Intelligence Artificielle et Fouille de données

questions ouvertes et perspectives pour les thématiques des JFPDA

Thomas Guyet avec des idées de
René Quiniou et Alexandre Termier

Agrocampus-Ouest/Inria/IRISA

JFPDA, July 2015, Rennes

Objectifs de la présentation

- Présentation des thématiques de recherche locales
  - ⇒ équipe DREAM en transition vers une équipe plus centré sur les thèmes de la fouille de données
- Objectif ambitieux de pointer des rapprochements entre
  - les thématiques des JFPDA (décision, plannification, méthodes formelles, etc) dont je connais pas grand chose !
  - les thématiques de *pattern mining* qui m'occupe la plupart de mon temps

⇒ Les questions sont principalement portées dans le sens de l'intérêt des approches d'intelligence artificielle pour améliorer les méthodes de fouille de données

⇒ Deux objectifs auquelles elles peuvent contribuer

- extraire de "meilleurs" motifs en intégrant du raisonnement et des connaissances
- améliorer l'utilisation des motifs extraits en les transformant en connaissances "utilisables"

Outline

DREAM Team

A team working on artificial intelligence tools to bring novel tools to diagnose, to monitor and to model dynamic systems.

- D : Diagnosis
  - $\Rightarrow$ causal reasoning, logic programming, ASP
- RE : Recommending

  A : Action
  - $\Rightarrow$ argumentation, logic programming
- M : Modelling
  - $\Rightarrow$ automatons, model checking approaches,
  - $\Rightarrow$ data mining, machine learning, stream mining, spatial and temporal

Monitoring dynamical systems

Dynamical system

- A **dynamical system** is a system that evolves in time,
- The evolution of the system depends on the **context**, the context may change over the time,
- Some systems **behaviours** are *faulty*,
- A dynamical system generates traces that inform about the system state along time.

Monitoring/diagnosing dynamical systems

- Monitoring a dynamical system is deciding, at each instant, if the dynamical is (or will be) in a faulty behaviour or not.
- Diagnosing is identifying the explanation of a (faulty) behaviour

Some system examples

- living systems : patients in ICU [1], cows, ...
  - ⇒ monitoring the physiologic signals of the patient.
  - ⇒ early detection of major events (vélages, chaleurs).
- environmental systems : vegetation, watershed ...
  - ⇒ monitoring the evolution of the environment from satellite images
- technical systems : engines, electronic chips, electrical networks, ...
  - ⇒ prediction of regional electrical consumption from individual instantaneous consumption records
  - ⇒ detection of anomalies in CHIPSET execution traces (Go/minutes of data)
- digital systems : software, web server, ...
  - ⇒ detection of intrusions in web servers from logs

1. Intensive Care Units

Model-based diagnosis/monitoring

Definition

Model-based diagnosis uses explicit models of the system to be diagnosed.

- monitoring is comparing the current state to the models,
  - model(s) of faulty behaviours : require to know the faulty behaviours (!), very sensitive to context changes,
  - model(s) of normal behaviours : difficult to construct, generate more false negatives,
- reasoning can use general knowledge about the system,

Building the model

- The models are difficult to construct
- The richer is the model, the more it is difficult to learn
- The context changes require on-line model building

Using trace patterns as model

- The system execution trace reveals the state of the system : we assume that traces holds enough information to discriminate *normal* and *faulty*
- Execution traces : numerical/symbolic, regular sampling or event based,
- Patterns in traces are interesting to characterize the system behaviours.
  1. patterns are mapped to behaviours
  2. a set of patterns is mapped to a behaviour (more robust)

Traces as streams of itemsets

Traces are timestamped discrete events : stream of itemsets

- numerical signals have to be discretized
- timestamped are discrete values
- several events may append at the same time

RISA

Using trace patterns as model : two classical tasks

Monitoring the system with patterns

- The objective is to use "patterns" (or an formal object constructed from the patterns) to decide about the state of the system
- Current difficulties :
  - compare the current state of the system with the patterns
  - concept drift : the (normal/faulty) behavior of the system may change over time
  - efficiently extracting patterns on the fly (datastream mining)

Supporting the expert to interpret data

- The objective is to use the "patterns" to give an abstract view of the system behaviors and to support the analysis of the data.
- The expert will then create appropriate models to
- difficulties
  - being able to process more and more data
  - being able to extract more accurate patterns
  - being able to integrate the expert knowledge in the mining process

RISA

Example : Electrocardiogram (ECG)

Objective

Extracting patterns that are characteristic to some cardiac diseases.

