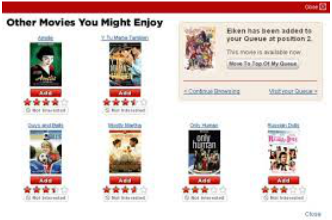
Granularity of co-Evolutions Patterns in Dynamic Attributed Graphs

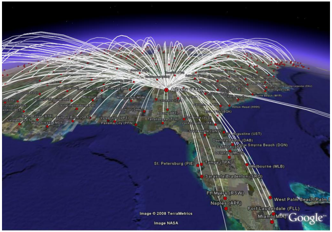
Elise Desmier, Marc Plantevit, Céline Robardet, Jean-François Boulicaut

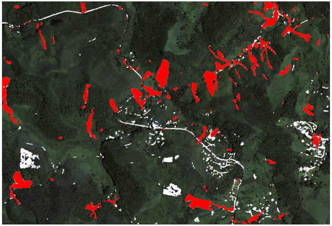
The Thirteenth International Symposium on Intelligent Data Analysis (IDA 2014), Leuven, Belgium.

Data: a new "natural ressource"

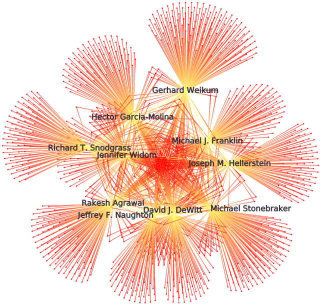
Potential increase of our knowledge

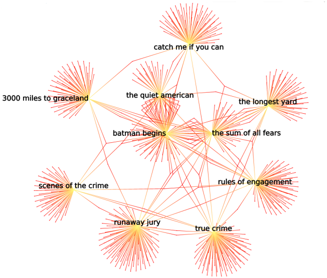


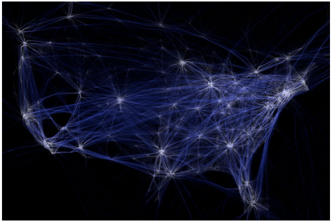


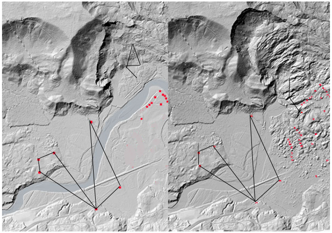


Viewed as dynamic attributed graphs

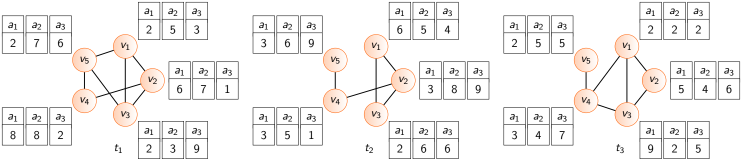






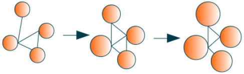


Objectives



Extract patterns from dynamic attributed graphs.

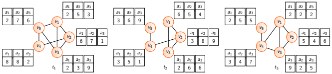
- Entities
- Interactions (social, spatial, etc.)
- Dynamic
- Entity information



Pattern that describes a local temporal phenomena

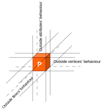
We propose to mine *maximal dynamic attributed sub-graphs* that satisfy some constraints on the *graph topology* and on the *attribute values*.

Overview of our proposal



Co-evolution patterns

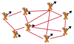
Interestingness Measures

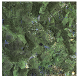


(Desmier et al., ECML/PKDD 2013)

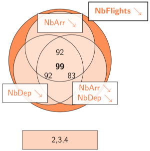
Experimental results

DBLP US flights Brazil landslides

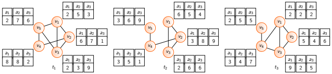




- Some obvious patterns are discarded ...
- ... but some patterns need to be generalized

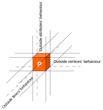


Overview of our proposal



Co-evolution patterns

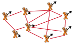
Interestingness Measures

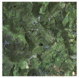


(Desmier et al., ECML/PKDD 2013)

