Title

Basic component of analysis data mining social network graph

(Assignment -3 …Spring -2022)

(July 14, 2022)

BY

Name: **Zeeshan Ali**

Roll No: **43**

Course Code (Course title):

BS Computer Science/IT:

Submitted To

Name

Sir Kamran

Department: CS/IT

Minhaj University Lahore

Table of Content

Table of Contents

Introduction:

- ✓ **Data mining**

Data mining is **the process of sorting through large data sets to identify patterns and relationships that can help solve business problems through data analysis**. Data mining techniques and tools enable enterprises to predict future trends and make more-informed business decisions.

- ✓ **Data Mining Architecture Components**
    - Sources of Data. The place where we get our data to work upon is known as the data source or the source of the data. ...
    - Database or Data Warehouse Server. ...
    - Data Mining Engine. ...
    - Modules for Pattern Evaluation. ...
    - GUI or Graphical User Interface. ...
    - Knowledge Base.

- ✓ **What is a Social Network?**

When we think of a social network, we think of Facebook, Twitter, Google+, or another website that is called a "social network," and indeed this kind of network is representative of the broader class of networks called "social." The essential characteristics of a social network are:

1. There is a collection of entities that participate in the network. Typically, these entities are people, but they could be something else entirely. We shall discuss some other examples in Section 10.1.3.

2. There is at least one relationship between entities of the network. On

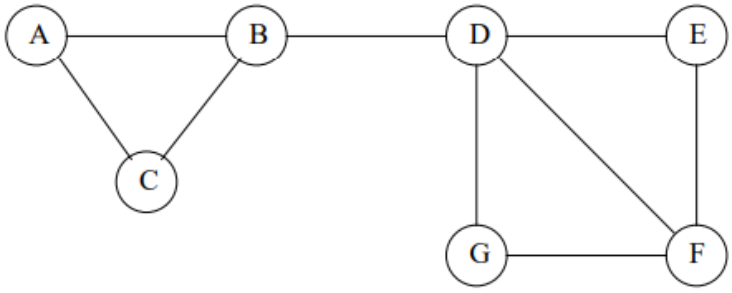
Facebook or its ilk, this relationship is called friends. Sometimes the relationship is all-or-nothing; two people are either friends or they are not. However, in other examples of social networks, the relationship has a degree. This degree could be discrete; e.g., friends, family, acquaintances, or none as in Google+. It could be a real number; an example would be the fraction of the average day that two people spend talking to each other.

3. There is an assumption of nonrandomness or locality. This condition is

the hardest to formalize, but the intuition is that relationships tend to cluster. That is, if entity A is related to both B and C, then there is a higher probability than average that B and C are related.

1.1. Social Networks as Graphs

Social networks are naturally modeled as graphs, which we sometimes refer to as a social graph. The entities are the nodes, and an edge connects two nodes if the nodes are related by the relationship that characterizes the network. If there is a degree associated with the relationship, this degree is represented by labeling the edges. Often, social graphs are undirected, as for the Facebook friends graph. But they can be directed graphs, as for example the graphs of followers on Twitter or Google+.



1.1.1 Varieties of Social Networks

There are many examples of social networks other than “friends” networks.

Here, let us enumerate some of the other examples of networks that also exhibit

locality of relationships.

1.1.2 Telephone Networks

Here the nodes represent phone numbers, which are really individuals. There is an edge between two odes if a call has been placed between those phones in some fixed period of time, such as last month, or “ever.” The edges could be weighted by the number of calls made between these phones during the

period. Communities in a telephone network will form from groups of people that communicate frequently: groups of friends, members of a club, or people working at the same company, for example.

