Mining patterns in Attributed Dynamic Graphs

Céline ROBARDET

Mon, 29.09.
Different kinds of torrents of data
Potential increase of our knowledge
Viewed as attributed dynamic graphs
Mining network data

Network data brings several questions:

• Working with network data is messy
  • Not just “wiring diagrams” but also dynamics and data (features, attributes) on nodes and edges

• Computational challenges
  • Large scale network data

• Algorithmic models as vocabulary for expressing complex scientific questions
  • Social science, physics, biology

Understanding how network structure and node attribute values relate and affect each other.
Database queries: retrieve transactions that match a search criteria

Data mining queries: retrieve sets of data, called patterns, that match some criteria

the criteria is computed on individual data or on sets of data
Constraint-based pattern mining

Developed for extracting itemsets in transaction databases

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1 3 4</td>
</tr>
<tr>
<td>200</td>
<td>2 3 5</td>
</tr>
<tr>
<td>300</td>
<td>1 2 3 5</td>
</tr>
<tr>
<td>400</td>
<td>2 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Itemset</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1}</td>
<td>2</td>
</tr>
<tr>
<td>{2}</td>
<td>3</td>
</tr>
<tr>
<td>{3}</td>
<td>3</td>
</tr>
<tr>
<td>{5}</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Itemset</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1 3}</td>
<td>2</td>
</tr>
<tr>
<td>{1 4}</td>
<td>1</td>
</tr>
<tr>
<td>{3 4}</td>
<td>1</td>
</tr>
<tr>
<td>{2 3}</td>
<td>2</td>
</tr>
<tr>
<td>{2 5}</td>
<td>3</td>
</tr>
<tr>
<td>{3 5}</td>
<td>2</td>
</tr>
<tr>
<td>{1 2}</td>
<td>1</td>
</tr>
<tr>
<td>{1 5}</td>
<td>1</td>
</tr>
</tbody>
</table>
Can be used to analyze graphs

Graphs that are often

- **Dynamic**: when nodes and edges appear/disappear through time
- **Attributed**: when another relation describes the nodes or the edges themselves

Relational graph

Nodes are uniquely identified through time
Some inductive queries on graphs

- What are the vertex attributes that strongly co-vary with the graph structure?
- What are the sub-graphs whose vertex attributes evolve similarly?

Co-authors that published at ICDE with a high degree and a low clustering coefficient

Airports whose arrival delays increased over the three weeks following Katrina hurricane
Outline of the talk

Constraint-based pattern mining framework

Topological patterns in static attributed graphs

Mining dynamic graphs

Trend dynamic sub-graphs

Triggering patterns of topology changes

Conclusions and perspectives
Constraint-based pattern mining framework
Pattern mining

A Pattern $\varphi$ describes a subgroup of the data $\mathcal{D}$
- observed several times
- or associated with characteristic properties

The pattern shape is fixed: $\varphi \in \mathcal{L}$

whose cardinality is exponential or infinite in the size of the data

$\mathcal{C}$ evaluates the adequacy of the pattern to the data

$$\mathcal{C}(\varphi, \mathcal{D}) \rightarrow \text{Boolean}$$

Pattern mining task: Find all interesting subgroups

$$Th(\mathcal{L}, \mathcal{D}, \mathcal{C}) = \{ \varphi \in \mathcal{L} \mid \mathcal{C}(\varphi, \mathcal{D}) \text{ is true} \}$$
Pattern constraints

Constraints are needed for:

- only retrieving patterns that describe an interesting subgroup of the data
- making the extraction feasible

Constraint properties are used to infer constraint values on (many) patterns without having to evaluate them individually.

They are defined up to the partial order $\preceq$ used for listing the patterns.
Pattern constraints

Constraints are needed for:

- only retrieving patterns that describe an interesting subgroup of the data
- making the extraction feasible

Constraint properties are used to infer constraint values on (many) patterns without having to evaluate them individually.

They are defined up to the partial order $\preceq$ used for listing the patterns.
Pattern constraints

Constraints are needed for:

- only retrieving patterns that describe an interesting subgroup of the data
- making the extraction feasible

Constraint properties are used to infer constraint values on (many) patterns without having to evaluate them individually.

