Community Detection:Hierarchical clustering Algorithms

Anupama Chowdhary
Principal, Keen College
Bikaner (Rajasthan)
India

Abstract **–** To understand the complex and huge networks we need to extract information from them. For this purpose community structure or clustering feature of graphs representing real networks is very significant. The organization of vertices in clusters with many edges joining vertices of the same cluster and comparatively few edges joining vertices of different clusters or communities, are basically independent segment of a graph. Various hierarchical clustering algorithms are used for community detection. We have discussed hierarchical clustering algorithms and software's implementing them for community detection in this paper.

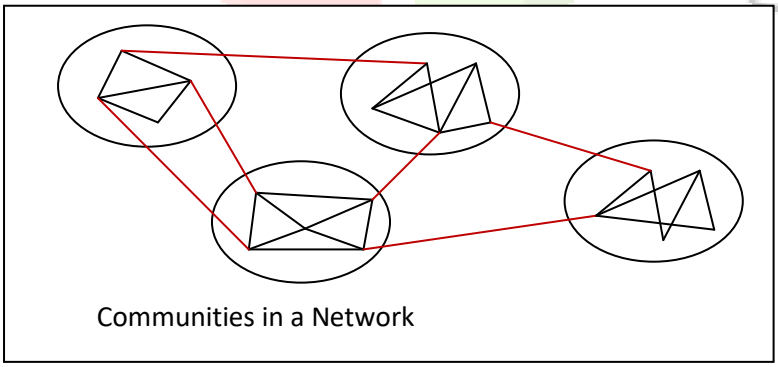
Index terms **–**data mining, community detection, hierarchical clustering.

I. INTRODUCTION

The first analysis of community structure was carried out by Weiss and Jacobson [1], who searched for work groups within a government agency. The authors studied the matrix of working relationships between members of the agency, which were identified by means of private interviews. Work groups were separated by removing the members working with people of different groups, which act as connectors between them. This idea of cutting the bridges between groups is at the basis of several modern algorithms of community detection.

What is a community?

A community could be loosely described as a collection of vertices within a graph that are densely connected among themselves while being loosely connected to the rest of the graph.[2]



What is community detection?

It is a process of discovering groups in a network where group memberships of individual vertices are not explicitly given.

Why to detect communities?

Communities can have concrete applications. Clustering Web clients who have similar interests and are geographically near to each other may improve the performance of services provided on the World Wide Web, in that each cluster of clients could be served by a dedicated mirror server [3]. Identifying clusters

of customers with similar interests in the network of purchase relationships between customers and products of online retailers (like, e.g., www.amazon.com) enables one to set up efficient recommendation systems [4], that better guide customers through the list of items of the retailer and enhance the business opportunities. Clusters of large graphs can be used to create data structures in order to efficiently store the graph data and to handle navigational queries, like path searches [5][6].

II. GENERAL CONCEPTS

As there is no single definition of "community," this means that the community detection problem is ill-defined. "There are no universal protocols on the fundamental ingredients, like thedefinition of community itself, nor on other crucial issues, like the validation of algorithms and the comparison of their performances."[7]. Different algorithms performed on the same networkcan produce different clusters of nodes, depending on their sets of assumptions.

It is important to stress that the identification of structural clusters is possible only if graphs are sparse, i.e. if the number of edges m is of the order of the number of nodes n of the graph. If m >> n, the distribution of edges among the nodes is too homogeneous for communities to make sense (This is not necessarily true if graphs are weighted with a heterogeneous distribution of weights. In such cases communities may still be identified as sub-graphs with a high internal density of weight). In this case the problem turns into something rather different, close to data clustering [8], which requires concepts and methods of a differentnature. The main difference is that, while communities in graphs are related, explicitly or implicitly, to the concept of edge density (inside versus outside the community), in data clustering communities are sets of points which are ``close" to each other, with respect to a measure of distance or similarity, defined for each pair of points.

