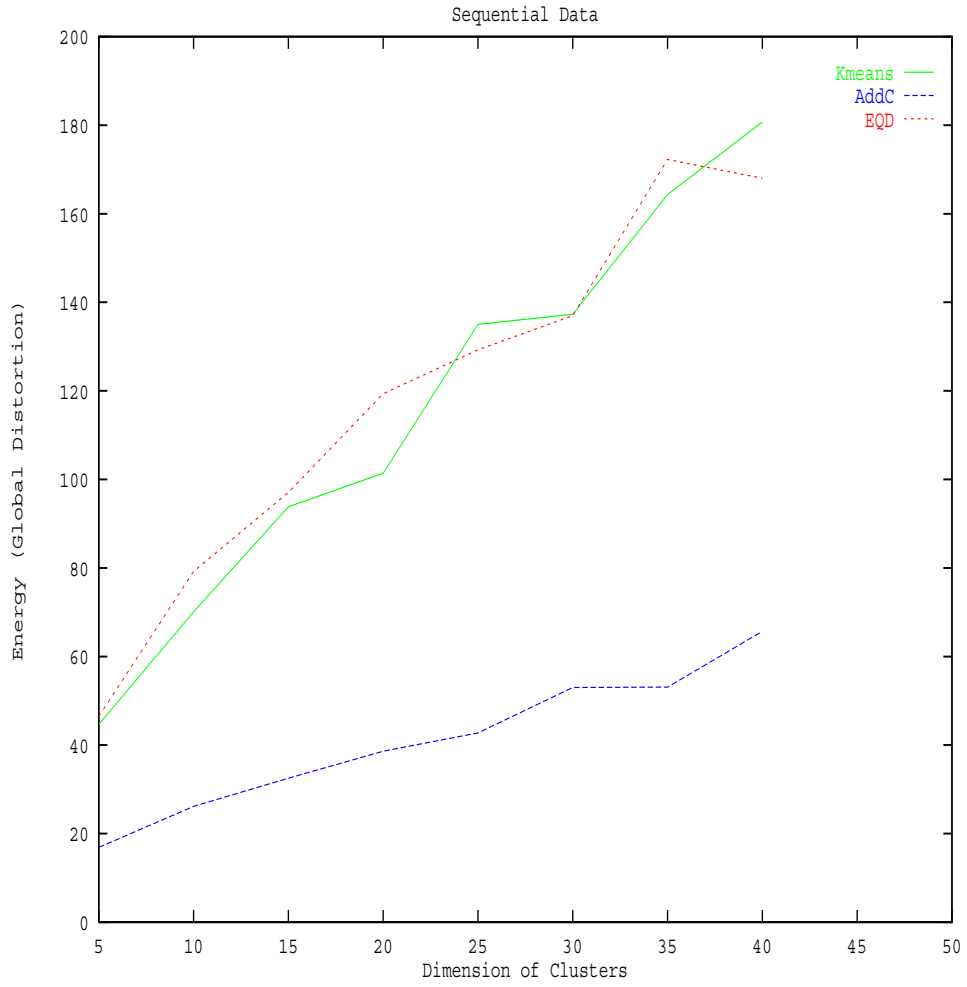

Figure 8: A comparison of the performance of the different non-stationary methodologies tested. Plot of the global distortion as a function of the dimensionality of the data. Ten gaussian clusters were generated with dimensions 5 through 40 at increments of 5. The number of points in each cluster was fixed.