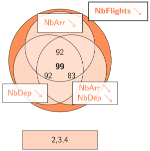
Experimental results

DBLP | US flights | Brazil landslides





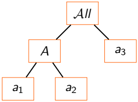
- Some obvious patterns are discarded ...
- ... but some patterns need to be generalized



Hierarchical co-evolution patterns

Take benefits from a hierarchy over the vertex attributes to :

- return a more concise collection of patterns;
- discover new hidden patterns;



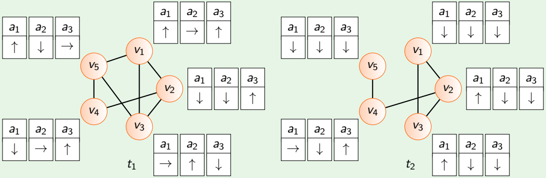
Talk Outline

Dynamic Attributed Graphs

A dynamic attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{T}, \mathcal{A})$ is a sequence over $\mathcal{T}$ of attributed graphs $G_t = (\mathcal{V}, E_t, A_t)$, where:

- $\mathcal{V}$ is a set of vertices that is fixed throughout the time,
- $E_t \in \mathcal{V} \times \mathcal{V}$ is a set of edges at time t,
- A_t is a vector of numerical values for the attributes of $\mathcal{A}$ that depends on t.

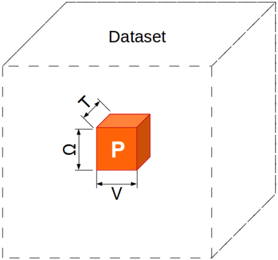
Example



Co-evolution Pattern

Given $\mathcal{G} = (\mathcal{V}, \mathcal{T}, \mathcal{A})$, a co-evolution pattern is a triplet $P = (V, T, \Omega)$ s.t.:

- $V \subseteq \mathcal{V}$ is a subset of the vertices of the graph.
- $T \subset \mathcal{T}$ is a subset of not necessarily consecutive timestamps.
- Ω is a set of signed attributes, i.e., $\Omega \subseteq A \times S$ with $A \subseteq \mathcal{A}$ and $S = \{+, -\}$ meaning respectively a $\{\textit{increasing}, \textit{decreasing}\}$ trend.



Predicates

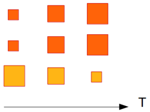
A co-evolution pattern must satisfy two types of constraints:

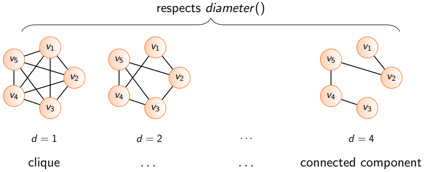
Constraint on the evolution:

- Makes sure attribute values co-evolve
- We propose δ-**strictEvol**.
- $\forall v \in V$, $\forall t \in T$ and $\forall a^s \in \Omega$ then $\delta\text{-}trend(v, t, a) = s$

Constraint on the graph structure:

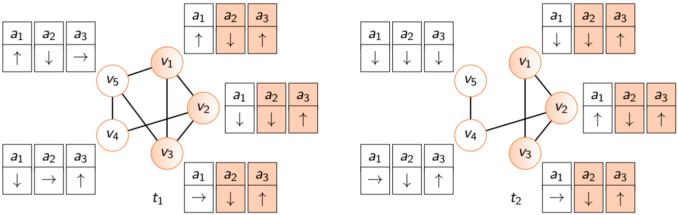
- Makes sure vertices are related through the graph structure.
- We propose **diameter**.
- $\Delta\text{-}diameter(V, T, \Omega) = true \Leftrightarrow \forall t \in T\ diam_{G_t(V)} \leq \Delta$



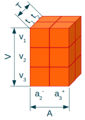


Example

$$P = \{(v_1, v_2, v_3)(t_1, t_2)(a_2^-, a_3^+)\}$$



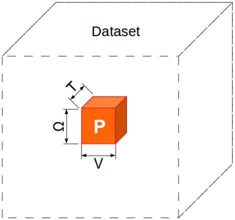
- 1-Diameter(P) is true,
- 0-strictEvol(P) is true.

