GRAVITATIONAL CLUSTERING: A NEW APPROACH BASED ON THE SPATIAL DISTRIBUTION OF THE POINTS

Sukhamay Kundu

Computer Science Dept, Louisiana State University
Baton Rouge, LA 70803, USA
(email: kundu@bit.csc.lsu.edu)

ABSTRACT. We describe here a new clustering method based on the notion of a force of attraction between each pair of points, and without using a "similarity" measure. The clusters are formed by allowing each point to move slowly under the resultant effect of all the forces on it, and by merging two points when they become too close to each other. This model can be regarded as a refinement of both the nearest-neighbor method and the fuzzy c-means method, and it avoids some of the pitfalls of both of those methods. It produces a complete cluster-hierarchy in $O(N^3)$ time for N points. We illustrate the method with several examples.

Keywords: Clustering, spatial distribution.

1. INTRODUCTION

The problem of clustering a set of data points $P = \{P_i, 1 \le i \le N\}$ into two or more classes occurs in many diverse applications [Ref(1)–(13)]. In particular, it is of fundamental importance in machine learning and data-mining, where a point P_i represents a collection of measurements of an object, which may be a string of text or an image or some other entity. Clustering is often used as an intermediate step in forming rules.

Almost all existing clustering methods make a direct or indirect use of an informal† measure of similarity $\sigma(P_i, P_j)$ between two points P_i and P_j, which is based on the distance $d_{ij} = dist(P_i, P_j)$. It is clear that $\sigma(P_i, P_j)$ should depend on not just d_{ij} but at least on the whole family of distances $D_i = \{d_{ik}: P_k \in P\}$ and $D_j = \{d_{jk}: P_k \in P\}$. It should also depend on the direction of the lines P_iP_k and P_jP_k to account for the spatial distribution of the points in P. The nearest-neighbor type clustering methods use only a small part of the information in D_i and D_j, but they do not use the directional information. This is a major weakness of the nearest-neighbor type methods, and it can cause clusters that are too elongated. In contrast, the fuzzy c-means method models each cluster by a single "center-point" to avoid elongated clusters. A point P_i can now "partially" belong to several clusters simultaneously. For a given number of clusters K, one determines the cluster centers and the membership of each P_i in those clusters using a certain optimality criteria. Some of the generalizations of the fuzzy c-means method involve replacing the one-point center of a cluster by a set of centers (multi-centered cluster). One can also use curves (or surfaces) of certain known types to act as "centers" [2, 3]. One drawback of the fuzzy c-means method is that one has to specify the number of clusters K to begin with. Also, the fuzzy clusters cannot be always converted to non-fuzzy (crisp) clusters in a meaningful way, without a significant deviation from the optimality. The fuzzy c-means method ignores the directional information of the lines P_iP_j although it uses all D_i's. A very different approach, based on the statistical estimation of the peaks in a noisy multi-modal data, is used in [4] for determining point-centered clusters. Unlike the fuzzy c-means and nearest-neighbor type methods, it does not use an approximate similarity measure $\sigma(P_i, P_j)$, which is also the case for our method.

The key idea in our clustering method is the notion of a force of attraction $\vec{F}_{ij}$ on P_i due to P_j, which depends both on d_{ij} and the direction of the line P_iP_j. Each point P_i is allowed to move slowly under the influence of the resultant force $\vec{F}_i$ of $\{\vec{F}_{ij}: j \ne i\}$ on P_i. If two points become very close to each other, then

† Since there is no known method for defining the values $\sigma(P_i, P_j)$ of a formal similarity measure [5, 6] that match our intuition without knowing the clusters, one is forced to use an informal or approximate form of a similarity measure.

we postulate that they are "similar", and we merge such points together to form a cluster. Note that we cannot use the force vectors $\vec{F}_i$ to obtain a formal similarity measure on the points P_i for the same reason that we cannot use the position vectors of P_i for that purpose. In our method, we do not require the number of clusters K to be preset; an appropriate value of K can be determined from the hierarchical clusters generated by our method.

2. SOME PRELIMINARY EXAMPLES

We begin with a few simple examples to give a quick overview of the clustering process in our method, including a comparison of the clusters obtained by our method with those obtained by the nearest-neighbor method and the fuzzy c-means method. We often use the short notation $P_{i\text{-}k}$ for the cluster-set $\{P_i, P_{i+1}, \cdots, P_k\}$; similarly, $P_{i,j}$ for $\{P_i, P_j\}$ and $P_{i,j,k}$ for $\{P_i, P_j, P_k\}$, etc. We refer to the original points P_i and also the points formed by merging other points (and thus representing clusters) simply as points; this should not cause any confusion.

Example 1. Consider a set of equally spaced points $P = \{P_i: 1 \le i \le 20\}$ on a line L as shown in Fig. 1. Each P_i moves along L by the resultant force $\vec{F}_i$ (defined in Section 3) on it due to the other points. We show the position $P_i^{(t)}$ of P_i at the move-step t in Fig. 1 at a height proportional to t from the line L to make the relative motions of the points easier to understand. Initially, each point moves toward the center. However, since the leftmost point P_1 moves relatively faster towards the center than its neighbor P_2, it soon becomes very close to P_2. As a result, P_2 begins to move towards P_1 and away from the center and finally it merges with P_1 to form the cluster $P_{1\text{-}2}$. A similar situation arises with P_{20} and P_{19}, and they form the cluster $P_{19\text{-}20}$. These clusters, in turn, begin to move toward the center and subsequently attract their neighboring points towards them and away from the center to form the clusters $P_{1\text{-}3}$ and $P_{18\text{-}20}$. The process continues, with the set of points as a whole always moving towards the center. Other clusters such as $P_{9\text{-}10}$ and $P_{11\text{-}12}$ are also formed in the central region in the same way. A single cluster $P_{1\text{-}20}$ is formed at the end. Note that the points P_6 and P_7 initially move closer to each other, but subsequently they are pulled away into different clusters $P_{5\text{-}6}$ and $P_{7\text{-}8}$, only to be united later again into a larger intermediate cluster $P_{1\text{-}8}$. The last non-trivial clustering phase consists of the clusters $P_{1\text{-}8}$, $P_{9\text{-}12}$, and $P_{13\text{-}20}$, and they form a natural $K = 3$ cluster-set. Since there is no clustering-phase that has just two clusters, we conclude that it is not reasonable to form $K = 2$ clusters like $P_{1\text{-}10}$ and $P_{11\text{-}20}$ from P (see Example 2). ♣