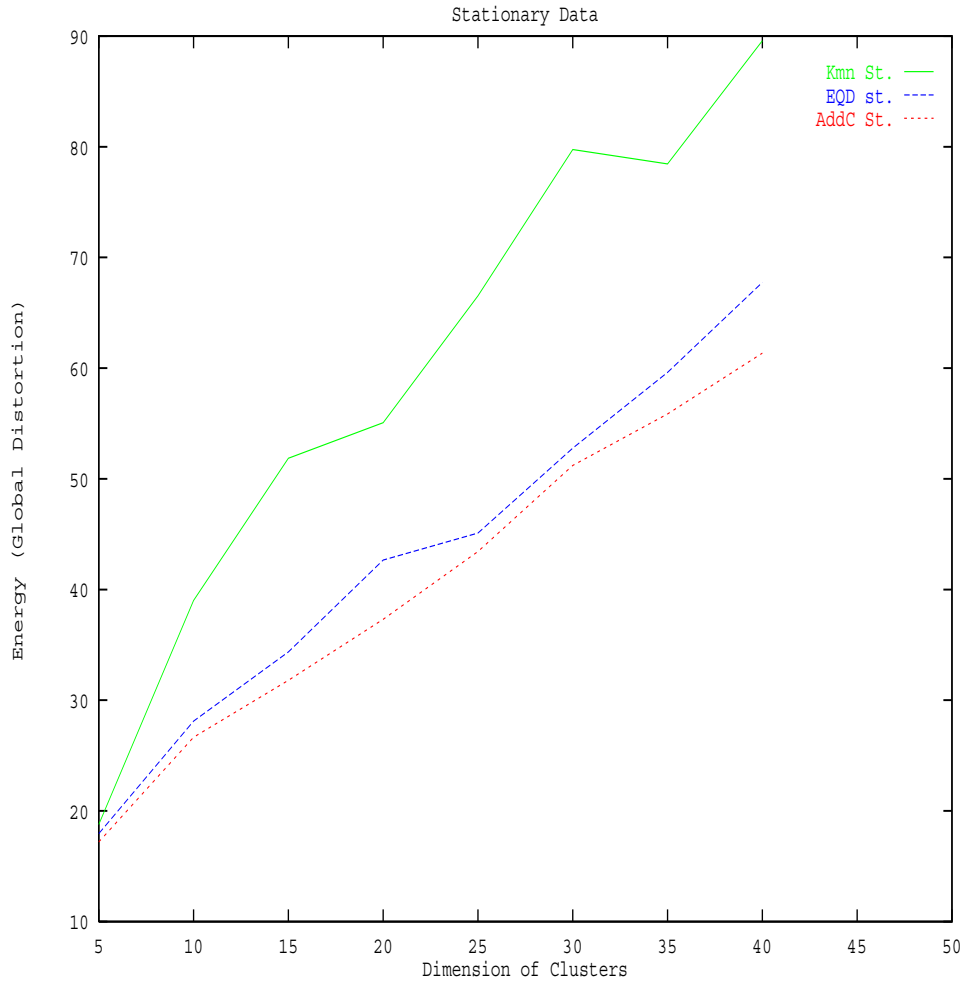

Figure 9: A comparison of the performance of the different stationary methodologies tested. Plot of the Energy (Global Distortion) of the system (averaged over ten runs) as a function of the dimensionality of the data. Ten gaussian clusters were generated with dimensions 5 through 40 at increments of 5. The number of points in each cluster was fixed.

