

Research Article

Neural Gas Clustering Adapted for Given Size of Clusters

Iveta Dirgová Luptáková, Marek Šimon, Ladislav Huraj, and Jiří Pospíchal

Department of Applied Informatics and Mathematics, University of SS. Cyril and Methodius, J. Herdu 2, 917 01 Trnava, Slovakia

Correspondence should be addressed to Iveta Dirgová Luptáková; iveta.dirgova@ucm.sk

Received 19 April 2016; Revised 8 September 2016; Accepted 26 October 2016

Academic Editor: Dan Simon

Copyright © 2016 Iveta Dirgová Luptáková et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Clustering algorithms belong to major topics in big data analysis. Their main goal is to separate an unlabelled dataset into several subsets, with each subset ideally characterized by some unique characteristic of its data structure. Common clustering approaches cannot impose constraints on sizes of clusters. However, in many applications, sizes of clusters are bounded or known in advance. One of the more recent robust clustering algorithms is called neural gas which is popular, for example, for data compression and vector quantization used in speech recognition and signal processing. In this paper, we have introduced an adapted neural gas algorithm able to accommodate requirements for the size of clusters. The convergence of algorithm towards an optimum is tested on simple illustrative examples. The proposed algorithm provides better statistical results than its direct counterpart, balanced k -means algorithm, and, moreover, unlike the balanced k -means, the quality of results of our proposed algorithm can be straightforwardly controlled by user defined parameters.

1. Introduction

Data amount in various disciplines, ranging from bioinformatics to web documents, increases nonlinearly each year. However, to exploit these data and to extract knowledge from them, their effective processing is necessary. Big data analysis contains cluster analysis together with clustering algorithms as its major topic. The goal of unsupervised clustering as a data mining task is to separate an unlabelled dataset of “observations” into several sets, where each separate set is ideally characterized by its unique hidden data structure. Since a definition of the principle underlying such a data structure is subjective, there does not exist the best clustering algorithm or the best definition of a cluster. Among major approaches to clustering belong hierarchical, partitional, neural network-based or kernel-based clustering [1].

In many applications, sizes of clusters are bounded or known in advance. Examples can be viewed in student study size segmentation [2] or could be used in testing for division into test groups [3] (e.g., searching in parts of a network for intrusions [4]), in customer segmentation for sales groups in marketing, with given/bounded capacities of each team, in job scheduling problem, where machines have given capacities, in document clustering constrained by storage spaces

[5–7], or in divide and conquer methods, where the divide part is controlled by clustering [8]. However, traditional clustering techniques cannot impose such constraints on the sizes of clusters. Nevertheless, a few attempts occurred recently to modify classical clustering algorithms like k -means to accommodate such requirements as equal cluster size [5, 8, 9], using also application of linear programming optimization techniques like in [7].

The k -means algorithm, whose core was suggested already about 60 years ago [10], separates observations into k clusters; each observation belongs to the cluster with the nearest mean. The problem is NP-hard, but there are available fast heuristic algorithms for its solution. These converge to local optima which sometimes can produce counterintuitive results. A standard algorithm starts with a set of centroids and then repeatedly assigns each of the input “observations” to the closest centroids and recalculates the centroid of each set in the partition. Unluckily selected initial positions of the centroids may cause the algorithm to fall to local optima instead of the global optimum.

The linear programming technique used in one version of k -means to include constraints [7] allows for relaxing constraints. Since it involves fixing weights for the importance of these constraints, these results cannot be directly compared

with our method described further. Our method, similar to [9], does not include weighting of constraint satisfaction as it produces their exact satisfaction.

In this paper, we shall be using neural gas clustering algorithm which was first introduced in [11] and it has not been used for clustering with constrained sizes of clusters yet. Its inspiration comes from a type of neural network called SOM (self-organizing map) [12]. Neural gas is similar to both neural network-based clustering and partitional one. Neural gas gained popularity thanks to its robust convergence compared to online k -means clustering. It is mostly used in speech recognition and image processing for data compression or vector quantization.

Neural gas, similarly to SOM, can be used for putting together related data, or for clustering. It finds optimal data representations based on feature vectors (“observations,” represented by data points in multidimensional space) and is typically used in pattern recognition.