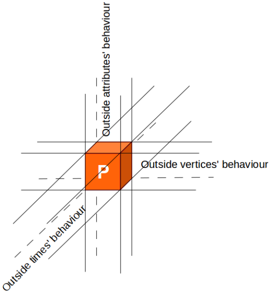


Density Measures

Intuition

Discard patterns that depict a behaviour supported by many other elements of the graph. We propose : **vertex specificity**, **temporal dynamic** and **trend relevancy**.





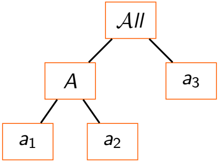
Talk Outline

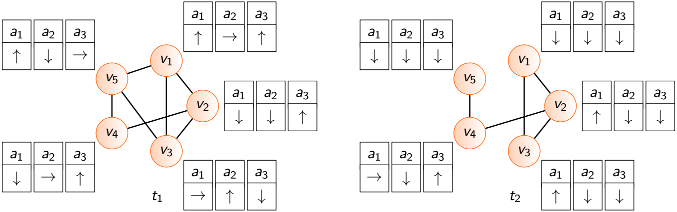
1. Co-evolution patterns
2. **Hierarchical co-evolution patterns**
3. Empirical study
4. Conclusion and perspectives

Hierarchy

A hierarchy $\mathcal{H}$ on $\mathcal{A}$ is a tree where:

- the edges are a relation is_a,
- the node $\mathcal{A}ll$ is the root of the tree,
- the leaves are attributes of $\mathcal{A}$,
- dom($\mathcal{H}$) is all the nodes except the root.

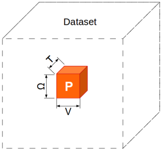




Hierarchical co-evolution Patterns

Given $\mathcal{G} = (\mathcal{V}, \mathcal{T}, \mathcal{A})$ and $\mathcal{H}$, a hierarchical co-evolution pattern is a triplet $P = (V, T, \Omega)$ s.t.:

- $V \subseteq \mathcal{V}$ is a subset of the vertices of the graph.
- $T \subset \mathcal{T}$ is a subset of not necessarily consecutive timestamps.
- Ω is a set of signed attributes, i.e., $\Omega \subseteq A \times S$ with $A \subseteq dom(\mathcal{H})$ and $S = \{+, -\}$ meaning respectively a $\{increasing, decreasing\}$ trend.

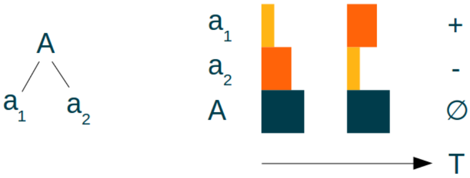


It must respect the following constraints:

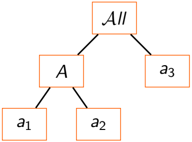
1. Constraint on the evolution.
2. Constraint on the graph structure.

Evolution Constraint

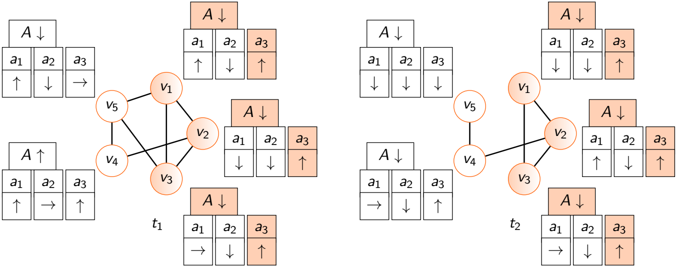
For an attribute A, its evolution is computed from the evolution of the leaves it covers.



Example



$$P = \{(v_1, v_2, v_3)(t_1, t_2)(A^-, a_3^+)\}$$

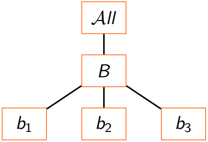


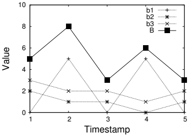
- 1-Diameter(P) is true,
- 0-strictEvolHierarchical(P) is true.