→ They are defined up to the partial order $\preceq$ used for listing the patterns
**Constraint properties - 1**

**Monotone constraint**
\[
\forall \varphi_1 \preceq \varphi_2, C(\varphi_1, \mathcal{D}) \Rightarrow C(\varphi_2, \mathcal{D})
\]

**Anti-monotone constraint**
\[
\forall \varphi_1 \preceq \varphi_2, C(\varphi_2, \mathcal{D}) \Rightarrow C(\varphi_1, \mathcal{D})
\]

\[
C(\varphi, \mathcal{D}) \equiv b \in \varphi \lor c \in \varphi
\]

\[
C(\varphi, \mathcal{D}) \equiv a \notin \varphi \land c \notin \varphi
\]
Constraint properties - 2

Convertible constraints

$\leq$ is extended to the prefix order $\leq$ so that $\forall \varphi_1 \leq \varphi_2, C(\varphi_2, D) \Rightarrow C(\varphi_1, D)$

Loose AM constraints

$C(\varphi, D) \Rightarrow \exists e \in \varphi : C(\varphi \setminus \{e\}, D)$

$C(\varphi, w) \equiv \text{avg}(w(\varphi)) > \sigma$

$w(a) \geq w(b) \geq w(c) \geq w(d) \geq w(e)$

$C(\varphi, w) \equiv \text{var}(w(\varphi)) \leq \sigma$

Pei and Han - 2000

Bonchi and Lucchese - 2007
Enumeration strategy

Binary partition: the element 'a' is enumerated
Enumeration strategy

Binary partition: the element 'a' is enumerated

\[ a \in \mathcal{R}^\vee \setminus \mathcal{R}^\wedge \]
Constraint evaluation

Monotone constraint

\[ R_\vee \quad \mathcal{C}(R_\vee, D) \text{ is false} \]

empty

\[ R_\wedge \]

Anti-monotone constraint

\[ R_\vee \quad \mathcal{C}(R_\wedge, D) \text{ is false} \]

empty

\[ R_\wedge \]
**Monotone constraint**

\[ \mathcal{C}(\mathcal{R}^\lor, \mathcal{D}) \text{ is false} \]

**Anti-monotone constraint**

\[ \mathcal{C}(\mathcal{R}^\land, \mathcal{D}) \text{ is false} \]
Constraint evaluation

Monotone constraint

\[ \mathcal{R}^\vee \mathcal{C}(\mathcal{R}^\vee, \mathcal{D}) \text{ is false} \]

\[ \text{empty} \]

\[ \mathcal{R}^\wedge \]

Anti-monotone constraint

\[ \mathcal{R}^\vee \mathcal{C}(\mathcal{R}^\wedge, \mathcal{D}) \text{ is false} \]

\[ \text{empty} \]

\[ \mathcal{R}^\wedge \]
Constraint evaluation

Monotone constraint

\( \mathcal{R}^\vee \)

\( \mathcal{C}(\mathcal{R}^\vee, \mathcal{D}) \) is false

empty

\( \mathcal{R}^\wedge \)

Anti-monotone constraint

\( \mathcal{R}^\vee \)

empty

\( \mathcal{C}(\mathcal{R}^\wedge, \mathcal{D}) \) is false

\( \mathcal{R}^\wedge \)
A new class of constraints

Piecewise monotone and anti-monotone constraints

1. \( C \) involves \( p \) times the pattern \( \varphi \): \( C(\varphi, D) = f(\varphi_1, \cdots, \varphi_p, D) \)

2. \( f_{i,\varphi}(x) = (\varphi_1, \cdots, \varphi_{i-1}, x, \varphi_{i+1}, \cdots, \varphi_p, D) \)

3. \( \forall i = 1 \cdots p \), \( f_{i,\varphi} \) is either monotone or anti-monotone:

   \[
   \forall x \preceq y, \quad \left\{ \begin{array}{ll}
   f_{i,\varphi}(x) \Rightarrow f_{i,\varphi}(y) & \text{iff } f_{i,\varphi} \text{ is monotone} \\
   f_{i,\varphi}(y) \Rightarrow f_{i,\varphi}(x) & \text{iff } f_{i,\varphi} \text{ is anti-monotone}
   \end{array} \right.
   \]
An example

• $\forall e, \ w(e) \geq 0$

• $\mathcal{C}(\varphi, w) \equiv \text{avg}(w(\varphi)) > \sigma \equiv \frac{\sum_{e \in \varphi} w(e)}{|\varphi|} > \sigma$.

$\mathcal{C}(\varphi, D)$ is piecewise monotone and anti-monotone with

$$f(\varphi_1, \varphi_2, D) = \frac{\sum_{e \in \varphi_1} w(e)}{|\varphi_2|}$$

$\forall x \preceq y$,

• $f_{1,\varphi}$ is monotone: $f(x, \varphi_2, D) = \frac{\sum_{e \in x} w(e)}{|\varphi_2|} > \sigma \Rightarrow \frac{\sum_{e \in y} w(e)}{|\varphi_2|} > \sigma$

• $f_{2,\varphi}$ is anti-monotone:

$$f(\varphi_1, y, D) = \frac{\sum_{e \in \varphi_1} w(e)}{|y|} > \sigma \Rightarrow \frac{\sum_{e \in \varphi_1} w(e)}{|x|} > \sigma$$
Piecewise constraint exploitation

Evaluation

If \( f(\mathcal{R}^\vee, \mathcal{R}^\wedge, D) = \frac{\sum_{e \in \mathcal{R}^\vee} w(e)}{|\mathcal{R}^\wedge|} \leq \sigma \)

then \( \mathcal{R} \) is empty.

