Community Detection: From Plain to Attributed Complex Networks

Martin Atzmueller

University of Kassel, Research Center for Information System Design
Ubiquitous Data Mining Team, Chair for Knowledge and Data Engineering

Web Science 2016, Hannover - 2016-05-22
Exploratory Data Analysis

- Different aspects & perspectives
  - Hypothesis generating
  - Visualization & Analytics
  - Semi-automatic & Interactive
  - Detect local models

- Approaches & methods
  - Local exceptionality detection
  - Community detection
  - Description-oriented community detection
Pattern

- Merriam Webster: "A repeated form or design especially that is used to decorate something"

- Oxford: "An arrangement or design regularly found in comparable objects"

- Pattern in data mining [Bringmann et al. 2011]
  - Captures regularity in the data
  - Describes part of the data
Attributed Graphs

- Additional information (on nodes, edges)
- E.g., "knowledge graph"
Homophily (i.e. "Love of the same")

- Sociology: "Birds of a feather flock together" (Lazarsfield & Merton 1954)

- Social Networks: "Similarity breeds connection": A connection between similar people occurs at a higher rate than between dissimilar ones. (McPherson et al. 2001)
FIG. 4 The two best studied information networks. Left: the citation network of academic papers in which the vertices are papers and the directed edges are citations of one paper by another. Since papers can only cite those that came before them (lower down in the figure) the graph is acyclic—it has no closed loops. Right: the World Wide Web, a network of text pages accessible over the Internet, in which the vertices are pages and the directed edges are hyperlinks. There are no constraints on the Web that forbid cycles and hence it is in general cyclic.

(Newman 2003)
Real-World System I: BibSonomy

- Users assign tags to resources
  - Organize
  - Share
  - Categorize

http://www.bibsonomy.org

### Bookmarks
- TunedIT - Data mining & machine learning data sets, algorithms, challenges
- KDD 2011: 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining
- Memory-Verwaltung von R

### Publications
- A Case Study: Data Mining Applied to Student Enrollment
- Mining Rare Association Rules from e-Learning Data
- Educational data mining: A survey from 1995 to 2005
Real-World System II: Conferator

- Social Conference Guidance System
  - GI: Lernen – Wissen – Adaptivität (LWA) 2010 + 2011 + 2012
  - ACM Hypertext 2011
  - INFORMATIK 2013
  - UIS 2015

- Based on RFID-Technology (smart badges)

- Management of social contacts, personalization of conference schedule

- Localization

www.conferator.org
Conferator - Live Interaction
Conferator

- Social interaction networks:
  - Friend network
  - Contact network
  - Picked/Visited talks
  - Co-location network
Agenda

■ Motivation
■ Basics: Graphs & Attributes
■ Subgroup Discovery & Analytics
■ Cohesive Subgroups & Communities
■ Community Detection on Attributed Graphs
■ Applications & Tools
■ Summary & Outlook
Terminology

Network ➔ Graphs

- Set of atomic entities (actors) ➔ nodes, vertices
- Set of links/edges between nodes ("ties")
- Edges model pairwise relationships
- Edges: Directed or undirected
- Social network [Wasserman & Faust 1994]
  - Social structure capturing actor relations
  - Actors, links given by dyadic ties between actors (friendship, kinship, organizational position, ...)
    ➔ Set of nodes and edges
  - Abstract object – independent of representation
Variables [Wassermann & Faust 1994]

■ Structural
  ■ Measure ties between actors (➔ links)
  ■ Specific relation
  ■ Make up **connections** in graph/network

■ Compositional
  ■ Measure actor attributes
    ■ Age
    ■ Gender
    ■ Ethnicity
    ■ Affiliation
    ■ ...
  ■ **Describe** actors
Attributed Graphs

- Graph: edge attributes and/or node attributes
  - Structure: ties/links (of respective relations)
- Attributes - additional information
  - Actor attributes (node labels)
  - Link attributes (information about connections)
  - Attribute vectors for actors and/or links
  - ... can be mapped from/to each other
- Integration of heterogeneous data (networks + vectors)
- Enables simultaneous analysis of relational + attribute data
Subgroups & Cohesive subgroups

■ Subgroup
  ■ Subset of actors (and all their ties)
■ Define subgroups using specific criteria (homogeneity among members)
  ■ Compositional – actor attributes
  ■ Structural – using tie structures
■ Detection of cohesive subgroups & communities ➔ structural aspects
■ Subgroup discovery ➔ actor attributes
■ … attributed graph ➔ can combine both

[Wasserman & Faust 1994]
Cohesive Subgroups  

Components: Simple, detect "isolated" islands

Based on (complete) mutuality
  - Cliques
  - n-Cliques
  - Quasi-cliques

Based on nodal degree
  - K-plex
  - K-core

[Wasserman & Faust 1994]
Compositional Subgroups

- Detect subgroups according to specific compositional criteria
  - Focus on actor attributes
  - Describe actor subset using attributes
- Often hypothesis-driven approaches: Test specific attribute combinations
- In contrast: Subgroup discovery  [Atzmueller 2015]
  - Hypothesis-generating approach
  - Exploratory data mining method
  - Local exceptionality detection
Agenda

■ Motivation
■ Basics: Graphs & Attributes
■ **Subgroup Discovery & Analytics**
■ Cohesive Subgroups & Communities
■ Community Detection on Attributed Graphs
■ Applications & Tools
■ Summary & Outlook
Subgroup Discovery & Analytics

Task:
„Find descriptions of subsets in the data, that differ significantly for the total population with respect to a target concept.“

Examples:

"45% of all men aged between 35 and 45 have a high income in contrast to only 20% in total."