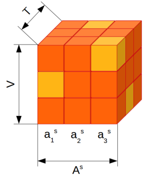
Purity of the pattern

Is the pattern described with the good level of granularity?

Purity computes the proportion of valid triplet (v, t, a^s) with regard to the number of possible triplets.



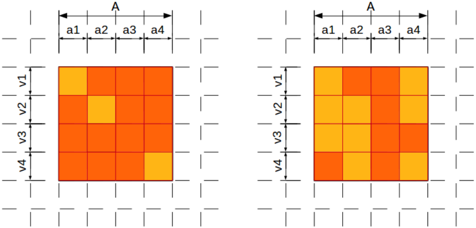




$$purity(P) = \frac{\sum_{v \in V} \sum_{t \in T} \sum_{a^s \in leaf(\Omega)} \delta_{a^s(v,t)}}{|V| \times |T| \times |leaf(\Omega)|}$$

Gain of purity

Does the pattern need to be specialized or kept generalized?



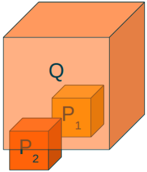
Keep attribute A Specialize A to a_3

Gain measure compares the purity of the "children" pattern P with regard to the purity of its "parent" patterns.

$$gain(P) \Leftrightarrow \frac{purity(P)}{max_{P_i \in parent(P)}(purity(P_i))} \geq \gamma$$

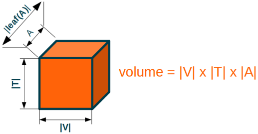
What about other measures/constraints?

Maximality:

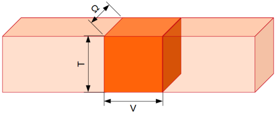


Size measures:

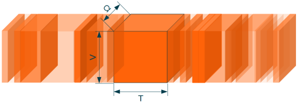
- $|leaf(A)| \geq min_A$,



Vertex specificity:



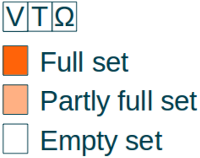
Temporal dynamicity:



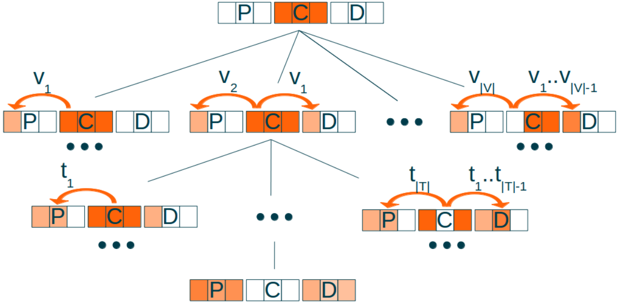
No trend relevancy with hierarchies.

- What level of hierarchy do we consider?
- What about attributes discarded because of a too small purity gain?

Algorithm: Enumeration



- *In* (P): the pattern in construction
- *Cand* (C) : the elements not yet enumerated
- *Out* (D): the elements enumerated as non member of the pattern



Algorithm: Constraint Properties

How to use the properties of the constraints to reduce the search space?

Anti-monotonicity:

- strictEvol
- volume (bound with H)
- size measures

Piecewise anti-monotonicity:

- purity (bound)
- vertex specificity (bound with H)
- temporal dynamic (bound with H)
- trend relevancy (bound)

