A Study on the Hierarchical Data Clustering Algorithm Based on Gravity Theory

Yen-Jen Oyang, Chien-Yu Chen, and Tsui-Wei Yang

Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan
yjoyang@csie.ntu.edu.tw
cychen@mars.csie.ntu.edu.tw
lotte@mars.csie.ntu.edu.tw

Abstract. This paper discusses the clustering quality and complexities of the hierarchical data clustering algorithm based on gravity theory. The gravity-based clustering algorithm simulates how the given N nodes in a K-dimensional continuous vector space will cluster due to the gravity force, provided that each node is associated with a mass. One of the main issues studied in this paper is how the order of the distance term in the denominator of the gravity force formula impacts clustering quality. The study reveals that, among the hierarchical clustering algorithms invoked for comparison, only the gravity-based algorithm with a high order of the distance term neither has a bias towards spherical clusters nor suffers the well-known chaining effect. Since bias towards spherical clusters and the chaining effect are two major problems with respect to clustering quality, eliminating both implies that high clustering quality is achieved. As far as time complexity and space complexity are concerned, the gravity-based algorithm enjoys either lower time complexity or lower space complexity, when compared with the most well-known hierarchical data clustering algorithms except single-link.

Keywords: data clustering, agglomerative hierarchical clustering, gravity force.

1 Introduction

Data clustering is one of the most traditional and important issues in computer science [4, 7, 9, 10]. In recent years, due to emerging applications such as data mining and document clustering, data clustering has attracted a new round of attention in computer science research communities [3, 5, 6, 11, 14, 17, 19]. One traditional taxonomy of data clustering algorithms that work on data points in a K-dimensional continuous vector space is based on whether the algorithm yields a hierarchical clustering dendrogram or not [10]. One major advantage of the hierarchical clustering algorithms is that a hierarchical dendrogram is generated. This feature is very important for applications such as in biological, social, and behavior studies, due to the need to construct taxonomies [9]. Furthermore, as Jain, Murty, and Flynn summarized [10], hierarchical clustering algorithms are more versatile than non-hierarchical algorithms, or so-called partitional algorithms. For example, most partitional algorithms work well only on data sets containing isotropic clusters. Nevertheless, hierarchical clustering algorithms suffer higher time and space complexities [10]. Therefore, a latest

L. De Raedt and A. Siebes (Eds.): PKDD 2001, LNAI 2168, pp. 350-361, 2001.

trend is to integrate hierarchical and partitional clustering algorithms such as in BIRCH[19], CURE[5], and Chameleon[11]. In the kernel of these algorithms, a hierarchical clustering algorithm can be invoked to derive a dendrogram and to improve clustering quality. Due to this trend, it is expected that hierarchical clustering algorithms will continue to play an important role in applications that require a dendrogram. Furthermore, clustering quality becomes the prevailing concern in comparing various hierarchical clustering algorithms.

This paper discusses the clustering quality and complexities of the hierarchical data clustering algorithm based on gravity theory in physics. The gravity theory based clustering algorithm simulates how the given N nodes in a K-dimensional continuous vector space will cluster due to gravity force, provided that each node is associated with a mass. The idea of exploiting gravity theory in data clustering was first proposed by W. E. Wright in 1977 [16]. In the article, Wright discussed several factors that may impact clustering quality. Nevertheless, one crucial factor that was not addressed in Wright's article is the order of the distance term in the denominator of the gravity force formula. As we know, the order of the distance term is 2 for the gravity force. However, there are natural forces of which the magnitude of influence decreases much rapidly as distance increases. One such force is the strong force in atom nuclei. This observation inspired us to investigate the effect of the order of the distance term. In this paper, we still use the term "gravity force", even though we employ various orders of the distance term in the simulation model.

