www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 3 March 2017, Page No. 20441-20445
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i3.07

E-Shopping Community Structure Analysis Using Data Clustering

Sreeja B P [1], Saratha Devi G[2]
[1,2] Assistant Professor,
Department of Information Technology,
Karpagam College of Engineering,Coimbatore,Tamilnadu,India.
sreejabp@gmail.com.
sarathadevi.techno@gmail.com.

Abstract: ***The E-Shopping Experience has opened the new ways of business and shopping. The conventional terms of shopping have been changed and new terms to shop online emerge into customers' online shopping behaviors and preferences. Extort interesting shopping patterns from ever increasing data is not a inconsequential mission. It require intelligent association rule mining of the available data, that can be practically knowledgeable for the online retail stores, so that they can make viable business decisions .The fast development of online shopping, the ability to segment e-shoppers basing on their preferences and characteristics has become a key source of competitive advantage for firms. This paper presented the pragmatic algorithms for clustering e-shoppers in e-commerce applications. Various multi-dimensional range search is presented to solve the range-searching problem..In addition, in this paper, the global clustering algorithm is presented which is an incremental approach to clustering that dynamically adds one cluster center at a time through a deterministic global search procedure The basic idea underlying the proposed method is that an finest solution for a clustering problem with other clusters can be obtained using a series of location based clustering and segmentation.***

Keywords: onlineshopping,e-shoppers, clustering,finestsolution.

1. Introduction

The task of grouping a set of objects in the same group is called a cluster . It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image-analysis, information-retrieval bioinformatics, data compression and computer graphics. Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It is often necessary to modify data preprocessing and model parameters until the result achieves the desired properties. Clustering analysis has been an emerging research issue in data mining due its variety of applications. With the advent of many data clustering algorithms in the recent few years and its extensive use in wide variety of applications, including image processing, computational biology, mobile communication, medicine and economics, has lead to the popularity of this algorithms. Main problem with the data clustering algorithms is that it cannot be standardized. Algorithm developed may give best result with one type of data set but may fail or give poor result with data set of other types. Although there has been many attempts for standardizing the algorithms which can perform well in all case of scenarios but till now no major accomplishment has been achieved. Many clustering algorithms have been proposed so far. However, each algorithm has its own merits and demerits and cannot work for all real situations. Before exploring various clustering algorithms in detail let's have a brief overview about what is clustering.

Data Clustering

Social networks are represented by people as nodes and their relationships by edges; and biological networks are usually represented by biochemical molecules as nodes and the reactions between them by edges. Most of the research in the recent past focused on understanding the evolution and organization of such networks and the effect of network topology on the dynamics and behaviors of the system. Finding community structures in networks is another step toward understanding the complex systems they represent. The goal of a community detection algorithm is to find groups of nodes of interest in a given network. The network partitioning problem is in general defined as the partitioning of a network into a fixed constant groups of approximately equal sizes, minimizing the number of edges between groups. This problem is NP-hard and efficient heuristic methods have been developed over years to solve the problem. Much of this work is motivated by engineering applications including very large scale integrated circuit layout designs and mapping of parallel computations. Thompson showed that one of the important factors affecting the minimum layout area of a given circuit in a chip is its

bisection width. Also, to enhance the performance of a computational algorithm, where nodes represent computations and edges represent communications, the nodes are divided equally among the processors so that the communications between them are minimized. The goal of a network partitioning algorithm is to divide any given network into approximately equal size groups irrespective of node similarities. Community detection, on the other hand, finds groups that either have an inherent or an externally specified notion of similarity among nodes within groups. Furthermore, the number of communities in a network and their sizes are not known beforehand and they are established by the community detection algorithm.

2. Related Works

The availability of such a vast collection of clustering algorithms in the literature can easily confound a user attempting to select an algorithm suitable for the problem at hand. In Dubes and Jain [1976], a set of admissibility criteria defined by Fisher and Van Ness [1971] are used to compare clustering algorithms. These admissibility criteria are based on: (1) the manner in which clusters are formed, (2) the structure of the data, and (3) sensitivity of the clustering technique to changes that do not affect the structure of the data. These issues have motivated this survey, and its aim is to provide a perspective on the state of the art in clustering methodology and algorithms. With such a perspective, an informed practitioner should be able to confidently assess the tradeoffs of different techniques, and ultimately make a competent decision on a technique or suite of techniques to employ in a particular application. There is no clustering technique that is universally applicable in uncovering the variety of structures present in multidimensional data sets. Humans perform competitively with automatic clustering procedures in two dimensions, but most real problems involve clustering in higher dimensions. It is difficult for humans to obtain an intuitive interpretation of data embedded in a high-dimensional space. In addition, data hardly follow the "ideal" structures (e.g., hyperspherical, linearthe large number of clustering algorithms which continue to appear in the literature; each new clustering algorithm performs slightly better than the existing ones on a specific distribution of patterns.

