JOURNAL OF INDUSTRIAL AND
MANAGEMENT OPTIMIZATION
Volume **3**, Number **4**, November **2007**

Website: http://AIMsciences.org

pp. **619–624**

A NEW MULTIMEMBERSHIP CLUSTERING METHOD

Yongbin Ou and Cun-Quan Zhang

Department of Mathematics, West Virginia University
Morgantown, WV 26506-6310, U.S.A.

(Communicated by Yuzhong Zhang)

Abstract. Clustering method is one of the most important tools in statistics. In a graph theory model, clustering is the process of finding all dense subgraphs. In this paper, a new clustering method is introduced. One of the most significant differences between the new method and other existing methods is that this new method constructs a much smaller hierarchical tree, which clearly highlights meaningful clusters. Another important feature of the new method is the feature of *overlapping clustering or multi-membership*. The property of multi-membership is a concept that has recently received increased attention in the literature (Palla, Derényi, Farkas and Vicsek, *(Nature 2005)*; Pereira-Leal, Enright and Ouzounis, *(Bioinformatics, 2004)*; Futschik and Carlisle, *(J. Bioinformatics and Computational Biology 2005)*)

1. **Introduction.** Clustering method is an important statistical tool for data mining. It has been applied in many areas such as social science, internet network, transportation, bioinformatics, image processing, etc. Clustering is the process of detecting all dense subgraphs. A variety of different clustering algorithms have been developed and implemented in popular statistical software packages. A general review of cluster analysis can be found in many references, for instance, [8], [5], [6], etc.. Although many algorithms for graph clustering have clearly demonstrated their usefulness in applications, numerous scholars have raised a number of important questions, such as:

> *"· · · problems related to robustness, uniqueness, and optimality of linear ordering which complicates the interpretation of the resulting hierarchical relationships" and issues of "how to determine the optimal number of clusters"* (Lukashin and Fuchs, 2001 [7] p.405);
>
> Concerns that *"none of these algorithms can, in general, rigorously guarantee to produce a globally optimal clustering for non-trivial objective functions"* (Xu, Olman and Xu, 2002 [15] p.536);
>
> Regarding the need for efficient clustering methods that allow for multimembership, Shepard and Arabie (1979 [13] p.91) write that *"the requirement of having the clusters hierarchically nested seems to be unduly limiting".*

2000 *Mathematics Subject Classification.* 90C26, 90C46.

Key words and phrases. Clustering, multimembership clustering, overlap clustering, hierarchical clustering, dense subgraph.

This work was supported by the West Virginia University Research Corporation, the National Security Agency under Grant MDA904-01-1-0022 and by WV EPSCoR under Grant EPS2006-37.

> And, finally, *"there are no completely satisfactory methods for determining the number of population clusters for any type of cluster analysis"* (SAS/STAT User's Guide, 2003 [12], *"Introduction to Clustering"*).

In this paper, a new clustering method is introduced. One of the most significant differences between the new method and other existing methods is that the "quasi-clique merger" method constructs a much smaller hierarchical tree, which clearly highlights meaningful clusters. The hierarchical trees produced by most existing hierarchical clustering methods are instead binary. A fair amount of guessing is often required to determine meaningful clusters.

Another important feature of the new method that we introduce is the *overlapping clustering or multi-membership*. The property of multi-membership is a concept that has recently received increased attention in the literature (Palla, Derényi, Farkas and Vicsek, 2005 [10]; Pereira-Leal, Enright and Ouzounis, 2004 [11]; Futschik and Carlisle, 2005 [3]). The concept of overlapping clusters is not new per se.

In the next section, the mathematical details of the new algorithm are described.