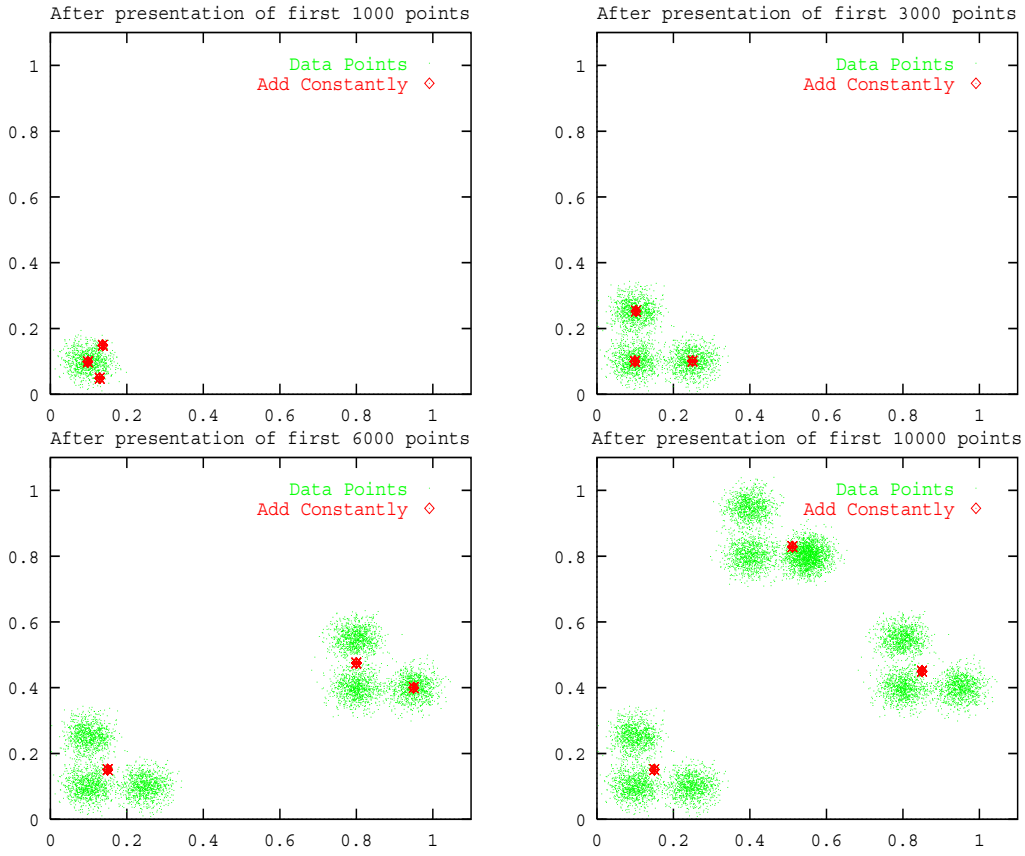

Figure 10: Sequential presentation of data from Figure 1. Four stages of the clustering by the proposed algorithm are presented. $K_{max} = 4$.

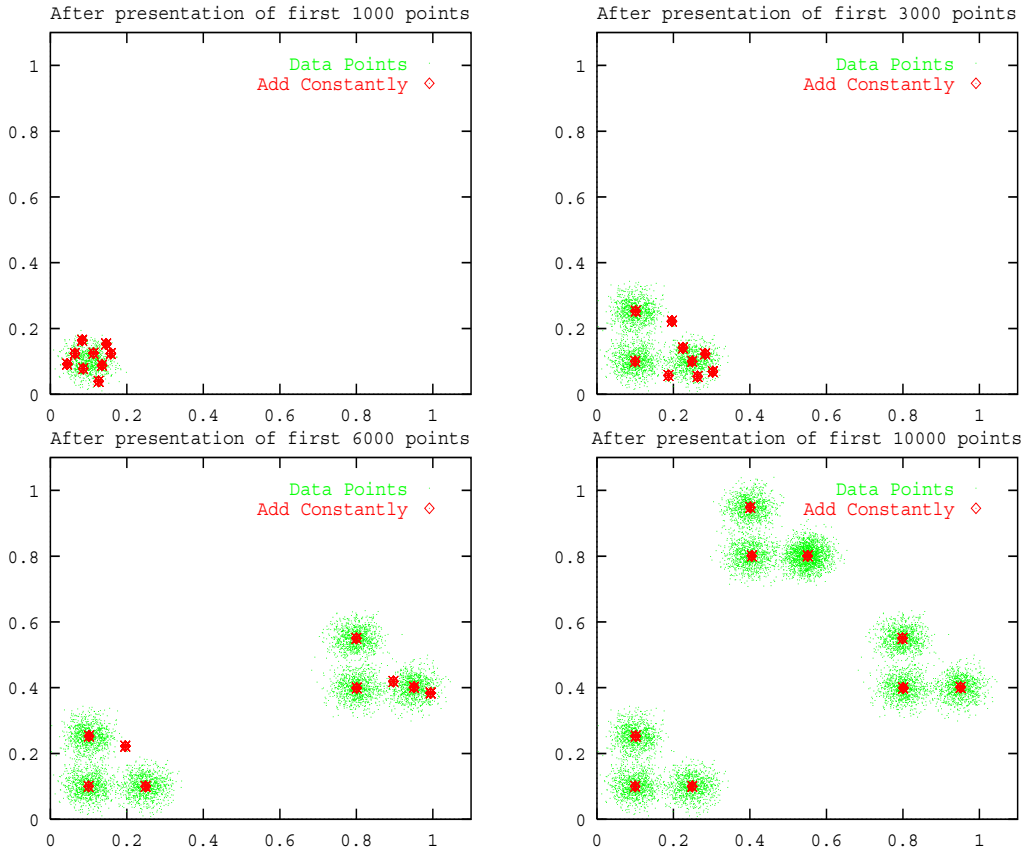

Figure 11: Sequential presentation of data from Figure 1. Four stages of the clustering by the proposed algorithm are presented. $K_{max} = 10$.

