Review on Practical Graph Mining with R

May 18, 2015

1 Chapter 1 on May 18, 2015

1. A graph is a collection of individual objects interconnected in some way. Graph data analytics refer to the extraction of insightful and actionable knowledge from graph data.

2. Applications of graph mining

   - Web graphs, e.g.., page ranks.
   - Social science graphs, which models the relationships among individuals and organizations. Links represent friendship, political alliance, professional collaboration, etc.
   - Computer networking graphs represent interconnections among various routers across the Internet.
   - Homeland security and cybersecurity graphs
   - Biological graphs
   - Chemical graphs
   - Finance graphs. The structure of stock markets and trading records can be represented using graphs, e.g., nodes are brokers, banks, and customers; links capture the financial trading information. A sequence of such graphs over a period of time can be mined to detect people involved in financial frauds, to predict which stocks will be on the rise, and to distinguish stock purchasing patterns that may lead to profits or losses.
   - Healthcare graphs

3. Dimensionality reduction in general refers to the problem of reducing the data amount while preserving the essential or more salient characteristics of the data. In terms of graph, dimensionality reduction refers to the problem of transforming graphs into low-dimensional vectors so that graph features and similarities are preserved.

2 Chapter 2 on May 18, 2015

1. A graph is a theoretical construct composed of points (i.e., vertices) connected by lines (i.e., edges). Graphs are structural data. The vertices of a graph symbolize discrete pieces of information, while the edges of a graph symbolize the relationships between those pieces.

2. A vertex is also called a node. It is a single point in a graph. Vertices are usually labeled. An edge can be regarded as a line connecting two vertices. Edges may have labels as well. Vertices and edges are the basic building blocks of graphs.

   - A graph G is composed of two sets, i.e., a set of vertices, denoted as $V(G)$; and a set of edges, denoted as $E(G)$.
   - An edge in a graph G is an unordered pair of two vertices (v_1, v_2) such that $v_1 \in V(G)$ and $v_2 \in V(G)$.
   - An edge is said to join its two vertices. Likewise, two vertices are said to be adjacent if and only if there is an edge between them. Two vertices are said to be connected if there is a path between one to the other via any number of edges.
   - A loop is an edge that joins a vertex to itself.
   - An edge is a multiple edge if there is another edge in $E(G)$ which joins the same pair of vertices.
   - Note that multiple edges and loops often make manipulating graphs more difficult. Many proofs and algorithms in graph theory require them to be excluded.
   - A *simple* graph is a graph with no loops or multiple edges.
   - An edge always has exactly two vertices, but a single vertex can be an endpoint for zero, one, or many edges. The *degree* of a vertex v, denoted as $degree(v)$, is the number of times v occurs as an endpoint for the edges $E(G)$. It indicates that the dgree of a vertex is the number of edges leading to it. Note that a loop adds 2 to the degree of a vertex.

3. A subgraph S of a graph G is a set of vertices.

   - A set of vertices $V(S) \subset V(G)$
   - A set of edges $E(S) \subset E(G)$. Every edge in $E(S)$ must be an unordered pair of vertices (v_1, v_2) such that $v_1 \in V(S)$ and $v_2 \in V(S)$. It indicates that an edge can only be part of a subgraph if both its endpoints are part of the subgraph.

4. A subgraph can be induced (i.e., induced graph) in two ways, i.e., by vertices and edges.

5. Two graphs G and H are *isomorphic*, denoted as $G \simeq H$, if there exists a bijection $f : V(G) \to V(H)$ such that an edge $(v_1, v_2) \in E(G)$ if and only if $(f(v_1), f(v_2)) \in E(H)$. Informally, it indicates that two graphs are isomorphic if they can be drawn in the same shape. If G and H are isomorphic, the bijection f is said to be an isomorphism between G and H and between H and G. The isomorphism class of G is all graphs isomorphic to G.

6. Note that when vertices and edges have labels, the notion of *sameness* and *isomorphism* are different. Two graphs having the same structure and thus *isomorphic*, but have different labels, and are thus not exactly the same. Labeled graphs are i*isomorphic* if their underlying unlabeled graphs are *isomorphic*.

7. An automorphism between graphs G and H is an isomorphism f that maps G onto iteself. The automorphism class of G is all graphs automorphic to G. It indicates that the graph structures are the same and the labels are the same too. All automorphisms are isomorphisms, but not all isomorphisms are automorphisms.

8. One common problem that is often encountered in graph mining is the subgraph isomorphism problem. The subgraph isomorphism problem asks if, given two graphs G and H, does G contain a subgraph isomorphic to H. That is, given a larger graph G and a smaller graph H, whether there is a subgraph in G that is the same shape as H. The problem is NP-complete, meaning it is computationally expensive.