Propagation

- \( \exists e \in \mathcal{R}^\vee \setminus \mathcal{R}^\wedge \) such that \( f(\mathcal{R}^\vee \setminus \{e\}, \mathcal{R}^\wedge, D) \leq \sigma \), then \( e \) is moved in \( \mathcal{R}^\wedge \)

- \( \exists e \in \mathcal{R}^\vee \setminus \mathcal{R}^\wedge \) such that \( f(\mathcal{R}^\vee, \mathcal{R}^\wedge \cup \{e\}, D) \leq \sigma \), then \( e \) is removed from \( \mathcal{R}^\vee \)
Algorithmic principles

Function Generic_CBPMEnumeration($\mathcal{R}^\wedge, \mathcal{R}^\vee$)

1: if Check_constraints($\mathcal{R}^\wedge, \mathcal{R}^\vee$) then
2:   ($\mathcal{R}^\wedge, \mathcal{R}^\vee$) ← Constraint_Propagation($\mathcal{R}^\wedge, \mathcal{R}^\vee$)
3:   if $\mathcal{R}^\wedge = \mathcal{R}^\vee$ then
4:     output $\mathcal{R}^\wedge$
5:   else
6:     for all $e \in \mathcal{R}^\vee \setminus \mathcal{R}^\wedge$ do
7:       Generic_CBPMEnumeration($\mathcal{R}^\wedge \cup \{e\}, \mathcal{R}^\vee$)
8:     end for
9:     Generic_CBPMEnumeration($\mathcal{R}^\wedge, \mathcal{R}^\vee \setminus \{e\}$)
10:   end if
11: end if
Case studies

Mining of

- Formal concepts [IDA journal 05]
- Fault-tolerant patterns [KDID 05, ICCS 06]
- Closed patterns in \( n \)-ary relations [SDM 08]
- Parallel episodes [SDM 09]
- Subspace clustering [SDM 09]
- Topological patterns in static attributed graphs [TKDE 13]
- Evolution patterns in dynamic graphs [ICDM 09]
- Trend dynamic sub-graphs [DS 12, PKDD 13]
- Triggering patterns [ASONAM 14]
Topological patterns in static attributed graphs
Outline

Constraint-based pattern mining framework

Topological patterns in static attributed graphs
  Topological properties
  Rank correlation constraint

Mining dynamic graphs

Trend dynamic sub-graphs

Triggering patterns of topology changes

Conclusions and perspectives
Mining attributed static graphs

- Composed by two relations
- Are these two relations linked?
  - Do the attribute values depend on the role played by the vertex in the graph?

New pattern domain that identifies sets of node attributes that co-vary with the graph structure

- the graph structure is captured by topological properties
Topological properties are derived from the edges of the graph

- Direct neighborhood
  - degree, clustering
- Connection to all other vertices
  - Centrality measures

Microscopic View

- Vertex attributes
- Degree cent.
- Clust. Coeff.
- #quasi-cliques involving v
- Size of the largest quasi-cliques

Macroscopic View

- Closeness Cent.
- Betw. Cent.
- EigenVector Cent.
- PageRank
- Size of the community
- #quasi-cliques involving v

Resource-aware Machine Learning
Topological patterns in static attributed graphs
26 / 78
Topological patterns

Topological pattern language

- $\mathcal{M}$: set of all properties
- $\mathcal{M} \times \{+, -\}$
  - $m^s \equiv (m, s) \in \mathcal{M} \times \{+, -\}$
- $\mathcal{L} = 2^{\mathcal{M} \times \{+, -\}}$

Betweenness centrality
Pattern examples

Co-authorship network

- Node attributes: number of publications in some conferences (KDD, ICDM, VLDB, etc.).

Patterns that can be discovered

- The higher the number of publications at VLDB and SIGMOD, the higher the centrality value.
- The higher the number of publications at SAC
  - The lower the number of publications at KDD and ICDM.
  - The lower the centrality value.
How to extract such patterns?

“Propositionalization”: the topological properties are associated to each node.