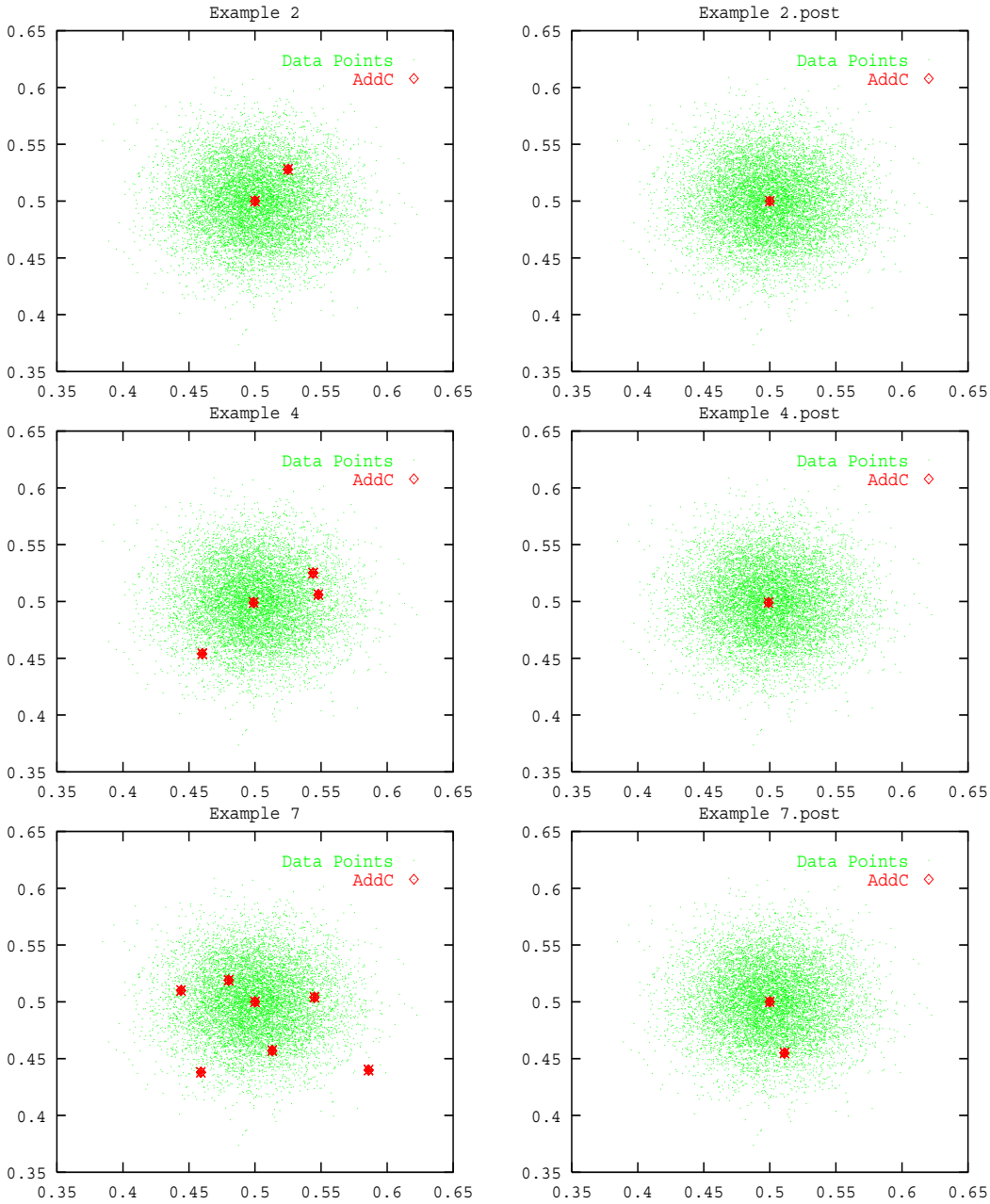

Figure 12: An example of the lack of sensitivity of the proposed algorithm to the choice of K_{max}. A single Gaussian centroid (stationary) was clustered with K_{max} equal 2 through 16. After the clustering process all centroids which represented less then 0.5% of the number of points were merged.