9. A directed graph or digraph D is composed of two sets

   - A set of vertices $V(D)$
   - A set of edges $E(D)$, such that each edge is an ordered pair of vertices (t, h). The first vertex t is called the tail, while the latter vertex h is called the head. The edge in a directed graph is usually drawn as an arrow with the arrow-head pointing towards the head vertex.
   - Indegree of a vertex v is the number of edges in $E(D)$ which have v as the head.
   - Outdegree of a vertex v is the number of edges in $E(D)$ which have v as the tail.
   - Digraph isomorphism specifies that two digraphs J and K are isomorphic if and only if their underlying undirected graphs are isomorphic. So similar to labeled graphs, the direction of edges are not considered when considering isomorphism. Two digraphs are isomorphic if you can change all the directed edges to undirected edges and then draw them in the same shape.

10. Families of graphs

- A clique is a set of vertices which are all connected to each other by edges. A set of vertices C is a clique in the graph G if for all pairs of vertices $v_1 \in C$ and $v_2 \in C$, there exists an edge $(v_1, v_2) \in E(G)$. If begin at one vertex in a clique, you can get to any other member of that clique by following only one edge.
- A complete graph with n vertices, denoted as K_n, is a graph such that $V(K_n)$ is a clique.
- A path of length n, denoted as P_n, in a graph G is an ordered set of edges $(v_0, v_1), (v_1, v_2), (v_3, v_4), \cdots, (v_{n-1}, v_n)$ such that each edge $e \in E(G)$. A path is sometimes called a walk. Note that there may be more than one path between the same two vertices in a graph.
- A path is closed if its first and last vertices are the same. A cycle of length n, denoted as C_n, in a graph G is a closed path of length n. Note that a simple cycle is the same as a simple closed path, with the exception that it may visit one vertex exactly twice, i.e., the vertex which is both the start and end of the cycle.
- Trees are graphs that obey certain structural rules and have many appealing mathematical properties. A tree graph has exactly one vertex as the root, i.e., parent. It can have any number of children vertices adjacent to it. Those children can in turn, be parent of their own children vertices. A vertex having no children is called a leaf.
  - A graph G is a tree if and only if there is exactly one simple path from each vertex to every other vertex. Similarly, it can be defined as a graph G if and only if it is a connect graph with no cycles. Trees are often modeled as directed graphs. Hence, the root has an indegree of 0. All the other vertices have indegree of exactly 1. Leaf vertices have an outdegree of 0.

11. A weighted graph W is composed of two sets
    - A set of vertices $V(W)$
    - A set of edges $E(W)$ such that each edge is a pair of vertices v_1 and v_2 and a numeric weight w.
    - Weighted graphs are often used in path-finding problems.
12. Graph representations
    - Adjacency list
      - It is the simplest and most compact way to represent a graph
      - Given a graph G such that $V(G) = \{v_1, v_2, \cdots, v_n\}$, the adjacency list representation of G is a list of length n such that the i^{th} element of the list is a list that contains one element for each vertex adjacent to v_i.

- Adjacency matrix: It is a $n \times b$ matrix and the cell entry is either 1 or 0, indicating whether two vertices are adjacent to each other or not.
- Incidence matrix: Given a graph G such that $V(G) = \{v-1, v-2, \cdots, v_n\}$ and $E(G) = \{e_1, e_2, \cdots, e_m\}$, the incidence matrix is an $n \times m$ matrix. Let $a_{r,c}$ represent the matrix value at row r and column c.
  - If G is undirected, $a_{r,c} = 1$ if v_r is the head or tail of e_c; otherwise $a_{r,c} = 0$.
  - If G is directed, $a_{r,c} = -1$ if v_r is the tail of e_c; $a_{r,c} = 11$ if v_r is the head of e_c; $a_{r,c} = 0$ if e_c is a loop; otherwise $a_{r,c} = 0$.

3 Chapter 4 on May 18, 2015

1. Kernel methods refers to the transformation of data in a problem's input space into a high-dimensional feature space, allowing the algorithms to be performed on the transformed space, i.e., feature space.
2. The implicit transformation can be used by replacing the operations in the analysis algorithm itself with operations corresponding to a different feature space. Noted that the inner product of vectors in feature space can be obtained by using the inner product of the original space. The algorithm can be represented using the inner products and thereby, there is no need to compute the actual feature space at all. This is the kernel trick, and the function used to computes the inner products is the kernel function. Kernel function must be symmetric, i.e., $K(d_i, d_j) = K(d_j, d_i)$ and postive semi-definite. It can be interpreted as a measurement of similarity between pairs of input data.
   - Polynomial $(x \cdot y + \theta)^d$
   - Gaussian RBF $e^{\frac{|x-y|^2}{c}}$
   - Sigmoidal $tanh(\alpha(x \cdot y) + \theta)$