Many clustering algorithms or problems related to clustering are NP-hard. If an algorithm has a polynomial complexity,it may still be too slow to tackle large systems of interest. In all such cases it is common to use approximation algorithms, i.e.methods that do not deliver an exact solution to the problem at hand, but only an approximate solution, with the advantageof a lower complexity. Approximation algorithms are often non-deterministic, as they deliver different solutions for thesame problem, for different initial conditions and/or parameters of the algorithm. In computer science and operations research, approximation algorithms are efficient algorithms that find approximate solutions to NP-hard optimization problems with provable guarantees on the distance of the returned solution to the optimal one [9]. Approximation algorithms are commonly used for optimization problems, in which one wants to find the maximum or minimum value of a given cost function over a large set of possible system configurations.

Many algorithms are able to identify a subset of meaningful partitions, ideally one or just a few, whereas some others, like techniques based on hierarchical clustering, deliver a large number of partitions. That does not mean that the partitions found are equally good. Therefore it is helpful (sometimes even necessary) to have a quantitative criterion to assess the goodness of a graph partition. A quality function is a function that assigns a number to each partition of a graph. In this way one can rank partitions based on their score given by the quality function. Partitions with high scores are ``good", so the one with the largest score is by definition the best. Nevertheless, one should keep in mind that the question of when a partition is better than another one is ill-posed, and the answer depends on the specific concept of community and/or quality function adopted[10].

III. HIERARCHICAL CLUSTERING ALGORITHMS

Finding communities within an arbitrary network can be a computationally difficult task. The number of communities, if any, within the network is typically unknown and the communities are often of unequal size and/or density. Despite these difficulties, however, several methods for community finding have been

developed and employed with varying levels of success [11]. In this paper Hierarchical Clustering Algorithms are discussed.

In this method one defines a similarity measure quantifying some (usually topological) type of similarity between node pairs. Some commonly used measures are

- **The cosine similarity**: The cosine of two non-zero vectors can be derived by using the Euclidean dot product formula:

$$p.q = ||p||_2||q||_2 \cos\theta$$

Let P and Q be two vectors of attributes, then the cosine similarity is represented as

$$\textbf{\textit{Similarity}} = \cos(\theta) = \frac{P.Q}{||P||_2||Q||_2} = \frac{\sum_{i=1}^{n} P_i Q_i}{\sqrt{\sum_{i=1}^{n} P_i^2}\sqrt{\sum_{i=1}^{n} Q_i^2}}$$

The range of similarity is [-1,1].

Value of similarity	**Description**
-1	exactly opposite
1	exactly the same
0	indicating orthogonality (decorrelation)
In-between	intermediate similarity or dissimilarity

- **The Jaccard index**: It is a statistic used for comparing the similarity and diversityofsamplesets. The Jaccard coefficient measures similarity between finite sample sets[12][13]. Let P and Q be such sets, then

$$J(P,Q) = \frac{|P \cap Q|}{|P \cup Q|} = \frac{|P \cap Q|}{|P|+|Q|-|P \cap Q|}$$

$0 \leq J(P,Q) \leq 1$.If P and Q are empty then J(P,Q)=1.

Jaccard distance measures dissimilarity between sample sets and is calculated as

$$d_J(P,Q) = 1 - J(p,Q)$$

- **The Hamming distance between rows of the adjacency matrix**: Let A(G) be the adjacency matrix of a graph G. The rows of A(G) corresponding to a vertex v of G, denoted by s(v), is thestring which belongs toZ_2^n, a set of n-tuples. The Hamming distance between the vertices u and v is the number of positions in which s(u) and s(v) differ.

These measures are used to group similar nodes into communities. Mostly hierarchical clustering methods adopt agglomerative “Bottom-up” clustering process. In the beginning of the agglomerative clustering process, each element is in a cluster of its own. The clusters are then sequentially combined into larger clusters, until all elements end up being in the same cluster. At each step, the two clusters separated by the shortest distance are combined. The definition of 'shortest distance' is what differentiates between the different agglomerative clustering methods. Another approach is Divisive “Top-Down” analysis clustering, here initially all the nodes in a network are placed in a single cluster based on the intuition that edges with the highest betweenness values are 'bridges' between communities, groups are subdivided into two group recursively until each node of the network forms a separate cluster of its own.