The experiments conducted in this study shows that the order of the distance term does have a significant impact on clustering quality. In particular, with a high order of the distance term, the gravity-based clustering algorithm neither has a bias towards spherical clusters nor suffers the well-known chaining effect [10,17]. Figure 1 exemplifies how bias towards spherical clusters impacts clustering quality. In Fig. 1, the data points at the two ends of the two dense regions are clustered. As will be shown in this paper, except the single-link algorithm, all the conventional hierarchical clustering algorithms studied in this paper as well as the gravity-based clustering algorithm with a low order of the distance term have a bias toward spherical clusters. On the other hand, the single-link algorithm suffers the well-known chaining effect. Since bias towards spherical clusters and the chaining effect are two common problems with respect to clustering quality, avoiding both implies that high clustering quality is achieved.

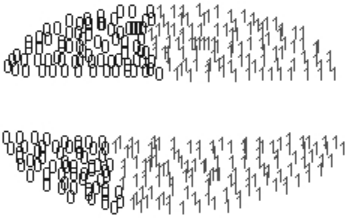


Fig. 1. An example of how a clustering algorithm with bias towards spherical clusters suffers poor clustering quality. The data points in the two clusters are marked by 0 and 1, respectively.

As Wright did not address time and space complexities of the gravity-based clustering algorithm, we conduct a detailed analysis in this paper. Table 1 compares the time complexity and space complexity between the gravity-based clustering algorithm and the most well-known hierarchical clustering algorithms [2, 10] that work in spaces of any degrees of dimension. The hierarchical clustering algorithms that work only in low-dimensional spaces [1, 12] are not included for comparison. The time and space complexities of the gravity-based algorithm reported in Table 1 are based on the simulation model employed in this paper, which is slightly different from the model employed in Wright's paper. Though Wright did not analyze the time and space complexities of his algorithm, our simulation results show that Wright's simulation model has the same orders of complexities as the simulation model employed in this paper. As Table 1 reveals, the gravity-based clustering algorithm enjoys either lower time complexity or space complexity, when compared with the most well-known hierarchical clustering algorithms except single-link.

In the following part of this paper, Sect. 2 elaborates how the gravity-based clustering algorithm works. Section 3 analyzes its time complexity and space complexity. Section 4 reports the experiments conducted to compare the gravity-based algorithm with the most well-known hierarchical clustering algorithms. Finally, concluding remarks are presented in Sect. 5.

Table 1. Time and space complexities of the gravity-based clustering algorithm and the most well-known hierarchical clustering algorithms.

Clustering Algorithm	Time complexity	Space complexity
The gravity-based algorithm	$O(N^2)$	$O(N)$
Single-Link [2]	$O(N^2)$	$O(N)$
Complete-Link [10, 13]	$O(N^2 \log N)$	$O(N^2)$
Centroid [2], Group Average [2]	$O(N^2 \log^2 N)$	$O(N)$

2 The Gravity-Based Clustering Algorithm

The simulation model that the gravity-based data clustering algorithm assumes is an analogy of how a number of water drops move and interact with each other in the cabin of a spacecraft. The main difference between the simulation model employed in this paper and that employed in Wright's paper is the order of the distance term in the denominator of the gravity force formula. This paper studies how different orders of the distance term impact clustering quality. Another difference is that the effect of air resistance is taken into account in this paper for guaranteeing termination of the algorithm, which was not addressed in Wright's paper.

Now, let us elaborate the simulation model employed in this paper. Due to the gravity force, the water drops in the cabin of a spacecraft will move toward each other. When these water drops move, they will also experience resistance due to the air in the cabin. Whenever two water drops hit, which means that the distance be-

tween these two drops is less than the lumped sum of their radii, they merge to form one new and larger water drop. In the simulation model, the merge of water drops corresponds to forming a new, one-level higher cluster that contains two existing clusters. The air resistance is intentionally included in the simulation model in order to guarantee that all these water drops eventually merge into one big drop regardless of how these water drops spread in the space initially. Before examining the details of the simulation algorithm, let us first discuss some important observations based on our physical knowledge.