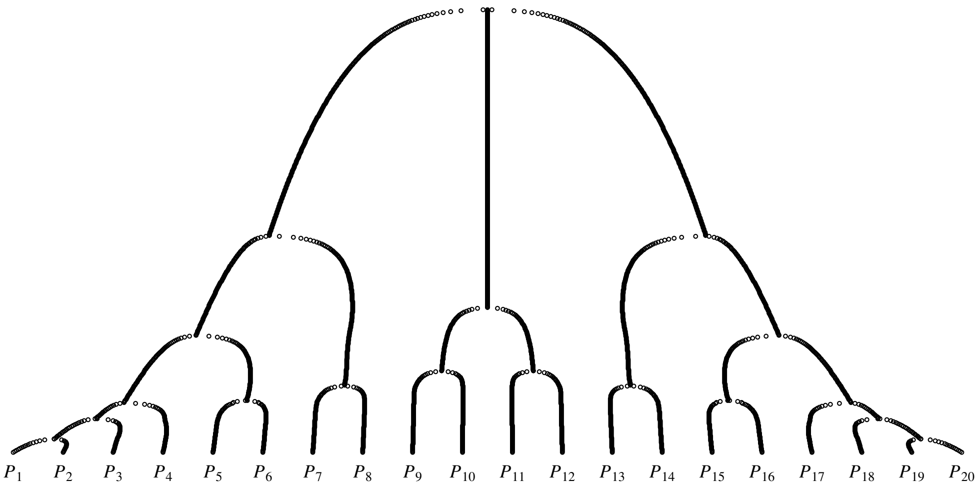


Figure 1. The formation of clusters as the points P_i move under the influence of a mutual force of attraction between pairs of points.

If we had an odd number of points in Fig. 1, the structure of the clusters remains basically the same. For $N = 21$, the final non-trivial clusters are $P_{1\text{-}8}$, $P_{9\text{-}13}$, and $P_{14\text{-}21}$ ($K = 3$), of sizes 8, 5, and 8. For $N = 30$, the situation is again similar and the final non-trivial clusters clusters have the sizes 12, 6, and 12 (K = 3). When we increase N to 40, the final non-trivial clustering-phase has K = 3 clusters (of sizes 16, 8, and 16), but this phase has now a significantly large gap from its previous clustering-phase, which has 5 clusters. This indicates that K = 5 clusters are more natural here; they are $P_{1\text{-}10}$, $P_{11\text{-}18}$, $P_{19\text{-}22}$, $P_{23\text{-}30}$, and $P_{31\text{-}40}$, with sizes 10, 8, 4, 8, and 10.

The nearest-neighbor like methods for the points in Fig. 1 gives a very large number of alternative clusterings of the form $\{P_{1\text{-}i}, P_{(i+1)\text{-}j}, P_{(j+1)\text{-}20}\}$, $1 \le i < j \le 20$, for K = 3 because of the uniform spacing of the points P_i. One of these clusterings is the one obtained by our method. On the other hand, if we apply the fuzzy c-means for K = 3 to the points in Fig. 1 and approximate the fuzzy clusters by crisp clusters, then we get the clustering $\{P_{1\text{-}7}, P_{8\text{-}13}, P_{14\text{-}20}\}$, with the cluster sizes 7, 6, and 7. This is comparable to the clusters obtained above by our method. Since the fuzzy c-means method tends to cluster points in a way that reduces the sum of variances for the clusters (with respect to the weights related to the membership values), the cluster sizes are now more uniform than those obtained by our method. Two other possible non-fuzzy solutions obtained by the fuzzy c-means method have clusters of sizes 6, 7, and 7 from the left to right, and a similar one from right to left; these clusters are clearly less desirable than the one obtained by our method.

Example 2. Consider the points in Fig. 2, obtained by throwing away a group of consecutive points in Fig. 1. Here, the clustering $\{P_{1\text{-}2}, P_{5\text{-}20}\}$ is clearly reasonable, and the final non-trivial clustering-phase in our method gives these clusters. Its previous clustering-phase gives a set of K = 3 clusters $\{P_{1\text{-}2}, P_{5\text{-}10}, P_{11\text{-}20}\}$ and the clustering-phase prior to that gives a set of 4 clusters, where the points P_5 to P_{20} are divided in a way similar to that in Fig. 1. Note that although the slight pull of the cluster $P_{1\text{-}2}$ delays the merging of the clusters $P_{5\text{-}10}$ and $P_{11\text{-}14}$ than the merging of $P_{15\text{-}20}$ and $P_{11\text{-}14}$, it does not alter the cluster-structure on the points P_5 to P_{20} (cf. Fig. 1). A close look at the gaps between successive clustering-phases, indicated by the dashed horizontal lines, suggests that K = 4 is more natural here than K = 3. ♣

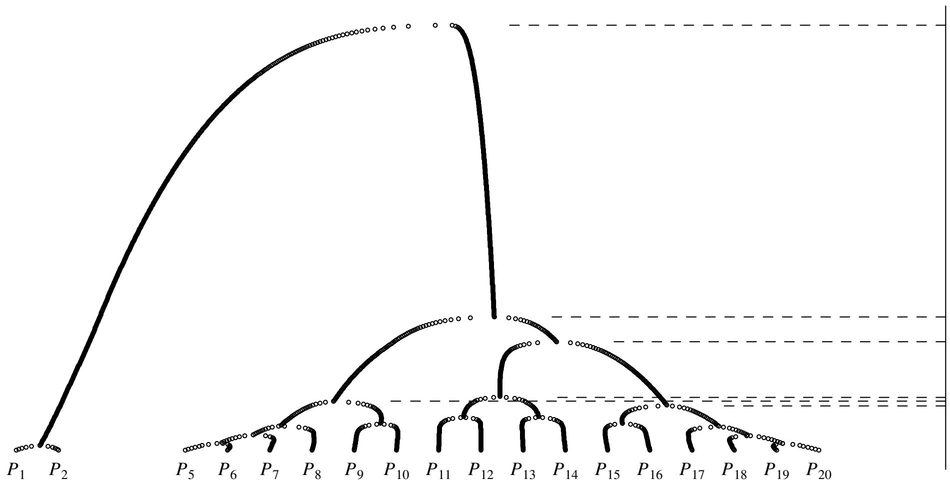
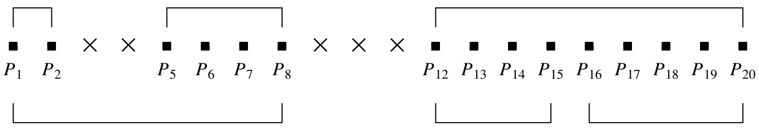


Figure 2. The formation of clusters on elimination of P_3 and P_4 in Fig. 1.