<table>
<thead>
<tr>
<th></th>
<th>KDD</th>
<th>PKDD</th>
<th>PODS</th>
<th>...</th>
<th>Page Rank</th>
<th>...</th>
<th>ClustCoef</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author1</td>
<td>14</td>
<td>14</td>
<td>15</td>
<td>...</td>
<td>0.95</td>
<td>...</td>
<td>0.55</td>
</tr>
<tr>
<td>Author2</td>
<td>12</td>
<td>15</td>
<td>14</td>
<td>...</td>
<td>0.85</td>
<td>...</td>
<td>0.65</td>
</tr>
<tr>
<td>Author3</td>
<td>15</td>
<td>13</td>
<td>13</td>
<td>...</td>
<td>0.75</td>
<td>...</td>
<td>0.45</td>
</tr>
<tr>
<td>Author4</td>
<td>11</td>
<td>12</td>
<td>12</td>
<td>...</td>
<td>0.65</td>
<td>...</td>
<td>0.3</td>
</tr>
<tr>
<td>Author5</td>
<td>13</td>
<td>11</td>
<td>11</td>
<td>...</td>
<td>0.55</td>
<td>...</td>
<td>0.1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Tabular data
Topological constraint

Sets of properties that behave in a similar manner for a large number of vertex pairs

- Kendall tau rank correlation coefficient
  - based on the ranks of pattern property values
  - counts the number of vertex pairs that are ordered similarly on all pattern properties

Age

Income
Example

\[ P = \{KDD^+, ClusCoef^-\} \]

- Supported by 5 pairs
- Ex: (A2,A1)

\[ \text{supp}_\tau(P) = \frac{\binom{5}{5}}{\binom{5}{2}} = \frac{1}{2} \]

<table>
<thead>
<tr>
<th></th>
<th>KDD</th>
<th>PKDD</th>
<th>PODS</th>
<th>Clus Coef</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>14</td>
<td>14</td>
<td>15</td>
<td>0.55</td>
</tr>
<tr>
<td>A2</td>
<td>12</td>
<td>15</td>
<td>14</td>
<td>0.65</td>
</tr>
<tr>
<td>A3</td>
<td>15</td>
<td>13</td>
<td>13</td>
<td>0.45</td>
</tr>
<tr>
<td>A4</td>
<td>11</td>
<td>12</td>
<td>12</td>
<td>0.3</td>
</tr>
<tr>
<td>A5</td>
<td>13</td>
<td>11</td>
<td>11</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Constraint properties

- Let \( P \in \mathcal{L} \), \( \text{Supp}_\tau(P) = \frac{|\{(u,v)\in V^2 \mid \forall m^s \in P: m(u) \triangleright_s m(v)\}|}{\binom{n}{2}} \) with

\[ \triangleright_s \equiv \begin{cases} < & \text{when } s \text{ is } + \\ > & \text{when } s \text{ is } - \end{cases} \]

and \( n \) is the number of vertices

\( C_{\text{topo}}(P, D) \equiv \text{Supp}_\tau(P) \geq \sigma \)

- \( C_{\text{topo}} \) is anti-monotone

- Redundant symmetrical patterns

\( \text{Supp}_\tau(A^+, B^-) = \text{Supp}_\tau(A^-, B^+) \)

- Support computation quadratic on the number of vertices:
  - An upper bound to avoid, in linear time, some support computation
  - ECLAT mining algorithm with range trees to ease the search of supporting vertices
Constraint properties

- Let $P \in \mathcal{L}$, $\text{Supp}_\tau(P) = \frac{|\{(u,v)\in V^2 \mid \forall m^s \in P \colon m(u) \mathbin{\triangleright}_s m(v)\}|}{\binom{n}{2}}$ with $\triangleright_s \equiv \begin{cases} < & \text{when } s \text{ is } + \\ > & \text{when } s \text{ is } - \end{cases}$ and $n$ is the number of vertices

\[ C_{\text{topo}}(P, \mathcal{D}) \equiv \text{Supp}_\tau(P) \geq \sigma \]

- $C_{\text{topo}}$ is anti-monotone
- Redundant symmetrical patterns

$$\text{Supp}_\tau(A^+, B^-) = \text{Supp}_\tau(A^-, B^+)$$

- Support computation quadratic on the number of vertices:
  - An upper bound to avoid, in linear time, some support computation
  - ECLAT mining algorithm with range trees to ease the search of supporting vertices
Two case studies

Movie Graph

• Netflix and IMDb (1998-2005).
• Users rate movies (1 to 5 stars).
• 5,972 nodes (movies).
• 64,338 edges (common actor).
• 5 local attributes:
  - Release year, num_users (that rate the movie), avg_rating,
  - stdev_rating, et num_actors
• 9 topological properties

DBLP Graph

• 42,252 nodes.
• 210,320 edges (common publication).
• 29 local attributes:
  - Nb of publications (1990-2013) in 29 journals and conference venues
• 9 topological properties
Some results

- \( \{ \text{avg\_rating}^+, \text{num\_customers}^+ \} \)

"People rate the films they like."