Similarly to SOM and a few other artificial neural networks, adaptation of neural gas repeats competitive learning and mapping. “Learning” moves cluster centers in the feature space by a competitive process called vector quantization using “observations.” “Mapping” assigns the “observations,” where each observation is assigned to the cluster with the closest center (Euclidean distance is used). Neural gas is composed of N neurons (defining centers of clusters, their number is fixed in advance), and position of each neuron i is defined by weight vector \mathbf{w}_i of the same dimension as the “observations” data vectors. The positions of neurons move around abruptly during the training, similar to gas molecules movements, which gave the algorithm its name.

In our testing cases, each weight vector defines the coordinates of its neuron in 1D or 2D space. After the adaptation, the coordinates of neurons should ideally correspond to the centers of clusters.

Initially, each neuron is associated with a random point (vector) from the “observations” data. Then, a randomly selected observation point \mathbf{x} is presented to the neural gas network. Euclidean distances of the selected point \mathbf{x} to all the weight vectors of neurons are calculated, and these centers are sorted by their distance from the selected point, from closest i_1 to most distant i_N . Then, each weight vector of the ordered sequence is adapted by

$$\mathbf{w}_i^{\text{new}} = \mathbf{w}_i^{\text{old}} + \varepsilon \cdot e^{-k_i/\lambda} (\mathbf{x} - \mathbf{w}_i^{\text{old}}), \quad i = 1, \dots, N, \quad (1)$$

where k_i is the number of neurons with weight vector closer to the current point \mathbf{x} than the current weight \mathbf{w}_i (i.e., index of its vector in ordered sequence, minus 1), ε is an adaptation step size, and λ is neighbourhood range. The parameters ε and λ are reduced with increasing number of presented points by the following equations:

$$\begin{aligned} \varepsilon_{\text{iter}} &= \varepsilon_{\text{start}} \cdot \left(\frac{\varepsilon_{\text{final}}}{\varepsilon_{\text{start}}} \right)^{(\text{iter}/\text{itermax})} \\ \lambda_{\text{iter}} &= \lambda_{\text{start}} \cdot \left(\frac{\lambda_{\text{final}}}{\lambda_{\text{start}}} \right)^{(\text{iter}/\text{itermax})}, \end{aligned} \quad (2)$$

where iter is the number of points presented so far and itermax is the total number of presented points.

The changes during the learning can be compared to the gradient descent method in the most typical multilayered perceptron neural networks, where the difference of the network output from the ideal output is proportional to the size of the adaptation change of the weights. Unlike SOM, not only winning neuron changes its position, but also all the other neurons move, and the more distant neurons move less.

2. Materials and Methods

2.1. Changes in the Neural Gas Learning. The adapted learning must take into account the fact that sometimes a data point or observation is closer to the center of one cluster, but since this cluster is already “saturated” by other data points assigned to it and achieved its full size, the point is assigned to another closest cluster, which still has a “free capacity.”

In the first iteration of the learning, the assignment of observations to clusters is not yet ready, so the adaptation of the vectors assigned to the centers remains the same as in the original neural gas algorithm, described by equations (1) and (2).

However, after the mapping step, each observation point is assigned to one of the clusters so that during the adaptation of a center of a cluster we can take the points already assigned to this cluster more seriously than the points assigned to another cluster.

This principle has been applied by changing the index value of k_i in (1). First, we sort the centers by the distance from the selected point in the same way as in the classical neural gas algorithm, but then we change the sequence by moving that center to the front of the sequence to which the currently selected point has been assigned in the previous iteration. The position of this center then changes the most.

In order to provide a most simple illustrative example, we use four points in 1D space placed at positions 0, 1, 2, and 10, which we clustered by the neural gas algorithm into two clusters. Each point during the learning iterations attracts both centers of cluster towards itself. In the beginning, the shifts in the positions are great and the centers shift quite abruptly, but due to the exponents in (2) which gradually change with iterations from 0 to 1, while their base is a positive number smaller than 1, the changes gradually diminish. At the top of the graphs in Figure 1, the two centers with each presented point move substantially to the left or right, but at the end of iterations in the bottom of figures, the proposed centers of clusters change only slightly. It means that with the increasing iteration number the positions of the centers of clusters converge to their final values. For our illustrative example, we use only 80 iterations, which is quite enough for our simple example. The initial value of $\varepsilon_{\text{start}}$ was set to $\varepsilon_{\text{start}} = 1$, its final value was set to $\varepsilon_{\text{final}} = 0.05$, and for λ these parameters were $\lambda_{\text{start}} = 10$ and $\lambda_{\text{final}} = 0.01$. Normally, the number of iterations is set to tens of thousands, and $\varepsilon_{\text{final}}$ would be much closer to 0 which would make better but lengthier convergence. Obviously, normally the two clusters of $n = 4$ points would be $C_1 = \{0, 1, 2\}$ and $C_2 = \{10\}$, with positions of cluster centers $\mathbf{w}_1 = 1$ and $\mathbf{w}_2 = 10$,