Observation 1: As time elapses, the system can not continue to stay in a state in which there are two or more isolated water drops and all these water drops stand still.

Reason:

The system may enter such a state but will leave that state immediately due to gravity forces among these water drops.

Observation 2: As long as the system still contains two or more isolated water drops, the lumped sum of the dynamic energy and potential energy in the system will continue to decrease as time elapses.

Reason:

Due to Observation 1, if the system still contains two or more isolated water drops, then these water drops can not all stand still indefinitely. As some water drops move, the dynamic energy will gradually dissipate due to air resistance. Actually, the dissipated dynamic energy is turned to another form of energy. That is heat. Furthermore, as the dynamic energy in the system continues to dissipate, the potential energy in the system will gradually convert to dynamic energy. Since the dissipation of dynamic energy is a non-stopping process as long as there are two or more isolated water drops in the system. The lumped sum of dynamic energy and potential energy in the system will continue to decrease as time elapses.

Observation 3: Regardless of how the water drops spread in the space initially, all water drops will eventually merge into one big drop.

Reason:

Assume that there is an initial spreading of water drops such that the system never reaches a state in which all water drops merge into one big water drop. Let *ENG(t)* denote the lumped sum of the potential energy and dynamic energy in the system. According to Observation 2, *ENG(t)* is a monotonically decreasing function as long as there are two or more isolated water drops in the system. Since $ENG(t) \geq 0$ at any time, there is a number $a \geq 0$ such that $\lim_{t\to\infty} ENG(t) = a$. $\lim_{t\to\infty} ENG(t) = a$ implies $\lim_{t\to\infty} \frac{dENG(t)}{dt} = 0$. Because the air resistance force experienced by a moving water

drop is proportional to the square of its moving velocity, $\lim_{t\to\infty}\frac{dENG(t)}{dt}=0$ implies the velocities of all water drops will approach 0 as time elapses. However, just like the reason for Observation 1 above, the velocities of water drops can not all approach 0 as time elapses, because the gravity forces will accelerate them. Therefore, a contradiction would occur, if our assumption held. Hence, the system will eventually reach a state in which all water drops merge into one big drop.

The physical observations discussed above implies that the gravity-based data clustering algorithm based on simulating the physical system discussed above will eventually terminate. Following is a formal description of the simulation model.

1. Each of the N initial nodes in the K-dimensional space is associated with a mass M_0.
2. There are two forces applied to the nodes in the system. The first one is gravity force and the second one is air resistance.
3. The gravity force F_g applied to two nodes apart by a distance r is equal to:
$$F_g=\frac{C_g\times M_1\times M_2}{r^k},$$
where C_g is a constant, M_1 and M_2 are the masses of these two nodes, and k is a positive integer.
4. The nodes will suffer air resistance when they move. The air resistance force F_r that a moving node experiences is equal to
$$F_r=C_r\times v^2,$$
where C_r is a constant and v is the velocity of the node.
5. At any time, if two nodes are apart by a distance less than the sum of their radii, then these two nodes will merge to form a new node with lumped mass. The radius of the new node is determined by the mass of the node and a given constant, which denotes the density of the material. As far as momentum is concerned, the momentum of the new node is equal to the addition of the momentums of the original nodes. The merge of two nodes corresponds to forming a new, one-level higher cluster that contains two existing clusters represented by the two original nodes.