Example 3. We now modify the data-set in Fig. 2 by creating more "holes" as shown in Fig. 3. Our method gives now a natural set of K = 3 clusters $\{P_{1\text{-}2}, P_{5\text{-}8}, P_{12\text{-}20}\}$, by taking advantage of both the closeness among the points and the holes among them. If we apply the fuzzy c-means method for K = 3, then the fuzzy membership values shown in Fig. 3 clearly indicate that there is considerable difficulty in determining the clusters for P_8 and P_{16}. The associated crisp clusters have the sizes 6, 4, and 5. (The fuzzy c-means method uses a parameter $m > 1$ as the exponent of the membership values in the objective function that is minimized. One typically uses $m = 2.0$; this is also the case in Fig. 3 and elsewhere in this paper. If m is close to 1.0, then each membership value tends to be close to either 1.0 or 0.0, i.e., the fuzzy clusters essentially become crisp clusters. For the points in Fig. 3, m = 1.1 gives the same crisp clusters as those obtained by crisp approximation.) ♣

3 clusters obtained by the gravitational clustering
(each '×' indicates a point that has been thrown away)



Crisp approximation of the 3 fuzzy clusters
obtained by the fuzzy c-means method

Clusters	Membership values for 3 clusters obtained by the fuzzy c-means method														
	P_1	P_2	P_5	P_6	P_7	P_8	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}	P_{17}	P_{18}	P_{19}	P_{20}
C#1	0.9	0.9	1.0	1.0	0.8	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C#2	0.1	0.1	0.0	0.0	0.1	0.3	0.9	1.0	1.0	0.8	0.4	0.1	0.0	0.0	0.1
C#3	0.0	0.0	0.0	0.0	0.1	0.1	0.1	0.0	0.0	0.2	0.6	0.9	1.0	1.0	0.9

Figure 3. Illustration of the superior performance of the gravitational clustering method compared to the fuzzy c-means method.

In [7], we described a method which gives the optimal crisp clusters for a linear set of points (for a given K) such that the sum of variances is minimized. This method gives the same clusters as the crisp approximation of the fuzzy clusters (using the same K = 3), which indicates that the variance-based optimal clustering may not always be the right choice.

3. THE NEW METHOD

We model the attraction-force between a pair of points P_i and P_j by a variant of Newton's law of gravitation. We associate a unit mass $m_i = 1$ with each point P_i to begin with. When two or more points (clusters) are merged, the sum of their masses becomes the mass of the resulting cluster; the mass of a cluster then simply equals the number of original points P_i in it. Although each cluster is represented by a single point in the clustering operations in our algorithm, this "point" has a very different role here compared to that of the center-point of a cluster in the fuzzy c-means method.

The attraction-force $\vec{F}_{ij}$ on a unit-mass at P_i due to a unit-mass at P_j is given by (1), where d_{ij} = $dist(P_i, P_j)$ and $\mu > 0$ is a fixed constant ($\mu = 2.0$ in Newton's law of gravitation). The total force $\vec{F}_i$ on the unit-mass at P_i is then given by the weighted vector sum in (2). In particular, if the initial spatial distribution of the points relative to P_i is the same as that relative to P_j, then $\|\vec{F}_i\| = \|\vec{F}_j\|$, where $\|\cdot\|$ denotes the length of a vector.

$$\vec{F}_{ij} = \frac{1}{d_{ij}^{\mu}}\text{, in the direction } P_i \text{ to } P_j \ (\mu > 0) \tag{1}$$

$$\vec{F}_i = \sum_{j \neq i} [m_j \vec{F}_{ij}] \tag{2}$$

Several important remarks are due here. If we translate each point P_i by a fixed amount Q, i.e., replace P_i by $T(P_i) = P_i + Q$, then there is no change in $\vec{F}_i$ and in the clustering. Likewise, if we replace each P_i by $R(P_i)$, where R is a rotation operator, then $\vec{F}_i$ is replaced by $R(\vec{F}_i)$ and once again there is no change in the clustering. A similar remark holds if we use scaling, i.e., replace each P_i by cP_i ($c \neq 0$); each $\vec{F}_i$ is now multiplied by $c^{-\mu}$, and there is no change in the direction of $\vec{F}_i$ or in the clustering. If two points

P_i and P_j are very close to each other, then both $\vec{F}_i$ and $\vec{F}_j$ are very large in magnitude and they will have more or less the opposite directions towards each other. (An exceptional case is, for example, when the points are uniformly spread on the boundary of a small circle.) The points P_i and P_j then become even closer when they move, and we ultimately merge them into a cluster. Note that in this case P_i and P_j look very dissimilar in terms of $\vec{F}_i$ and $\vec{F}_j$, but the differences in $\vec{F}_i$ and $\vec{F}_j$ are overcome by the physical proximity of P_i and P_j, which leads to their merging together into a cluster. The underlying notion of similarity in our clustering method is thus determined by the global motions of all the points, which also take into account the approximate notion of similarity based on the nearness of the points when the points are very close to each other.

3.1. The Role of μ

For μ close to 0, each $\|\vec{F}_{ij}\|$ is close to 1 and therefore $\vec{F}_i$ is determined mostly by the directions of the lines P_iP_j from P_i, with the distances d_{ij} having comparatively a less important role. Fig. 4(i) shows the clustering process for the points $P - \{P_3, P_4\}$ as in Fig. 2 using $\mu = 0.001$. Note that there are very few clustering phases here. The first set of clusters $P_{1\text{-}2}$, $P_{5\text{-}12}$ (which consists of the left half of the points P_5 to P_{20}), and $P_{13\text{-}20}$ are formed at the same clustering phase and in the very next step the last two clusters are merged into a single cluster $P_{5\text{-}20}$. Finally, all points are merged into a single cluster, as is always the case. In general, the clustering-phases spread out more and more as μ is increased. For example, if μ close to 1, the clusters are formed primarily at the extremities. For large μ, the points P_j which are further away from P_i have a smaller impact on $\vec{F}_i$, and therefore both the value and the direction of $\vec{F}_i$ are determined mostly by the nearest neighbor(s) of P_i and hence both the clustering process and the shape of the clusters can be quite different now. This is illustrated in Fig. 4(ii), where the clusters are spatially more spread out compared to those in Fig. 1, which is based on $\mu = 2.0$ (as is the case in Figs. 2 and 3).

3.2. Learning Rate

A *move-step* in our clustering process consists of computing the current position of each P_i according to (3), where η is a small constant > 0, called the *learning rate*. We imagine here that the points are moving (say, in a very viscous medium) with negligible velocities. Since both the direction and the magnitude of each $\vec{F}_i$ change as the points move, a smaller value of η gives a closer approximation of the actual path followed by each P_i. This, however, requires a larger number of move-steps to form the clusters. Thus we keep $\eta \geq \eta_0$ (> 0) so that the clustering process terminates in a reasonable number of move-steps; η_0 is called the *minimum learning rate*. The actual value of η is adjusted in each move-step as described in the next two sections. Note that replacing η by η/c^{μ} is equivalent to the scaling of the points by the factor $c > 0$.