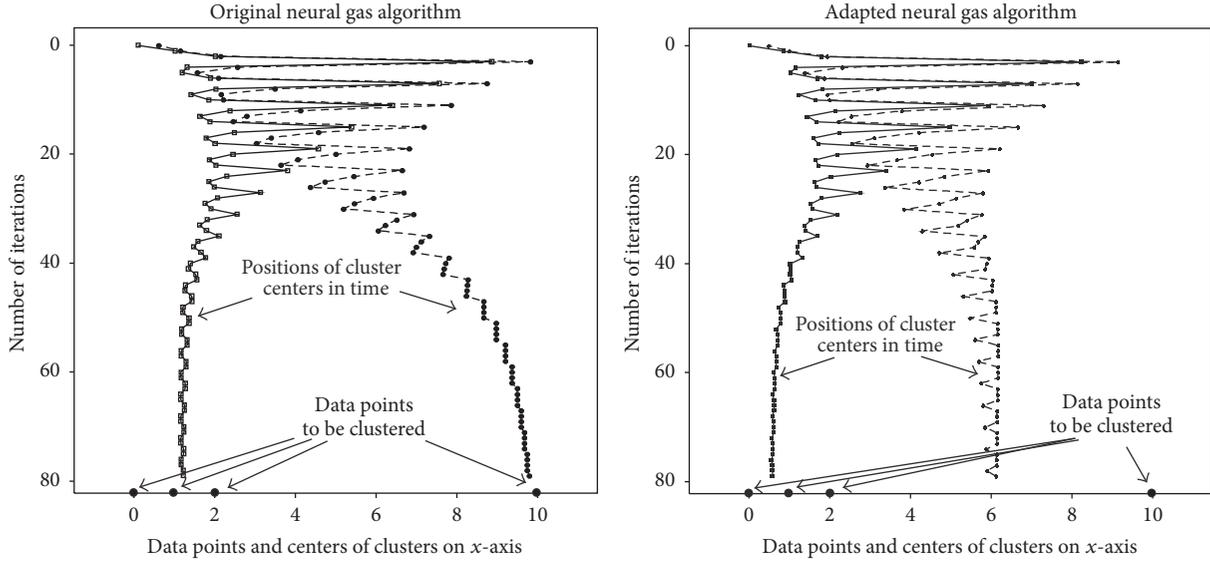


FIGURE 1: Changes in positions of cluster centers during learning iterations.

which would produce the minimal Mean Square Error $MSE = \sum_{j=1}^k \sum_{\mathbf{x}_i \in C_j} (\|\mathbf{x}_i - \mathbf{w}_j\|/n)$. This convergence of positions of cluster centers with increasing iterations towards values $\mathbf{w}_1 = 1$ and $\mathbf{w}_2 = 10$ can be seen in the left part of Figure 1 depicting the convergence of the original neural gas algorithm. The right part of Figure 1 shows a convergence of the adapted neural gas algorithm, where the size of both clusters was set to 2. Therefore, the 4 points were clustered with minimum MSE into clusters $C_1 = \{0, 1\}$ and $C_2 = \{2, 10\}$, whose centers would be $\mathbf{w}_1 = 0.5$ and $\mathbf{w}_2 = 6$.

2.2. Changes in the Mapping of Observations to Clusters.

In the classical k -means algorithm, the observations (data points) are assigned to the cluster with minimum Euclidean distance to its center. Here, we should assign each data point to the cluster, whose center yields the least within-cluster sum of squares, while the constraints concerning the cluster sizes are satisfied. We shall further simplify the examples by requiring balanced clustering, where each cluster has the same cardinality (i.e., the same number of data points is assigned to it). To achieve these occasionally controversial assignments, the adapted k -means algorithms [7, 8] used various strategies, from size regularized cut, which includes weights of satisfaction of constraints, up to relatively primitive assignment assigning the closest points, until each cluster achieves its preset size.

