A Generalized Affinity Propagation Clustering Algorithm for Nonspherical Cluster Discovery

Teng Qiu,   Yongjie Li*

Key Laboratory of NeuroInformation, Ministry of Education of China, School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu, 610054, China
*Corresponding author. Email: liyj@uestc.edu.cn

Abstract: Clustering analysis aims to discover the underlying clusters in the data points according to their similarities. It has wide applications ranging from bioinformatics to astronomy. Here, we proposed a Generalized Affinity Propagation (G-AP) clustering algorithm. Data points are first organized in a sparsely connected in-tree (IT) structure by a physically inspired strategy. Then, additional edges are added to the IT structure for those reachable nodes. This expanded structure is subsequently trimmed by affinity propagation method. Consequently, the underlying cluster structure, with separate clusters, emerges. In contrast to other IT-based methods, G-AP is fully automatic and takes as input the pairs of similarities between data points only. Unlike affinity propagation, G-AP is capable of discovering nonspherical clusters.

1 Introduction
1.1 Physically inspired in-tree (IT) structure and several IT-based methods

In (*1*), we proposed a physically inspired method to resolve the evolutionary behavior of complex system (consisting of data points), and further used it to solve an old and fundamental problem of data clustering. In that method, each data point is viewed as a basic particle with an isotropic potential around it. Potentials from different data points can be superposed, thus forming a non-uniform potential atmosphere. Consequently, the non-uniform feature of space triggers the evolution of the point system, that is, data points tend to move from higher potential areas to lower ones (denoted as the "descending direction").

However, instead of an analytic solution for the trajectory of the moving path of each point, we devised a simple rule to mimic the moving process, that is, let each point i "descend" (referring to the descending direction) to the nearest neighbor point j. In other words, point i is linked to point j by a directed edge, for which nodes i and j are the start and end nodes of the directed edge, respectively. Consequently, the smooth trajectory of each point can be approximated by a directed path consisting of a sequence of zigzag "hops" (i.e., directed edges), similar to the case in Isomap (*2*). The descending direction in the rule guarantees the order of the network, that is, each node has only one directed path to reach the node (root node) of the globally lowest potential.

As demonstrated in (*1*), the constructed network is precisely the IT structure in graph theory. However, something special for the IT structure here is that not only the edges are weighted (referring to the similarities of the connected nodes), but also the

nodes are weighted (referring to the potential values of all nodes). This IT structure, though imperfect (due to the undesired edges across different clusters), still looks extremely attractive (fig. S1 and fig. S2), since, on one hand, the structure is generally in line with the clusters' forms, on the other hand, the undesired edges are distinguishable (thus can be easily removed) due to several salient features of them: (F1) they are usually much longer than their surroundings; (F2) they are usually started from the dense part of the clusters, in contrary to their surroundings which direct towards the dense part of the clusters; (F3) they are usually started from the nodes associated with locally lowest potentials.

According to these features, different methods, similar to the repair mechanisms as in cell replication (3)(4), were proposed in (1) to determine the undesired edges in IT structure. Based on F1, the K longest edges can be determined as the undesired edges (K-Cut). Based on F1, F2 and F3, users can interactively identify those undesired edges (Int-Cut). Based on F1, one can make a combination with semi-supervised clustering (Sup-Cut). See (5) for a more efficient semi-supervised cutting strategy. Based on F1 and F3, we can follow a method recently proposed by Rodriguez and Laio (6), that is, at first, interactively identifying as the start nodes of the undesired edges the pop-out points in a 2-dimensional scatter plot where each point is coordinated by just two features, the potential magnitude (x-axis) on it and the length (y-axis) of the edge started from it. Consequently, the undesired edges started from those points are in fact indirectly identified. Besides the above interactive way (Int-DCC-Cut), those start nodes of the undesired edges can also be determined by the K (prespecified) nodes with the largest values of the product term of the x and y coordinates in the above plot (K-DCC-Cut).

See an application for all the above IT-based clustering methods in fig. S3 and the detailed analysis of their advantages and disadvantages in (1).

1.2 Affinity propagation

Affinity propagation (**AP**) (7) aims to find a set of optimal data points (called exemplars) from the dataset to represent clusters, and consequently, the sum of the similarities (or distances) for all data points to their corresponding exemplars is maximal (or minimal).