1.1.3 Email Networks

The nodes represent email addresses, which are again individuals. An edge represents the fact that there was at least one email in at least one direction between the two addresses. Alternatively, we may only place an edge if there were emails in both directions. In that way, we avoid viewing spammers as “friends” with all their victims. Another approach is to label edges as weak or strong. Strong edges represent communication in both directions, while weak edges indicate that the communication was in one direction only. The communities seen in email networks come from the same sorts of groupings we mentioned in connection with telephone networks. A similar sort of network involves people who text other people through their cell phones.

1.1.4 Collaboration Networks

Nodes represent individuals who have published research papers. There is an edge between two individuals who published one or more papers jointly. Optionally, we can label edges by the number of joint publications. The communities in this network are authors working on a

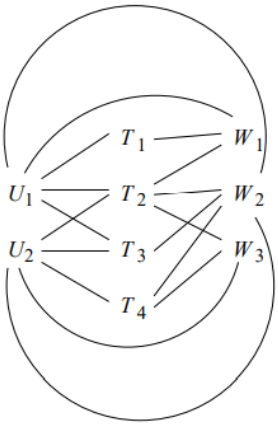
particular topic. An alternative view of the same data is as a graph in which the nodes are papers. Two papers are connected by an edge if they have at least one author in common. Now, we form communities that are collections of papers on the same topic. There are several other kinds of data that form two networks in a similar way. For example, we can look at the people who edit Wikipedia articles and the articles that they edit. Two editors are connected if they have edited an article in common. The communities are groups of editors that are interested in the same subject. Dually, we can build a network of articles, and connect articles if they have been edited by the same person. Here, we get communities of articles on similar or related subjects.

1.1.5 Other Examples of Social Graphs

Many other phenomena give rise to graphs that look something like social graphs, especially exhibiting locality. Examples include: information networks (documents, web graphs, patents), infrastructure networks (roads, planes, water pipes, powergrids), biological networks (genes, proteins, food-webs of animals eating each other), as well as other types, like product co-purchasing networks (e.g., Groupon).

1.1.6 Graphs With Several Node Types

There are other social phenomena that involve entities of different types. We just discussed under the heading of "collaboration networks," several kinds of graphs that are really formed from two types of nodes. Authorship networks can be seen to have author nodes and paper nodes. In the discussion above, we built two social networks by eliminating the nodes of one of the two types, but we do not have to do that. We can rather think of the structure as a whole. For a more complex example, users at a site like del.icio.us place tags on Web pages. There are thus three different kinds of entities: users, tags, and pages. We might think that users were somehow connected if they tended to use the same tags frequently, or if they tended to tag the same pages. Similarly, tags could be considered related if they appeared on the same pages or were used by the same users, and pages could be considered similar if they had many of the same tags or were tagged by many of the same users.



1.1.7 Clustering of Social-Network Graphs

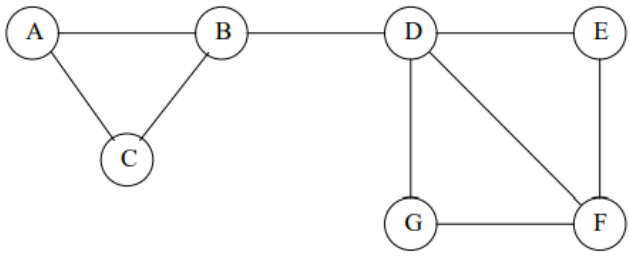
An important aspect of social networks is that they contain communities of entities that are connected by many edges. These typically correspond to groups of friends at school or groups of researchers interested in the same topic, for example. In this section, we shall consider clustering of the graph as a way to identify communities. It turns out that the techniques we learned in Chapter 7 are generally unsuitable for the problem of clustering social-network graphs.