The following algorithm has been introduced for this purpose:

- (1) If the number of unassigned data points is zero, finish.
- (2) If the number of clusters with free capacity (i.e., those with the number of assigned points to them smaller than their desired size) is one, assign all the remaining data points to it and finish (we assume that the sum of the desired sizes of clusters equals the number of data points).

- (3) Order data points in descending order by the distance to their second nearest available center of cluster minus the distance to their nearest available center of the cluster (i.e., biggest benefit of the best over the second best assignment). If the nearest center of cluster has already achieved its preset size (and is therefore unavailable), add half of the difference of the data point to the second nearest and the nearest center to the already computed difference for available centers.
- (4) Assign data points according to their ordering from point 3 to their nearest cluster, until any of the clusters has achieved a preset size.
- (5) Remove the assigned points and the full cluster from further consideration and continue to point 1.

While this algorithm is not optimal (the whole problem is NP-hard), the further testing results will show that it works well. Its main contribution is in point 3. Let us illustrate it with a simple 1D example, with the same four data points as in Figure 1 placed in positions \mathbf{x}_i 0, 1, 2, and 10. Let us say that initial centers of clusters are set randomly to $\mathbf{w}_1 = 1.5$ and $\mathbf{w}_2 = 7$. Then, the distances are shown in Table 1.

If we would simply assign the closest points first to their respective centers, then the first center with $\mathbf{w}_1 = 1.5$ would be assigned first points $\mathbf{x}_2 = 1$ and $\mathbf{x}_3 = 2$, and the points $\mathbf{x}_3 = 10$ and $\mathbf{x}_1 = 0$ would be assigned to the second center $\mathbf{w}_2 = 7$. However, if we follow our algorithm described above, then we first assign $\mathbf{x}_3 = 10$ to $\mathbf{w}_2 = 7$, then $\mathbf{x}_1 = 0$ to $\mathbf{w}_1 = 1.5$, then $\mathbf{x}_2 = 1$ to $\mathbf{w}_1 = 1.5$, and finally $\mathbf{x}_3 = 2$ to $\mathbf{w}_2 = 7$.

Preferential assignment of the closest points to the centers would be too greedy and could assign the last data points totally wrong; it is evident in assigning the point at 0 to a faraway cluster center (see Figure 2).

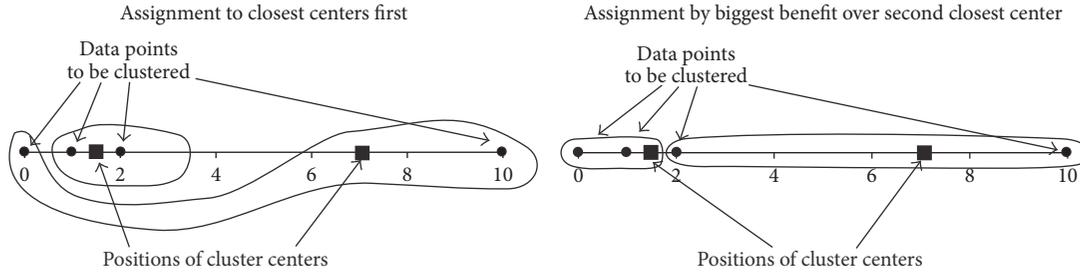


FIGURE 2: Assignment by biggest benefit over the second closest center is better.

TABLE 1: Distances of data points from centers of clusters.

| Position of point x_i | Distance of point from center $w_1 = 1.5$ | Distance of point from center $w_2 = 7$ | $ w_1 - x_i ^2 - w_2 - x_i ^2 $ |
|-------------------------|---|---|---------------------------------------|
| 0 | 1.5 | 7 | $ 1.5 - 0 ^2 - 7 - 0 ^2 = 46.75$ |
| 1 | 0.5 | 6 | $ 1.5 - 1 ^2 - 7 - 1 ^2 = 35.75$ |
| 2 | 0.5 | 5 | $ 1.5 - 2 ^2 - 7 - 2 ^2 = 24.75$ |
| 10 | 8.5 | 3 | $ 1.5 - 10 ^2 - 7 - 10 ^2 = 63.25$ |

3. Results and Discussion: Testing of the Adapted Neural Gas Algorithm

The main problem with testing the adapted neural gas algorithm is that there are no proper testing data. The classical clustering testing data have no results for clustering with constraints, and the published works on clustering with constraints mostly consider various constraints (like some data points must be together and some must not be together). Even the papers with size constraints take various measures, either implicit or explicit, to balance two goals in this multiobjective optimization, to achieve a minimum difference from desired cluster sizes while achieving minimum within-clusters sums of distances.