The operation of a hierarchical clustering algorithm is illustrated using the two-dimensional data set.Most hierarchical clustering algorithms are variants of the single-link [Sneath and Sokal 1973], complete-link [King 1967], and minimum-variance [Ward 1963; Murtagh 1984] algorithms. Of these, the single-link and completelink algorithms are most popular. These two algorithms differ in the way they characterize the similarity between a pair of clusters. In the single-link method, the distance between two clusters is the minimum of the distances between all pairs of patterns drawn from the two clusters (one pattern from the first cluster, the other from the second). In the complete-link algorithm, the distance between two clusters is the maximum of all pairwise distances betweenpatterns in the two clusters. In either case, two clusters are merged to form a larger cluster based on minimum distance criteria. The complete-link algorithm produces tightly bound or compact clusters [Baeza-Yates 1992]. The single-link algorithm, by contrast, suffers from a chaining effect [Nagy 1968]. It has a tendency to produce clusters that are straggly or elongated.

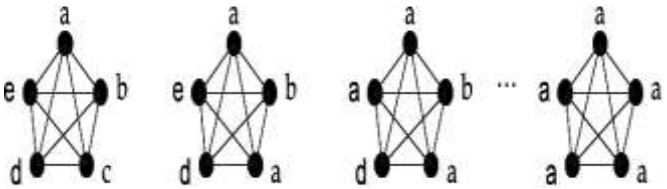
Four scenarios of Web Intelligence data which possess different characteristics are put under test in the experiment, with K-Means and Clustering with PSO (C-PSO) for benchmarking references plus four latest nature-inspired clustering algorithms namely, Clustering with Fireflies (C-Firefly), Clustering with Cuckoos (C-Cuckoo), Clustering with Bats (C-Bat) and Clustering with Wolves (C-WSA). The representative Web data include the datasets of Page Blocks (PB), Internet Usage (IU), Ipod auctions on e-Bay (IE) and Spambase (SB). The datasets are extracted from real-world applications which are available for download from UCI Machine Learning Repository.UCI Dataset Archive is a popular place for researchers downloading publicly available data for testing machine learning algorithms. The datasets are described briefly below. Page Blocks (PB) is a typical Web page classification problem that consists of classifying all the blocks of the page layout of a document that has been detected by a segmentation process. This is an essential step in document analysis in order to separate text from graphic areas. The dataset is released in 1995. There is 5473 instance and 10 attributes, which comes from 54 distinct documents, all attributes are numeric. The attributes describe a variety of characteristics of the block, for example, dimension and area of the block, the ratio of pixels within the block, etc. One application is to try clustering the data objects to different groups of text, horizontal lines, vertical lines, graphics and pictures depending on the attribute values. Internet Usage (IU): Internet Usage (IU) dataset was released in 1999. Data were collected from a survey provided by the Graphics and Visualization Unit at Georgia Tech in 1997. It contained 72 discrete attributes of user's personal information and interests of using Internet. For a data mining utility, we use these data to build a segmentation model of user's occupations so that relevant advertising information will be delivered to approximate users.

The main idea behind the label propagation algorithm is the following. Suppose that a node x has neighbors x1,x2,...,xk and that each neighbor carries a label denoting the community to which they belong. Then x determines its community based on the labels of its neighbors. We assume that each node in the network chooses to join the community to which the maximum numbers of its neighbors belong, with ties broken uniformly randomly. We initialize every node with unique labels and let the labels propagate through the network. As the labels propagate, densely connected groups of nodes quickly reach a consensus on a unique label see Fig. 2. When many such dense consensus groups are created throughout the network, they continue to expand outwards until it is possible to do so. At the

end of the propagation process, nodes having the same labels are grouped together as one community. We perform this process iteratively, where at every step, each node updates its label based on the labels of its neighbors. The updating process can either be synchronous or asynchronous. In synchronous updating, node x at the iteration updates its label based on the labels of its neighbors at iteration t−1.

3. Near Linear Time Algorithm

Community detection and analysis is an important methodology for understanding the organization of various real-world networks and has applications in problems as diverse as consensus formation in social communities or the identification of functional modules in biochemical networks. Currently used algorithms that identify the community structures in large-scale real-world networks require a priori information such as the number and sizes of communities or are computationally expensive. In this paper we investigate a simple label propagation algorithm that uses the network structure alone as its guide and requires neither optimization of a pre-defined objective function nor prior information about the communities.