"66% all all woman aged between 50 and 60 have a high centrality value in the corporate network"

Descriptive patterns for subgroup

- Gender = Female ∧ Age = [50; 60] → Centrality = high
- {flickr, delicious}, {library, android}, {php, web} → Centrality = high
Subgroup Discovery

• Given – INPUT:
  – Data as set of cases (records) in tabular form
  – Target concept (e.g. „high centrality“)
  – Quality function (interesting measure)

• OUTPUT - Result: Set of the best k Subgroups:
  – Description, e.g., sex=female ∧ age= 50-60
    ➜ Conjunction of selectors
  – Size n, e.g., in 180 of 1000 cases
  – Deviation
    (p = 60% in the subgroup vs. p₀=10% in all cases)
  ➜ "Quality" of the subgroup: weight size and deviation
Subgroup Quality Functions [Atzmueller 2015]

- Consider size and deviation in the target concept

\[ n^a \cdot (p - p_0) \]

- Weighted Relative Accuracy \((a = 1)\)
- Simple Binomial \((a = 0.5)\)
- Added Value \((a = 0)\)

- Continuous: Mean value \( (m, m_0) \) of target variable

\[ q_{CWRACC} = \frac{n}{N} \cdot (m - m_0), \quad q_{CPS} = n \cdot (m - m_0) \]
Efficient Search

■ Heuristic: Beam Search

■ Exhaustive Approaches:
  ■ Basic idea: Efficient data structures + pruning
  ■ SD-Map – based on FP-Growth [Atzmueller & Puppe 2006]
  ■ SD-Map* – Utilizing optimistic estimates (branch & bound) [Atzmueller & Lemmerich 2009]
Pruning

■ Optimistic Estimate
Pruning – Branch & Bound

■ Optimistic Estimate: Upper bound for the quality of a pattern and all its specializations

→ Top-K Pruning

■ Remove path starting at current pattern, if optimistic estimate for current pattern (and all its specializations) is below quality of worst result of top-k results
Extensions

- Numeric features
- Very large data
  - Distributed Algorithms: Local (several cores) vs. network
  - Sampling
- Non tabular data
  - Text
  - Sequences
  - Networks/Graphs (community detection)
### Example: Binary target

<table>
<thead>
<tr>
<th>Income</th>
<th>Sex</th>
<th>Age</th>
<th>Education level</th>
<th>Married</th>
<th>Has Children</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>M</td>
<td>&gt;50</td>
<td>High</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>High</td>
<td>M</td>
<td>&gt;50</td>
<td>Medium</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>High</td>
<td>F</td>
<td>40-50</td>
<td>Medium</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Medium</td>
<td>M</td>
<td>&gt;50</td>
<td>High</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Medium</td>
<td>M</td>
<td>30-40</td>
<td>Medium</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>High</td>
<td>M</td>
<td>40-50</td>
<td>Low</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Low</td>
<td>M</td>
<td>&lt;30</td>
<td>High</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Medium</td>
<td>F</td>
<td>&lt;30</td>
<td>Medium</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Low</td>
<td>F</td>
<td>40-50</td>
<td>Low</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Low</td>
<td>M</td>
<td>40-50</td>
<td>Medium</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Medium</td>
<td>F</td>
<td>&gt;50</td>
<td>Medium</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Low</td>
<td>F</td>
<td>&lt;30</td>
<td>Low</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Low</td>
<td>F</td>
<td>30-40</td>
<td>Medium</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Low</td>
<td>F</td>
<td>40-50</td>
<td>Low</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Low</td>
<td>M</td>
<td>&lt;30</td>
<td>Low</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Medium</td>
<td>F</td>
<td>30-40</td>
<td>Medium</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

**Target concept: 'Income' = 'High'**

Quality function: \( q = n \times (p - p_0) \)

\( N = 16 \); \( p_0 = 0.25 \)

\( q = 0.0625 \)

**SG 1: 'Married' = 'Y'**

\( n = 8 \); \( p = 0.375 \) \( \Rightarrow q = 0.0625 \)

**SG 2: 'Sex' = 'M' \& 'Age' = '<30'**

\( n = 2 \); \( p = 0 \) \( \Rightarrow q = -0.03125 \)
Numeric Features

- Discretization:
  "While only 20% of the total population have an income > 60.000, in subgroup X it can be observed in more than 45% of all cases."

- Mean-Value:
  "While the average income in the total population is 45.000, it is more than 65.000 in subgroup Y."