Since our approach provides the desired cluster sizes with no allowance for error, the results are not directly comparable with most of the other published algorithms. Furthermore, since the quality of achieved clustering strongly depends on the number of iterations, which is in neural gas set in advance, the complexity of the algorithm cannot be exactly measured, similar to evolutionary algorithms.

We have therefore generated artificial “mouse” data in two-dimensional space, similar to those occasionally used in testing, where the dataset can be divided into three clearly visible clusters with data encompassed by circles denoting the “face” of the Mickey Mouse and its “ears.” The “face” circle consists of 300 randomly generated points with uniform distribution, the center of the circle in $(0, 0)$, and its radius equal to 1, while the “ears” circles have centers $(1, 1)$ or $(-1, 1)$ and the radius is set to $1/3$.

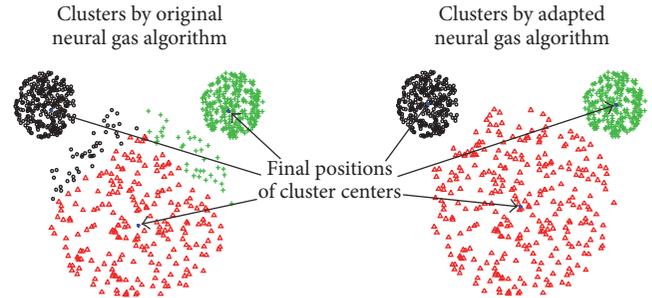


FIGURE 3: Mouse data clustered by original and adapted neural gas algorithm.

The results of the algorithm with 5000 iterations, 3 desired clusters, $\epsilon_{\text{start}} = 0.5$, $\epsilon_{\text{final}} = 0.005$, and $\lambda_{\text{start}} = 10$, $\lambda_{\text{final}} = 0.01$ are given in Figure 3, when for the adapted algorithm the sizes of clusters were set to 300. The figure clearly shows that all the data points by adapted neural gas on the right-hand side were assigned correctly to clusters (which show points assigned to them as red triangles, green crosses, and black circles). On the other hand, results on the left-hand side of the classical neural gas algorithm provided similar counterintuitive results as would classical k -means algorithm, assigning “upper” data points from “face” to “ears” clusters with closer centers.

Figure 4 shows that at the beginning of learning the algorithm produces a significant MSE error which moreover substantially fluctuates as the centers of the clusters flit around like in Figure 1, only in two dimensions instead of one dimension. At the second half of iterations, the error approaches its minimum value. Smoothed maximum, average, and minimum MSE values collected from 10 runs are shown in Figure 5.

Similar results can be seen in Figure 6 which shows smoothed maximum, average, and minimum numbers of points wrongly assigned to clusters collected from 10 runs. It is apparent that these numbers fluctuate in the early iterations even more wildly than MSE error, but in the second half of iterations, the number of wrongly assigned points drops to zero. If necessary, there exist approaches designed to damp the fluctuations [14] and thus increase the convergence speed of the optimization, but since this would increase a danger to be stuck in a local optimum, we decided not to use them.

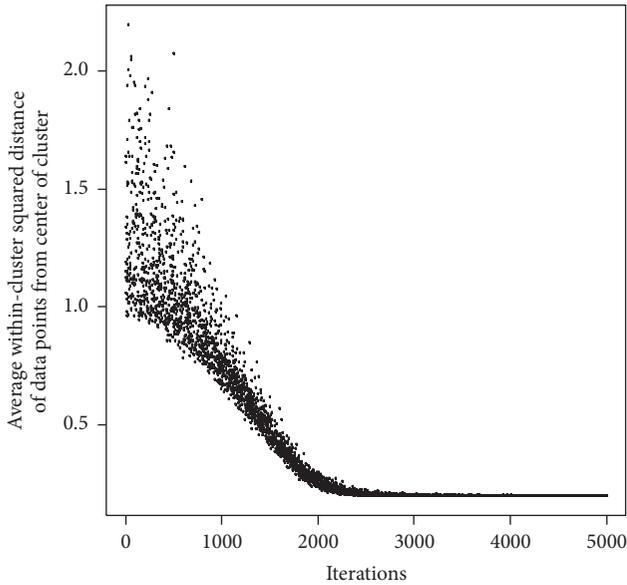


FIGURE 4: Convergence of Minimum Square Error during a typical run.