AP takes as input two kinds of real-valued input: (i) the similarity $s(i, j)$ of any pair of data points; (ii) a real number $s(i, i)$, called "preference", for each node i. $s(i, j)$ signals how well-suited node j can be the exemplar of point i. $s(i, i)$ indicates how likely point i is to be an exemplar. The number of clusters is automatically determined by the input $s(i, i)$. Usually, a median value of the similarities is suggested to be the shared preference for all points so as to produce a moderate number of clusters.

Based on the above input, AP uses a message-passing strategy to update the network (usually a fully connected graph) until convergence. This process requires only simple and local computations, and is much more effective than k-centers clustering method. One run of AP can in some case be superior to more than 10000 runs of k-centers clustering.

Despite the above great advantages of AP over k-centers, both of them face one

inherent weakness, i.e., they are not capable of detecting the nonspherical clusters (*6*).

2 Motivation

The IT structure with potential values on all nodes, actually contains all the input for AP: (i) the potential values can serve as a priori to specify the preference of each node as an exemplar; (ii) the weights of the directed edges can specify the likelihood of each node to be the exemplar of other nodes.

Therefore, AP is expected to lead to an effectively automatic and unsupervised IT-based clustering method. In turn, due to the sparseness and effectiveness of the IT structure, the inherent weakness of AP could be solved, and the message-passing procedure of AP can be much faster to converge.

3 The proposed algorithm G-AP
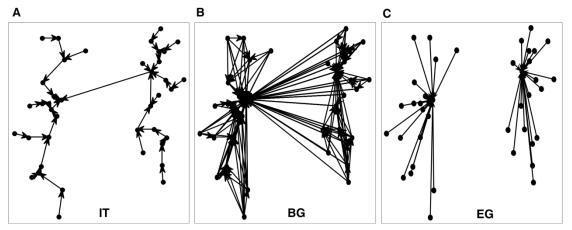
The proposed clustering algorithm consists of 3 steps:

Fig. 1. An illustration for the graphs obtained in each step. (**A**) The in-tree structure ($\sigma = 10$). (**B**) The belief graph. (**C**) The exemplar graph ($\alpha = -10$).

Step 1, construct the IT structure

First, Let the potential P at each node i be the superposition of the isotropic Gaussian potentials (negative value) exerted from all points:

$$P_i = -\sum_{j=1}^{N} e^{-\frac{d_{i,j}^2}{\sigma}} \tag{1}$$

where the parameter σ is of positive value. Then, the end node I_i of the directed edge started from any node i is defined as the nearest node among the ones with lower potentials:

$$I_i = \arg\min_{k \in K_i} d_{i,k} \tag{2}$$

where $K_i = \{k \mid P_k < P_i\}$, called the optional node set of node i. This definition for I_i is a simple version to obtain the IT structure (Fig. 1A) and in practice, a more elaborate one was given in (*1*).

Step 2, construct the belief graph

Any node i (start node) is linked with node j (end node) by a directed edge if node i can reach node j along a directed path $\Gamma_{i,j}$ in IT. This will result in a directed graph. We call it belief graph (**BG**) (Fig. 1B). The edge weight $W_{BG}(i, j)$ is defined as the sum of the lengths of all edges in $\Gamma_{i,j}$, rather than the Euclidian distance between node i and j.

Thanks to the IT structure, there is always only one directed path from any node i to its reachable node j (thus the end node j is easy to determine), in contrast to Isomap (*2*) for which a time-consuming process is required to search for an appropriate (e.g., shortest) path among several choices in an undirected K (requiring to be specified beforehand) nearest neighborhood (KNN) graph.

Step 3, identify exemplars

The similarity of nodes i and j is defined as

$$s(i, j) = e^{-\frac{W_{BG}(i,j)}{\sigma}}$$
(3)

if they are reachable; otherwise $s(i, j) = -\infty$. The initial preference $s(i, i)$ at node i is set proportional to the sum of the similarities between node i with all the nodes that can reach it. The proportional constant, denoted as α, should be a negative number (its magnitude is preferred to be larger than σ). Then, based on the input, the sparse version (*8*) of AP is used and thus messages are propagated only between the connected nodes (i.e., $s(i, j) \neq -\infty$).