➤ Both can be useful,
  Mean does not require threshold,
  Is it easier to understand?
Local Exceptionality Detection

- Exceptional Model Mining
  - Identification of Patterns
  - showing an "interesting behavior" for a certain "model"
    - Mean test (e.g., influence factors for increased centrality)
    - Linear regression (e.g., different centrality measures)
    - Correlation Coefficient (e.g., factors for role analysis)
    - Variance (e.g., degree, clustering coefficient, ...)
    - ...

- Algorithms:
  - Beam-Search: Heuristic (!) [Duivestein et al. 2015]
  - GP-Growth [Lemmerich et al. 2012]
    - Faster by multiple orders of magnitude compared to standard methods
    - Fastest exhaustive algorithm so far
EMM - Example Linear Regression

Subgroup:
\[ \text{drive} = 1 \land \text{nbath} > 2 \]

Total population

[1 - Example Linear Regression]

[2 - Example Linear Regression]
Exploratory Analytics

- Semi-automatic & Interactive
- Hypothesis generating
- Detect local models for description & prediction
  - Local exceptionality detection
    - Subgroup discovery
    - Exceptional model mining
    - ... community detection ...
  - Applicable also for big data (with Map/Reduce, ...)
Subgroup & Pattern Analytics

- VIKAMINE: Open-source tools for pattern mining and subgroup analytics
  
  www.vikamine.org

- R package: Algorithms of VIKAMINE
  
  www.rsubgroup.org
Agenda

- Motivation
- Basics: Graphs & Attributes
- Subgroup Discovery & Analytics
- **Cohesive Subgroups & Communities**
- Community Detection on Attributed Graphs
- Applications & Tools
- Summary & Outlook
Cohesive Subgroups

- Identify cohesive subgroups of actors

- Cohesive subgroup
  (Wassermann & Faust, p. 249):
    - Subsets of actors
      - Relatively strong, direct, intense, frequent or positive ties

- Social cohesion – primary criterion based on internal ties

- Extension: Social structure
  (→ communities!)
Subgroups – Local Definitions

[Clique: Subset of nodes of a graph, such that all nodes are adjacent to each other]

- Triangles
- Clique detection in graphs NP-Complete

Definition:

- Usually too conservative/strict
- Usually not found in sparse networks
- May not reflect real social groups

[Wasserman & Faust 1994]
Extension – K-Clique

■ K-Clique:
  ■ Maximal subgroup, where
    ■ largest geodesic distance between any pair of nodes is not greater than k

■ 1-Clique is a clique

■ 2-Clique: Subgraph, where all pairs of actors are connected with a path not longer than 2

[Wasserman & Faust 1994]
Extension – Quasi-Clique

■ Generalize clique to dense subgraph
■ Different definitions (degree, density)
■ Subset of nodes is quasi-clique, if
  ■ Nodal degree: every node in induced subgraph is adjacent to at least $\gamma (n - 1)$ other nodes in the subgraph
  ■ Edge density: Number of edges in subgraph is at least $\lambda n(n - 1)/2$

(with $n$ : number of nodes in subgraph)
K-Core

- Maximal subgraph
- Each node has at least degree $k$
- Hierarchy of cores
  - Iteratively, eliminate lower-order cores
  - Until: Relatively dense subgroups remain

[Wasserman & Faust 1994]
Communities

- Cohesive subgroups – structure within group
- Basic idea of communities
  - Tightly-knit groups
  - Consider both internal and external ties in network
- In general:
  - High number of internal ties (high density within)
  - Low number of external ties (lower density between)
Zachary's Karate Club [Zachary, 1977]

- Members of university karate club
- Conflict between club president (34) and karate instructor (1)
- Result: Split-up of the network according to friendship ties
Karate Club – 2 Factions
Finding Communities

- Given a network/graph, find "modules"
  - Single network [Newman 2002]
  - Multiplex networks [Bothorel 2015]

- Community structures [Fortunato 2010]
  - Graph Clustering ➔ disjoint communities
  - Hierarchical organization [Lancichinetti 2009]
  - Overlapping communities [Xie et al. 2013]

- Questions:
  - What is "a community"?
  - What are "good" communities?
  - How do we evaluate these?
Community: Definition & Properties

- No universally accepted definition
- Informally:
  - Intuition: Densely connected group of nodes
  - Subset of nodes such that there are more edges inside the community than edges linking the nodes with the rest of the graph
- Intra Cluster Density
- Inter Cluster Density
- Connectedness
Global View

- Communities can also be defined with respect to the whole graph.
- Graph has community structure, if it is different from random graph.
- Random graph: Not expected to have community structure.
  - Here: Any two vertices have the same probability to be adjacent.
  - Define null model; use it for investigating if we can observe community structure in a graph.
Community Evaluation Measures

- **Modularity** [Newman 2006]
  \[ MOD(S) = \frac{1}{2m} \sum_{i,j} (A_{i,j} - \frac{d(i)d(j)}{2m}) \delta(C_i, C_j) \]
  Compares the number of edges within a community with the expected such number in a corresponding null model

- **Conductance** [Kannan et al. 2004]
  \[ CON(C) = \frac{\bar{m}_C}{2m_C + \bar{m}_C} \]
  Compares the number of edges within a community and the number of edges leaving the community

  \[ COIN(C) = 1 - CON(C) = \frac{2m_C}{\sum_{u \in C} d(u)} \]
Community Evaluation Measures

■ Inverse Average Out-Degree Fraction (IAODF) [Leskovec et al. 2010]

\[
\text{IAODF}(C) := 1 - \frac{1}{n_C} \sum_{u \in C} \frac{\bar{d}_C(u)}{d(u)}
\]

compares the number of inter-edges to the number of all edges of a community, and averages this for the whole community by considering the fraction for each individual node.