1.2 Distance Measures for Social-Network Graphs

If we were to apply standard clustering techniques to a social-network graph, our first step would be to define a distance measure. When the edges of the graph have labels, these labels might be usable as a distance measure, depending on what they represented. But when the edges are unlabeled, as in a "friends" graph, there is not much we can do to define a suitable distance. Our first instinct is to assume that nodes are close if they have an edge between them and distant if not. Thus, we could say that the distance d(x, y) is 0 if there is an edge (x, y) and 1 if there is no such edge. We could use any other two values, such as 1 and ∞, as long as the distance is closer when there is an edge. Neither of these two-valued "distance measures" – 0 and 1 or 1 and ∞ – is a true distance measure. The reason is that they violate the triangle inequality when there are three nodes, with two edges between them. That is, if there are edges (A, B) and (B, C), but no edge (A, C), then the distance from A to C exceeds the sum of the distances from A to B to C. We could fix this problem by using, say, distance 1 for an edge and distance 1.5 for a missing edge. But the problem with two-valued distance functions is not limited to the triangle inequality, as we shall see in the next section.

1.2.1 Applying Standard Clustering

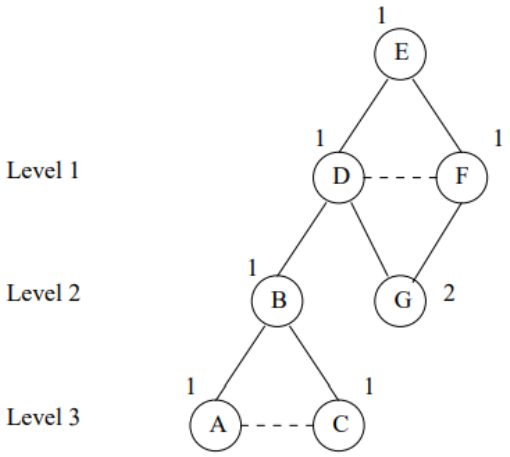
Methods Recall from Section 7.1.2 that there are two general approaches to clustering: hierarchical (agglomerative) and point-assignment. Let us consider how each of these would work on a social-network graph. First, consider the hierarchical methods covered in Section 7.2. In particular, suppose we use as the intercluster distance the minimum distance between nodes of the two clusters. Hierarchical clustering of a social-network graph starts by combining some two nodes that are connected by an edge. Successively, edges that are not between two nodes of the same cluster would be chosen randomly to combine the clusters to which their two nodes belong. The choices would be random, because all distances represented by an edge are the same.



1.2.2 The Girvan-Newman Algorithm

In order to exploit the betweenness of edges, we need to calculate the number of shortest paths going through each edge. We shall describe a method called the Girvan-Newman (GN)

Algorithm, which visits each node X once and computes the number of shortest paths from X to each of the other nodes that go through each of the edges. The algorithm begins by performing a breadth-first search (BFS) of the graph, starting at the node X. Note that the level of each node in the BFS presentation is the length of the shortest path from X to that node. Thus, the edges that go between nodes at the same level can never be part of a shortest path from X. Edges between levels are called DAG edges ("DAG" stands for directed, acyclic graph). Each DAG edge will be part of at least one shortest path from root X. If there is a DAG edge (Y, Z), where Y is at the level above Z (i.e., closer to the root), then we shall call Y a parent of Z and Z a child of Y , although parents are not necessarily unique in a DAG as they would be in a tree.



Example : Figure 10.4 is a breadth-first presentation of the graph of Fig. 10.3, starting at node E. Solid edges are DAG edges and dashed edges connect nodes at the same level. ✷ The second step of the GN algorithm is to label each node by the number of shortest paths that reach it from the root. Start by labeling the root 1. Then, from the top down, label each node Y by the sum of the labels of its parents.

1.3 Direct Discovery of Communities

In the previous section we searched for communities by partitioning all the individuals in a social network. While this approach is relatively efficient, it does have several limitations. It is not possible to place an individual in two different communities, and everyone is assigned to a community. In this section, we shall see a technique for discovering communities directly by looking for subsets of the nodes that have a relatively large number of edges among them. Interestingly, the technique for doing this search on a large graph involves finding large frequent itemsets.