Consequently, for every node i in BG, only one of the exemplar candidates (the nodes on the directed path from node i to the root node) will be automatically determined as its exemplar node, or in other words, the directed edges, started from node i yet not ended at its identified exemplar, are trimmed. We call the trimmed graph the exemplar graph (**EG**), as shown in Fig. 1C.

4 Experiments

For the datasets (*9-12*) in Fig. 2, false clustering assignments occur (Fig. 2, left column) for AP, whereas G-AP can accurately detect all those clusters (Fig. 2, right column).

Although, for AP, one can avoid the clustering error by increasing the shared preference so as to make an over-dividing (i.e., a large cluster number) of the nonspherical clusters, what are really detected by AP are still spherical clusters, since AP specializes only in detecting spherical clusters (*6*) (revealed also by the clear line-style decision margin between two close clusters).

We also applied G-AP to cluster mushrooms (*13*) (Fig. 3A). It has been showed in (*1*) that the IT structure for this dataset is insensitive to a large range value of the parameter σ from 0.001 to 1000. Here, we choose one ($\sigma = 4$) of them for instance. It shows a trend in Fig. 3B that, as the magnitude of α increases, the cluster number decreases and the error rate of the clustering assignment increases. Specifically, $\alpha = 4$ leads to a result (28 clusters at an error rate of 0.1%) very close to the underlying

cluster structure saliently revealed by Rodriguez and Laio's method (*6*) (fig. S4). Compared with AP, G-AP is superior in clustering result (Fig. 3C), and requires much less time to converge (Fig. 3D) due to the sparse feature (*14*) of the input.
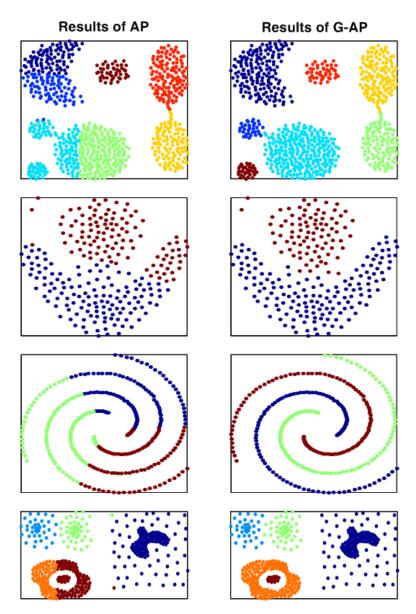

Fig. 2. A comparison between AP (left) and G-AP (right) for nonspherical cluster detection. Left column (the original dataset are directly processed by AP): from up to bottom, the shared preferences are −1600, −2000, −5000, −1000, respectively. Right column: from up to bottom, σ = 1, 1, 1, 2 (the corresponding IT structures are given in fig. S2), and α= −400, −400, −140, −10. In fact, a large range of values for α can obtain the same results in the right column, e.g. for the second dataset, the magnitude of α can vary from 4 to 4600.
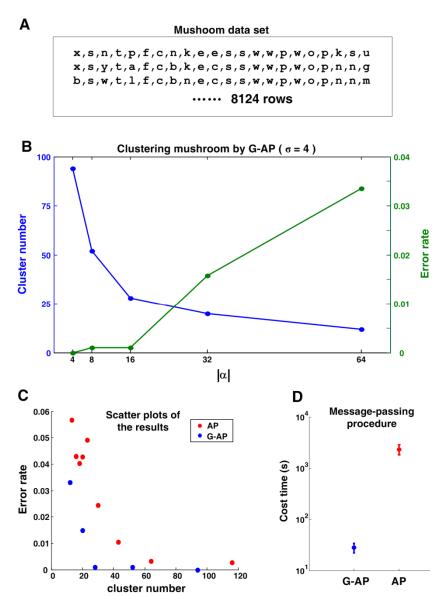
Fig. 3. Clustering mushrooms. (**A**) A small portion of mushroom dataset. Each mushroom X_i is a vector consisting of 22 characters (each row). The distance $d_{i,j}$ between any pair of mushroom X_i and X_j is measured by $\sum_m 1\{X_i^m \neq X_j^m\}$, where $1\{X_i^m \neq X_j^m\}$ equals 1 if the m-th attributes are different, else 0. (**B**) The clustering results, cluster number (blue) and error rate (green), achieved by G-AP, versus the magnitude of the parameter α. (**C**) A comparison of the scatter plot of clustering results (error rate and cluster number) between G-AP and AP. The results (blue points) of our method lies in the bottom-left of that (red points) of AP method, being more close to the origin of the coordinates, indicating that G-AP can achieve a lower false clustering assignment when the cluster numbers are comparable, or a less cluster number in the same error rate. (**D**) Comparison of the time cost for the message-passing procedure. G-AP: 28.1 ± 6.0 s (mean ± std); AP: 2342.7 ± 506.2 s (mean ± std).