■ Segregation Index (SIDX) [Freeman 1978]

\[
\text{SIDX}(C) = \frac{E(\bar{m}_C) - \bar{m}_C}{E(\bar{m}_C)} = 1 - \frac{\bar{m}_C n (n - 1)}{2mn_C (n - n_C)}
\]

compares the number of expected inter-edges to the number of observed inter-edges, normalized by the expectation.
Community Criteria

Several possible community criteria

- **Node-Centric Community**: Each node in a group satisfies certain properties, e.g., reachability, clique-based

- **Group-Centric Community**: Consider the connections within a group as a whole. Group has to satisfy certain properties, e.g., minimal density, Quasi-clique ...

- **Network-Centric Community**: Partition the whole network into several disjoint sets, e.g., graph clustering, modularity maximization

- **Hierarchy-Centric Community**: Construct a hierarchical structure of communities

- **Descriptive Community Detection**: Identifies communities and description at the same time

  ➔ Especially for exploratory community detection
Clique Percolation Method (CPM) [Palla et al. 2005]

- Clique is a very strict definition, unstable
- Normally use cliques as a core or a seed to find larger communities
- CPM: Detect overlapping communities

**Input**
- A parameter k, and a network

**Procedure**
- Find out all cliques of size k in a given network
- Construct a clique graph. Two cliques are adjacent if they share k-1 nodes
- Each connected component in the clique graph forms a community
CPM Example

 Cliques of size 3:
  \{1, 2, 3\}, \{1, 3, 4\}, \{4, 5, 6\},
  \{5, 6, 7\}, \{5, 6, 8\}, \{5, 7, 8\},
  \{6, 7, 8\}

 Communities:
  \{1, 2, 3, 4\}
  \{4, 5, 6, 7, 8\}
Network-Centric Community Detection

- Network-centric criterion needs to consider the connections within a network globally
- Goal: partition nodes of a network into disjoint sets
- Approaches:
  - Clustering based on vertex similarity [Zhou et al. 2009]
  - Latent space models [Raftery et al. 2002]
  - Block model approximation [Karrer & Newman 2011]
  - Spectral clustering [Ma & Gao 2011]
  - Modularity maximization [Newman 2006]
Agglomerative Hierarchical Clustering

- Initialize each node as a community
- Merge communities successively into larger communities following a certain criterion
  - E.g., based on modularity increase

[Clauset et al. 2004]
Divisive Hierarchical Clustering

- Divisive clustering
  - Partition nodes into several sets
  - Each set is further divided into smaller ones
  - Network-centric partition can be applied for the partition

- One particular example: recursively remove the “weakest” tie
  - Find the edge with the least strength
  - Remove the edge and update the corresponding strength of each edge

- Recursively apply the above two steps until a network is discomposed into desired number of connected components.

- Each component forms a community

[Girvan & Newman 2002]
Agenda

■ Motivation
■ Basics: Graphs & Attributes
■ Subgroup Discovery & Analytics
■ Cohesive Subgroups & Communities
■ Community Detection on Attributed Graphs
■ Applications & Tools
■ Summary & Outlook
Combining Structure and Attributes

- Data sources
  - Structural variables (ties, links)
  - Compositional variables
    - Actor attributes
    - Represented as attribute vectors
- Edge attributes
  - Each edge has an assigned label
  - Multiplex graphs
    - Multiple edges (labels) between nodes
Communities/Edge-Attributed Graphs

- Clustering edge-attributed graphs
  - Reduce/flatten to weighted graph
    - [Bothorel et al. 2015]
      - Derive weights according to number of edges where nodes are directly connected [Berlingerio et al. 2011]
      - Standard graph clustering approaches can then be directly applied
  - Frequent-itemset based [Berlingerio et al. 2013]
  - Subspace-oriented [Boden et al. 2012]
Node-Attributed Graphs

- Non-uniform terminology
  - Social-attribute network
  - Attribute augmented graph
  - Feature-vector graph, vertex-labeled graph
  - Attributed graph
  - ...