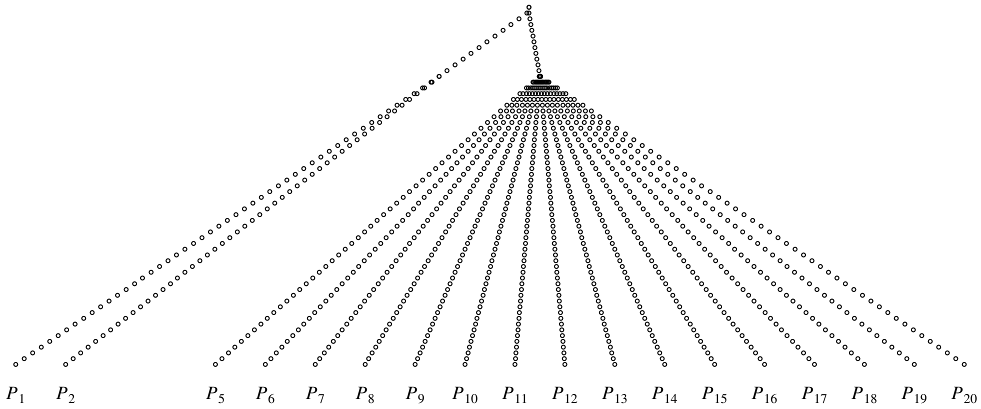
$$P_i^{new}(\eta) = P_i^{old} + \eta\vec{F}_i \tag{3}$$

3.3. Merging of Points

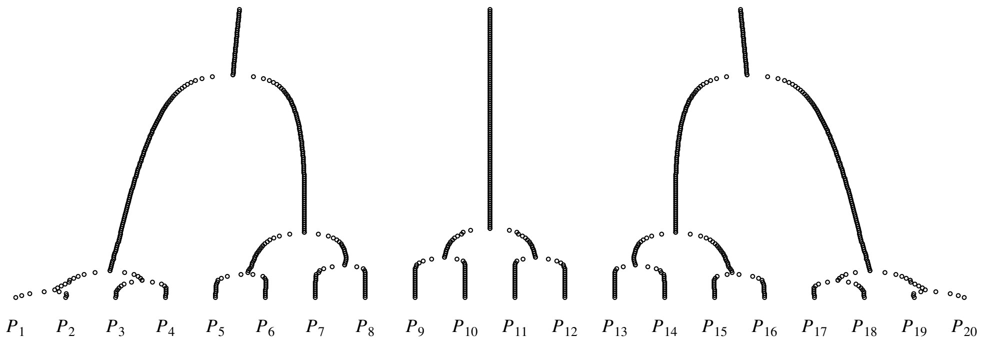
There are two ways in which two points P_i and P_j may get merged. The simplest case is when the distance d_{ij} between P_i and P_j falls below a certain threshold $\delta > 0$. In particular, if $d_{ij} < \delta$ and $d_{jk} < \delta$, then we merge all of P_i, P_j, and P_k into a single point even if $d_{ik} > \delta$. It is for this reason that in Fig. 4(i) all 8 points P_5 to P_{12} (and likewise for the points P_{13} to P_{20}) are merged simultaneously into a single point. The set of points to be merged at a particular clustering phase is determined as follows. If M_{dist} is the matrix defined below and M^*_{dist} is its transitive closure, then each equivalence class of M^*_{dist} corresponds to a set of points which are merged into a single point forming a cluster.

$$M_{dist}[i, j] \begin{array}{l} = 1, \text{ if } d_{ij} < \delta\, (d_{ii} = 0), \text{ and} \\ = 0, \text{ otherwise.} \end{array}$$

The second kind of merging occurs when one point "crosses" another point in a move-step in the following sense. In Fig. 5, we show two different situations involving the relative positions of P_i and P_j before and after a move-step for a certain η. In each case, the projection of P_j in the direction of the force $\vec{F}_i$ at P_i lies to the right of P_i before the move, and after the move the projection of new P_j lies to the left of new P_i. We say that P_i has *crossed* P_j. The projections of P_i on $\vec{F}_j$ at P_j for the case on the leftside in Fig. 5 do not indicate a crossing although for the case on the rightside in Fig. 5 they indicate a crossing. The



(i) Clustering of 18 points as in Fig. 2 for $\mu = 0.001$.



(ii) Clustering of 20 points as in Fig. 1 for $\mu = 3.0$; shown here are all but the last clustering phase where the clusters $P_{1\text{-}8}$, $P_{9\text{-}12}$, and $P_{13\text{-}20}$ are merged into one cluster.

Figure 4. Illustration of the effect of μ on the nature of clusters and the clustering process.

crossing relationship is therefore not symmetric, in general. One can also show that the crossing relationship need not be transitive. The crossing between P_i and P_j clearly depends on: (1) the relative positions of P_i and P_j, (2) the directions and magnitudes of the forces $\vec{F}_i$ and $\vec{F}_j$, and (3) the minimum learning-rate η_0. The simplest way of avoiding a crossing is to use a sufficiently small η; however, this may not be achievable because $\eta \geq \eta_0$.

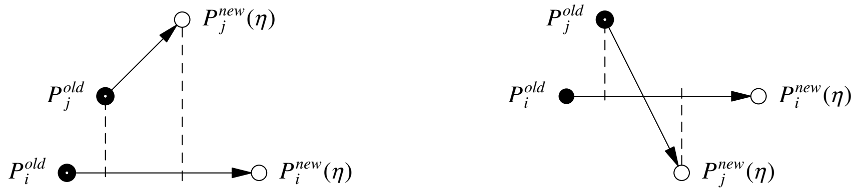


Figure 5. Illustration of P_i crossing P_j due to a move-step.

If we define the matrix M_{cross} as shown below and let M^*_{cross} be its transitive closure, then each equivalence class of M^*_{cross} corresponds to a set of points which are merged into a single point due to the crossing.

$$M_{cross}[i, j] \begin{array}{l} = 1, \text{ if } i = j \text{ or one of } P_i \text{ and } P_j \text{ crosses the other for } \eta \geq \eta_0, \text{ and} \\ = 0, \text{ otherwise.} \end{array}$$

The crossing between a pair of points is typically caused by their closeness (i.e., d_{ij} being small). In particular, a small value of d_{ij} can make $\vec{F}_i$ large and this may cause crossings between P_i and many other P_k's even when d_{ik} is not so small. To avoid unnecessary merging, we take several precautionary measures, which have proven useful in our experiments. For each P_i, we test for crossing between P_i and one of its nearest neighbor P_j (which is selected arbitrarily, if there are more than one nearest neighbors) if $d(P_i, P_j) \geq \delta$. We define the matrix M by (4.1) in which we combine the information related to M_{cross} and M_{dist}.

$$M[i, j] = 1 = M[j, i] \quad \text{if} \begin{cases} P_j \text{ is a nearest neighbor of } P_i, \text{ and} \\ d(P_i, P_j) < \delta \text{ or} \\ \text{there is crossing between } P_i \text{ and } P_j \end{cases} \tag{4.1}$$