1: Initialize the labels at all nodes in the network. For a given node x, $C_x(0) = x$.

2: Set $t = 1$.

3: Arrange the nodes in the network in a random order and set it to X.

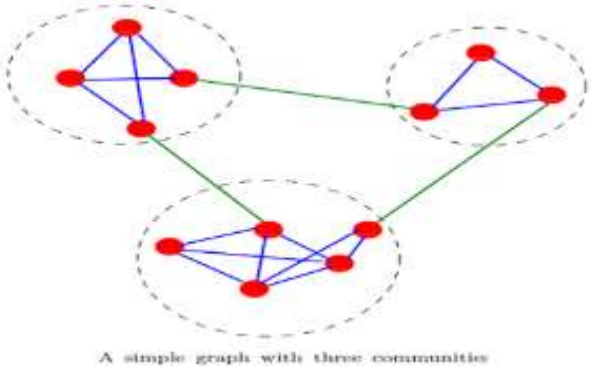
4: For each $x \in X$ chosen in that specific order, let $C_x(t) = f(C_{xi1}(t), ..., C_{xim}(t), C_{xi(m+1)}(t-1), ..., C_{xik}(t-1))$.
here returns the label occurring with the highest frequency among neighbors and ties are broken uniformly randomly.

5. If every node has a label that the maximum number of their neighbors have, then stop the algorithm. Else, set $t = t + 1$ and go to (3).

In this algorithm every node is initialized with a unique label and at every step each node adopts the label that most of its neighbors currently have. In this iterative process densely connected groups of nodes form a consensus on a unique label to form communities.

3.1 Community Deduction Algorithm

The community structure of the network is interpreted as the spin configuration that minimizes the energy of the spin glass with the spin states being the community indices. We elucidate the properties of the ground state configuration to give a concise definition of communities as cohesive subgroups in networks that is adaptive to the specific class of network under study. Further, we show how hierarchies and overlap in the community structure can be detected.

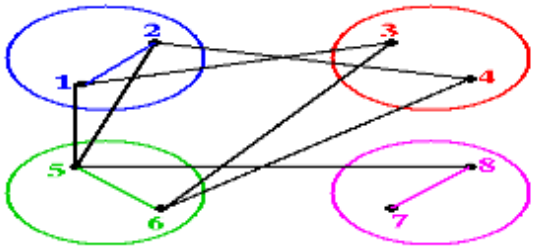


A simple graph with three communities

Computationally efficient local update rules for optimization procedures to find the ground state are given. We show how the ansatz may be used to discover the community around a given node without detecting all communities in the full network and we give benchmarks for the performance of this extension.

3.2 Network Partitioning Algorithm

A key step in network analysis is to partition a complex network into dense modules. Currently, modularity is one of the most popular benefit functions used to partition network modules. However, recent studies suggested that it has an inherent limitation in detecting dense network modules. In this study, we observed that despite the limitation, modularity has the advantage of preserving the primary network structure of the undetected modules.



Thus, we have developed a simple iterative Network Partition (iNP) algorithm to partition a network. The iNP algorithm provides a general framework in which any modularity-based algorithm can be implemented in the network partition step.

3.3 Localized Community Deduction Algorithm

Local network community detection is the task of finding a single community of nodes concentrated around few given seed nodes in a localized way. Conductance is a popular objective function used in many algorithms for local community detection. This paper studies a continuous relaxation of conductance. We show that continuous optimization of this objective still leads to discrete communities. We investigate the relation of conductance with weighted kernel k-means for a single community, which leads to the introduction of a new objective function, σ-conductance. Conductance is obtained by setting σ to 0.

Algorithm : PGDc,

Input: A set S of seeds of seeds, a graph G, a constant $\sigma \geq 0$.

```
s ← [S]
c(0) ← s
t ← 0
repeat
    γ(t) ← LineSearch(c(t))
    c(t+1) = p(c(t) − γ(t)∇ φσ(c(t)))
    t ← t + 1
until c(t−1) = c(t)
C ← {i ∈ V | c_i(t) ≥ 1/2}
```

function LineSearch(c)

```
γ* ← 0,  φ* ← φσ(c)
g ← ∇ φσ(c)
γ ← 1/ max(|g|)
repeat
    c' ← p(c − γg)
    if φσ(c') < φ* then
        γ* ← γ,  φ* ← φσ(c')
    end if
    γ ← 2γ
until c'_i ∈ {0, 1} for all i with g_i 6= 0
return γ*
```

Two algorithms, EMc and PGDc, are proposed to locally optimize σ-conductance and automatically tune the parameter σ. They are based on expectation maximization and projected gradient descent, respectively. We prove locality and give performance guarantees for EMc and PGDc for a class of dense and well separated communities centered around the seeds.