2. A new clustering algorithm.

A graph or network is one of the most commonly used models to present real-valued relationships of a set of input items. Let $G = (V, E)$ be a graph with the vertex set V and the edge set E with weight $w(e)$ on every edge e. Models with un-weighted graphs (the weight of every edge is set to 1) have been extensively studied in graph theory. In an un-weighted graph G, a subgraph H of G is defined as a *clique* if every pair of vertices of H is joined by one edge (see for instance, [1, 2, 14]). It is well-known that the search of maximum cliques in graphs is an NP-complete problem [4]. Therefore, it is not practical to define cliques as clusters. Furthermore, there is no appropriate definition for a clique in a weighted graph. However, in order to closely represent the nature and the real situation of the inputs in most applications (different degrees of similarity for clustering problems), we should use weighted graph models which are much more appropriate than un-weighted models. For simplification or other practical reasons, many designers of clustering methods may set a specific threshold, such as that any edge with weight below the threshold is deleted and the remaining ones have no associated weight. However, one may not be able to expect a good output since the cut-off (by threshold) may cause a loss of important information.

For a subgraph C, we define the density of C by

$$d(C) \;=\; \frac{2\sum_{e\in E(C)} w(e)}{|V(C)|(|V(C)|-1)}.$$

As seen above, if $w(e) = 1$ and $d(C) = 1$ for every edge e in C, then the subgraph C induces a clique. For a weighted graph, a subgraph C is called a Δ-quasi-clique if $d(C) \geq \Delta$ for some positive real number Δ.

Clustering is a process that detects all dense subgraphs in G, and construct a hierarchically nested system to illustrate their inclusion relation.

2.1. Algorithm (Quasi-Clique Merger v.2).

A heuristic process is applied here for finding all quasi-cliques with density in various levels. The core of the algorithm is deciding whether or not to add a vertex to an already selected dense subgraph

C. For a vertex $v \notin V(C)$, we define the contribution of v to C by

$$c(v, C) = \frac{\sum_{u \in V(C)} w(uv)}{|V(C)|}.$$

A vertex v is added into C if $c(v, C) > \alpha d(C)$ where α is a function of some user specified parameters.

Instance: $G = (V, E)$ is a graph with $w : E(G) \mapsto R^+$.

Question: Detects Δ-quasi-cliques in G with various levels of Δ, and construct a hierarchically nested system to illustrate their inclusion relation.

Algorithm

Step 0. $\ell \leftarrow 1$ where ℓ is the indicator of the levels in the hierarchical system.

$w_0 \leftarrow \gamma \max\{w(e) : \forall e \in E(G)\}$ where γ $(0 < \gamma < 1)$ is a user specified parameter.

Step 1. (The initial step)

Sort the edge set $\{e \in E(G) : w(e) \geq w_0\}$ as a sequence $S = e_1, \cdots, e_m$ such that $w(e_1) \geq w(e_2) \geq \cdots \geq w(e_m)$.

$\mu \leftarrow 1$, $p \leftarrow 0$, and $\mathbf{L}_\ell \leftarrow \emptyset$.

Step 2. (Starting a new search). $p \leftarrow p + 1$, $C_p \leftarrow V(e_\mu)$. $\mathbf{L}_\ell \leftarrow \mathbf{L}_\ell \cup \{C_p\}$.

Step 3. (Grow)

Substep 3.1. If $V(G) - V(C_p) = \emptyset$, then go to Step 4, otherwise continue:

Pick $v \in V(G) - V(C_p)$ such that $c(v, C_p)$ is a maximum.

If

$$c(v, C_p) \geq \alpha_n d(C_p) \tag{1}$$

where $n = |V(C_p)|$ and $\alpha_n = 1 - \frac{1}{2\lambda(n+t)}$ with $\lambda \geq 1$ and $t \geq 1$ as user specified parameters, then $C_p \leftarrow C_p \cup \{v\}$ and go back to Substep 3.1.

Substep 3.2. $\mu \leftarrow \mu + 1$. If $\mu > m$ go to Step 4.

Substep 3.3. Suppose $e_\mu = xy$. If at least one of $x, y \notin \bigcup_{i=1}^{p-1} V(C_i)$ then go to Step 2, otherwise go to Substep 3.2.

Step 4. (Merge)

Substep 4.1.

List all members of $\mathbf{L}_\ell$ as a sequence $C_1, \cdots, C_s$ such that $|V(C_1)| \geq |V(C_2)| \geq \cdots \geq |V(C_s)|$ where $s \leftarrow |\mathbf{L}_\ell|$.

$h \leftarrow 2$, $j \leftarrow 1$.