In particular, if the distance of P_i to its nearest neighbor is $< \delta$, then we do not need to compute the total force $\vec{F}_i$ at P_i and do not need to perform the test for P_i crossing some other point. Based on the transitive closure M^*, we perform merging and define the coordinates of a merged point to be the centroid of the points merged as shown in (4.2). We then repeat the process until there is no more merging. This repeated merge-operation has the effect of merging all points P_i of a set in which each point is close to some point in the set into a single cluster, although we process only one of the nearest neighbor of each P_i in any iteration. On the other hand, it avoids merging two points P_i and P_j which are far apart and yet have a crossing relationship between them, which may happen if $\vec{F}_i$ and $\vec{F}_j$ are nearly perpendicular.

$$P_{merged} = \frac{\sum [m_i P_i]}{\sum m_i} \quad \text{(both sums taken over the of points } P_i \text{ merged)} \tag{4.2}$$

3.4. Adjusting the Learning Rate

When there is no crossing between any pair of points, we define for each P_i its *safe-learning* rate η_i ($\geq \eta_0$) as the maximum η that prevents crossings between P_i and any other P_j. The overall safe-learning rate is then given by (5). We choose the final learning rate for the current move-step, following the merging operations (if any), according to (6) and each P_i is now moved according to (3) using $\eta = \eta_{curr}$. The constant f in (6) is called the learning-rate *adjustment factor*. A simple alternative to (6) is (7), where $\eta_{\max}$ is a fixed upper limit on the learning rate. The examples presented in this paper are based on the scheme (7) for simplicity. We sometimes refer to a move-step as a *clustering phase* if there are one or more merging operations before the points are moved. The algorithm MOVE summarizes the steps described above. It also computes the height $h(P_i)$ of each point (cluster) P_i to be the sum of all η_{curr} for all previous and the current move-step. In Section 3.5, we shall use these heights in determining the appropriate number K of clusters.

$$\eta_{safe} = \min\{\eta_i\} \text{ (over all } i) \tag{5}$$

$$\eta_{curr} = (1 - f)\eta_0 + f\eta_{safe} \quad (0 < f < 1) \tag{6}$$

$$\eta_{curr} = \min\{\eta_{safe}, \eta_{\max}\} \tag{7}$$

Algorithm MOVE:

Input: The coordinates of the points P_i, the mass m_i at P_i, the height $h(P_i)$ for each P_i, the minimum learning-rate η_0, and the maximum learning-rate $\eta_{\max}$.

Output: The next position of the points, including formation of one or more clusters by merging disjoint subsets of points into single points each, if necessary, and the heights $h(P_i)$ of each new point (all P_i have the same height).

1. Initialize the matrix $M = I$ (the identity matrix).
2. For (each P_i) do the following:
   (a) Find a point P_j, $j \neq i$, which is at the shortest distance from it.
   (b) If $d(P_i, P_j) \geq \delta$, then determine the total force $\vec{F}_i$ on P_i.
   (c) Let $M[i, j] = M[j, i] = 1$ if the conditions in (4) hold.
3. If $(M \neq I)$, then compute the transitive closure M^* and perform merging based on the equivalence classes of M^* and go back to step (2).
4. Otherwise, each point P_i which is the result of merging two or more points in the iterations of step (2) corresponds to a new cluster in the cluster hierarchy; in that case $h(P_i)$ is simply the height of any of the points in the cluster. Compute η_i for each P_i and compute η_{curr} by (5) and (7).
5. For each point P_i, compute its new position $P_i + \eta_{curr}\vec{F}_i$ and let $h(P_i) = h(P_i) + \eta_{curr}$. ♣

Theorem 1. The total time to construct the complete cluster hierarchy by repeated application of the algorithm MOVE is $O(N^3)$.

Proof. Each repetition of step (2) reduces the number of points by at least one. Also, each point moves at most a constant number of times, depending on δ, the diameter = $\max\{d_{ij}: 1 \leq i < j \leq N\}$ of the initial point set P, η_0, and $\eta_{\max}$ before it is merged with some other point. It follows that the total number of applications of algorithm MOVE is at most $O(N)$ and the total number of repetitions of step (2) in all those applications of MOVE is also $O(N)$. For each P_i, it takes $O(N)$ time to compute $\vec{F}_i$ and a nearest neighbor of it, and $O(N)$ time for determining the crossing relationship between P_i and all other P_j. Each repetition of step (2) thus takes $O(N^2)$ time. The equivalence classes of M^* are simply the connected components of the underlying undirected graph on the points P_i corresponding to M, and hence they can be computed in $O(N^2)$ time [8]. The steps (4) takes $O(N^2)$ time and step (5) takes $O(N)$ time. This proves the theorem. ♣

3.5. Hierarchical Clusters

The clusters obtained by our method form a hierarchical tree structure. This tree structure is clear for Figs. 1-2 and also for Fig. 4 (which do not show the root node). A node in the cluster hierarchy may have an arbitrary number of children. The nodes at the same tree-level, i.e., having the same number of intermediate nodes on the path from the root to that node, need not correspond to clusters formed at the same move-step or clustering phase. A simple method of determining an appropriate number of clusters is to first determine an appropriate clustering-phase and then select the clusters in that phase. We can select the appropriate clustering-phase as follows. If there is a large change in height between the *ith* and the $(i+1)$*th* clustering-phases, then the *ith* clustering-phase is a possible good choice. A more precise criteria is to consider the *gain ratio* defined by (8). We select the clustering-phase (other than the last one) that has the maximum gain ratio or has a sudden increase in the gain ratio.

$$\text{gain ratio of a clustering phase} = \frac{\text{height of the clustering phase}}{\text{total number of clusters (size} \geq 1\text{) in that phase}} \tag{8}$$

An alternative method is to select individual clusters one by one, which may come from different clustering-phases. For this purpose, we define the gain-ratio of a cluster by (9). We compute the gain-ratio for each cluster in the hierarchy in a top-down fashion and select a cluster as soon as its gain-ratio falls below the gain-ratio of the root by a certain factor g $(0 < g < 1)$, i.e., gain-ratio(cluster) $\leq g$(gain-ratio of the root node). In this way, we can assure that each cluster in the set has a reasonable gain-ratio.

$$\text{gain ratio of a cluster} = \frac{\text{height of the cluster}}{\text{size of the cluster}} \tag{9}$$

Since a large value of η_0 forces all the points P_i to be merged into a single node (= root of the hierarchy) in a few iterations, it gives a short cluster hierarchy, with only few intermediate nodes in the hierarchy and hence little room for choosing an appropriate clustering-phase (or individual clusters). A good strategy is to choose a small value of η_0 such as 0.01 and let $\eta_{\max} = 10\eta_0$ or choose an f in the range $5 \leq f \leq 10$.

