Some Notes on Graph Mining

May 24, 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`

6 Community detection algorithms in igraph

1. Algorithms

   - `edge.between.community`, 2004
   - `fashgreedy.community`, 2004, i.e., modularity optimization method
   - `label.propagation.community`, 2007
   - `leading.eigenvector.community`, 2006

- `multilevel.community`, 2008, i.e., the Louvain method
- `optimal.community`, 2008
- `singlass.community`, 2006
- `walkstrap.community`, 2005
- `infomap.community`, 2008

2. Evaluation criteria

   - `modularity`
   - `conductance`
   - `cut ratio`
   - `expansion`

7 Graph mining and graph kernels

1. FSM algorithms

   - Apriori-based approaches, e.g., AGM/AcGM, FSG, PATH, FFSM, FTOSM
   - Pattern growth approaches, e.g., SUBDUE, gSpan, MoFa, Gaston, CMTreeMiner, LEAP

2. For Apriori-based approaches, the logic behind is that if a graph is frequent, all of its subgraphes are frequent.

3. Properties of graph mining algorithms

   - Search order: breadth or depth, complete or incomplete
   - Candidate generation mechanism: apriori or pattern growth
   - Discovery order of patterns: DFS oder, or $path \rightarrow tree \rightarrow graph$
   - Elimination of duplicate subgraphs: passive or active
   - Support calculation: embedding store or not

4. Pattern summarization aims to use a small set of representative patterns which preserve most of the information to represent the original data

5. A frequent graph G is closed if there exists no supergraph of G that carries the same support of G.

6. A frequent graph G is maximal if there exists no supergraph of G that is frequent.

7. Graph kernels aim to compute similarity scores between graphs

8. Two graphs G_1 and G_2 are said to be isomorphic if there is a mapping function f, such that for each edge (x, y) in G_1, there is a corresponding edge in G_2 and it is $(f(x), f(y))$. f is said to be the isomorphism. Note that there is no polynomial-time algorithm in solving graph isomorphism. It is known to be NP-complete.

9. Subgraph isomorphism asks if there is a subset of edges and vertices of G_1 that is isomorphic to a smaller graph G_2. Subgraph isomorphism is also NP-complete.

10. Graph edit distances

    - The principle is to count the operations to transform G_1 to G_2. Assign costs to different types of operations, including edge/node insertion or deletion, modification of labels.
    - It can partially capture the similarities between graphs. It allos for noise in the nodes, edges and their labels.
    - One disadvantage is that it has to contain a subgraph isomorphism chec step as one intermediate step. In addition, choosing cost function for different operations can be difficult.

11. Topological descriptors

    - The principle is to map each graph to a feature vector and use distances and metrics on vector for learning in graphs.
    - The advantage is that it can utilize the tools for feature vectors
    - The disadvantage is that the feature vector transformation leads to information loss

12. Polynomial alternatives

    - Graph kernels compare the substructures in polynomial time
    - A good graph kernel should be expressive, efficient to compute, positive definite, and applicable to wide range of graphs