1.3.1 Finding Cliques

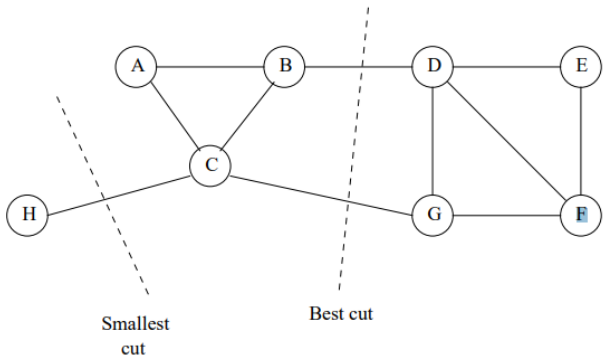
Our first thought about how we could find sets of nodes with many edges between them is to start by finding a large clique (a set of nodes with edges between any two of them). However, that task is not easy. Not only is finding maximal cliques NP-complete, but it is among the hardest of the NP-complete problems in the sense that even approximating the maximal clique is hard. Further, it is possible to have a set of nodes with almost all edges between them, and yet have only relatively small cliques.

1.4 Partitioning of Graphs

In this section, we examine another approach to organizing social-network graphs. We use some important tools from matrix theory ("spectral methods") to formulate the problem of partitioning a graph to minimize the number of edges that connect different components. The goal of minimizing the "cut" size needs to be understood carefully before proceeding. For instance, if just joined Facebook, you are not yet connected to any friends. We do not want to partition the friends graph with you in one group and the rest of the world in the other group, even though that would partition the graph without there being any edges that connect members of the two groups. This cut is not desirable because the two components are too unequal in size.

What Makes a Good Partition? Given a graph, we would like to divide the nodes into two sets so that the cut, or set of edges that connect nodes in different sets is minimized. However, we also want to constrain the selection of the cut so that the two sets are approximately equal in size. The next example illustrates the point.

Example: Recall our running example of the graph in Fig. 10.1. There, it is evident that the best partition puts {A, B, C} in one set and {D, E, F, G} in the other. The cut consists only of the edge (B, D) and is of size 1. No nontrivial cut can be smaller.



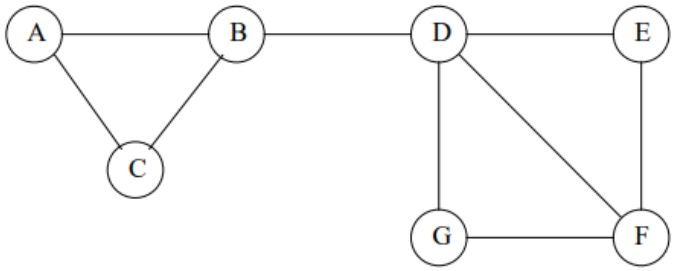
In Fig. is a variant of our example, where we have added the node H and two extra edges, (H, C) and (C, G). If all we wanted was to minimize the size of the cut, then the best choice would be to put H in one set and all the other nodes in the other set. But it should be apparent that if we reject partitions where one set is too small, then the best we can do is to use the cut consisting of edges (B, D) and (C, G), which partitions the graph into two equal-sized sets {A, B, C, H} and {D, E, F, G}