- Different representations

[Bothorel et al. 2015]
Community Detection – Attribute Extensions

■ Utilize structural + attribute information

■ Different roles of a description
  ■ Methods aiding community detection using attribute information
    ■ "Dense structures" - connectivity
    ■ But no "perfect" attribute homogeneity (purity)
  ■ Methods generating explicit descriptions, i.e., descriptive community patterns
    ■ "Dense structures" – connectivity
    ■ Concrete descriptions, e.g., conjunctive logical formula
Attributes for Aiding Community Detection

  - Abstraction into similarities between nodes
    - Edge weights
    - Apply standard community detection algorithm,
  - Specifically, distance-based community detection methods


Weight modification [Steinhaeuser & Chawla 2008]

- Use attribute-based distance measure

```plaintext
1: for each node $i = 1 \ldots n$ do
2:  for each node $j = 1 \ldots \text{neighbors}(i)$ do
3:    $w(i, j) = 0$
4:    for each node attribute $a$ do
5:      if $a$ is nominal and $i.a = j.a$ then
6:        $w(i, j) = w(i, j) + 1$
7:      else if $a$ is continuous then
8:        $w(i, j) = w(i, j) + 1 - \alpha |i.a - j.a|$
9:      end if
10:    end for
11:  end for
12: end for
```

- Community detection: Group nodes according to threshold $t$, i.e., given $t \in (0, 1)$ place any pair of nodes whose edge weight exceeds the threshold into the same community
- Evaluate final partitioning using Modularity
Entropy Minimization

- For a partition, optimize entropy using Monte-Carlo.
- Integrate entropy step into Modularity optimization algorithm.

```latex
\textbf{Require:} \ C, \ imax, \ PoV_{F^*} \\
1: \quad H_0 \leftarrow \mathcal{H}_C^0 \\
2: \quad i \leftarrow 0 \\
3: \quad \textbf{while } i < \text{imax } \textbf{and } \text{More possible changes do} \\
4: \quad \quad i \leftarrow i + 1 \\
5: \quad A \leftarrow \text{random cluster from } C \\
6: \quad x \leftarrow \text{random node: } x \in A \\
7: \quad A(x, -) \\
8: \quad B \leftarrow \text{random cluster from } C \setminus \{D_A \cup A\} \\
9: \quad B(x, +) \\
10: \quad H_i \leftarrow \mathcal{H}_C^i \\
11: \quad \textbf{if } H_i \geq H_{i-1} \textbf{ then} \\
12: \quad \quad B(x, -) \\
13: \quad \quad A(x, +) \\
14: \quad \textbf{end if} \\
15: \quad \textbf{end while} \\
16: \quad \textbf{return } \ C_H \{A \text{ new partition with a reduced entropy}\}
```

[Cruz et al. 2011] [Blondel et al. 2008]
Model-based/MDL

■ In general: Model edge & attribute values using mixtures of probability distributions

■ Use MDL to select clusters w.r.t. attribute value similarity & connectivity similarity
  ■ Data compression of connectivity & attribute matrices (PICS algorithm)
  ■ Lossless compression $\rightarrow$ MDL cost-function

■ Resulting node groups
  ■ Homogeneous both in node & attribute matrix
  ■ Nodes - similar connectivity & high attribute coherence

[Akoglu et al. 2013]
Descriptive Community Patterns

- Community mining scenario
  - Discover "densely connected groups of nodes"
  - Communities should have explicit description
  - Community (evaluation) space: network/graph

- Goal:
  - Often: Discover top-k communities
  - Maximize some community quality function
Examples: Community Patterns

- Social tagging system:
  - \{work, flickr, delicious\}
  - \{business, production, sales\}
  - \{php, web, internet\}, \{innovation, business, forschung\}
  - \{work, flickr, delicious\}, \{library, android, emulation\}, \{php, web, internet\}
Finding Explicit Descriptions

- Cluster transformed node-attribute similarity graph & extract pure clusters
- Mine frequent itemsets (binary attributes) & analyze communities  [Adnan et al. 2009]
- Apply correlated pattern mining  [Silva et al. 2012]
- Interleave community detection & redescription mining  [Pool et al. 2014]
- Adapt local exceptionality detection (using subgroup discovery) for communities  [Atzmueller & Mitzlaff 2011, Atzmueller et al. 2015]
Subspace-Clustering & Dense Subgraphs [Günnemann et al. 2011]

- Twofold cluster $O$: Combine subspace-clustering & dense subgraph mining (GAMer algorithm)
  - $O$ fulfills subspace property (maximal distance threshold w.r.t. node attribute values in $O$) with minimal number of dimensions
  - $O$ fulfills quasi-clique property, according to nodal-degree and threshold $\gamma$
  - Induced subgraph of $O$ is connected, and fulfills minimal size threshold

- Quality function: $\text{Density} \cdot \text{Size} \cdot \text{#Dimensions}$
- Pruning using subspace & quasi-clique properties
- Includes Redundancy-optimization step (Overlapping communities)
Correlated Pattern Mining  [Silva et al. 2011]

- Structural correlation pattern mining (SCPM)
  - Correlation between node attribute set and dense subgraph, induced by the attribute set
  - Quality measure: Comparison against null model
    - Size of the pattern
    - Cohesion of the pattern (density of quasi-clique)
  - Compare against expected structural correlation of attribute set (in random graph)