IV. AGGLOMERATIVE APPROACH

1. **Single-linkage (nearest neighbour)**: It is based on grouping clusters in bottom-up fashion (agglomerative clustering). In single-linkage clustering, the distance between two clusters is determined by a single element pair, namely those two elements (one in each cluster) that are closest to each other. The shortest of these links that remains at any step causes the fusion of the two clusters whose elements are involved. The result of the clustering can be visualized as a dendrogram, which shows the sequence of cluster fusion and the distance at which each fusion took place[14].The linkage function

$$D(P, Q) = \min_{p \in P, q \in Q} d(p, q)$$

Here, D(P,Q) : Distance between clusters P and Q
P and Q: any two sets of elements considered as clusters
d(p,q) : denotes the distance between the two elements p and q.

Time complexity: $O(n^3)$[15]

Improved algorithms:

- R. Sibson proposed SLINK algorithm with time complexity $O(n^2)$ and space complexity $O(n)$. The SLINK algorithm represents a clustering on a set of numbered items by two functions (λ, π), these functions are both determined by finding the smallest cluster C that contains both item i and at least one larger-numbered item.

  λ: It maps item i to the largest-numbered item in cluster C

  π : It maps item i to the distance associated with the creation ofcluster C

  Storing these functions in two arrays takes space O(n), and this information is sufficient to determine the clustering itself. As Sibson shows, when a new item is added to the set of items, the updated functions representing the new single-linkage clustering for the augmented set, represented in the same way, can be constructed from the old clustering in time O(n). The SLINK algorithm then loops over the items, one by one, adding them to the representation of the clustering[16][17].
- Gower and Ross [18] used Prism's algorithm, in a variation without binary heaps, that takes time $O(n^2)$ and space $O(n)$ to construct the minimum spanning tree (but not the clustering) of the given items and distances. Then, applying Kruskal's algorithm to the sparse graph formed by the edges of the minimum spanning tree produces the clustering itself in an additional time $O(n \log n)$ and space $O(n)$.

A drawback of this method is that it tends to produce long thin clusters in which nearby elements of the same cluster have small distances, but elements at opposite ends of a cluster may be much farther from each other than to elements of other clusters. This may lead to difficulties in defining classes that could usefully subdivide the data[19].

2. **Complete-linkage (diameter, farthest neighbour clustering**): In complete-linkage clustering, the link between two clusters contains all element pairs, and the distance between clusters equals the distance between those two elements (one in each cluster) that are farthest away from each other. The shortest of these links that remains at any step causes the fusion of the two clusters whose elements are involved. The result of the clustering can be visualized as a dendrogram, which shows the sequence of cluster fusion and the distance at which each fusion took place[20][21][22].The linkage function

$$D(P, Q) = \max_{p \in P, q \in Q} d(p, q)$$

Here, D(P,Q) : Distance between clusters P and Q

P and Q: any two sets of elements considered as clusters

d(p,q) : denotes the distance between the two elements p and q.

Time complexity: $O(n^3)$[15]

Improved algorithms:
R. Defays proposed CLINK algorithm with time complexity $O(n^2)$ inspired by the similar algorithm SLINK[23].

3. **Group average (average link, UPGMA(U**nweighted **P**air **G**roup **M**ethod with **A**rithmetic Mean**))**: Group average algorithm constructs a rooted tree (dendrogram) that reflects the structure present in a pairwise similarity matrix (or dissimilarity matrix). At each step, the nearest two clusters are combined into a higher-level cluster [24]. The linkage function

$$D(P,Q) = \frac{1}{|P|.|Q|}\sum_{p\in P}\sum_{q\in Q} d(p,q)$$

Here, D(P,Q) : Distance between clusters P and Q

P and Q: any two sets of elements considered as clusters
d(p,q) : denotes the distance between the two elements p and q.