One can also use the following traditional method for evaluating a clustering-phase. For a cluster C_k, let q_k be its centroid and let $V(C_k)$ be the variance of $dist(P_i, q_k)$ for $P_i \in C_k$. Then, $\sum V(C_k)$, summed over all clusters C_k in a clustering-phase, is taken to be its evaluation. We can now select the proper clustering-phase which maximizes a criteria similar to (8), with height replaced by $\sum V(C_k)$. Note that both height and $V(C_k)$ increase in value as we go up in the cluster hierarchy. Alternatively, one can select individual clusters C_k from the hierarchy which have the largest $V(C_k) \leq gV(P)$.

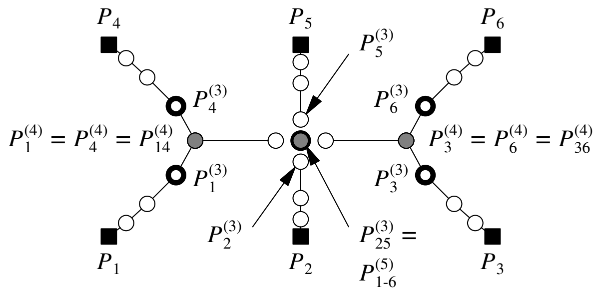
4. SOME COMPLEX EXAMPLES

Example 4. Fig. 6(i) shows the initial positions of 6 points $\{P_1, P_2, \cdots, P_6\}$, indicated by solid squares; the small circles show their successive positions $P_i^{(t)}$ at iterations $t = 1, 2, \cdots$, based on $\mu = 2$, $\eta_0 = 0.01$, $\eta_{\max} = 0.05$, and the scheme (7) for choosing η_{curr} in each move-step. We show a line joining $P_i^{(t)}$ and $P_i^{(t+1)}$ except when the circles for them are overlapping. If $P_i^{(t)}$ is merged with one or more other $P_j^{(t)}$, then we do not show the a line joining $P_i^{(t)}$ to the merged point. (The same convention is used in Fig. 7(i).) Each of the first three iterations $t = 1$, 2, and 3 has $\eta_{curr} = \eta_{\max}$. The positions $P_2^{(3)}$ and $P_5^{(3)}$ are then merged based on the crossing-property, and they are replaced by the point $P_{25}^{(3)}$ (shown as a shaded circle) at the centroid of $P_2^{(3)}$ and $P_5^{(3)}$. The height of the cluster $\{P_2, P_5\}$ is therefore 0.05+0.05+0.05 = 0.15, as indicated in Fig. 6(ii). The positions of the remaining four points in this clustering-phase are shown as thick circles. In the next move-step, we have $\eta_{curr} = 0.02$; here, $P_1^{(4)}$ and $P_4^{(4)}$ coincide at the point shown as $P_{14}^{(4)}$ and they are merged into a cluster, and a similar situation arises with $P_3^{(4)}$ and $P_6^{(4)}$. The height of both $P_{14}^{(4)}$ and $P_{36}^{(4)}$ is therefore 0.17. The point $P_{25}^{(3)}$ does not move, i.e., $P_{25}^{(3)} = P_{25}^{(4)}$. In the next move-step, we have $\eta_{curr} = 0.05$ and each of $P_{14}^{(5)}$ and $P_{36}^{(5)}$ becomes too close to $P_{25}^{(5)}$, causing all three of them to be merged together into a single point $P_{1\text{-}6}$. The height of $P_{1\text{-}6}$ is therefore 0.22. The set $\{P_{1,4}, P_{2,5}, P_{3,6}\}$ corresponding to the height 0.17 gives a satisfactory clustering of P. We get the same clusters based on the gain-ratio (8). Note that there is no clustering-phase corresponding to the hypothetical clusters $P_{1\text{-}3}$ and $P_{4\text{-}6}$ (cf. Example 5).

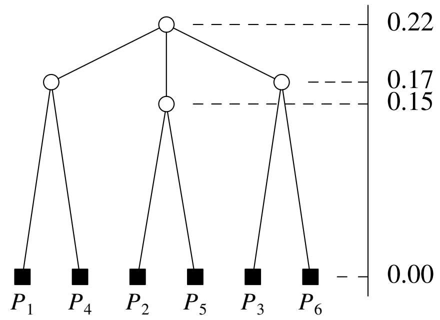
Interestingly, the fuzzy c-means method does not provide a good set of two clusters for the points in Fig. 6(i) either. The membership values shown in Fig. 6(iii) for $K = 2$ indicate a difficulty in clustering P_2 and P_5. (These membership values arise even if we initialize the cluster centers at P_2 and P_5 in the fuzzy c-means method, hoping that we might get the clusters $P_{1\text{-}3}$ and $P_{4\text{-}6}$.) The above membership values strongly indicate that P_1 and P_4 should be in the same cluster, and similarly for P_3 and P_6, which is in agreement with the clusters obtained by our method. Fig. 6(iv) shows the membership values for $K = 3$ clusters obtained by the fuzzy c-means method. Even here considerable difficulty arises in classifying P_1, P_3, and P_5 (and, equivalently, in classifying P_4, P_6, and P_2 for the alternative symmetric solution). The situation is no better if we initialize the cluster centers at the mid-point of P_1 and P_4, the mid-point of P_2 and P_5, etc; the same membership values as in Fig. 6(iv) are obtained. ♣

Example 5. Fig. 7 shows a slightly different spatial distribution of 6 points compared to Fig. 6(i); the vertical separation between the two rows is now one and half times the minimum distance between two points in a row. Our method gives now two clusters $P_{1\text{-}3}$ and $P_{4\text{-}6}$ as the most natural choice. The memberships values obtained by the fuzzy c-means method for $K = 2$ are, however, mostly not too close to 0 or 1 although the crisp approximation to the fuzzy clusters are the same as those obtained by our method, ♣

Example 6. Fig. 8(i) shows a complex set of 14 points and its clustering-phases for $\mu = 2$, $\eta_0 = 0.01$, and $\eta_{\max} = 0.05$. The individual moves of the points are now too complex to be shown here. There are 6 clustering phases in all and we indicate the clusters in these phases by different shaded areas. As seen from the left half of the table in Fig. 8(ii), the large increase in height from 0.614 to 1.376 between the last two clustering phases indicates that $K = 3$ clusters is the most appropriate, with the clusters $P_{1\text{-}2}$, $P_{3\text{-}12}$, and $P_{13\text{-}14}$. The gain-ratio (8) also gives the same choice of the clusters. Note that the points P_3 and P_{12} do not get merged into the cluster $P_{3\text{-}12}$ until very late, indicating the difficulty in determining their grouping. This is also partly supported by the membership values shown in Fig. 8(iii), which are obtained by the fuzzy c-means method for $K = 3$. Interestingly, the membership values for P_4 and P_{11} indicate even more difficulty in their classification although these points are put in a single cluster quite early in our method. Our method merges P_3 with $P_{4\text{-}11}$ instead of with $P_{1\text{-}2}$, and this is more reasonable because although the distance between P_3 and its nearest point in $P_{1\text{-}2}$ is the same as that between P_3 and its nearest neighbor in $P_{4\text{-}11}$, the distance to the second nearest point in $P_{4\text{-}11}$ is smaller than that for $P_{1\text{-}2}$. A similar situation arises for P_{12}. The right half of the table in Fig. 8(ii) shows a slightly different clustering phases for the larger value $\eta_{\max} =$



(i) The positions of the points P_i, $1 \le i \le 6$, in the successive move-steps.