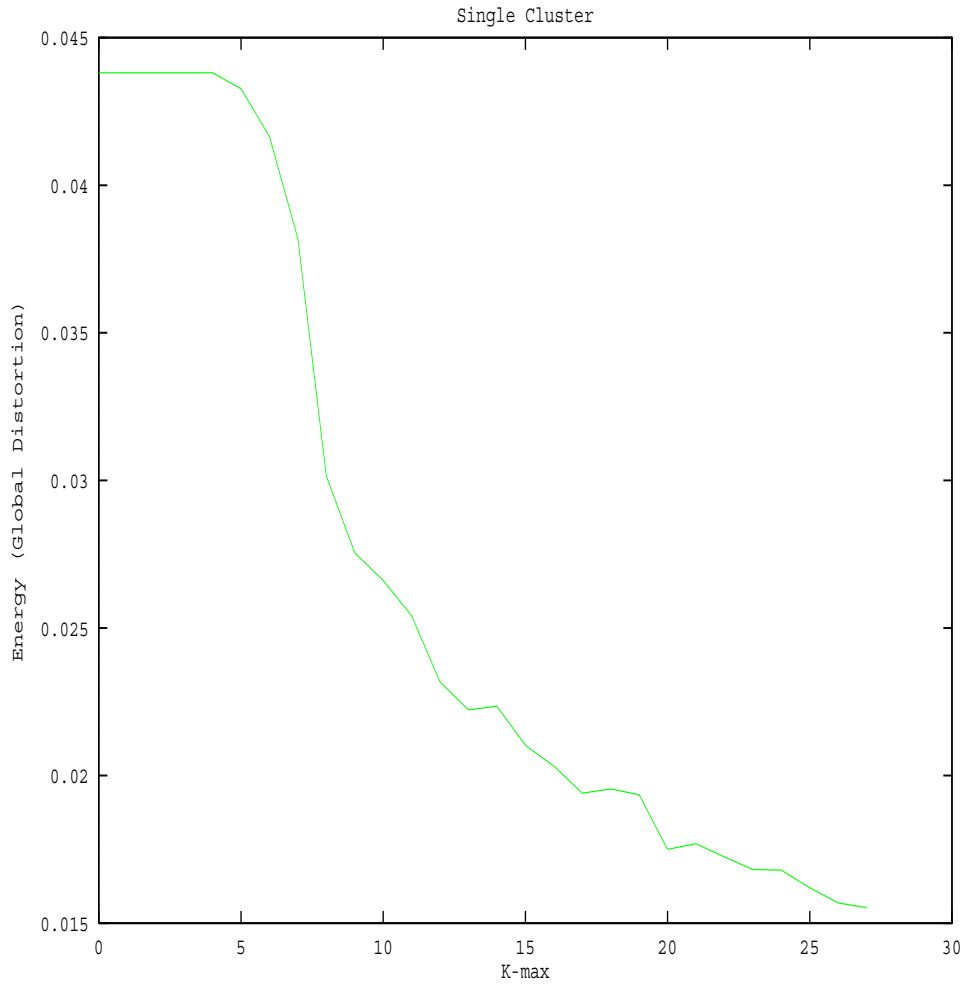

Figure 13: Plot of the Energy (Global Distortion) as a function of the addition of K_{max}. A single gaussian cluster was clustered with the proposed method at different K_{max}. Next, all centroids which represented less then 0.5% of the total number of points where merged. This was averaged over ten runs. The Energy (Global Distortion) function demonstrates that there is a clear plateau in which there is no change in the solutions found. This is in contrast to methods which minimize the Energy and would utilize all the centroids available.