5 Discussion

There are several reasons that may facilitate G-AP, even dealing with nonspherical clusters, to identify as the exemplars the start points of the undesired edges: (i) these points obtain high support or responsibility to be the exemplars of other points, since, according to F2, they are usually from the center (or dense) areas; (ii) these points send low support to other nodes to be as their exemplars, since, according to F1, the edges started from these points are generally longer; (iii) the IT structure provides us with a graph in general revealing the underlying nonlinear structure (fig. S1 and S2), which makes effective the computation of the distance in step 2; (iv) according to step 2, the candidate exemplars for each node i are effectively constrained to the nodes only on the directed path from node i to the root node, which avoids the unexpected result as shown in fig. S5.

Something strange is that, according to the preferences defined in step 3, the nodes with low potentials also have low preferences, whereas in order to make the start nodes of the undesired edges be more likely to become exemplars, theoretically speaking, the nodes with low potentials should have high preferences. A possible reason why the strange fact can lead to the good performance for G-AP may be that the similarities defined in step 3 are of positive value, in contrast to the general case (negative value).

6 References and Notes

1.   T. Qiu, K. Yang, C. Li, Y. Li, *arXiv preprint arXiv:1412.5902*, (2014).

2.   J. B. Tenenbaum, V. De Silva, J. C. Langford, *Science* **290**, 2319 (2000).

3.   T. Lindahl, R. D. Wood, *Science* **286**, 1897 (1999).

4.   The idea behind this process is to first use a simple rule which allows the emergence of error and then repair the error by a repair mechanism so as to obtain a reliable output.

5.   T. Qiu, Y. Li, *arXiv preprint arXiv:1412.7625*, (2014).

6.   A. Rodriguez, A. Laio, *Science* **344**, 1492 (2014).

7.   B. J. Frey, D. Dueck, *Science* **315**, 972 (2007).

8.   Available at http://www.psi.toronto.edu/index.php?q=affinity%20propagation

9.   A. Gionis, H. Mannila, P. Tsaparas, *ACM Trans. Knowl. Discovery Data* **1**, 4 (2007).

10.   L. Fu, E. Medico, *BMC Bioinf.* **8**, 3 (2007).

11.   H. Chang, D.-Y. Yeung, *Pattern Recognit.* **41**, 191 (2008).

12.   C. T. Zahn, *IEEE Trans. Comput.* **100**, 68 (1971).

13.   From http://archive.ics.uci.edu/ml/.

14.   The IT structure contains N − 1 directed edges (N is the number of the nodes), even sparser than the K (= 1) nearest neighbor graph (containing N undirected edges). Although this edge number in IT is slightly increased in belief graph (in 2nd step), the number of edges facing the 3rd step of G-AP is still much less than the edge number (N − 1) × N facing AP if it directly deals with the original data.
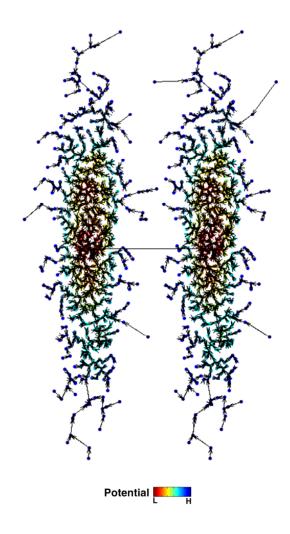
Supplementary Material: fig. S1~S5