Time complexity: $O(n^3)$[15]

Improved algorithms:
Using a heap for each cluster to keep its distances from other cluster reduces its time to $O(n^2 \log n)$. FionnMurtagh presented some other approaches for special cases, a $O(k3^k n^2)$ time algorithm by Day and Edelsbrunner[25] for k-dimensional data that is optimal $O(n^2)$ for constant k, and another $O(n^2)$ algorithm for restricted inputs, when "the agglomerative strategy satisfies the reducibility property."[26]

4. **McQuitty'smethod(WPGMA(W**eighted **P**air **G**roup **M**ethod with **A**rithmetic Mean)**)**: It is a simple agglomerative method, generally attributed to Sokal and Michener[24]. This algorithm constructs a rooted tree (dendrogram) that reflects the structure present in a pairwise similarity matrix (or dissimilarity matrix). At each step, the nearest two clusters, namely ***p*** and ***q*** , are combined into a higher-level cluster **p** ∪ **q** . Then, its distance to another cluster ***s*** is simply the arithmetic mean of the distances between ***s*** and members of **p** ∪ **q**:

$$\mathbf{d_{p\cup q,s}} = \frac{\mathbf{d_{p,s}} + \mathbf{d_{q,s}}}{\mathbf{2}}$$

Time complexity: $O(n^3)$[15]
Improved algorithms:
Similar improved algorithms are there as for UPGMA.

5. **Centroid (UPGMC(U**nweighted**P**air **G**roup **M**ethod **C**entroid)**)**: The centroid is the mean of all points in a cluster. When two clusters C_i *and* C_j are merged, the distance to the third cluster C_k is calculated as

$$D\left(C_k,(C_i,C_j)\right) = \frac{n_i}{n_i+n_j}D(C_k,C_i) + \frac{n_j}{n_i+n_j}D(C_k,C_j) - \frac{n_i n_j}{(n_i+n_j)^2}D(C_i,C_j)$$

Time complexity: $O(n)$[27]

6. **Median method (Gower's, WPGMC(W**eighted **P**air **G**roup **M**ethod **C**entroid)**)**: It is very similar to the centroid method, but here equal weight is given to the clusters to be merged.

$$D\left(C_k,(C_i,C_j)\right)=\frac{1}{2}D(C_k,C_i)+\frac{1}{2}D(C_k,C_j)-\frac{1}{4}D(C_i,C_j)$$

Time complexity: $O(n)$[27]

7. **Ward's method (minimum variance, error sum of squares)**: Ward's is the only one among the agglomerative clustering methods that is based on a classical sum-of-squares criterion, producing groups that minimize within-group dispersion at each binary fusion. Ward's method shares the total error sum of squares criterion with K-means partitioning, which is widely used to directly cluster observations in Euclidean space, hence to create a partition of the observation set. This clustering is done without any structural constraint such as cluster embeddedness, represented by a hierarchy. A more direct and computer-efficient approach is to first apply Ward's minimum variance agglomerative clustering to the data, identify the partition of the objects into K groups in the dendrogram, and then use that partition as the starting approximation for K-means partitioning[28][29][30].
The initial cluster distances in Ward's minimum variance method are the squared Euclidean distance between points:

$$\mathrm{d_{ij}}=\mathrm{d}(\{\mathrm{X_i}\},\{\mathrm{X_j}\})=||\mathrm{X_i}-\mathrm{X_j}||^2$$

Time complexity: $O(n^2)O(n\ \log n)$[15][27]
Improved algorithms:
Ward's minimum variance method can be defined and implemented recursively by a Lance–Williams algorithm. At each step, it is necessary to optimize the objective function (find the optimal pair of clusters to merge). The recursive formula simplifies finding the optimal pair.