Figure 2 shows the pseudo-code of the gravity-based clustering algorithm. Basically, the algorithm iteratively simulates the movement of each node during a time interval T and checks for possible merges. The algorithm terminates when all nodes merge into one big node.

```
W: the set containing all disjoint nodes.  At the beginning, W
contains all initial nodes.
Repeat
   For every w_i ∈ W {
      calculate the acceleration of w_i based on the gravity
         forces applied on w_i by other nodes in W and the mass of
         w_i ;
      calculate the new velocity of w_i based on its current ve-
         locity and acceleration;
      calculate the new position of w_i based on its current ve-
         locity;
   };
   For every pair of nodes w_i , w_j ∈ W {
      if ( w_i and w_j hit during the given time interval T){
         create a new cluster containing the clusters represented
            by w_i and w_j ;
         merge w_i and w_j to form a new node w_k with lumped
            masses and merged momentum;
      };
   };
Until (W contains only one node);
```

Fig. 2. Pseudo-code of the gravity-based data clustering algorithm

3 Time and Space Complexity of the Gravity-Based Algorithm

We will employ a probability model to prove that the time complexity of the gravity-based data clustering algorithm is $O(N^2)$, where N is the number of nodes initially. The proof is based on the observation that the expected number of disjoint nodes remaining after each iteration of simulation decreases exponentially.

Assume that these N initial nodes randomly spreading in a K-dimensional Euclidian space bounded by $[X_{1l}, X_{1h}],[X_{2l}, X_{2h}],.....,[X_{kl}, X_{kh}]$, where X_{jl} and X_{jh} are the lower bound and upper bound in the *j-th* dimension, respectively. Depending on how these N nodes initially spread in the bounded space, the number of disjoint nodes remained after the *i-th* iteration of the gravity-based data clustering algorithm may differ. Let N_i denote the random variable that corresponds to the number of disjoint nodes after the *i-th* iteration of the gravity-based algorithm. It has been proved that all the nodes will eventually merge into one big node. Therefore, if the number initial nodes N and the boundary in the K-dimensional space are determined, then there exists an integer number S such that all nodes merge into one big node after S iterations of the gravity-based algorithm regardless of how these N nodes spread in the

bounded space initially. Let $E[N_i]$ denote the expected value of random variable N_i and q = Maximum($\frac{E[N_{i+1}]}{E[N_i]}$), where $0 \leq i < S\text{-}1$ and $E[N_0] = N$, $E[N_S] = 1$. Then, we have $0 < q < 1$ and

$$E[N_i] \leq N \times q^i \tag{1}$$

One important attribute of q that we will exploit later is that q decreases as the number of initial nodes N increases, as long as the boundary in the K-dimensional space in which the initial nodes spread does not change with the value of N. As the number of nodes in a fixed-size space increases, the probability that two nodes hit during a time interval increases. As a result, q decreases as N increases.

To determine the time complexity of the graivity-based data clustering algorithm, we need to determine the number of operations performed in each iteration of the algorithm. In each iteration of the algorithm, we need to compute the distance between each pair of disjoint nodes and check for possible merges. The complexity of carrying out these two operations is in quadratic order. The complexities of all other operations executed in one iteration are in lower orders and thus can be ignored in determining time complexity. Therefore, the time complexity of the gravity-based data clustering algorithm is equal to

$\sum_{i=0}^{S-1} E[C \times {N_i}^2]$, where C is a constant.

$$\sum_{i=0}^{S-1} E[C \times {N_i}^2] = C \times \sum_{i=0}^{S-1} E[{N_i}^2] = C \times \sum_{i=0}^{S-1} \sum_{l=1}^{N} \text{Probability}(N_i = l) \times l^2$$

$$\leq C \times \sum_{i=0}^{S-1} \sum_{l=1}^{N} \text{Probability}(N_i = l) \times l \times N = C \times N \times \sum_{i=0}^{S-1} \sum_{l=1}^{N} \text{Probability}(N_i = l) \times l$$

$$= C \times N \times \sum_{i=0}^{S-1} E[N_i] \leq C \times N \times \sum_{i=0}^{S-1} N \times q^i \text{ , according to (1) above}$$

$$= C \times N^2 \times \sum_{i=0}^{S-1} q^i = C \times \frac{1-q^S}{1-q} \times N^2$$

As elaborated earlier, q decreases as N increases and $0<q<1$. Therefore, term $\frac{1-q^S}{1-q}$ decreases as N increases and the time complexity of the gravity-based data clustering algorithm is $O(N^2)$. The space complexity of the algorithm is $O(N)$, because the space complexity of the hierarchical dendrogram built by the clustering algorithm is $O(N)$ and in each iteration we need to compute and store the location, velocity, and acceleration of each disjoint node.