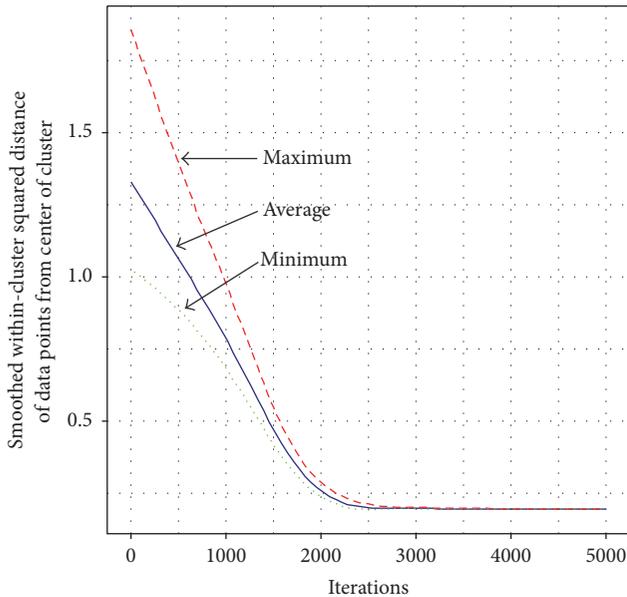


FIGURE 5: Smoothed convergence of maximum, average, and minimum values of Minimum Square Error from 10 runs.

From Figures 4, 5, and 6, we can deduce that the adapted neural gas algorithm converges without problems. The number of iterations of the neural gas algorithm is controlled by its parameters; more iterations mean longer time but better results. Similar to evolutionary algorithms, complexity of the algorithm therefore cannot be easily measured, contrary to k -means clustering.

In order to prove the efficiency of our proposed algorithm, we have also compared the results of our algorithm with balanced k -means algorithm [8]. For the comparison, the same artificial testing dataset generated and tested in

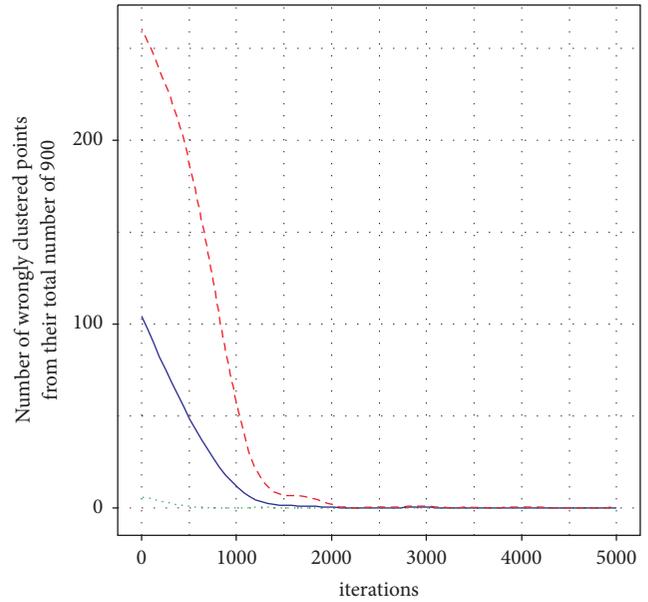


FIGURE 6: Smoothed convergence of maximum, average, and minimum values of the number of wrongly clustered points from 10 runs.

[8, 15] with 15 clusters was used where 5 clusters were of the forced size 34 and 10 clusters of forced size 33. Furthermore, we have used a dataset att532 of 532 cities located in the (continental) United States [13], which does not have any natural partition into clusters. This dataset was partitioned into 4, 7, 14, and 19 equally sized clusters, and the results were compared again with balanced k -means algorithm [8]. The result of the best clustering for adapted neural gas algorithm can be seen in Figure 7. When we used the same running time by reducing the number of iterations of our algorithm to 2000, the adapted neural gas algorithm achieved the same value for minimum of errors from one hundred runs. Moreover, it achieved much better values for the mean of errors and for the standard deviation of errors than balanced k -means algorithm for higher numbers of clusters (see Table 2). With more iterations, the mean and standard deviation would be even substantially better; it can be done easily by changing one parameter.