Ward's minimum variance method can be implemented by the Lance–Williams formula. For disjoint clusters C_i, C_j, C_k with sizes n_i, n_j, n_k respectively:

$$d(C_i\cup C_j,C_k)=\frac{n_i+n_k}{n_i+n_j+n_k}d(C_i,C_k)+\frac{n_j+n_k}{n_i+n_j+n_k}d(C_j,C_k)-\frac{n_k}{n_i+n_j+n_k}d(C_i,C_j)$$

Hence Ward's method can be implemented as a Lance–Williams algorithm with

$\alpha_l=\frac{n_i+n_k}{n_i+n_j+n_k}$, $\beta=\frac{-n_k}{n_i+n_j+n_k}$, $\gamma=0$

where $(\alpha_i,\alpha_j,\beta,\gamma)$ are parameters, which may depend on cluster sizes, that together with the cluster distance function d_{ij} determine the clustering algorithm.

The popularity of the Ward's method has led to variations of it. For instance, $\mathrm{Ward_p}$ introduces the use of cluster specific feature weights, following the intuitive idea that features could have different degrees of relevance at different clusters[31].

Algorithm for agglomerative clustering
Algorithm steps:

1. Represent each point as a cluster.
2. Calculate the proximity matrix.
3. Merge a pair of cluster with the minimal distance.
4. Repeat steps 2 and step 3 until only one cluster left.
5. Generate the clusters by cutting dendrogram at an appropriate level

V. DIVISIVE APPROACH

Divisive clustering algorithms provide clearer insights of the main structure of the data, but the larger clusters are generated at the early stage of the clustering process, so we have to rely on heuristic methods

1. **DIANA (Di**visive **Ana**lysis**)**: It considers only a part of all the possible divisions. DIANA consists of a series of iterative steps in order to move the closer objects into the splinter group, which is seeded with the object that is farthest from the others in the cluster to be divided. The cluster with the largest diameter, defined as the largest distance between any pair of objects, is selected for further division [32]. Suppose that cluster C_k is going to be split into clusters C_i and C_j

   1. Start with $C_i = C_k$ and C_j empty
   2. For each data object x_m in C_i
      a) For the first iteration, compute its average distance to all the other objects

$$d(x_m, C_i\{x_m\}) = \frac{1}{N_{C_i} - 1} \sum_{\substack{x_p \in C_i \\ p \neq m}} d(x_m, x_p)$$

      b) For the remaining iterations, compute the difference between the average distance to C_i and the average distance to C_j

$$d(x_m, C_i\{x_m\}) - d(x_m, C_j) = \frac{1}{N_{C_i} - 1} \sum_{\substack{x_p \in C_i \\ p \neq m}} d(x_m, x_p) - \frac{1}{N_{C_j} - 1} \sum_{x_q \in C_j} d(x_m, x_q)$$

   3. Decide whether to move element x_m to cluster C_i or keep it in cluster C_i
      a) For the first iteration, move the object with the maximumvalue to C_j (that is, theelement farther from every otherelement is separated from the others)
      b) For the remaining iterations, if the maximum differencevalue is greater than zero, move the data object with the maximum difference into C_j, then repeat steps 2b and 3b. Ifthe maximum value is less than zero, stop.

   Time complexity: $O(2^n)$ [32]

2. **Girvan-Newman Algorithm:** It is a divisive hierarchical clustering community detection algorithm given by Michelle Girvan and Mark Newman[33][34]. The steps of the algorithm are:
   1. Computation of the centrality for all edges.
   2. Removal of edge with largest centrality: in case of ties with other edges, one of them is picked at random.
   3. Recalculation of centralities on the running graph.
   4. Iteration of the cycle from step 2.

   Girvan and Newman focused on the concept of betweenness, which is a variable expressing the frequency of the participation of edges to a process. Edge betweenness is the number of shortest paths between all vertex pairs that run along the edge. It is an extension to edges of the popular concept of site betweenness, introduced by Freeman in 1977[35]. In the calculation of site betweenness,

   - If there are two or more geodesic paths with the same endpoints that run through an edge, the contribution of each of them to the betweenness of the edge must be divided by the multiplicity of the paths, as one assumes that the signal/information propagates equally along each geodesic path. Thebetweenness of all edges of the graph can be calculated in a time that scales as $O(mn)$, or $O(n^2)$on a sparse graph, with techniques based on breadth-first-search[34][36][37].