- \( \{ \text{num\_customers}^+, \text{Degree}^+ \} \) "People rate movies with major actors." (e.g., R de Niro, S. Connery, and T. Hanks)

- \( \{ \text{stdev\_rating}^+, \text{PageRank}^- \} \)
  "Controversial movies are isolated."
Some results

- \{avg\_rating^+, num\_customers^+\}

- \{num\_customers^+, Degree^+\} “People rate movies with major actors.” (e.g., R de Niro, S. Connery, and T. Hanks)

- \{stdev\_rating^+, PageRank^-\} “Controversial movies are isolated.”
Some results

- \{\text{avg\_rating}^+, \text{num\_customers}^+\} \\

“People rate the films they like.”

- \{\text{num\_customers}^+, \text{Degree}^+\} “People rate movies with major actors.” (e.g., R de Niro, S. Connery, and T. Hanks)

- \{\text{stdev\_rating}^+, \text{PageRank}^-\}

“Controversial movies are isolated.”
Is publishing at SAC penalizing?

- \( \{ \text{SAC}^+, \text{ECML/PKDD}^- \} \), \( \{ \text{SAC}^+, \text{KDD}^- \} \), \( \{ \text{SAC}^+, \text{VLDB}^- \} \)
- \( \{ \text{SAC}^+, \text{PageRank}^- \} \)
- Of course not!
- Bias in the data (SAC has a much wider spectrum than databases and data mining).
Is publishing at SAC penalizing?

- \{SAC^+, ECML/PKDD^-\}, \{SAC^+, KDD^-\}, \{SAC^+, VLDB^-\}
- \{SAC^+, PageRank^-\}
- Of course not!
- Bias in the data (SAC has a much wider spectrum than databases and data mining).
Results on DBLP - 2

\{Prk^+, \text{Deg}^+, \text{Betw}^+, \text{ClusCoef}^-\}

- Dense part \equiv database
- Other parts \equiv NLP, ML

**Conclusion:** These behaviors are not specific to a community!
PageRank and conferences

Top 5 publications related to the emergence of \{\text{Deg}^+\} and \{\text{Betw}^+\} for Prk^+ (A) and the top 5 authors (B)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Deg</th>
<th>Publication</th>
<th>Factor</th>
<th>Between</th>
<th>Publication</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>ECML/PKDD^+</td>
<td>2.5</td>
<td></td>
<td>PVLDB^+</td>
<td>5.67</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>IEEE TKDE^+</td>
<td>2.28</td>
<td></td>
<td>EDBT^+</td>
<td>5.11</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>PAKDD^+</td>
<td>2.21</td>
<td></td>
<td>VLDB J.^+</td>
<td>4.35</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>DASFAA^+</td>
<td>2.09</td>
<td></td>
<td>SIGMOD^+</td>
<td>4.25</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>ICDM^+</td>
<td>1.95</td>
<td></td>
<td>ICDE^+</td>
<td>3.42</td>
</tr>
</tbody>
</table>

(A)

<table>
<thead>
<tr>
<th>Prk^+</th>
<th>Deg</th>
<th>Publication</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECML/PKDD^+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PVLDB^+</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(B)

<table>
<thead>
<tr>
<th>Prk^+</th>
<th>Between</th>
</tr>
</thead>
<tbody>
<tr>
<td>PVLDB^+</td>
<td></td>
</tr>
</tbody>
</table>

Authors:
- Christos Faloutsos
- Gerhard Weikum
- Jiawei Han
- Jiawei Han
- Philip S. Yu
- David Maier
- Philip S. Yu
- Hector Garcia-Molina
- C. Lee Giles
Mining dynamic graphs
Outline

Constraint-based pattern mining framework

Topological patterns in static attributed graphs

**Mining dynamic graphs**
- Constrained subgraphs in static graphs
- Temporal relationships

Trend dynamic sub-graphs

Triggering patterns of topology changes

Conclusions and perspectives
Objective

- Study the evolution of a graph over time
- Capture strong interactions in the graph and their evolution over time
- Evolving communities of social dynamic graphs
The data

Focus on the **microscopic level** and propose a constraint-based mining approach to uncover **evolving patterns**.
Mine subgraphs in each static graph

Time 1

Time 2

Time 3

Compute highly connected subgraphs that are also isolated.
Subgraphs of interest are usually those made of vertices that have a high density of edges.