4. Results Analysis

To better understand the effectiveness of the proposed algorithms near linear time, community deduction algorithm, localizied community deduction algorithm, network partitioning algorithm extensive experimental results are reported in fig1

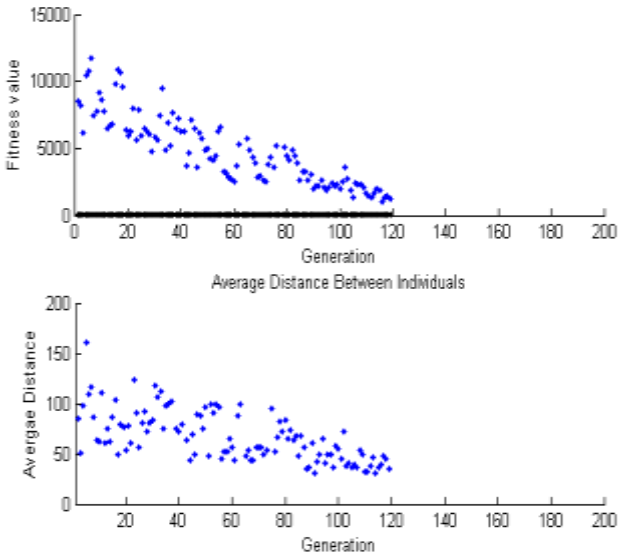


Fig 1: Distance between individuals

A. Multiple Community Structures:

To find the percentage of nodes classified in the same group in two different solutions, we form a matrix M, where Mij is the number of nodes common to community i in one solution and community j in the other solution. Then we calculate $fsame = 1/2 \sum imaxjMij + \sum jmaxiMij100/n.$

Given a network whose communities are already known, a community detection algorithm is commonly evaluated based on the percentage or number of nodes that are grouped into the correct communities [22,26]. fsame is similar, whereby fixing one solution we evaluate how close the other solution is to the fixed one and vice versa. While fsame can identify how close one solution is to another, it is, however, not sensitive to the seriousness of errors. For example, when few nodes from several different communities in one solution are fused together as a single community in another solution, the value of fsame does not change much. Hence we also use Jaccard's index which has been shown to be more sensitive to such differences between solutions [35]. If a stands for the pairs of nodes that are classified in the same community in both solutions, b for pairs of nodes that are in the same community in the first solution and different in the second, and c vice versa, then Jaccard's index is defined as a(a+b+c). It takes values between 0 and 1, with higher values indicating stronger similarity between the two solutions. Below Figure shows the similarities between solutions obtained from applying the algorithm five different times on the same network.

-	92.3	94.6	93.4	91.6
0.61	-	81.6	83.4	91
0.73	0.69	-	79.4	93.1
0.65	0.61	0.60	-	81.9
0.83	0.71	0.69	0.58	-

a) Product Network
Q=0.457-0.475

-	91.4	96.2	91.4	90.6
0.51	-	80.5	82.4	96
0.67	0.59	-	78.4	94.1
0.71	0.76	0.61	-	82.5
0.80	0.61	0.65	0.52	-

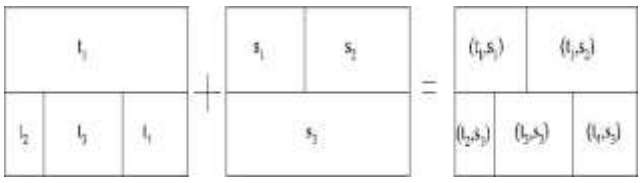
b) World Wide Web
Q=0.823 – 0.813

The given network, the ij[th] entry in the lower triangle of the table is the Jaccard index for solutions i and j, while the ij[th] entry in the upper triangle is the measure fsame for solutions i and j. We can see that the solutions obtained from the different runs are similar, implying that the proposed label propagation algorithm can effectively identify the community structure of any given network. Moreover, the tight range and high values

of the modularity measure Q obtained for the solutions it suggest that the partitions denote significant community structures.