(ii) The hierarchy of resulting clusters and their heights.

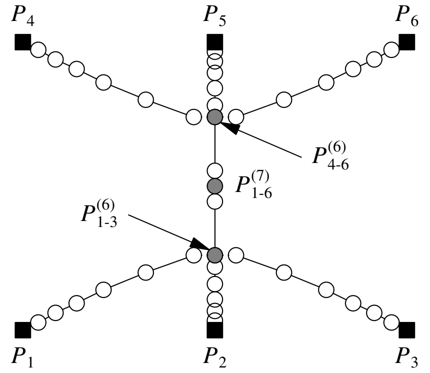
Clusters	P_1	P_2	P_3	P_4	P_5	P_6
C#1	0.92	0.50	0.08	0.92	0.50	0.08
C#2	0.08	0.50	0.92	0.08	0.50	0.92

(iii) Membership values for $K = 2$ clusters obtained by the fuzzy c-means method.

Clusters	P_1	P_2	P_3	P_4	P_5	P_6
C#1	0.61	0.01	0.08	0.93	0.32	0.02
C#2	0.08	0.01	0.60	0.02	0.32	0.94
C#3	0.31	0.98	0.32	0.04	0.36	0.04

(iv) Membership values for $K = 3$ clusters obtained by the fuzzy c-means method.

Figure 6. The move-steps for a set of points $P = \{P_i;\ 1 \le i \le 6\}$, its hierarchical cluster-tree, and a comparison with the clusters obtained by the fuzzy c-means method.



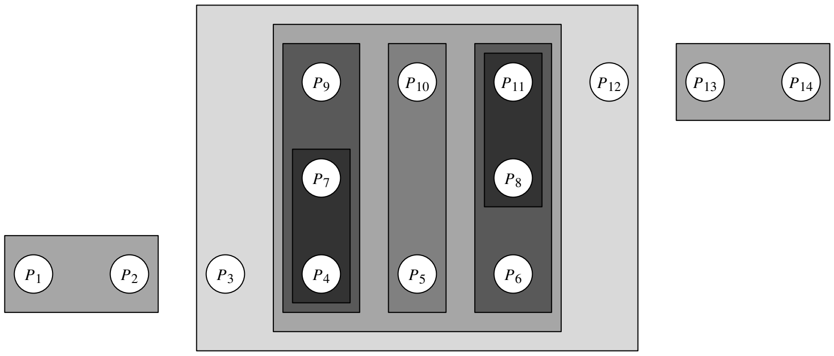
(i) The positions of the points P_i, $1 \le i \le 6$, in the successive move-steps.

Clusters	P_1	P_2	P_3	P_4	P_5	P_6
C#1	0.76	1.00	0.74	0.26	0.00	0.24
C#2	0.24	0.00	0.26	0.74	1.00	0.76

(ii) The membership values for $K = 2$ clusters obtained by the fuzzy c-means method.

Figure 7. A slight variation of the data set used in Fig. 6, giving $K = 2$ natural clusters.

0.1, giving the same set of clusters as before and providing additional support for that solution. (If we use a smaller value of the parameter m close to 1, say, $m = 1.1$, then the fuzzy clusters become almost crisp clusters, with each membership value > 0.99 or < 0.01; the associated crisp clusters in this case are given by $P_{1\text{-}3}$, $P_{4\text{-}11}$, and $P_{12\text{-}14}$.) It is clear that there is no natural way of partitioning the points into just 2 clusters. Note that $\eta_{max} = 1.0$ tends to give a larger height of a cluster compared to that for $\eta_{max} = 0.05$. This is because a larger $\eta_{\max}$ makes a more crude approximation to the movements of the points P_i by moving them in larger steps of $\eta = \eta_{curr}$ using a smaller value of the current force $\vec{F}_i$. ♣



(i) The hierarchical clusters for $\mu = 2$, η_0 = 0.01, and $\eta_{\max} = 0.05$

Clustering phase	Move-step (height)	Clusters of size > 1 for $\eta_{\max} = 0.05$	Move-step (height)	Clusters of size > 1 for $\eta_{\max} = 0.10$
1.	5 (0.229)	$P_{4,7}$, $P_{8,11}$	1 (0.100)	$P_{5\text{-}6}$, $P_{9\text{-}10}$
2.	6 (0.279)	$P_{4,7,9}$, $P_{6,8,11}$	3 (0.258)	$P_{4,7}$, $P_{8,11}$, $P_{5\text{-}6}$, $P_{9\text{-}10}$
3.	7 (0.299)	$P_{5,10}$, $P_{4,7,9}$, $P_{6,8,11}$	4 (0.355)	$P_{4,7,9,10}$, $P_{5,6,8,11}$
4.	8 (0.314)	$P_{1\text{-}2}$, $P_{4\text{-}11}$, $P_{13\text{-}14}$	5 (0.369)	$P_{1\text{-}2}$, $P_{4\text{-}11}$, $P_{13\text{-}14}$
5.	14 (0.614)	$P_{1\text{-}2}$, $P_{3\text{-}12}$, $P_{13\text{-}14}$	8 (0.667)	$P_{1\text{-}2}$, $P_{3\text{-}12}$, $P_{13\text{-}14}$
6.	30 (1.376)	$P_{1\text{-}14}$	16 (1.410)	$P_{1\text{-}14}$

(ii) Comparison of the clustering-phases for $\eta_{\max} = 0.05$ and $\eta_{\max} = 0.10$;
shown here are only the clusters of size ≥ 2.

Clusters	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}
C#1	0.908	0.997	0.849	0.340	0.101	0.095	0.175	0.049	0.194	0.069	0.064	0.023	0.001	0.023
C#2	0.069	0.002	0.128	0.596	0.830	0.709	0.776	0.771	0.710	0.828	0.590	0.124	0.002	0.070
C#3	0.023	0.001	0.023	0.063	0.069	0.196	0.049	0.180	0.095	0.102	0.346	0.853	0.997	0.907

(iii) Membership values for $K = 3$ clusters obtained by the fuzzy c-means method.

Figure 8. The gravitational clustering of a complex set of 14 points; both $\eta_{\max} = 0.05$ and $\eta_{\max} = 0.10$ give the same choice of 3 clusters having comparable heights.