- If the signals flow across random the betweenness of an edge is given by the frequency of the passages across the edge of a random walker running on the graph (random-walk betweenness). Calculation of random-walk betweenness requires the inversion of an$n \times n$ matrix (once), followed by obtaining and averaging the flows for all pairs of nodes. The first task requires a time$O(n^3)$, the second$O(mn^2)$, for a total complexity $O(n^3)$for a sparse matrix [38].

Improved algorithms:

- Tyler et al. proposed a modification of the Girvan-Newman algorithm, to improve the speed of the calculation. They proposed to calculate the contribution to edge betweenness only from a limited number of centres, chosen at random, deriving a sort of Monte Carlo estimate. The algorithm proceeds just like that of Girvan and Newman. The stopping criterion is different, though, as it does not require the calculation of modularity on the resulting partitions, but relies on a particular definition of community. By repeating the calculation many times, the method gives good results on a network of gene co-occurrences, with a substantial gain of computer time. The technique has been also applied to a network of people corresponding via email [39][40].
- Rattigan et al. lower the complexity of the algorithm to$O(m)$. A quick approximation of the edge betweenness values is carried out by using a network structure index, which consists of a set of vertex annotations combined with a distance measure. Basically one divides the graph into regions and computes the distances of every vertex from each region. This version of the Girvan-Newman algorithm gives good results on the benchmark graphs proposed by Brandes et al. , as well as on a collaboration network of actors and on a citation network. [41][42].
- Chen and Yuan proposed to count only non-redundant paths, i.e. paths whose endpoints are all different from each other: the resulting betweenness yields better results than standard edge betweenness for mixed clusters on the benchmark graphs of Girvan and Newman[43].
- Holme et al. have used a modifiedversion of the algorithm in which vertices, rather than edges, are removed. A centrality measure for the vertices, proportional to their site betweenness, and inversely proportional to their indegree, is chosen to identify boundary vertices, which are then iteratively removed with all their edges [44].
- Pinney and Westhead have proposed a modification of the algorithm in which vertices can be split between communities, to find overlapping communities [45].Gregory has proposed a similar approach, named CONGA (Cluster Overlap Newman-Girvan Algorithm), he go for the split that yields the maximum of a new centrality measure, called split betweenness, which is the number of shortest paths that would run between two parts of a vertex if the latter were split. Worst-case complexity of algorithm is $O(n^2)$ [46].

Software's implementing hierarchical clustering algorithms	
Open source	**Commercial**
Cluster 3.0	MATLAB
ELKI	SAS
Octave	Mathematica
Orange	NCSS
R	SPSS
SCaViS	Qlucore
Scikit-learn	Stata
Weka	

VI. CONCLUSIONS

By studying both agglomerative and divisive algorithms it was found out that approximately all the algorithms used different distance measures to discover various clusters in a large graph. The most popular method is divisive algorithm given by Newman and Girvan as this method returns results of reasonable

quality and is implemented in a number of standard software packages. More over this algorithm is modified by many people according to the requirement. Ward's is also another popular agglomerative method its variations are mostly used. Although mostly software packages provides SLINK, and CLINK algorithms too for community detection. The time complexity of most of the algorithms is $O(n^3)$ but in some special cases it comes down to $O(n^2)$.

REFERENCES

[1] R.S. Weiss, E. Jacobson, "A method for the analysis of the structure of complex organizations", Am. Sociol.Rev. 20 (1955) 661668.

[2] J. Bagrow and E. Bolt, "Local method for detecting communities".Physical Rev. E., Volume 72 No. 4, p 046108, 2015.

[3] B. Krishnamurthy, J. Wang, "On network-aware clustering of web clients", SIGCOMM Comput.Commun.Rev. 30 (4) (2000) 97110.