(a) Vertex attributes
(b) Graph
(c) Dense subgraph
(d) Dense subgraph
Algorithm 2 SCPM Algorithm

Require: \( G, \sigma_{\min}, \gamma_{\min}, \text{min}_\text{size}, \epsilon_{\min}, \delta_{\min}, k \)
Ensure: \( \mathcal{P} \)

1: \( \mathcal{P} \leftarrow \emptyset \)
2: \( \mathcal{T} \leftarrow \emptyset \)
3: \( \mathcal{I} \leftarrow \text{frequent attributes from } G \)
4: for all \( S \in \mathcal{I} \) do
5: \( \epsilon \leftarrow \text{structural correlation of } S \)
6: if \( \epsilon \geq \epsilon_{\min} \text{ AND } \epsilon/\epsilon_{\exp}(S) \geq \delta_{\min} \) then
7: \( Q \leftarrow \text{top-} k \text{ patterns from } G(S) \)
8: for all \( q \in Q \) do
9: \( \mathcal{P} \leftarrow \mathcal{P} \cup (S, q) \)
10: end for
11: end if
12: if \( \epsilon.\sigma(S) \geq \epsilon_{\min}.\sigma_{\min} \text{ AND } \epsilon.\sigma(S) \geq \delta_{\min}.\epsilon_{\exp}(\sigma_{\min}).\sigma_{\min} \) then
13: \( \mathcal{T} \leftarrow \mathcal{T} \cup S \)
14: end if
15: end for
16: \( \mathcal{P} \leftarrow \mathcal{P} \cup \text{enumerate-patterns}(\mathcal{T}, G, \sigma_{\min}, \gamma_{\min}, \text{min}_\text{size}, \epsilon_{\min}, \delta_{\min}, k) \)

- Thresholds: min. support (size), structural correlation, expected structural correlation
Description-driven Community Detection

- Find communities with concise descriptions (e.g., given by tags)
- Focus: Overlapping, diverse, descriptive communities
- Language: Disjunctions of conjunctive expressions
- Two-stage approach
  - Greedy hill-climbing step: Generate candidates for communities
  - Redescription generation: Induce description for each community, and reshape if necessary
- Heuristic approach, due to large search space

[Pool et al. 2014]
ALGORITHM 1: DCM

Input: Attributed graph $G$, parameters $k$ and $\eta$, and a set of candidate communities $C$.
Output: An approximation of $Q$, the top-$k$ communities.

1. $Q \leftarrow \emptyset$
2. for all $C \in C$ do
3.     while $C$ changes do
4.         $C \leftarrow \text{MAXIMIZE\_COMMUNITY\_SCORE}(C)$
5.         $C \leftarrow \text{FIND\_CONCISE\_QUERY}(C)$
6.     end while
7.     $Q \leftarrow Q \cup Q(C)$
8. end for
9. $Q \leftarrow \text{SELECT\_DIVERSE\_TOP\_K}(Q, k, \eta)$
10. return $Q$

- Starts with candidate communities
  - Domain knowledge
  - Partial communities
  - Start with single vertices (later being extended using hill-climbing approach)

- ReMine algorithm for deriving patterns for communities  [Zimmermann et al. 2010]
a. FLICKR community 1

b. FLICKR community 2

c. FLICKR community 3

d. LASTFM community 1

e. LASTFM community 2

[Pool et al. 2013]
Description-Oriented Community Detection

[Atzmueller et al., Information Science, 2016]

- Basic Idea: Pattern Mining for Community Characterization
  - Mine patterns in description space (tags/topics) ➔ Subgroups of users described by tags/topics
  - Optimize quality measure in community space ➔ Network/graph of users
  - Improve understandability of communities (explanation)
Direct Descriptive Community Mining

- **Goal**: Identification/description of communities with a high quality (exceptional model mining)
  - **Input**: Network/Graph + node properties (e.g., tags)
  - **Output**: k-best community patterns
- **Description language**: conjunctive expressions
- **COMODO algorithm**: Top-k pattern mining, based on SD-Map* algorithm for subgroup discovery
  - Discover k-best patterns
  - Search space: Conjunctions/tags
  - Apply standard community quality functions, e.g., Modularity [Newman 2004]

\[
MOD(S) = \frac{1}{2m} \sum_{i,j} (A_{i,j} - \frac{d(i)d(j)}{2m}) \delta(C_i, C_j)
\]
Community Detection on Attributed Graphs

- **Goal:** Mine patterns describing such groups

<table>
<thead>
<tr>
<th>Size</th>
<th>Community description</th>
<th>Size</th>
<th>Community description</th>
</tr>
</thead>
<tbody>
<tr>
<td>519</td>
<td>80s</td>
<td>32</td>
<td>psychedelic AND minimal</td>
</tr>
<tr>
<td>240</td>
<td>gregorian_chant AND 80s</td>
<td>16</td>
<td>psychedelic AND 80s</td>
</tr>
<tr>
<td>215</td>
<td>girl_groups AND 80s</td>
<td>10</td>
<td>psychedelic AND brit_rock AND classic_rock</td>
</tr>
<tr>
<td>171</td>
<td>atmospheric</td>
<td>10</td>
<td>death_rock AND minimal AND 80s</td>
</tr>
<tr>
<td>122</td>
<td>synth_pop</td>
<td>10</td>
<td>death_rock AND 80s AND doom_metal</td>
</tr>
</tbody>
</table>