Substep 4.2. If $|C_j \cap C_h| > \beta \min(|C_j|, |C_h|)$ (where β $(0 < \beta < 1)$ is a user specified parameter), then $C_{s+1} \leftarrow C_j \cup C_h$ and the sequence $\mathbf{L}_\ell$ is rearranged as follows

$$C_1, \cdots, C_{s-1} \leftarrow \text{deleting } C_j, C_h \text{ from } C_1, \cdots, C_{s+1}$$

and $s \leftarrow s - 1$, $h \leftarrow \max\{h - 2, 1\}$, and go to Substep 4.4.

Substep 4.3. $j \leftarrow j + 1$. If $j < h$ go to Substep 4.2.

Substep 4.4. $h \leftarrow h + 1$ and $j \leftarrow 1$. If $h \leq s$ go to Substep 4.2.

Step 5. Contract each $C_p \in \mathbf{L}_\ell$ as a vertex:

$$V(G) \leftarrow [V(G) - \bigcup_{p=1}^{s} V(C_p)] \bigcup \{C_1, \cdots, C_s\},$$

$$w(uv) \leftarrow w(C_{i'}, C_{i''}) = \frac{\sum_{e \in E_{C_{i'}, C_{i''}}} w(e)}{|E_{C_{i'}, C_{i''}}|}$$

if the vertex u is obtained by contracting $C_{i'}$ and v is obtained by contracting $C_{i''}$ where $E_{C_{i'}, C_{i''}}$ is the set of crossing edges which is defined as $E_{C_{i'}, C_{i''}} = \{xy : x \in C_{i'}, y \in C_{i''}, x \neq y\}$. For $t \in V(G) - \{C_1, \cdots, C_s\}$, define $w(t, C_{i'}) = w(\{t\}, C_{i'})$. Other cases are defined similarly.

If $|V(G)| \geq 2$ then go to Step 6, otherwise, go to END

Step 6. $\ell \leftarrow \ell + 1$, $\mathbf{L}_\ell \leftarrow \emptyset$,

$$w_0 \leftarrow \gamma \ \max\{w(e) : \ \forall e \in E(G)\}$$

where γ $(0 < \gamma < 1)$ is a user specified parameter and go to Step 1 (to start a new search in a higher level of the hierarchical system).

END.

3. **The bound of the density.** Let C_p^n be the subgraph generated in Step 3 after $(n-2)$-nd iteration of Substep 3.1 for $n \geq 2$ (that is $|V(C_p^n)| = n$). It is obvious that $d(C_p^2), d(C_p^3), \cdots, d(C_p^n)$ could be a decreasing sequence since $\alpha_n < 1$ for each $n \geq 2$. Therefore, it is necessary to verify that there is a positive constant K such that $d(C_p^n) \geq K w_0$.

In Substep 3.1, a vertex v is added into C_p^n if the contribution of v to C_p^n is

$$\frac{\sum_{u \in V(C_p^n)} w(vu)}{n} \ \geq \alpha_n \frac{\sum_{e \in V(C_p^n)} w(e)}{\frac{n(n-1)}{2}}$$

where

$$\alpha_n = 1 - \frac{1}{2\lambda(n+t)}$$

with $\lambda \geq 1$ and $t \geq 1$ as parameters of user's choice.

Here, let $f(n) = d(C_p^n) = \frac{\sum_{e \in E(C_p^n)} w(e)}{\frac{n(n-1)}{2}}$. Obviously, $\{f(2), f(3), \cdots, f(n)\}$ is a non-increasing sequence. We are to show that there is a constant K $(0 < K < 1)$ such that $f(n)$ has a lower bound $Kf(2)$ that guarantees the minimum density of C_p^n generated in Step 3. Here $f(2) \geq w_0$, and

$$\frac{n(n+1)}{2} f(n+1) \geq \frac{n(n-1)}{2} f(n) + \alpha_n f(n).$$

Hence,

$$\frac{f(n+1)}{f(n)} \geq \frac{\frac{n(n-1)}{2} + \alpha_n n}{\frac{n(n+1)}{2}} = \frac{n - 1 + 2\alpha_n}{n+1} = 1 - \frac{2(1-\alpha_n)}{n+1}$$

$$= 1 - \frac{2\frac{1}{2\lambda(n+t)}}{n+1} = 1 - \frac{1}{\lambda(n+1)(n+t)} = \frac{\lambda(n+1)(n+t) - 1}{\lambda(n+1)(n+t)}.$$