$f(P \cup C) > \sigma$ is anti-monotonic

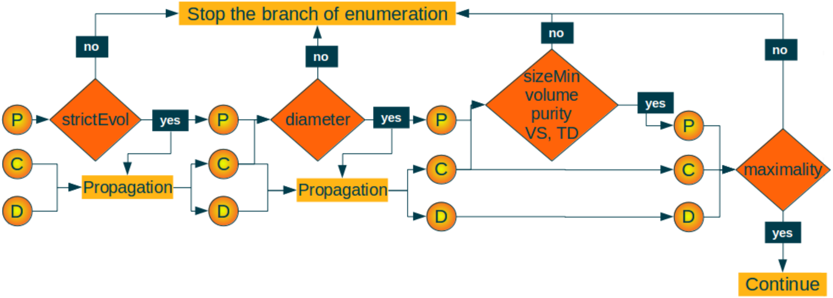
$f(P) > \sigma$ is monotonic

$f = \frac{a(P \cup C)}{b(P)} > \sigma$ is piecewise anti-monotonic

Algorithm: Pruning and Propagation

Use the constraints at each step of the enumeration to

- prune: stop the branch of enumeration,
- propagate: discard elements from C and D.



Talk Outline

Quantitative Experiments

Co-evolution Patterns

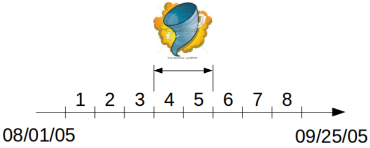
Using relevancy constraints

- ↘ Execution times
- ↘ Number of patterns

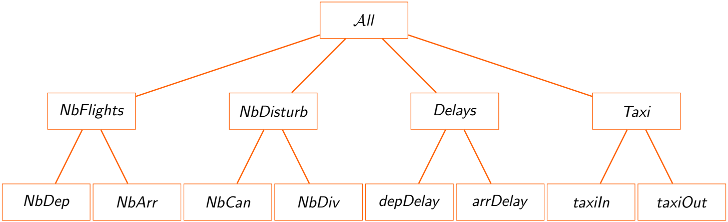
Hierarchical Co-evolution Patterns

- ↗ Execution times
- ↘ Number of patterns

US flights datasets



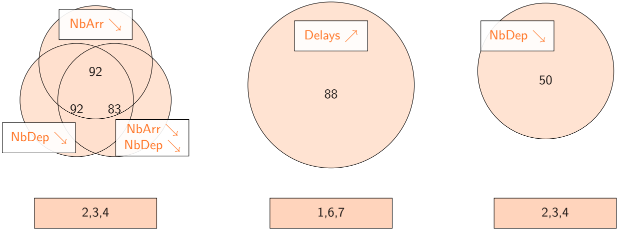
- Vertices: 280 airports.
- Times: 8 weeks around the Katrina hurricane.
- Attributes: number of departure/arrival/cancelled/deviated flights, departure/arrival delays and ground times.

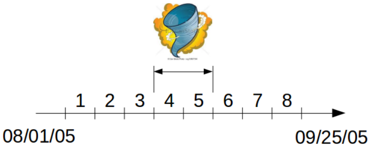


RITA "On-Time Performance" database.
(http://www.transtats.bts.gov)

Hierarchy impact

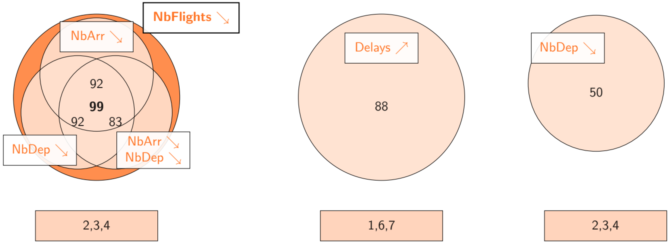
- 2 experiments with $\gamma = 1$ and $\gamma = 1.2$,
- Thresholds: min_V=40, min_T=min_A=ϑ=1, ψ=0.9, κ=0.2, τ=0.4.