Fig. S1 An IT structure. Different colors on nodes denote different potentials.
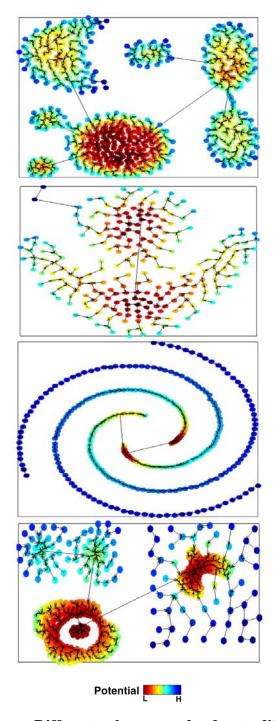
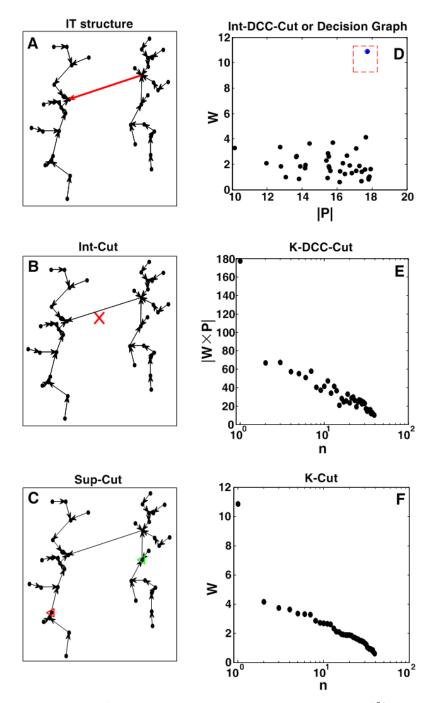**Fig. S2. IT structures. Different colors on nodes denote different potentials.**

Fig. S3. An illustration for the introduced IT-based methods. (**A**) An IT structure ($\sigma = 10$). The undesired edge (red) can be removed by any method from (B) to (F) . (**B**) Red cross records user's interactive operation, to which the closest edge will be removed. (**C**) Points surrounded by the colored triangles are the labeled data. Different colors of the triangles denote that the labeled data are of different labels. These labeled data function to supervise the cutting (in decreasing order of the edge length) procedure. This process stops when the points with different labels are in different clusters. (**D**) The scatter plot of all points, each featured by two variables: its potential magnitude $|P|$ and the length W of the edge started from it. One point saliently pops out, corresponding to the start node of the red edge in (A). (**E** and **F**) The plot of $|W \times P|$ and W (in decreasing order) of all points, respectively.
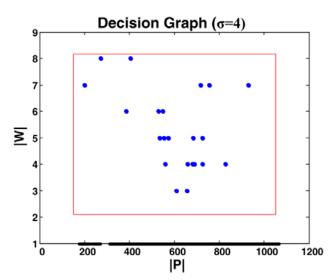
Fig. S4. The "Decision graph" of mushroom dataset. Instead of using the method proposed by Rodriguez and Laio, here we use a variant of it, i.e., the Int-DCC-Cut (see also Fig. S3D), by using just two variables in our in-tree structure, the magnitude of the potential P and the length W of the edge started from each point. The 22 points (blue points) popping out in this scatter plot are identified by user's interactive choice (red box). Consequently, the edges started from them will be removed and 23 clusters are thus obtained. No mushroom is falsely assigned. Since this decision graph is so salient, it is very likely that this dataset contains 23 clusters, although they are further classified into two classes (edible and poisonous) by human's favor.
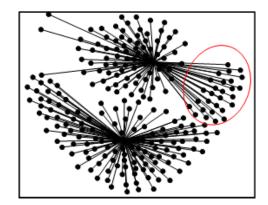

Fig. S5. A bad result of AP (shared preference = −2000). The points in red ellipse take a point in other cluster as their exemplars, whereas, according to the in-tree structure of this dataset in Fig. S2 and the rule in step 2, this case will never happen for the proposed G-AP method.