1.4.1 Normalized Cuts

A proper definition of a "good" cut must balance the size of the cut itself against the difference in the sizes of the sets that the cut creates. One choice 10.4. PARTITIONING OF GRAPHS 375 that serves well is the "normalized cut." First, define the volume of a set S of that serves well is the "normalized cut." First, define the volume of a set S of nodes, denoted Vol(S), to be the number of edges with at least one end in S. Suppose we partition the nodes of a graph into two disjoint sets S and T . Let Cut(S, T) be the number of edges that connect a node in S to a node in T . Then the normalized cut value for S and T is Cut(S, T) Vol(S) + Cut(S, T) Vol(T) Example 10.15 : Again consider the graph of Fig. 10.11. If we choose S = {H} and T = {A, B, C, D, E, F, G}, then Cut(S, T) = 1. Vol(S) = 1, because there is only one edge connected to H. On the other hand, Vol(T) = 11, because all the edges have at least one end at a node of T . Thus, the normalized cut for this partition is 1/1 + 1/11 = 1.09. Now, consider the preferred cut for this graph consisting of the edges (B, D) and (C, G). Then S = {A, B, C, H} and T = {D, E, F, G}. Cut(S, T) = 2, Vol(S) = 6, and Vol(T) = 7. The normalized cut for this partition is thus only 2/6 + 2/7 = 0.62.

1.4.2 Some Matrices That Describe Graphs

To develop the theory of how matrix algebra can help us find good graph partitions, we first need to learn about three different matrices that describe aspects of a graph. The first should be familiar: the adjacency matrix that has a 1 in row i and column j if there is an edge between nodes i and j, and 0 otherwise.



Example: We repeat our running example graph in Fig. 10.12. Its adjacency matrix appears in Fig. 10.13. Note that the rows and columns correspond to the nodes A, B, . . . , G in that order. For example, the edge (B, D) is reflected by the fact that the entry in row 2 and column 4 is 1 and so is the entry in row 4 and column 2. ✷ The second matrix we need is the degree matrix for a graph. This graph has nonzero entries only on the diagonal. The entry for row and column i is the degree of the ith node.

1.5 Finding Overlapping Communities

So far, we have concentrated on clustering a social graph to find communities. But communities are in practice rarely disjoint. In this section, we explain a method for taking a social graph and fitting a model to it that best explains how it could have been generated by a mechanism that

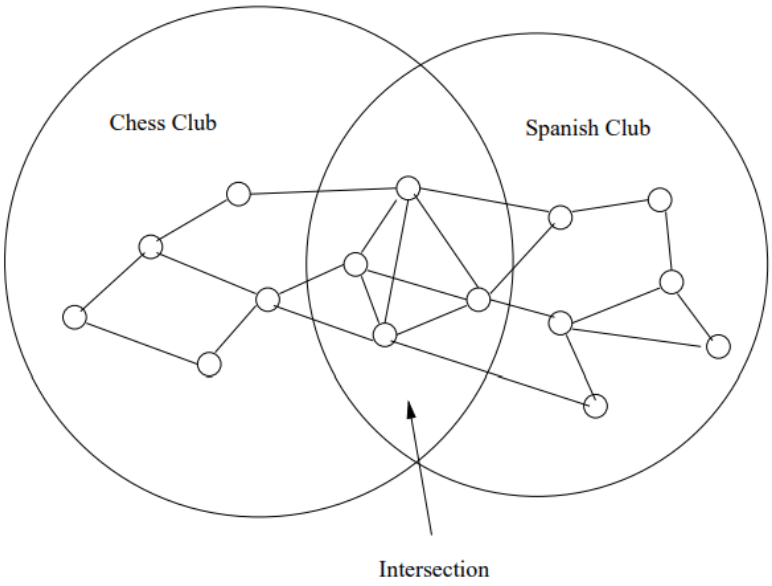
assumes the probability that two individuals are connected by an edge (are “friends”) increases as they become members of more communities in common. An important tool in this analysis is “maximum-likelihood estimation,” which we shall explain before getting to the matter of finding overlapping communities.

1.5.1 The Nature of Communities

To begin, let us consider what we would expect two overlapping communities to look like. Our data is a social graph, where nodes are people and there is an edge between two nodes if the people are “friends.” Let us imagine that this graph represents students at a school, and there are two clubs in this school: the Chess Club and the Spanish Club. It is reasonable to suppose that each of these clubs forms a community, as does any other club at the school. It is also reasonable to suppose that two people in the Chess Club are more likely to be friends in the graph because they know each other from the club. Likewise, if two people are in the Spanish Club, then there is a good chance they know each other, and are likely to be friends. What if two people are in both clubs? They now have two reasons why they might know each other, and so we would expect an even greater probability that they will be friends in the social graph. Our conclusion is that we expect edges to be dense within any community, but we expect edges to be even denser in the intersection of two communities, denser than that in the intersection of three communities, and so on. The idea is suggested by Fig. 10.19.