4. Conclusions

We have designed and tested an adapted neural gas algorithm which includes user provided constraints of the sizes of clusters. We have tested the algorithm with the requirement for balanced (i.e., all equal) sizes of clusters. The results showed that the algorithm converges without problems. Since we do not allow the cluster size constraints to be relaxed, we did not compare our adapted neural gas algorithm with a constrained k -means algorithm but we compared our algorithm with balanced k -means algorithm, where our algorithm produces better average results for higher number of clusters. Since the neural gas algorithm is generally considered to be more robust than the k -means algorithm, we have every reason to

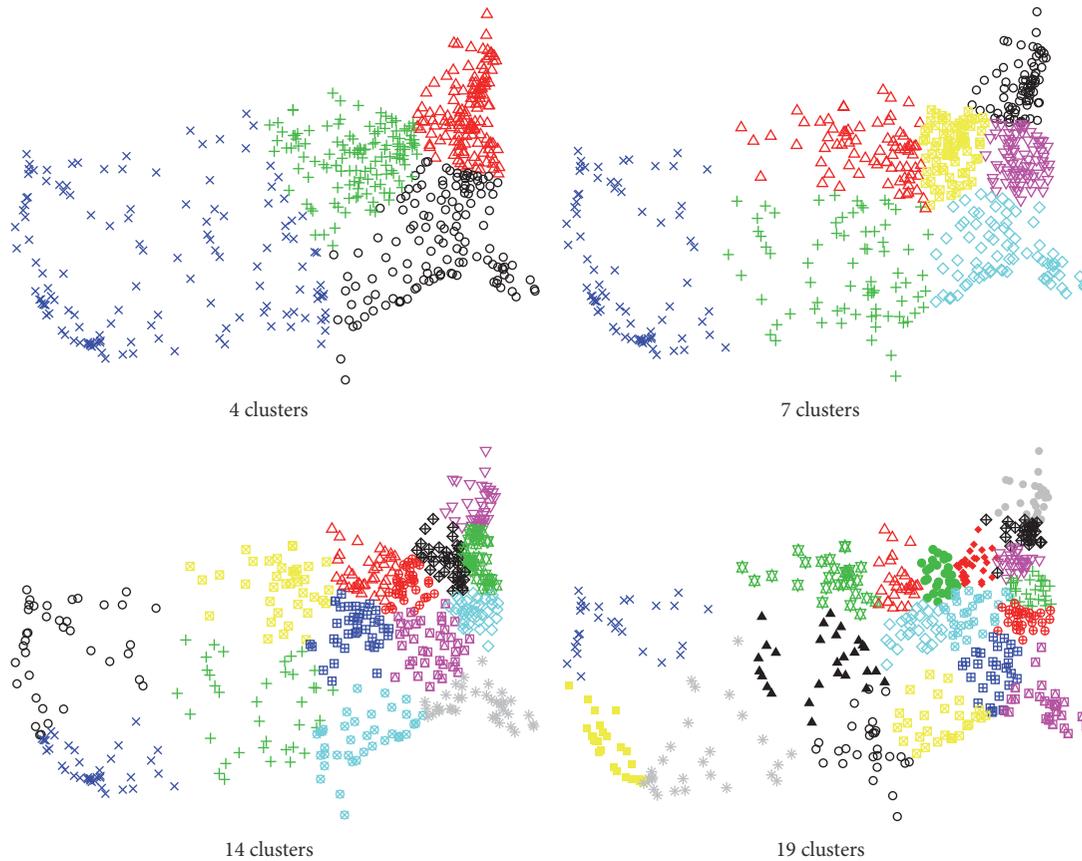


FIGURE 7: The dataset att532 [13] of 532 cities located in USA, used typically for travelling salesman problem, clustered by adapted neural gas into 4 clusters of 133 cities, 7 clusters of 76 cities, 14 clusters of 38 cities, and 19 clusters of 28 cities.