4 Experimental Results

This section reports the experiments conducted to study how the gravity-based clustering algorithm performs in practice. The first part of this section reports how various algorithms perform with respect to clustering quality. The second part of this section reports the execution times of the gravity-based algorithm when running on real data sets. The clustering algorithms included in the clustering quality comparison are as follows:

(1) the gravity-based clustering algorithm based on our simulation model with the order of the distance term set to 5;
(2) the gravity-based clustering algorithm based on Wright's model with the order of the distance term set to 2;
(3) four conventional hierarchical clustering algorithms: single-link[9, 10], complete-link[9, 10], group-average[9, 10], and centroid[9, 10].

Table 2 shows how the parameters in the gravity-based algorithms were set in these experiments. According to our experiences, the order of the distance term has the dominant effect on clustering quality. With the order of the distance term set to 5 or higher, the gravity-based clustering algorithm neither has a bias towards spherical clusters nor suffers the chaining effect. The settings of C_g , M_0, and T mainly affect how rapidly the nodes merge in iterations. Employing small values for these parameters may result in more iterations in simulation. Nevertheless, the effect does not change the order of the time complexity of the algorithm. It only affects a coefficient in the time complexity formula. The remaining two parameters, C_r and D_m, have essentially no effect on clustering quality or speed of converging as long as they are not set to some weird values. For example, the coefficient of air resistance should not be set so high that the nodes can hardly move and the material density of the nodes should not be set so high that all the nodes have virtually no volume and can hardly hit each other.

Table 2. The parameter settings in the gravity-based algorithms.

C_g : Gravity force coefficient	30
M_0 : Initial mass of each node	1
C_r : Air resistance coefficient	0.01
D_m : Material density of the node	1
T : Time interval of each iteration	1

(a) The simulation model employed in this paper

q : the order of the mass term	0
δ : the distance that the node with maximum velocity moves in one iteration	1

(b) Wright's simulation model

Figures 3~5 show three experiments conducted to study the clustering quality of different algorithms. These figures only plot the remaining clusters before the last merge of clusters is executed for better visualization quality. In these figures, different clusters are plotted using different marks. The experimental results presented in Fig. 3 show the effect caused by bias towards spherical clusters. As shown in Fig. 3, except the gravity-based algorithm with a high order of the distance term and the single-link algorithm, all other algorithms have a bias towards spherical clusters and generate clusters that contain data points from both separate dense regions. The ex-

perimental results presented in Fig. 4 show the well-known chaining effect. In this case, only the single-link algorithm suffers the chaining effect. As shown in Fig. 4b, the cluster containing the data points marked by "O" extends to both spheres. Fig. 5 shows a data set designed to test how each algorithm handle both bias towards spherical clusters and the chaining effect. As shown in Fig. 5, only the gravity-based algorithm with a high order of the distance term neither has a bias towards spherical clusters nor suffers the chaining effect.

We also use three 3-dimensional data sets to compare the clustering quality of various algorithms. Figure 6a depicts the shapes of the 3 data sets used in the experiments and the numbers of data points in these data sets, respectively. Figure 6b summarizes the clustering quality of various algorithms. Again, only the gravity-based algorithm with a high order of the distance term neither has a bias toward spherical clusters nor suffers the chaining effect.