The following is an estimation of $K = \frac{f(n+1)}{f(2)}$ with $\lambda = 1$ and $t = 1$. (Note that K is larger if λ and t are larger.)

$$K = \frac{f(n+1)}{f(2)} = \Pi_{\mu=2}^n \frac{f(\mu+1)}{f(\mu)} = \Pi_{\mu=2}^n \frac{(\mu+1)^2 - 1}{(\mu+1)^2}.$$

Let A_μ be the numerator $((\mu+1)^2-1)$ of $\frac{f(\mu+1)}{f(\mu)}$ and B_μ be the denominator $((\mu+1)^2)$ of $\frac{f(\mu+1)}{f(\mu)}$. Note that

$$\frac{A_\mu}{B_{\mu-1}} = \frac{(\mu+1)^2-1}{\mu^2} = \frac{\mu+2}{\mu}. \tag{2}$$

Hence,

$$\begin{aligned}
\frac{f(n+1)}{f(2)} &= A_2 \times \frac{A_3}{B_2} \times \frac{A_4}{B_3} \times \cdots \times \frac{A_n}{B_{n-1}} \times \frac{1}{B_n} \quad \text{by (2)} \\
&= 8 \times \frac{5}{3} \times \cdots \times \frac{n+2}{n} \times \frac{1}{(n+1)^2} \\
&= 8 \times \frac{(n+1)(n+2)}{3 \times 4} \times \frac{1}{(n+1)^2} \\
&= \frac{2(n+2)}{3(n+1)} > \frac{2}{3}.
\end{aligned}$$

4. **Complexity analysis.** A typical running the algorithm consists of loops from Step 2 through Step 5. First we look at the complexity of one loop. Let $G(V,E)$ be the graph at the beginning of the loop and let $\nu = |V|$. After Step 3 and before Step 4, each vertex is assigned to one or more clusters (for a vertex not in any cluster, treat it as a cluster with only one vertex). Let $m(v)$ be number of the clusters that contain v. Let V' be the multi-set obtained from V by replacing each $v \in V$ by $m(v)$ copies of v. The following practical and realistic assumption in needed in order to get an appropriate estimation of the complexity:

Assumption. $|V'| = O(\nu)$.

In practice, the overlappings of clusters are small relative to $|V|$. This assumption is easy to satisfy.

Now, let's analyze the complexity of each step one by one.

1. Initialization (Step 1)

   It takes $O(|E|\log|E|)$ to sort the edges, which is $O(\nu^2 \log \nu)$.

2. *Grow* (Step 3)

   Initially (in Step 2), when $|C_p| = |\{x,y\}| = 2$, $c(z,C_p) \leftarrow \frac{1}{2}(w(xz)+w(yz))$. Note that the contribution $c(z,C_p)$ $(\forall z \in V(G) - \bigcup_{\mu=1}^{p} V(C_\mu))$ is updated whenever a new vertex v is added into C_p:

$$c(z,C_p) \leftarrow \frac{|C_p|\ c(z,C_p) + w(v,z)}{|C_p|+1}$$

   for every $z \in V(G) - \bigcup_{\mu=1}^{p} V(C_\mu)$. Let G' be the complete graph with vertex set V'. Since the update is applied to every edge of G' at most once, it totally takes at most $O|E(G')|)$ time units to update the contribution.

   Each vertex in V' is added to some cluster at most once. For a given C_p (during the iteration of Step 3), and it takes at most $O(\nu)$ time to decide which vertex (with maximum contribution $c(v,C_p)$) should be added into the current cluster C_p.

   Therefore, the complexity of Step 3 is at most $O(|E(G')| + \nu^2)$, which is $O(\nu^2)$.