1.5.2 Maximum-Likelihood Estimation

Before we see the algorithm for finding communities that have overlap of the kind suggested in Section 10.5.1, let us digress and learn a useful modeling tool called maximum-likelihood estimation, or MLE. The idea behind MLE is that we make an assumption about the generative process (the model) that creates instances of some artifact, for example, “friends graphs.” The model has parameters that determine the probability of generating any particular instance of the artifact; this probability is called the likelihood of those parameter values. We assume that the value of the parameters that gives the largest value of the likelihood is the correct model for the observed artifact.



1.5.3 The Affiliation-Graph Model

We shall now introduce a reasonable mechanism, called the affiliation-graph model, to generate social graphs from communities. Once we see how the parameters of the model influence the likelihood of seeing a given graph, we can address how one would solve for the values of the parameters that give the maximum likelihood. The mechanism, called community-affiliation graphs.

1. There is a given number of communities, and there is a given number of individuals (nodes of the graph).

2. Each community can have any set of individuals as members. That is, the memberships in the communities are parameters of the model.

3. Each community C has a probability pC associated with it, the probability that two members of community C are connected by an edge because they are both members of C. These probabilities are also parameters of the model.

4. If a pair of nodes is in two or more communities, then there is an edge between them if any of the communities of which both are members justifies that edge according to rule (3).

1.6 Simrank

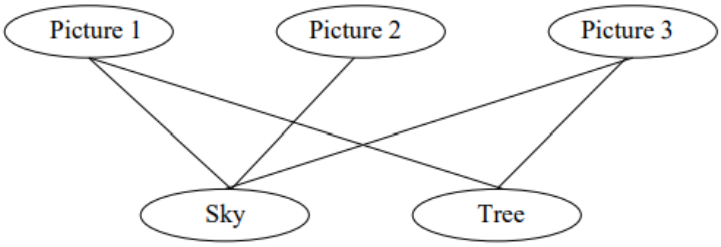
In this section, we shall take up another approach to analyzing social-network graphs. This technique, called “simrank,” applies best to graphs with several types of nodes, although it can in principle be applied to any graph. The purpose of simrank is to measure the similarity between nodes of the same type, and it does so by seeing where random walkers on the graph wind up when starting at a particular node. Because calculation must be carried out once for each starting node, it is limited in the sizes of graphs that can be analyzed completely in this manner.

1.6.1 Random Walkers on a Social Graph

Recall our view of PageRank in Section 5.1 as reflecting what a "random surfer" would do if they walked on the Web graph. We can similarly think of a person "walking" on a social network. The graph of a social network is generally undirected, while the Web graph is directed. However, the difference is unimportant. A walker at a node N of an undirected graph will move with equal probability to any of the neighbors of N (those nodes with which N shares an edge).

1.6.2 Random Walks with Restart

We see from the observations above that it is not possible to measure similarity to a particular node by looking at the limiting distribution of the walker. However, we have already seen, in Section 5.1.5, the introduction of a small probability that the walker will stop walking at random. Later, we saw in Section 5.3.2 that there were reasons to select only a subset of Web pages as the teleport set, the pages that the walker would go to when they stopped surfing the Web at random.



References

file:///C:/Users/Zeeshan%20Ali/Desktop/data%20mining.pdf

https://www.researchgate.net/publication/257495235_Data_Mining_of_Social_Networks_Represented_as_Graphs

https://preventviolentextremism.info/data-mining-social-networks-represented-graphs