[4] K.P. Reddy, M. Kitsuregawa, P. Sreekanth, S.S. Rao, "A graph based approach to extract a neighborhood customer community for collaborative filtering",in: DNIS '02: Proceedings of the Second International Workshop on Databases in Networked Information Systems, Springer-Verlag, London, UK, 2002,pp. 188200.

[5] R. Agrawal, H.V. Jagadish, "Algorithms for searching massive graphs", Knowl.Data Eng. 6 (2) (1994) 225238.

[6] A.Y. Wu, M. Garland, J. Han, "Mining scale-free networks using geodesic clustering", in: KDD '04: Proceedings of the Tenth ACM SIGKDD InternationalConference on Knowledge Discovery and Data Mining, ACM Press, New York, NY, USA, 2004, pp. 719724.

[7] Andrea Lancichinetti, MikkoKivelä, Jari , "Characterizing the Community Structure of Complex Networks". Saramäki, Santo Fortunato. PLOS One.(August 12, 2010) http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0011976

[8] G. Gan, C. Ma, J. Wu, "Data Clustering: Theory, Algorithms, and Applications". (ASA-SIAM Series on Statistics and Applied Probability), Society for Industrial and Applied Mathematics, Philadelphia, USA, 2007

[9] Vazirani, Vijay V. (2003). "Approximation Algorithms". Berlin: Springer. ISBN 3-540-65367-8.

[10] M.E.J. Newman, M. Girvan, "Finding and evaluating community structure in networks", Phys. Rev. E 69 (2) (2004) 026113.

[11] M. A. Porter; J.-P. Onnela; P. J. Mucha (2009). "Communities in Networks" . Notices of the American Mathematical Society. 56: 1082–1097, 1164–1166.

[12] Jaccard, Paul (1901), "Étude comparative de la distribution floraledansune portion des Alpeset des ura", Bulletin de la SociétéVaudoise des Sciences Naturelles, 37: 547–579.

[13] Jaccard, Paul (1912), "The distribution of the flora in the alpine zone", New Phytologist, 11: 37–50, doi:10.1111/j.1469-8137.1912.tb05611.x.

[14] Legendre, P. & Legendre, L. 1998. "Numerical Ecology".Second English Edition.853 pages.

[15] Murtagh, Fionn Contreras, Pedro (2012). "Algorithms for hierarchical clustering: an overview" (PDF). Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. Wiley Online Library. 2 (1): 86–97. doi:10.1002/widm.53.

[16] R. Sibson (1973). "SLINK: an optimally efficient algorithm for the single-link cluster method" (PDF). The Computer Journal.British Computer Society. 16 (1): 30–34. doi:10.1093/comjnl/16.1.30.

[17] Gan, Guojun (2007). Data clustering : theory, algorithms, and applications. Philadelphia, Pa. Alexandria, Va: SIAM, Society for Industrial and Applied Mathematics American Statistical Association. ISBN 9780898716238.

[18] Gower, J. C.; Ross, G. J. S. (1969), "Minimum spanning trees and single linkage cluster analysis", Journal of the Royal Statistical Society, Series C, 18 (1): 54–64, JSTOR 2346439, MR 0242315.

[19] Everitt, Brian (2011). Cluster analysis. Chichester, West Sussex, U.K: Wiley. ISBN 9780470749913.

[20] T. Sorensen (1948). "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons.". BiologiskeSkrifter. 5: 1–34.

[21] Legendre, P. & Legendre, L. 1998. Numerical Ecology.Second English Edition.853 pages.

[22] Brian S. Everitt; Sabine Landau; MorvenLeese (2001). "Cluster Analysis", (Fourth ed.). London: Arnold. ISBN 0-340-76119-9.

[23] D. Defays (1977). "An efficient algorithm for a complete link method" (PDF). The Computer Journal.British Computer Society. 20 (4): 364–366. doi:10.1093/comjnl/20.4.364.

[24] Sokal R and Michener C (1958). "A statistical method for evaluating systematic relationships". University of Kansas Science Bulletin. 38: 1409–1438.