3. *Merge* (Step 4)

   The number of clusters produced by Step 3 is at most $|V'|$, thus is at most $O(\nu)$. Each execution of Step 4.2 will reduce the number of clusters by 1, so there are at

most $O(\nu)$ executions of Step 4.2. Therefore the number of clusters (either produced by Step 3 or Step 4.2) on the list $\mathbf{L}_\ell$ is $O(\nu)$. For a cluster C_h on the list, there are two possible operations on it:
(a) For each $j < h$, test if C_h and C_j should be merged;
(b) Merge C_h and C_j for some $j < h$ if applicable.
Clearly (b) takes $O(\nu)$ time. For (a), we need to go through all the vertices in C_i $(1 \le i \le j)$ and C_h. By the assumption, it is also $O(\nu)$. It follows that the complexity of this step is $O(\nu^2)$.

4. *Contract* (Step 5)

Clearly it takes $O(\nu)$ to contract every cluster to a vertex. For the computation of the weights of the new graph, each pair (u, v) where $u, v \in V'$ is processed at most once. Therefore the complexity for this step is also $O(\nu^2)$.

From the above analysis, the complexity of one loop is $O(\nu^2 \log \nu)$. The number of the loops is equal to the height of the hierarchical structure h. Hence, the total complexity is no more than $O(h\nu^2 \log \nu)$ for the entire program.

In each loop some *grow* or *merge* must be done (that is, the number of vertices of the graph is an decreasing function as level goes up), there are at most $O(\nu)$ hierarchical levels (that is, $h \le O(\nu)$). However, in practice, the number h is usually $O(\log \nu)$.

REFERENCES

[1] J. A. Bondy and U. S. R. Murty, "Graph Theory with Applications," Macmillan, London, 1976.

[2] R. Diestel, "Graph Theory," 3rd Ed. Graduate Texts in Mathematics 173, Heidelberg, Springer, 2005.

[3] M. E. Futschik and B. Carlisle, *Noise-robust soft clustering of gene expression timecourse*, Journal of Bioinformatics and Computational Biology, **3** (2005), 965-988.

[4] M. R. Gary and D. S. Johnson, "Computers and Intractability," NY, Freeman, 1979.

[5] W. Härdle and L. Simar, "Applied Multivariate Statistical Analysis," Berlin, Springer, 2003.

[6] P. Hansen and B. Jaumard, *Cluster analysis and mathematical programming*, Mathematical Programming, **79** (1997), 191-215.

[7] A. V. Lukashin and R. Fuchs, *Analysis of temporal gene expression profiles: clustering by simulated annealing and determining the optimal number of clusters*, Bioinformatics, **17** (2001), 405-414.

[8] G. W. Milligan, *Cluster analysis*, in "Encyclopedia of Statistical Sciences" (S. Kotz (Ed.), New York, NY: Wiley, (1998), 120-125.

[9] YB. Ou, L. Guo and CQ. Zhang, *A new clustering method and its application to protemic profiling for colon cancer*, in "Proceeding of IASTED International Conference on Computational and Systems Biology", Dallas, TX., (2006), 68-72.

[10] G. Palla, I. Derényi, I. Farkas and T. Vicsek, *Uncovering the overlapping community structure of complex networks in nature and society*, Nature, Vil. 435, **7043** (2005), 814-818.

[11] J. B. Pereira-Leal, A. J. Enright and C. A. Ouzounis, *Detection of functional modules from protein interaction networks, PROTEINS: Structure, Function*, Bioinformatics, **54** (2004), 49-57.

[12] SAS Institute Inc., "Introduction to Clustering Procedures," Chapter 8 of *"SAS/STAT User's Guide"*. (SAS OnlineDoc^{TM}: Version 8) *http:// www.math.wpi.edu/ saspdf/ stat/ pdfidx.htm*

[13] R. N. Shepard and P. Arabie, *Additive clustering: representation of similarities as combinations of discrete overlapping properties*, Psychological Review, **86** (1979), 87-123.

[14] D. West, "Introduction to Graph Theory," Upper Saddle River, NJ, Prentice Hall, 1996.

[15] Y. Xu, V. Olman and D. Xu, *Clustering gene expression data using graph-theoretic approach: an application of minimum spanning trees*, Bioinformatics, **18** (2002), 536-545.

Received September 2006; revised December 2006 and April 2007.
E-mail address: cqzhang@math.wvu.edu