⇒ Cliques are subgraphs with maximal density
⇒ Pseudo-cliques relax this strong property using a user-defined threshold $\sigma \in [0, 1]$: 

$$S \text{ is a pseudo-clique iff } \frac{2|E_S|}{|S||S|-1} \geq \sigma$$

Properties

The pseudo-clique constraint is not anti-monotonic: Expanding a subgraph by adding a vertex could make the density increase or decrease.
Pseudo-cliques can always be grown from a smaller pseudo-clique with one vertex less [F. Zhu et al. (PAKDD 2007)]

- Let $S$ be a pseudo-clique
- Let $v^*$ be a vertex of $S$ having the smallest degree on $S$
- Thus $S \setminus \{v^*\}$ is also a pseudo-clique

$\sigma = \frac{2}{3}$
Other useful constraints

Maximality and isolated constraint

Not all the pseudo-cliques of a graph are of importance:

- Some are redundant (because non maximal)
- Others have many links to external vertices

The isolation constraint imposes a maximum to the average number of external links per vertex:

$$\sum_{u \in S} (\deg(u) - \deg_S(u)) \leq \gamma$$
Other useful constraints

\[ \sigma = 0.7 \quad \text{Without Isolated constraint} \quad \text{With } \gamma = 1 \]

![Diagram showing 9 patterns without isolated constraint and 1 pattern with \( \gamma = 1 \)]
Temporal relationships among subgraphs

Combine patterns from **consecutive time stamps** to construct a **global model** of the dynamic of the graphs.
Global model of dynamic graph

Objective
Structured the numerous pseudo-cliques obtained to answer the following questions:

- Do the strong interactions grow, diminish or remain stable over time?
- When do the change occur?

Temporal relationships between time consecutive subgraphs

- **Stability**: S remains the same between $t - 1$ and $t$
- **Growth**: S enlarges at time $t$
- **Diminution**: S shrinks at time $t$
- **Extinction**: S disappears at time $t$
- **Emergence**: S appears at time $t$
Lyon's shared bicycle system Velo'v

Velo'v system
- 340 stations spread in Lyon
- 4000 bikes available in those stations
- rental at any station, return it at any other one

Velo'v data
- More than 13 millions of bicycle trips
  - Time-stamps of the trip
  - Rent and return station IDs
Results for Velo'v graph

Evolving patterns ($\sigma = 0.8$ and $\gamma = 5$) of Velo'v stations and their localization in Lyon. Patterns are shadowed on the map.
Trend dynamic sub-graphs
Outline

Constraint-based pattern mining framework

Topological patterns in static attributed graphs

Mining dynamic graphs

Trend dynamic sub-graphs
  Definition of patterns
  Definition of constraints

Triggering patterns of topology changes

Conclusions and perspectives
Dynamic attributed graph

The data
\[ G_t = (V, E_t), \ t = 1, \ldots, t_{\text{max}} \] and A a set of ordinal attributes:

\[ a_i : V \times T \rightarrow \mathbb{D}_i, \text{ with } \mathbb{D}_i \text{ the domain of } a_i \]
What are the sub-graphs whose vertex attributes evolve similarly?

Mining maximal sub-graphs that satisfy some constraints on the graph topology and on the attribute values

- The connectivity of the dynamic subgraphs is constrained by a maximum diameter value.
- To be more robust towards intrinsic inter-individual variability, we do not compare raw numerical values, but their trends.
Language of patterns

\[ \mathcal{L} = \{(U, S, \Omega) \mid U \subset V, \; S = \langle t_1, \cdots, t_s \rangle, \; \Omega \subset A \times \{+, -\} \} \]

Example: \((\{A, C, D\}, \langle t_0, t_1 \rangle, \{a_1^+, a_3^-\}) \in \mathcal{L}\)
Definition of constraints

A pattern \((U, S, \Omega)\) satisfies

- \(C_{diam}((U, S, \Omega), D) \equiv \text{diameter}_{G_t(u)} \leq k, \ \forall t \in S\)

- \(C_{trend}((U, S, \Omega), D) \equiv \forall u \in U, \ \forall t \in S, \ \forall (a, m) \in \Omega\)
  
  \[
  \left\{
  \begin{array}{l}
  a(u, t) < a(u, t + 1), \ if \ m = + \\
  a(u, t) > a(u, t + 1), \ if \ m = -
  \end{array}
  \right.
  \]

- \(C_{max}((U, S, \Omega), D) \equiv U, \ S \text{ and } \Omega \text{ cannot be enlarged without invalidating one or both of the above constraints.}\)
Constraint properties