TABLE 2: Best MSE, mean MSE, and standard deviation of MSE of 100 runs for distances of data points from centers of clusters.

| Dataset | Size | Clusters | Algorithm | Best | Mean | St. dev. |
|-----------|------|----------|---------------------|-------|-------|----------------------|
| SI subset | 500 | 15 | Balanced k -means | 3.48 | 3.73 | 0.21 |
| | | | Adapted neural gas | 3.48 | 3.56 | 0.04 |
| att532 | 532 | 4 | Balanced k -means | 1.597 | 1.597 | 0 |
| | | | Adapted neural gas | 1.597 | 1.597 | 1.0×10^{-4} |
| att532 | 532 | 7 | Balanced k -means | 0.723 | 0.723 | 8.6×10^{-5} |
| | | | Adapted neural gas | 0.723 | 0.723 | 1.0×10^{-4} |
| att532 | 532 | 14 | Balanced k -means | 0.343 | 0.351 | 5.9×10^{-3} |
| | | | Adapted neural gas | 0.343 | 0.343 | 1.2×10^{-4} |
| att532 | 532 | 19 | Balanced k -means | 0.252 | 0.266 | 0.012 |
| | | | Adapted neural gas | 0.252 | 0.252 | 1.6×10^{-4} |

assume that our adapted neural gas algorithm is more robust than its modified k -means counterpart.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

References

- [1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.
- [2] Y. Pang, F. Xiao, H. Wang, and X. Xue, "A clustering-based grouping model for enhancing collaborative learning," in *Proceedings of the 13th International Conference on Machine*

- Learning and Applications (ICMLA '14)*, pp. 562–567, Detroit, Mich, USA, December 2014.
- [3] M. Hostovecky, M. Misut, and K. Pribilova, “Web based testing in science education,” in *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering*, T. Sobh and K. Elleithy, Eds., vol. 313 of *Lecture Notes in Electrical Engineering*, pp. 247–254, Springer, Berlin, Germany, 2015.
 - [4] M. Šimon, L. Huraj, and M. Čerňanský, “Performance evaluations of IPTables firewall solutions under DDoS attacks,” *Journal of Applied Mathematics, Statistics and Informatics*, vol. 11, no. 2, pp. 35–45, 2015.
 - [5] K. Lei, S. Wang, W. Song, and Q. Li, “Size-constrained clustering using an initial points selection method,” in *Knowledge Science, Engineering and Management*, M. Wang, Ed., vol. 8041 of *Lecture Notes in Computer Science*, pp. 195–205, Springer, Berlin, Germany, 2013.
 - [6] L. Cabyova and J. Ptacin, “Benchmarking comparison of marketing communication of universities in Slovakia,” *Communication Today*, vol. 5, no. 1, pp. 54–69, 2014.
 - [7] S. Zhu, D. Wang, and T. Li, “Data clustering with size constraints,” *Knowledge-Based Systems*, vol. 23, no. 8, pp. 883–889, 2010.
 - [8] M. I. Malinen and P. Fränti, “Balanced K -means for clustering,” in *Structural, Syntactic, and Statistical Pattern Recognition*, P. Fränti, G. Brown, M. Loog, F. Escolano, and M. Pelillo, Eds., vol. 8621 of *Lecture Notes in Computer Science*, pp. 32–41, Springer, Berlin, Germany, 2014.
 - [9] N. Ganganath, C.-T. Cheng, and C. K. Tse, “Data clustering with cluster size constraints using a modified k-means algorithm,” in *Proceedings of the 6th International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC '14)*, pp. 158–161, Shanghai, China, October 2014.
 - [10] H. Steinhaus, “Sur la division des corps matériels en parties,” *Bulletin L'Académie Polonaise des Science (BAPMAM)*, vol. 4, no. 12, pp. 801–804, 1957 (French).
 - [11] T. Martinetz and K. Schulten, “A neural gas network learns topologies,” in *Artificial Neural Networks*, T. Kohonen et al., Ed., pp. 397–402, Elsevier, 1991.
 - [12] T. Kohonen, M. R. Schroeder, and T. S. Huang, Eds., *Self-Organizing Maps*, Springer, New York, NY, USA, 3rd edition, 2001.
 - [13] TSPLIB by Gerhard Reinelt, Universitat Heidelberg, <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>.
 - [14] D. Gabriska, G. Michalconok, P. Tanuska, and T. Skulavik, “Analysis of the compensation algorithm stability of disturbance for the phase control systems,” in *Proceedings of the 11th IFAC/IEEE International Conference on Programmable Devices and Embedded Systems (PDeS '12)*, pp. 244–248, Brno, Czech Republic, May 2012.
 - [15] M. Malinen and P. Fränti, “Balanced k-means implementation,” University of Eastern Finland, <http://cs.uef.fi/sipu/soft/Balanced.zip>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