B.**Aggregate**

It is difficult to pick one solution as the best among several different ones. Furthermore, one solution may be able to identify a community that was not discovered in the other and vice versa. Hence an aggregate of all the different solutions can provide a community structure containing the most useful information. In our case a solution is a set of labels on the nodes in the network and all nodes having the same label form a community. Given two different solutions, we combine them as follows; let C^1 denote the labels on the nodes in solution 1 and C2 denote the labels on the nodes in solution 2. Then, for a given node x, we define a new label as $Cx = Cx^1, Cx^2$. Starting with a network initialized with labels C we perform the iterative process of label propagation until every node in the network is in a community to which the maximum number of its neighbors belongs. As and when new solutions are available they are combined one by one with the aggregate solution to form a new aggregate solution. Note that when we aggregate two solutions, if a community T in one solution is broken into two or more different communities S1 and S2 in the other, then by defining the new labels as described above we are showing preferences to the smaller communities S1 and S2 over T. This is only one of the many ways in which different solutions can be aggregated. For other methods of aggregation used in community detection refer to [26,36,37]. It shows the similarities between aggregate solutions. The algorithm was applied on each network 30 times and the solutions were recorded. An ij^{th} entry is the Jaccard index for for the aggregate of the first 2i solutions with the aggregate of the first 2j solutions. We observe that the aggregate solutions are very similar in nature and hence a small set of solutions in this case can offer as much insight about the community structure of a network as can a larger solution set. In particular, the WWW network which had low similarities between individual solutions Jaccard index range 0.4883–0.5931, shows considerably improved similarities. Jaccard index range 0.6604–0.719 between aggregate solutions.



C^1 + C^2 = C

Aggregating two community structure solutions. t_1, t_2, t_3 and t_4 are labels on the nodes in a network obtained from solution 1 and denoted as C^1. The network is partitioned into groups of nodes having the same labels. s_1 , s_2 and s_3 are labels on the nodes in the same network obtained from solution 2 and denoted as C^2. All nodes that had label t_1 in solution 1 are split into two groups with each group having labels s_1 and s_2 respectively. While all nodes with labels t_3 or t_4 or t_5 in solution 1 have labels s_3 in solution 2. C represents the new labels defined from C^1 and C^2 .

C. Time Complexity

It takes a near-linear time for the algorithm to run to its completion. Initializing every node with unique labels requires O(n) time. Each iteration of this algorithm takes linear time in the number of edges (O (m)). At each node x, we first group the neighbors according to their labels ($O(d_x)$). We then pick the group of maximum size and assign its label to x, requiring a worst-case time of $O(d_x)$. This process is repeated at all nodes and hence an overall time is O(m) for each iteration.

5. Conclusion

In this proposed system a localized community detection algorithm based on label propagation .Customer having the same location are grouped together as communities. Product delivery will be very faster compared to normal delivery because of clustering of customer places. Communication is minimized and easy access of community data. Information passing can be done easier by community detection algorithm**.**

References

[1] Isabel M Kloumann and Jon M Kleinberg Community membership identification from small seed sets. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1366–1375. ACM, 2014.

[2] H. Peng, F. Long, and C. Ding ,IEEE Transactions on Pattern Analysis and Machine Intelligence,27,1226-1238 (2005).

[3] S. Gomez, P. Jensen and A. Arenas,Physical Review E 80, 016114 (2009).

[4] R. Albert and A.-L. Barabási," Statistical mechanics of complex networks" Rev. Mod. Phys. 74, 47 2002.

[5] Leon Danon,Albert Diaz-Guilera,Jordi Duch and Alex Arenas "Comparing Community Structure Identification" Journal of Statistical Mechanics:Theory and experiment,Volume 1(2006).

[6] Ronald L Breiger , Scott A Boorman,Phipps Arabie " An Algorithm For Clustering Relational Data With Applications to Social Network Analysis and Comparison with multidimensional scaling" Journal of Mathematical psychology volume 12,issue 3,August,pages 328-383.

[7] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, Proceedings of National Academy of Sciences 101, 2658 (2004).

[8] J. Eckermann and E. Moses, "Proceedings of National Academy Sciences" Springer , 5825 0369-8203 (2002).

[9] G. Flake, S. Lawrence, and C. Giles, Proceedings of the 6^{TH} ACM SIGKDD pp. 150–160 (2000).

[10] Qiong Chen, Ting-Ting Wu, and Ming Fang. Detecting local community structures in complex networks based on local degree central nodes. *Physica A: Statistical* Mechanics and its Applications, 392(3):529–537, 2013.

[11] Haim Avron and Lior Horesh. Community detection using time-dependent personalized pagerank. In Proceedings of The 32nd International Conference on Machine Learning, pages 1795–1803, 2015.

[12] Ullas Gargi, Wenjun Lu, Vahab S Mirrokni, and Sangho Yoon. Large-scale community detection on youtube for topic discovery and exploration. In ICWSM, 2011.

[13] Suzanne Hoppins, et al. A mitochondrial-focused genetic interaction map reveals a scaffold-like complex required for inner membrane organization in mitochondria. The Journal of Cell Biology, 195(2):323–340, 2011.