- $C_{\text{trend}} \left( \left( U, S, \Omega \right), D \right)$ is anti-monotone
- $C_{\text{diam}} \left( \left( U, S, \Omega \right), D \right)$ is piecewise monotone:

$$diameter_{G_t(U)} \leq k \equiv \max_{v, w \in U} d_{G_t(U)}(v, w) \leq k$$

$$f(U_1, U_2, D) = \max_{v, w \in U_1} d_{G_t(U_2)}(v, w)$$

- $f_{1, U}$ is anti-monotone i.e. if it is satisfied on $U_1$, it is also satisfied for any of its subsets;
- $f_{2, U}$ is monotone i.e. if it is satisfied on $G_t(U_2)$, then, adding some vertices and edges to $G_t(U_2)$ will not increase its value

We can use the propagation mechanisms

- $C_{\text{max}} \left( \left( U, S, \Omega \right), D \right)$ is pushed using specific mechanisms
Katrina hurricane results

Top pattern w.r.t. time specificity (in red)

- 71 airports whose arrival delays increase over 3 weeks.
- Arrival delays never increased in these airports during another week.
- The hurricane strongly influenced the domestic flight punctuality.

Top pattern w.r.t. trend specificity (Yellow)

- 5 airports whose number of flights increased over 3 weeks
- Substitutions flights were provided from these airports during this period.
- This behavior is rather rare in the rest of the graph
Triggering patterns of topology changes
Outline

Constraint-based pattern mining framework

Topological patterns in static attributed graphs

Mining dynamic graphs

Trend dynamic sub-graphs

Triggering patterns of topology changes

Conclusions and perspectives
Context & motivation

Networks structurally change over time: 
How to describe these dynamics?

Intuition
Consider attributed graphs evolving through time
The variation of some attribute values (nodes attributes) of a node can 
lead in several cases to a structural change (topological properties).
Triggering pattern example: \( \langle \{a^+, b^+\}, \{c^-\}, \{deg^+\} \rangle \)

- \(a^+\)  Updating his status more often
- \(b^+\)  giving positive opinions about others
- \(c^-\)  receiving less negative opinions from the others
- \(deg^+\)  is often followed by an increase of user's popularity

Resource-aware Machine Learning

Triggering patterns of topology changes
Dynamic attributed graphs

Let $\mathcal{G} = \{G_1, \ldots, G_t\}$ be a sequence of $t$ static attributed graphs

- $G_i = (V_i, E_i, F_i)$ with $T = \{1, \ldots, t\}$

- $F$ the set of numerical attributes that map each vertex-time pair to a real value: $\forall f \in F, f : V \times T \to \mathbb{R}$. 

  ➤ $F$ gathers the node attributes and the node topological properties
Characterizing vertex behaviors

Vertex descriptive sequence

- A discretization function gives a variation symbol to a vertex/attribute/time triple, e.g.

\[
discr(v, f, i) = \begin{cases} 
+ & \text{if } f(v, i) - f(v, i - 1) \geq 2 \text{ and } i > 1 \\
- & \text{if } f(v, i) - f(v, i - 1) \leq -2 \text{ and } i > 1 \\
\emptyset & \text{otherwise}
\end{cases}
\]

- A vertex \( v \) is described by a sequence of itemsets

\[
\delta(v) = \langle \{discr(v, f, 1) \mid f \in F\}, \ldots, \{discr(v, f, t) \mid f \in F\} \rangle
\]

- \( \Delta = \{\delta(v) \mid v \in V\} \) is the set of all sequences.

Example

\[
\delta(u_1) = \langle \{a^+, b^+\}, \{c^-\}, \{deg^+\} \rangle
\]

\[
\Delta = \{\delta(u_1), \delta(u_2), \delta(u_3), \delta(u_4), \delta(u_5)\}
\]
Triggering patterns

A triggering pattern is a sequence \( P = \langle L, R \rangle \) with
- \( L \) a sequence of node attribute variations
- \( R \) a single topological property variation

Support measure

\[
\text{supp}(P, \Delta) = \{ v \in V \mid P \preceq \delta(v) \}
\]

where \( p \preceq q \) means that \( p \) is a super-sequence of \( q \)

Example

- \( L = \langle \{ a^+, b^+ \}, \{ c^- \} \rangle \)
- \( R = \langle \{ \text{deg}^+ \} \rangle \)
- \( \text{supp}(\langle L, R \rangle, \Delta) = \{ u_1, u_3 \} \)
Assessing the strength of a pattern

