Some Notes on Graph Mining

May 22, 2015

1 Into2GraphMining

1. A graph is said to be connected if there is path between every pair of vertices
2. Two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ are said to be isomorphic if they are topologically identicle, which means a mapping from V_1 to V_2 exists so that each edge E_1 is mapped to a single edge in E_2 and vice-versa.
3. Frequent subgraph mining (FSM)
   - Given a set of undirected and labeled graphs (D) and a support threshold σ, find all connected and undirected graphs that are subgraphs in at least $\sigma \times D$ of input graphs.

2 Complex networks tools for analyzing networks (R+igraph)

1. `igraph` can be used to handle undirected and directed graphs. It includes implementations for classic graph theory problems like minimum spanning trees and network flow and community structure search.
2. Procefures for analyzing network
   - Create a graph object
   - Layout the network: use `igraph: tkplot`
   - Ranking: use `igraph: page.rank`
   - Metrics
     - `igraph: diameter(g)`
     - `igraph: graph.density(g)`, i.e., $\frac{No.edges}{No.vertex \times (No.vertex-1)}$
     - `igraph: average.path.length(g)`
     - `igraph: transitivity(g)`
   - Community detection
   - Export

3 Practical statistical network analysis (with R and igraph)

1. `igraph` is for classic graph theory and network science. Its core functionality is implemented in C and has high level interfaces with *R* and *Python*.
2. Note that in the old version of `igraph`, vertices are always numbered from zero.
3. Name vertices: `V(g)$name`
4. Graph representations
   - Adjacency matrix
   - Edge list
   - Adjacency list
5. Some metrics
   - degree
   - closeness
   - betweenness
   - eigenvector centrality
   - page rank

4 Graph and web mining - motivation, applications and algorithms

1. The structure of the data is just as important as its content
2. The discovered pattern can be used as compact representation of the information, find strongly connected groups and etc.
3. Frequent patterns refer to a set of items, subsequences, and substructures that occur frequently in a data set.
4. Motivations for graph mining
   - Most of existing DM algorithms are based on flat transaction representation, i.e., sets of items.
   - Data with structures, layers, hierarchy or geometry often do not fit well in this flat transaction setting.
5. Graph mining is essentially the problem of discovering repetitive subgraphs occuring in the input graphs.

6. The main difference between association rules and graph patterns is that gaph patterns are topology-based, which means graph patterns have structure in addition to atomic values.

7. Graph mining

   - Frequent subgraph mining
     - Apriori-based, e.g., AGM, FSG, PATH
     - Pattern growth-based, e.g., gSpan, MoFa, GASTO, FFSM, SPIN
     - Approximate methods, e.g., SUBDUE, GBI
   - Variant subgraph pattern mining
     - Closed subgraph mining, e.g., CloseGraph
     - Coherent subgraph mining, e.g., CSA, CLAN
     - Dense subgraph mining, e.g., CloseCut, Splat, CODENS
   - Applications of FSM
     - Clustering
     - Classification, e.g., kernel methods (graph kernels)
     - Indexing and search, e.g., gIndex

5 Introduction to igraph

1. Creating a graph

   - Attributes include color and weight
   - `plot(g, edge.width=2+3*E(g)$weight, vertex.label=NA, vertex.size=2)`

2. Measuring graphs

   - `diameter`
   - `transitivity`: cluster coefficient or transitivity
   - `average.path.length`
   - `degree`
   - `degree.distribution`