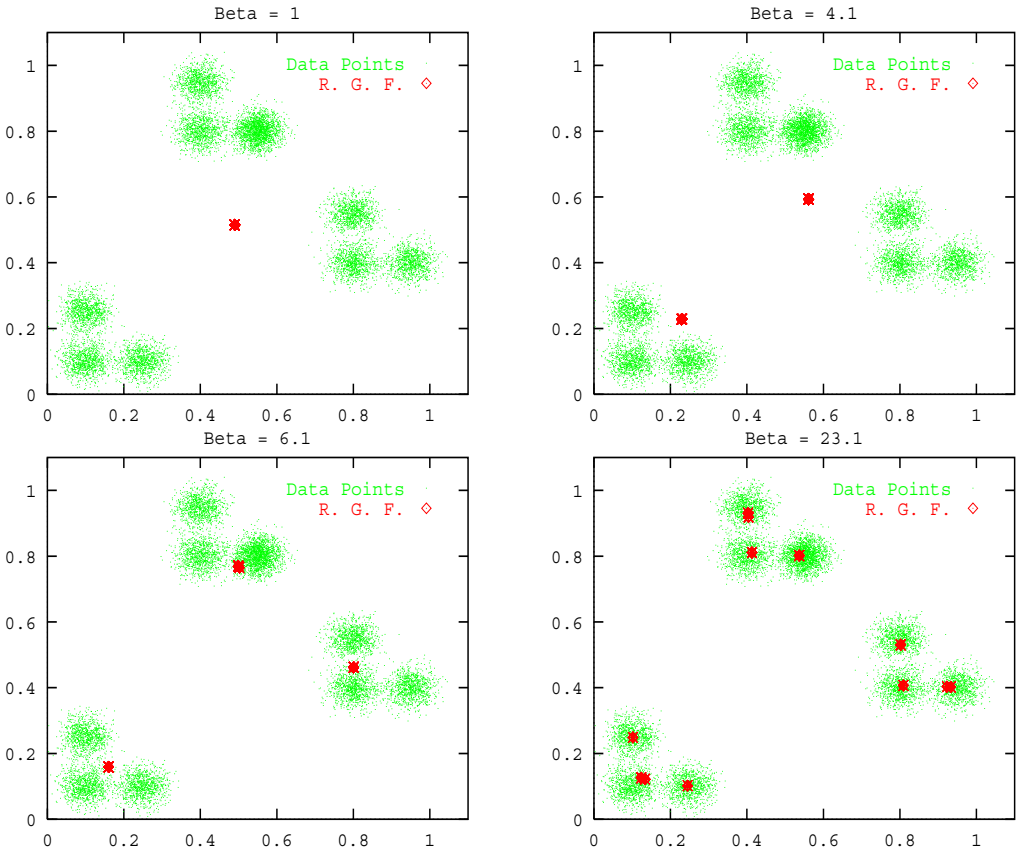

Figure 14: Clustering of data from Figure 1 by Deterministic Annealing. Four stages of the clustering at different β are presented.

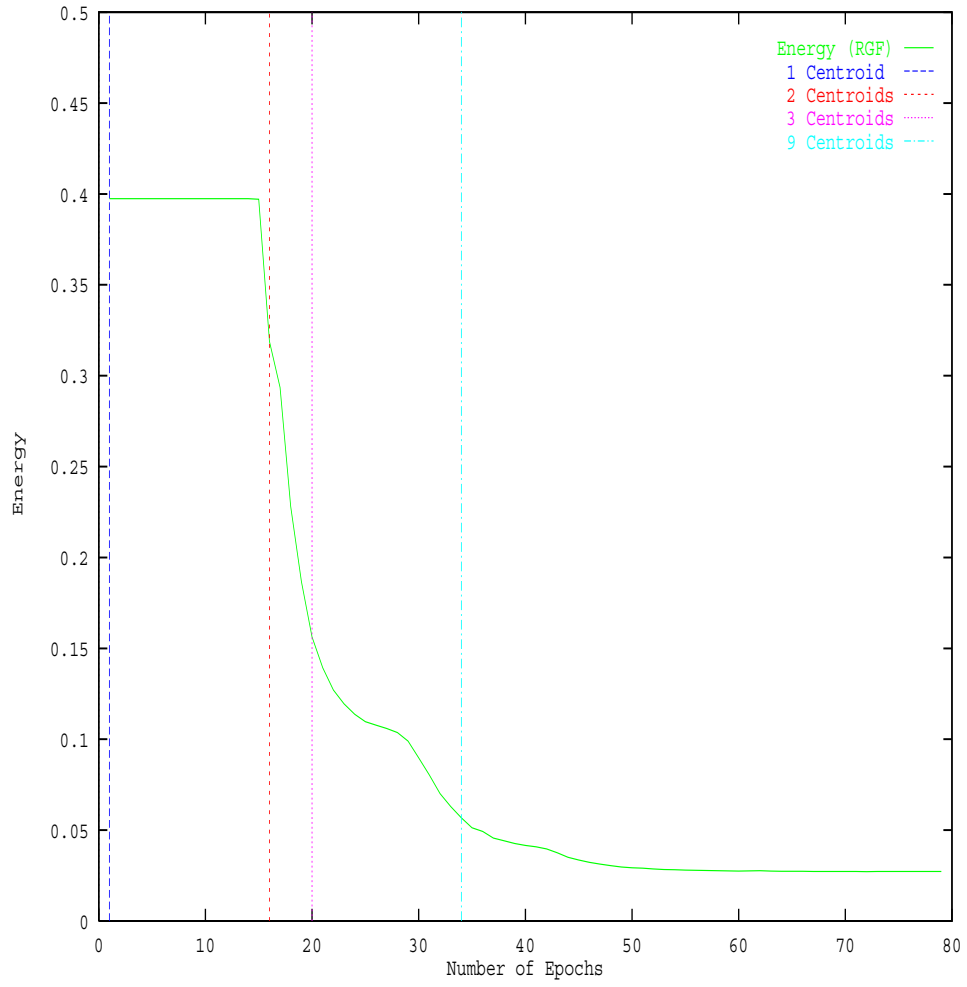

Figure 15: Energy as a function of time (decreasing temperature) during the Deterministic Annealing clustering of the data from Figure 1. The four stages of the clustering depicted in Figure 14 are noted by horizontal lines.