Triggering pattern growth rate

Let $P = \langle L, R \rangle$, we denote by $\Delta^R \subseteq \Delta$ the set of vertex descriptive sequences that contain $R$. The growth rate of $P$ is given by:

$$GR(P, \Delta^R) = \frac{|\text{supp}(L, \Delta^R)|}{|\Delta^R|} \times \frac{|\Delta \setminus \Delta^R|}{|\text{supp}(L, \Delta \setminus \Delta^R)|}$$

---

G. Dong and J. Li.
Efficient mining of emerging patterns: Discovering trends and differences.
In KDD, pages 43--52, 1999.
Quantitative results

(i) Runtimes and number of patterns, (ii) Distribution of execution times, (iii) Support distribution, (iv) Growth rate vs support, (v) Scalability.
The DBLP data

Detecting asynchronous events

\[ \{\text{eigenvector}^{++}_1\}, \{\text{VLDB}^{++}, \text{degree}^{++}_2\} \rightarrow \{\text{degree}^{++}_3\} \]
Synchronous events

- RITA1: daily in September 2001
  \{{\# Cancelled}^+, \{Deg^-, Close^-, NbCliq^-, Prk^-, Betw^-\} \rightarrow Deg^+\}
  Airports that absorb the traffic two days after

  \{{\# Cancelled}^+, \{\# Cancelled^-\}, \{nbCliq^-, Betw^+\} \rightarrow nbCliq^+\}
  A "back to normal" around March 2002

- RITA3: Aug./Sept. 2005 (Katrina Hurricane)
  \{{\# Cancelled}^+, \ DelayAtDep^+\}, \{\# Diverted^-, \# Depar^-, \# Arrival^-\} \rightarrow \{close^-\}
  All the airports supporting this pattern are located in the US West coast where Katrina raged.
Conclusions and perspectives
Conclusion

- Constraint-based pattern mining framework can be used to analyze dynamic graphs
- Patterns combine information about the topological structure, the vertex attribute values and their tendencies.
  - A wide variety of data can be mined
  - Provide new insights on the techniques that can be used to analyze dynamic graphs
Enhancing the quality of the extracted information

• by capturing changes in the graph structure
• while allowing changes in trend direction
  Combining sub-graph and sequence mining
Avoiding the known drawbacks of pattern mining approaches

- How to get rid of redundant patterns?
- How to fix the threshold values?

Returning patterns whose constraint values are surprising

- with respect to what could be expected from randomized data
- with respect to what happened in the past (real-time data mining)
  - Trigger alerts when changes in behavior
Avoiding the known drawbacks of pattern mining approaches

- How to get rid of redundant patterns?
- How to fix the threshold values?

Returning patterns whose constraint values are surprising

- with respect to what could be expected from randomized data
- with respect to what happened in the past (real-time data mining)
  - Trigger alerts when changes in behavior
Pattern without threshold values: Skyline patterns
Retrieves the patterns that are not dominated by any other pattern

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>A B C D E F</td>
</tr>
<tr>
<td>$t_2$</td>
<td>A B C D E F</td>
</tr>
<tr>
<td>$t_3$</td>
<td>A B</td>
</tr>
<tr>
<td>$t_4$</td>
<td>D</td>
</tr>
<tr>
<td>$t_5$</td>
<td>A C</td>
</tr>
<tr>
<td>$t_6$</td>
<td>E</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Patterns</th>
<th>freq</th>
<th>length</th>
<th>area</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCDEF</td>
<td>2</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>AB</td>
<td>3</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>AC</td>
<td>3</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>A</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>ABCD</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

A. Soulet, C. Raissi, M. Plantevit, B. Cremilleux - 2011
Ongoing research projects

Vél'innov - ANR INOV 2012 - labex IMU project

• Constraint-based pattern mining of Vélo'V data
  • Characterizing the usage of the bicycle sharing system
  • Modeling the system to simulate the impacts of local changes (in the city or in the system)

• Pitfall:
  • Vertex attributes are static (INSEE data) or derived from the edges (e.g., number of users of a considered category)
  
  Searching for relationships between attributes and the graph structure does not make sense

• Envisioned solutions:
  • Considering the user trajectories and looking for paths that characterize a user group
  • Considering an attributed static graph with temporal usage vectors associated to edges and seeking for sub-graphs homogeneous on vertex and edges attributes
Ongoing research projects - 1

GRAISearch: enhancing a location-based social media search engine

- FP7-PEOPLE-2013-IAPP: Industry-Academia Partnerships and Pathways
- Benefits for DM2L team
  - 40 researcher months
  - Access to data with high added value
  - An industrial cooperation with critical potential benefits for the company
Ongoing research projects - 2

GRAISearch: Research tasks

- Geo-localized demographic flow prediction using geo-tagged social media uploaded data

- Geo-localized event detection algorithm

- Integration into a geo-located social media recommender system
Acknowledgments