- Merge networks + descriptive features, e.g., characteristics of users
- **Target both**
  - Community structure (some evaluation function) &
  - Community description (logical formula, e.g., conjunction of features, see above)
Transformation & Mining (I)

■ Sources:
  ■ Database DB: Users described by attributes (e.g. used topics)
  ■ Graph G: Links between users (e.g. friend graph)

■ Goal:
  ■ Discover k best communities as subgroups of DB
  ■ Maximizing community evaluation function on G

■ Need to merge both data sources

User 1: \{work, flickr, delicious\}
User 2: \{business, production, sales\}
User 3: \{php, web, internet\},
  \{innovation, business, forschung\}
User 4: \{work, flickr, delicious\},
  \{library, android, emulation\},
  \{php, web, internet\}
…
Transformation & Mining (II)

- Dataset of edges connecting two nodes
  - Described by intersection of labels of the two nodes
  - Additionally: Store nodes, and respective degrees
- Apply top-k method w/ optimistic-estimate pruning (COMODO)
Algorithm 1 COMODO

procedure COMODO-Mine (cf. [17] for an extended description)

Input: Current community pattern tree \( CPT \), pattern \( \hat{p} \), priority queue \( \text{top-}k \), int \( k \) (max. number of patterns), int \( \text{maxLength} \) (max. length of a pattern), int \( \tau_n \) (min. community size)

1: \( \text{COM} = \text{new dictionary: basicpattern } \rightarrow \text{pattern} \)
2: \( \text{minQ} = \text{minQuality}(\text{top-}k) \)
3: for all \( b \) in \( CPT.\text{getBasicPatterns} \) do
   4: \( p = \text{createRefinement}(\hat{p}, b) \)
   5: \( \text{COM}[b] = p \)
   6: if size\( (p, CPT) \geq \tau_n \) then
      7: if quality\( (p, F) \geq \text{minQ} \) then
         8: \( \text{addToQueue(top-}k, p) \)
         9: \( \text{minQ} = \text{minQuality}(\text{top-}k) \)
   10: if length\( (\hat{p}) + 1 < \text{maxLength} \) then
      11: \( \text{refinements} = \text{sortBasicPatternsByOptimisticEstimateDescending(} \text{COM} \) \)
      12: for all \( b \) in refinements do
         13: if optimisticEstimate\( (\text{COM}[b]) \geq \text{minQ} \) then
            14: \( \text{CCPT} = \text{getConditionalCPT}(b, CPT, \text{minQ}) \)
            15: Call COMODO-Mine(\( \text{CCPT}, \text{COM}[b], \text{top-}k \) )

Algorithm utilizes special tree-structure & optimistic estimates for efficient processing
Optimistic Estimates

■ Problem: Exponential Search Space

■ Optimistic Estimate: Upper bound for the quality of a pattern and all its specializations

⇒ Top-K Pruning

---

**Last.fm friend graph**

**Delicious friend graph**
## Optimistic Estimate Pruning

[Atzmueller et al. 2015]

<table>
<thead>
<tr>
<th>Method</th>
<th>Top $k$</th>
<th>$\tau_n = 5$</th>
<th>$\tau_n = 10$</th>
<th>$\tau_n = 20$</th>
<th>$\tau_n = 50$</th>
<th>$\tau_n = 70$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODL</td>
<td>5</td>
<td>619</td>
<td>619</td>
<td>619</td>
<td>87</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1,654</td>
<td>1,654</td>
<td>1,654</td>
<td>87</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>4,104</td>
<td>4,104</td>
<td>4,104</td>
<td>87</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>48,460</td>
<td>30,287</td>
<td>47,945</td>
<td>87</td>
<td>15</td>
</tr>
<tr>
<td>dMODL</td>
<td>5</td>
<td>420</td>
<td>420</td>
<td>420</td>
<td>87</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1,393</td>
<td>1,393</td>
<td>1,393</td>
<td>87</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>3,479</td>
<td>3,688</td>
<td>3,676</td>
<td>87</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>34,620</td>
<td>51,041</td>
<td>50,236</td>
<td>87</td>
<td>15</td>
</tr>
<tr>
<td>COIN</td>
<td>5</td>
<td>79,219,778</td>
<td>60,421,374</td>
<td>1,215,426</td>
<td>86</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>79,219,778</td>
<td>61,722,695</td>
<td>1,373,936</td>
<td>87</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>79,219,778</td>
<td>62,340,121</td>
<td>1,463,930</td>
<td>87</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>79,219,778</td>
<td>63,052,402</td>
<td>1,555,210</td>
<td>87</td>
<td>15</td>
</tr>
<tr>
<td>No Pruning</td>
<td></td>
<td>79,375,495</td>
<td>69,971,088</td>
<td>3,205,222</td>
<td>87</td>
<td>15</td>
</tr>
</tbody>
</table>
Agenda