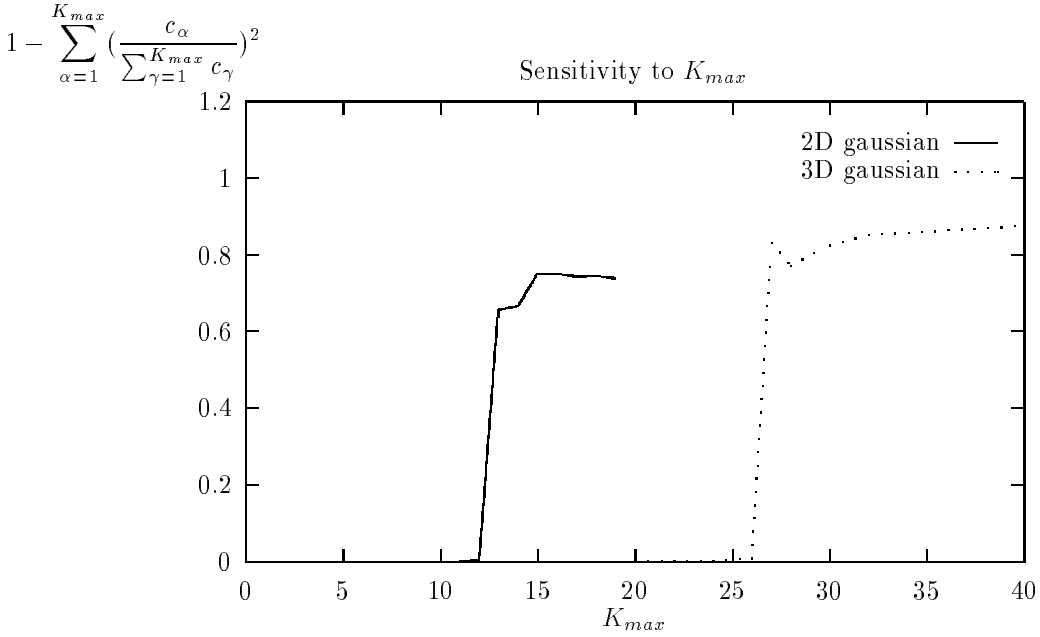

Figure 16: Plot of order parameter as a function of K_{max}. A single 2D or 3D gaussian with a million data points was presented randomly and clustered with the proposed method at different K_{max}'s. The order parameter shows clear phase transitions indicating the methods robustness to K_{max} The phase transition indicates a sudden change in the number of non-redundant centroids. Furthermore, as the dimensionality increases this occurs at a larger K_{max}.