We remark that the points P_1 and P_{14} in Fig. 8 are similar to each other in the sense that the spatial distribution of the other points relative to them are identical. The same remark holds for P_4 and P_{11}. However, since the distance between P_1 and P_{14} is significantly larger than that between P_4 and P_{11}, the latter pair of points are more similar to each other than the other pair. In our method, P_4 and P_{11} are thus brought into the same cluster long before the points P_1 and P_{14} (namely, at the very end).

Example 7. Our final example in Fig. 9 shows 40 uniformly spaced points on the boundary of a square and the move-steps for $\mu = 2.0$. Here, the most suitable number of clusters is 8. ♣

If one wants to regard "boundary" or "exterior" points differently from the other points so that they become part of different clusters (e.g., forcing P_1 and P_{20} in Fig. 1 to be in the same cluster), then we may

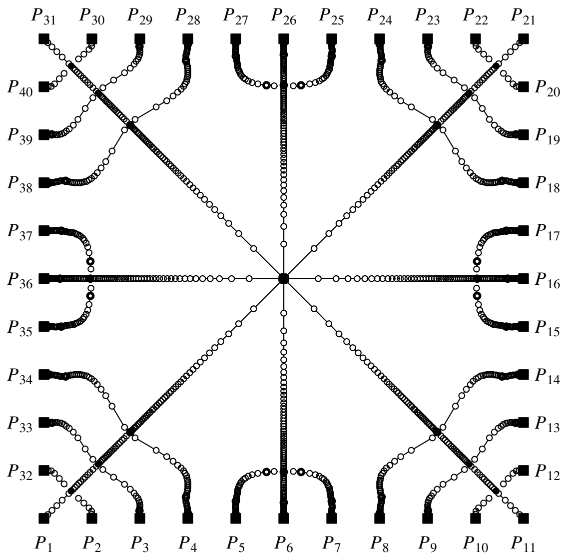


Figure 9. Clustering of 40 uniformly spaced points on the boundary of a square by the gravitational clustering method, giving 8 natural clusters.

first map each P_i to a new point P'_i whose coordinates consist of the distances $\langle d_{ij}\colon\ j \geq 1\rangle$ sorted in increasing order, say. To keep the dimensionality ($= N$) of the points P'_i small, one may keep only the first few and the last few coordinates of P'_i, and perhaps using the average and the variance of d_{ij}'s as two additional coordinates to compensate for the coordinates that are eliminated. One can also use the original coordinates of P_i as additional coordinates in P_i.

5. CLASSIFICATION OF A NEW POINT

Given a clustering $\{C_1, C_2, \cdots, C_K\}$ of a set of points $P = \{P_i\colon 1 \leq i \leq N\}$, we can classify a new point P_0 based on the forces $\vec{G}_j$ defined by (10) by selecting the cluster which gives the maximum magnitude $\|\vec{G}_j\|$. In case of a tie, we may break the tie in favor of the cluster which has the smaller num of points. Any remaining ties may be broken arbitrarily. If the sum in (10) is replaced by $\max\{|\vec{F}_{0i}|\colon P_i \in C_j\}$, then this amounts to classifying P_0 according to the nearest neighbor.

$$\vec{G}_j = \sum_{P_i \in C_j} \vec{F}_{0i} \tag{10}$$

6. GENERALIZATIONS

We can generalize our method in several ways. The simplest generalization is to replace $1/d_{ij}^{\mu}$ in (1) by a decreasing function $\delta(d_{ij})$ of d_{ij}. One can also modify the role of m_j in (2); e.g., if we want to discourage formation of clusters of size $> M$, then we can replace m_j by M/m_j (or some power of it) when $m_j > M$.

A different kind of generalization is obtained when we introduce a force of *repulsion* between certain pairs of points. Suppose that P consists of two types of points P_i and P'_j, where the points P_i of the first type are not to be clustered with points P'_j of the second type even if they are very close to each other. In particular, each cluster should be homogeneous and consist entirely of one type of points. Such a situation arises in supervised learning involving multiple types of data, where the clusters are used to form "disjoint" rules for each type of points. Here, we must cluster the different types of points simultaneously, and we should not cluster the points of each type independently. This can be done by introducing a repulsive force between point of different types. The form of the repulsive and attractive force may be different. A detailed study of clustering by mixing the attractive and repulsive forces is left to a future work.

7. CONCLUSION

We have presented here a new clustering method which accounts for the spatial distribution of the given points more accurately than either of the fuzzy c-means method and the nearest-neighbor method, which are the two most commonly used clustering methods. Unlike the fuzzy c-means method, our method produces crisp, non-fuzzy clusters, and a complete cluster-hierarchy. The proper number of clusters to be formed can be determined directly from the cluster-hierarchy. Our method performed better than both the fuzzy c-means method and the nearest-neighbor method in all of the cases we examined. However, if reducing the total cluster-variance is a major objective in the clustering, then the fuzzy c-means is the right choice. One important strength of our method is its possible adaptability in a wide variety of situations, where we may need to control the size of the clusters and where we have a mixture of different types of points such that one type of points are considered "unfit" for grouping with the other types of points.

8. REFERENCES

1. Everitt, B., *Cluster analysis* (2nd ed.), 1980, Halsted Press (John Wiley & Sons, Inc., New York).
2. Hathaway, R.J. and Bezdek, J.C., Switching regression models and fuzzy clustering, *IEEE Transaction on Fuzzy Systems,* 1(1993), pp. 195-204.
3. Kundu, S. and Chen, J., Fuzzy linear invariant clusters with applications to fuzzy control, *Proc. of NASA/NAFIPS,* San Antonio, TX, 1994, pp 196-200.
4. Wilson, R. and Spann, M., A new approach to clustering, *Pattern Recognition,* 23(1990), pp. 1413-1425.
5. Klir, G. and Youn, B., *Fuzzy sets and fuzzy logic,* 1995, Prentice Hall, NJ.
6. Novak, V., Fuzzy sets and applications, 1989, Adam Hilger Bristol.
7. Kundu, S., A solution to the histogram equalization and other related problems by shortest path method, *Pattern Recognition,* (in press).
8. Aho, A.V., Hopcroft, J.E., and Ullman, J.D., *THe design and analysis of algorithms,* 1974, Addison-Wesley, Reading Mass.
9. Jambu, M. and Lebeaux, M.-O., *Cluster analysis and data analysis,* 1983, North-Holland Publ., New York.
10. Miltenburg, J. and Zhang, W., A comparative evaluation of the well-known algorithms for solving the cell-formation problem in group technology, *J. of Operations Management,* 10(1991), pp. 44-72.
11. Raz, T. and Yaung, A., Application of clustering techniques in information systems design, *Information and Software Technology,* 37(1995), pp. 145-154.
12. Seifoddini, H. and Hsu, C.P., A comparative study of similarity coefficient and clustering algorithms, *J. Manufacturing Systems,* 13(1994), pp. 119-127.
13. Wilbur, W.J. and Coffee, L., The effectiveness of document neighboring in search enhancement, *Information Processing and Management,* 30(1994), pp. 253-266.