- Motivation
- Basics: Graphs & Attributes
- Subgroup Discovery & Analytics
- Cohesive Subgroups & Communities
- Community Detection on Attributed Graphs
- Applications & Tools
- Summary & Outlook
Descriptive Community Detection

- Example: Patterns from last.fm
  - Recommendation
  - Browsing
  - …

<table>
<thead>
<tr>
<th>Size</th>
<th>Community description</th>
</tr>
</thead>
<tbody>
<tr>
<td>519</td>
<td>80s</td>
</tr>
<tr>
<td>240</td>
<td>gregorian_chant AND 80s</td>
</tr>
<tr>
<td>215</td>
<td>girl_groups AND 80s</td>
</tr>
<tr>
<td>171</td>
<td>atmospheric</td>
</tr>
<tr>
<td>122</td>
<td>synth_pop</td>
</tr>
<tr>
<td>32</td>
<td>psychedelic AND minimal</td>
</tr>
<tr>
<td>16</td>
<td>psychedelic AND 80s</td>
</tr>
<tr>
<td>10</td>
<td>psychedelic AND brit_rock AND classic_rock</td>
</tr>
<tr>
<td>10</td>
<td>death_rock AND minimal AND 80s</td>
</tr>
<tr>
<td>10</td>
<td>death_rock AND 80s AND doom_metal</td>
</tr>
</tbody>
</table>
Conferator

- Interest profiles – Recommending conference participants
  - BibSonomy: User profiles
    - java AND android AND nfc
  - Conferator: Acquaint-O-Matic
    - recommender & analysis
    - mining & ubiquitous & social
Further Examples

- Behavioral social targeting
  - Apply domain knowledge
  - Use (explicit) descriptions

- Recommendations
  - Popular items in community
  - Deal with cold-start problems

- Exploratory analytics
  - First insights into data
  - Characterization of exceptional subgroups
Tool: VIKAMINE

- Visual, Interactive and Knowledge-intensive Analysis and semantic MINing Environment
  - Data mining
  - Visual analytics
  - Knowledge refinement
  - Semantic knowledge capture

- Option: Include background knowledge, semantic annotation, ontologies

- http://www.vikamine.org
  (R-Package: rsubgroup.org)
VIKAMINE Features

- Efficient automatic discovery algorithms
  - Subgroup discovery & analytics
  - Community detection
- Seamless integration of visualization methods
- Effective visualizations for ad-hoc analysis
- Ad-hoc formalization, utilization, and extension of background knowledge
- Plugin for description-oriented community detection
- Works also on big data (Map/Reduce)
Workbench
Community Detection Software

- **igraph (R):**
  - Different community detection methods (mostly methods for detecting disjoint communities)
    - Modularity maximization, Walktrap, Label propagation, INFOMAP, ...

- **Linkcomm (R):** detection of link communities (potentially overlapping)

- **CPM**
  - CFinder: http://www.cfinder.org/
  - Fast clique percolation (cp5): https://github.com/aaronmcdaid/MaximalCliques
Community Detection Software

- Overlapping communities:
  - COPRA: http://www.cs.bris.ac.uk/~steve/networks/copra/
  - MOSES: http://sites.google.com/site/aaronmcdaid/moses
- Description-oriented methods/attributed graphs
  - COMODO: www.vikamine.org
  - DCM: http://www.patternsthatmatter.org/software.php#dcm
  - GAMER: http://dme.rwth-aachen.de/de/gamer
- Bipartite networks: http://danlarremore.com/bipartiteSBM/
Agenda

- Motivation
- Basics: Graphs & Attributes
- Subgroup Discovery & Analytics
- Cohesive Subgroups & Communities
- Community Detection on Attributed Graphs
- Applications & Tools
- Summary & Outlook
Summary

- Subgroup discovery & community detection enable the identification of subgroups at different levels & dimensions
  - Compositional
  - Structural + compositional
  - Providing explicit descriptions
- Both can be combined for obtaining descriptive community patterns according to standard community quality functions
- Efficient tools for detection & analysis
Outlook

■ Challenges using ubiquitous & social data
  ■ Heterogeneous data & complex networks
  ■ Integration of multiplex networks & temporal information
  ■ Support for integration & analysis
  ■ Necessary: Efficient methods and tools for the mining of such data

■ Extensions: Effective exploratory methods for analytics. Integrated assessment, mining & inspection
Community Detection: From Plain to Attributed Complex Networks

Martin Atzmueller

University of Kassel, Research Center for Information System Design
Ubiquitous Data Mining Team, Chair for Knowledge and Data Engineering

Web Science 2016, Hannover - 2016-05-22


References


References (cont.)


References (cont.)

- [Atzmueller & Lemmerich 2013] M. Atzmueller and F. Lemmerich (2013) Exploratory Pattern Mining on Social Media using Geo-References and Social Tagging Information. IJWS, 2(1/2)


References (cont.)


References (cont.)


References (cont.)


References (cont.)