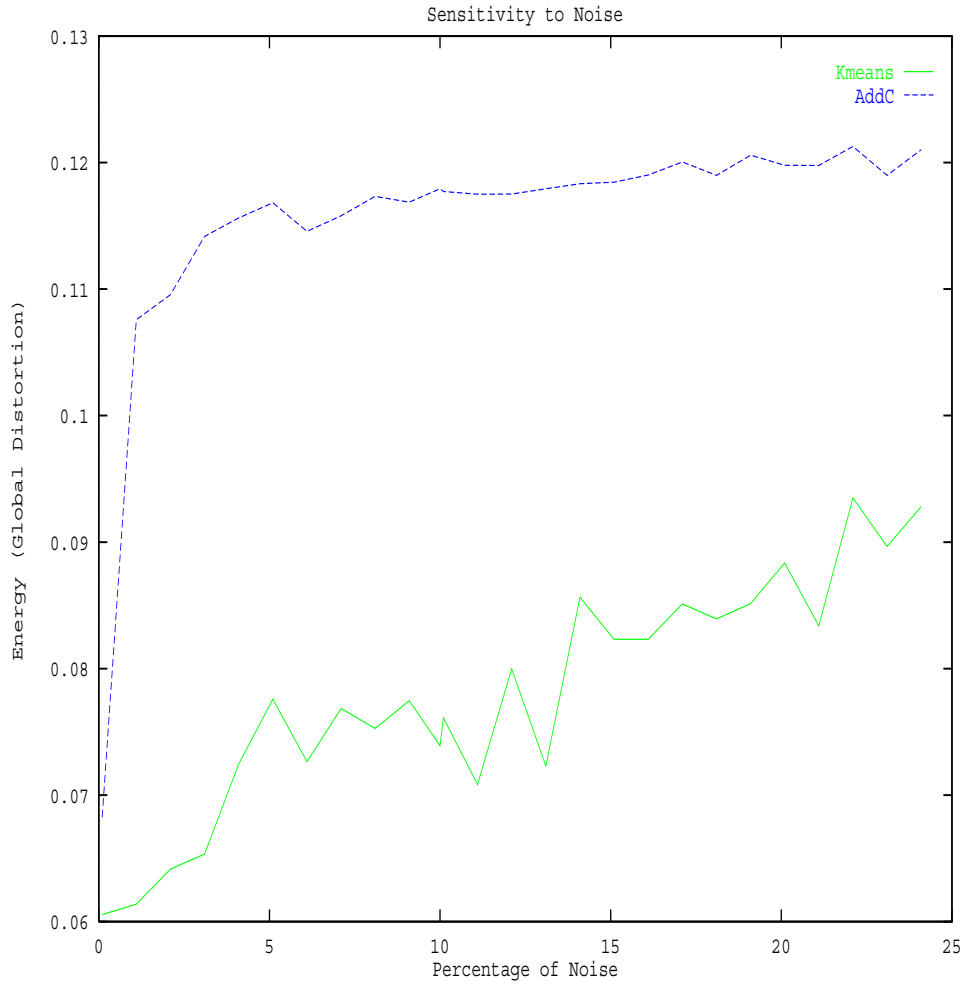

Figure 17: Plot of the Energy (Global Distortion) as a function of the addition of noise. The data from Figure 1 with the addition of noise was clustered with either Kmeans or AddC. Note, how the AddC method immediately reacts to the addition of noise, while the Kmeans method slowly degrades.

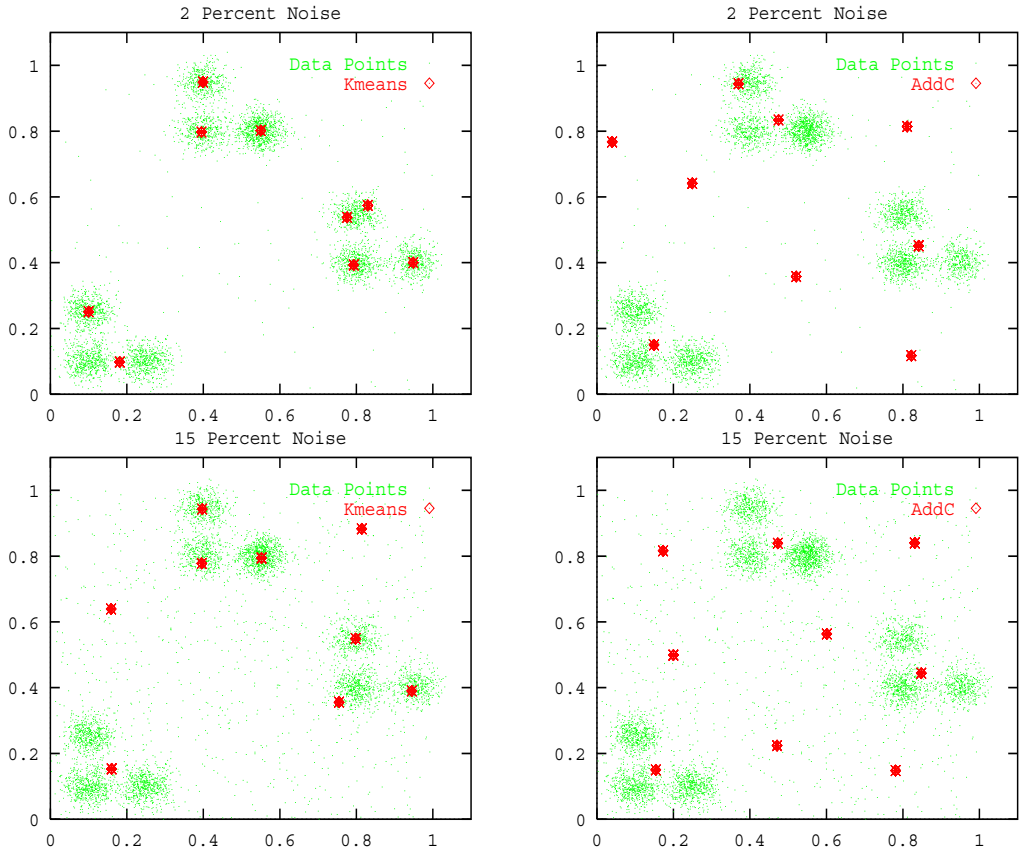

Figure 18: Performance of Kmeans and AddC in clustering data from Figure 1 with the addition of noise. Note, how the AddC method immediately reacts to the addition of noise, while the Kmeans method slowly degrades.