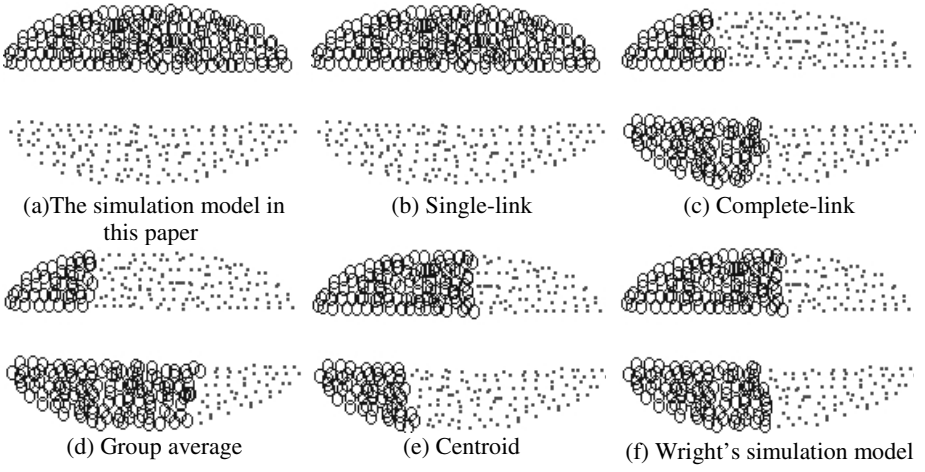


Fig. 3. Clustering results of the first experiment

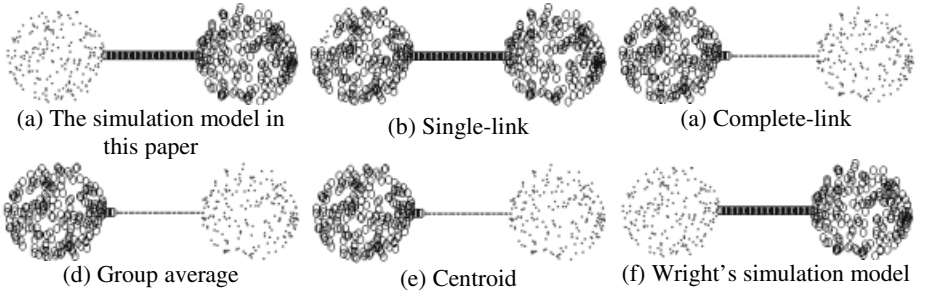
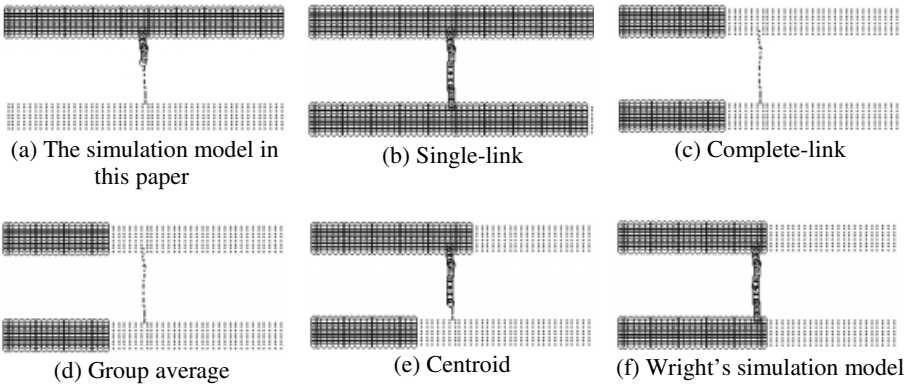