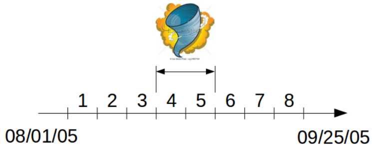




Hierarchy impact

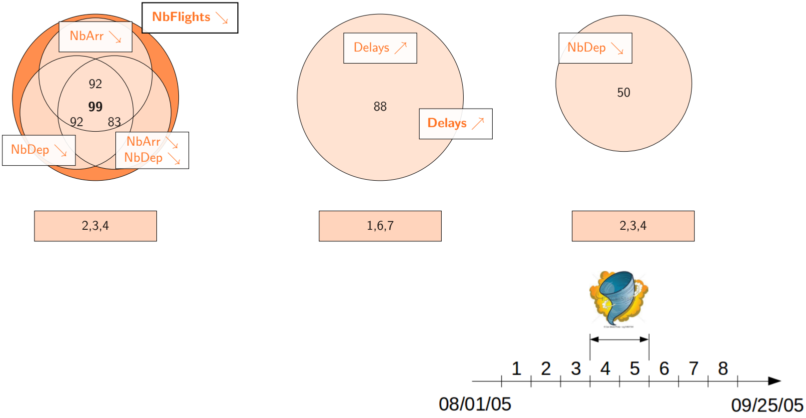
- 2 experiments with $\gamma = 1$ and $\gamma = 1.2$,
- Thresholds: min_V=40, min_T=min_A=ϑ=1, ψ=0.9, κ=0.2, τ=0.4.





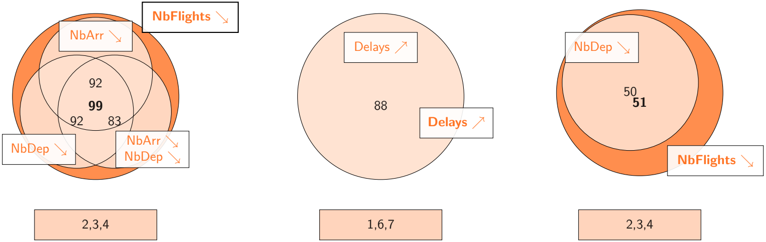
Hierarchy impact

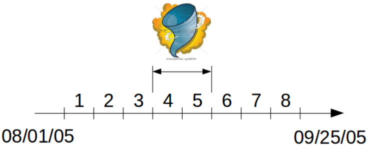
- 2 experiments with $\gamma = 1$ and $\gamma = 1.2$,
- Thresholds: min_V=40, min_T=min_A=ϑ=1, ψ=0.9, κ=0.2, τ=0.4.



Hierarchy impact

- 2 experiments with $\gamma = 1$ and $\gamma = 1.2$,
- Thresholds: min_V=40, min_T=min_A=ϑ=1, ψ=0.9, κ=0.2, τ=0.4.





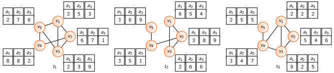
Talk Outline

Conclusion

We propose to extract hierarchical co-evolution patterns from a dynamic attributed graph and a hierarchy.

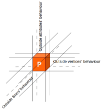
- These patterns are sets of vertices that are connected and that follow the same trends over a set of attributes over time, with attributes that are either those of the dataset or of the hierarchy.
- Definition of some constraints to reduce the execution time and increase the relevancy of the patterns, in particular according to hierarchy.
- Design of an algorithm to compute the complete set of patterns
- Experiments on a real-world dataset prove that this method extracts, in a feasible time, interesting patterns based on the user parametrized constraints.

Future Directions



Co-evolution patterns

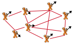
Interestingness Measures



(Desmier et al., ECML/PKDD 2013)

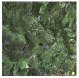
Experimental results

DBLP



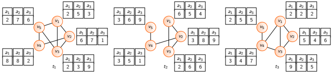
US flights

Brazil landslides



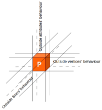
- Some obvious patterns are discarded ...
- ... but some patterns need to be generalized ✓
- Difficulties to set parameters.

Future Directions



Co-evolution patterns

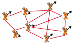
Interestingness Measures



(Desmier et al., ECML/PKDD 2013)

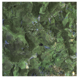
Experimental results

DBLP



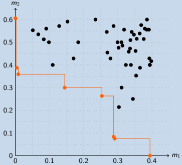
US flights

Brazil landslides



- Some obvious patterns are discarded ...
- ... but some patterns need to be generalized ✓
- Difficulties to set parameters.

⇒ Skyline Analysis



Thank you !

Follow @LirisDM2L