[25] Day, William H. E.; Edelsbrunner, Herbert (1984-12-01). "Efficient algorithms for agglomerative hierarchical clustering methods". Journal of Classification. 1 (1): 7–24. ISSN 0176-4268. doi:10.1007/BF01890115.

[26] Murtagh F (1984). "Complexities of Hierarchic Clustering Algorithms: the state of the art". Computational Statistics Quarterly. 1: 101–113.

[27] F. Murtagh, "Expected time complexity results for hierarchal clustering algorithms which use cluster centres", Information Processing letters 16(1983) 237-241, North-Holland.

[28] Gower, J. C. (1967). "A comparison of some methods of cluster analysis". Biometrics, 23, 623–637.

[29] FionnMurtagh, Pierre Legendre,"Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion?", Journal of Classification 31:274-295 (2014), DOI: 10.1007/s00357-014-9161-z

[30] Ward, J. H., Jr. (1963), "Hierarchical Grouping to Optimize an Objective Function", Journal of the American Statistical Association, 58, 236–244.
[31] R.C. de Amorim (2015). "Feature Relevance in Ward's Hierarchical Clustering Using the Lp Norm". Journal of Classification. 32 (1): 46–62. doi:10.1007/s00357-015-9167-1
[32] Kaufman, L., &Roussew, P. J. (1990). "Finding Groups in Data - An Introduction to Cluster Analysis".A Wiley-Science Publication John Wiley & Sons.
[33] Girvan M. and Newman M. E. J., "Community structure in social and biological networks", Proc. Natl. Acad. Sci. USA 99, 7821–7826 (2002)
[34] M.E.J. Newman, M. Girvan, "Finding and evaluating community structure in networks", Phys. Rev. E 69 (2) (2004) 026113.
[35] L.C. Freeman, "A set of measures of centrality based on betweenness",Sociometry 40 (1977) 35-41.
[36] U. Brandes, "A faster algorithm for betweenness centrality", J. Math. Sociol. 25 (2001) 163-177.
[37] T. Zhou, J.-G.Liu, B.-H. Wang," Notes on the calculation of node betweenness", Chin. Phys. Lett. 23 (2006) 2327-2329
[38] M.E.J. Newman," A measure of betweenness centrality based on random walks", Soc. Netw. 27 (2005) 39-54
[39] J.R. Tyler, D.M. Wilkinson, B.A. Huberman, "Email as spectroscopy: Automated discovery of community structure within organizations", in: Communities and Technologies, Kluwer, B.V., Deventer, The Netherlands, 2003, pp. 81-96.
[40] D.M. Wilkinson, B.A. Huberman," A method for finding communities of related genes", Proc. Natl. Acad. Sci. U.S.A. 101 (2004) 5241-5248.
[41] M.J. Rattigan, M. Maier, D. Jensen, "Graph clustering with network structure indices", in: ICML '07: Proceedings of the 24th International Conference on Machine Learning, ACM, New York, NY, USA, 2007, pp. 783-790
[42] M.J. Rattigan, M. Maier, D. Jensen, "Using structure indices for efficient approximation of network properties", in: T. Eliassi-Rad, L.H. Ungar,
M. Craven, D. Gunopulos (Eds.), SIGKDD'06: Proceedings of the 12th international conference on Knowledge Discovery and Data Mining, ACM, 2006, pp. 357-366
[43] J. Chen, B. Yuan," Detecting functional modules in the yeast protein-protein interaction network", Bioinformatics 22 (18) (2006) 2283-2290
[44] P. Holme, M. Huss, H. Jeong, "Subnetwork hierarchies of biochemical pathways", Bioinformatics 19 (4) (2003) 532-538
[45] J.W. Pinney, D.R. Westhead, "Betweenness-based decomposition methods for social and biological networks", in: Interdisciplinary Statistics and Bioinformatics, Leeds University Press, Leeds, UK, 2006, pp. 87-90.
[46] S. Gregory, "An algorithm to find overlapping community structure in networks", in: Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD 2007, Springer-Verlag, Berlin, Germany, 2007, pp. 91-102.