Fig. 4. Clustering results of the second experiment

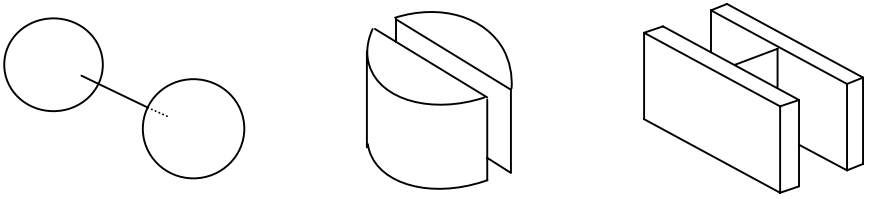


(a) The simulation model in this paper (b) Single-link (c) Complete-link

(d) Group average (e) Centroid (f) Wright's simulation model

Fig. 5. Clustering results of the third experiment

The experimental results above show that the gravity-based algorithm with a high order of the distance term is not biased towards spherical clusters. However, if the order of the distance term is low, then the situation may be different. We must resort to our physical intuition to explain this phenomenon. With a high order of the distance term, the influence of the gravity force decays rapidly as distance increases. Therefore, data points separated by a channel feel virtually no influence from each other.



Data set 1 (808 points) Data set 2 (1800 points) Data set 3 (4175 points)

(a) The 3 data sets used.

	Data set 1	Data set 2	Data set 3
Single-link	Poor (due to chaining effect)	Good	Poor (due to the chaining effect)
Complete-link	Good	Poor (bias towards spherical clusters)	Poor (bias towards spherical clusters)
Group average	Good	Poor (bias towards spherical clusters)	Poor (bias towards spherical clusters)
Centroid	Good	Poor (bias towards spherical clusters)	Poor (bias towards spherical clusters)
Wright's model	Good	Poor (bias towards spherical clusters)	Poor (bias towards spherical clusters)
The simulation model in this paper	Good	Good	Good

(b) Summary of clustering quality of various algorithms

Fig. 6. The experiments conducted on 3-D data sets

As far as execution time is concerned, Fig. 7 shows how the execution time of the gravity-based algorithm increases with the number of initial nodes to be clustered. The experiment was conducted on a machine equipped with a 700-MHz Intel Pentium-III CPU and 786 Mbytes main memory and running Microsoft Window 2000 operating system. The data set used is the Sequoia 2000 storage benchmark[15], which contains 62556 nodes in total. In this experiment, we randomly selected a subset of nodes from the benchmark dataset. The results in Fig. 7 confirm that the time complexity of the gravity-based clustering algorithm is $O(N^2)$.

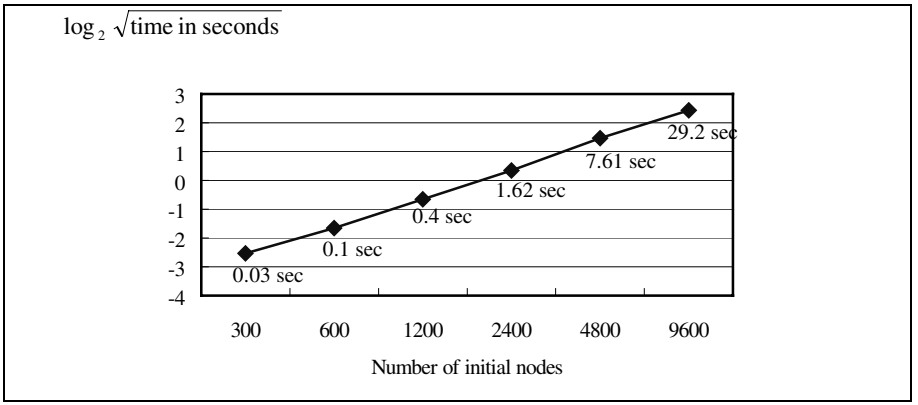


Fig. 7. Execution time of the gravity-based clustering algorithm versus the number of initial nodes to be clustered

5 Conclusions

This paper studies the clustering quality and complexities of a hierarchical data clustering algorithm based on gravity theory in physics. In particular, this paper studies how the order of the distance term in the denominator of the gravity force formula impacts clustering quality. The study reveals that with a high order of the distance term, the gravity-based clustering algorithm neither has a bias towards spherical clusters nor suffers the chaining effect. Since bias towards spherical clusters and the chaining effect are two major problems with respect to clustering quality, eliminating both implies that high clustering quality is achieved. As far as time complexity and space complexity are concerned, the gravity-based algorithm enjoys either lower time complexity or lower space complexity, when compared with the well-known hierarchical data clustering algorithms except single-link.

As discussed earlier, a latest trend in developing clustering algorithms is to integrate hierarchical and partitional algorithms. Since the general properties of the gravity-based algorithm with respect to clustering quality are similar to those of the density-based partitional algorithms such as DBSCAN [3] and DENCLUE [8], it is of interest to develop a hybrid algorithm that integrates the gravity-based algorithm and the density-based algorithm. In the hybrid algorithm, the gravity-based algorithm is

invoked to derive the desired dendrogram. This is the follow-up work that we have been investigating.

References

1. Choudry, S. and N. Murty, *A divisive scheme for constructing minimal spanning trees in coordinate space*, Pattern Recognition Letters, volume 11 (1990), number 6 , pp. 385-389
2. D. Eppstein, *Fast hierarchical clustering and other applications of dynamic closest pairs*, The ACM Journal of Experimental Algorithmics, 5(1):1-23, Jun 2000
3. M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *A density-based algorithm for discovering clusters in large spatial databases with noise*, Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), Aug. 1996.
4. B. Everitt, *Cluster analysis*, Halsted Press, 1980.
5. S. Guha, R. Rastogi, and K. Shim. *Cure: An efficient clustering algorithm for large databases*. In Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data(SIGMOD'98), pages 73-84, Seattle, WA, June 1998.
6. S. Guha, R. Rastogi, and S. Kyuseok. *ROCK: A robust clustering algorithm for categorical attributes*. In Proceedings of ICDE'99, pp. 512-521, 1999.
7. J. Han, M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, 2000
8. A. Hinneburg, and D. A. Keim, *An Efficient Approach to Clustering in Large Multimedia Databases with Noise,* Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, (KDD98), pp. 58-65, 1998.
9. A.K. Jain, R.C. Dubes, *Algorithms for clustering data*, Prentice Hall, 1988.
10. A.K. Jain, M.N. Murty, P.J. Flynn, *Data Clustering: A Review*, ACM Computing Surveys, Vol. 31, No. 3, pp.264-323, Sep. 1999.
11. G. Karypis, E.-H. Han, and V. Kumar. *CHAMELEON: A hierarchical clustering algorithm using dynamic modeling*. COMPUTER, 32:68-75, 1999
12. D. Krznaric and C. Levcopoulos, *Fast Algorithms for Complete Linkage Clustering*, Discrete & Computational Geometry, 19:131-145, 1998.
13. Kurita, T., *An efficient agglomerative clustering algorithm using a heap*, Pattern Recognition, volume 24 (1991), number 3 pp. 205-209
14. R.T. Ng, J. Han, *Efficient and Effective Clustering Methods for Spatial Data Mining*, VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, pp.144-155, Sep. 1994.
15. M. Stonebraker, J. Frew, K. Gardels and J. Meredith, *The Sequoia 2000 Storage Benchmark,* Proceedings of SIGMOD, pp. 2 – 11, 1993.
16. W.E. Wright, *Gravitational Clustering*, Pattern Recognition, 1977, Vol.9, pp. 151-166.
17. X. Xu, M. Ester, H.-P. Kriegel, J. Sander, *A distribution-based clustering algorithm for mining in large spatial databases*, In Proceedings of 14th International Conference on Data Engineering (ICDE'98), 1998.
18. Zamir, O. and O. Etzioni (1998). *Web document clustering: A feasibility demonstration*. In Proceedings of the 21th International ACM SIGIR Conference, pp. 46--54.
19. T. Zhang, R. Ramakrishnan, M. Livny, *BIRCH: An Efficient Data Clustering Method for Very Large Databases*, Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, pp.103-114, Jun. 1996.