Electrocardiogram segmentation

- electrocardiogram may be abstract by a sequence of events (three typical events : `P`, `QRS`, `T`),
- signal processing may automatically annotate time series.

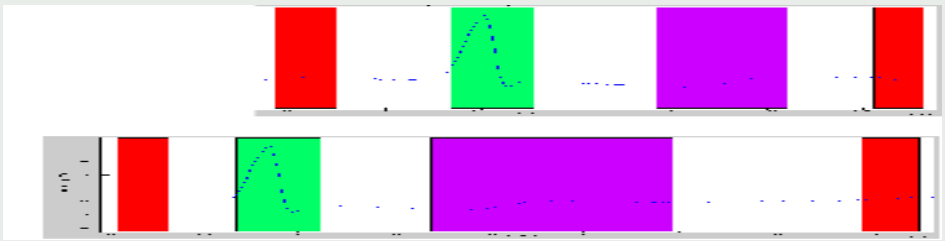


Figure : Two examples of ECG : normal (left), disease (right)

- monitoring usage : detecting on-line the first pattern triggers an alarm
- analyzing usage : automatically annotate ECG

Spatial data mining

Spatial data mining

Objectives : support the understanding of the impact of spatial organizations on agro-environmental processes.

- mining attributed graphs for characterizing and simulating agricultural landscapes.
- discriminant learning of spatial organization (PayTal), water and pesticide runoffs prediction (SACADEAU), urban sprawling prediction (PAYTAL)
- simulation of "realistic" landscape : mixing patterns and expert knowledge

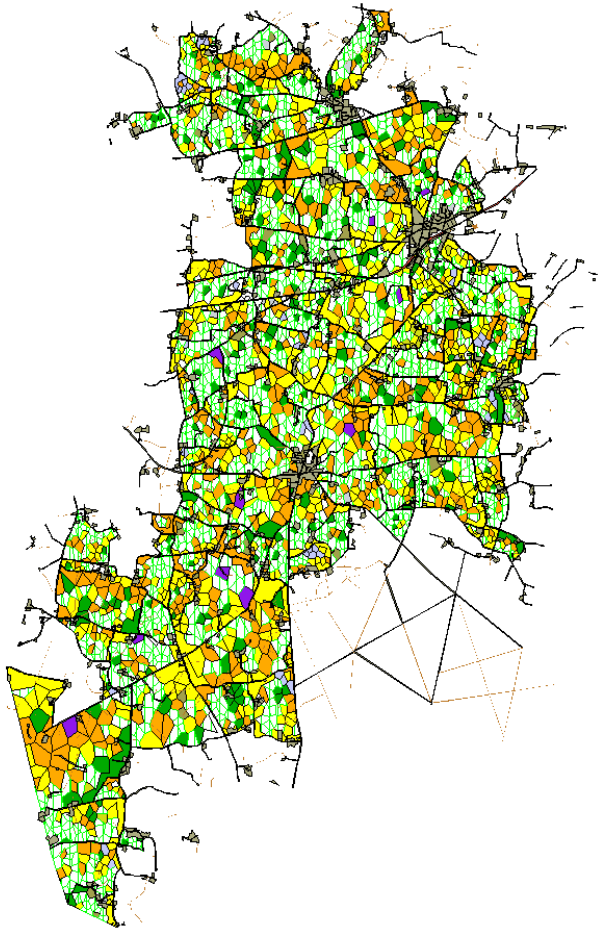


Figure : Simulated landscape from frequent spatial patterns

Temporal data mining

Temporal data mining

Objectives : modelling the behaviours of dynamical systems for online monitoring

- Data : time series, sequential data
- Modelling objective : building model for dynamical system monitoring
- Some challenges :
  - extracting multi-scale symbolic patterns [MGQ13]
  - extracting frequent behaviours from traces
  - online adaptation to concept-changes

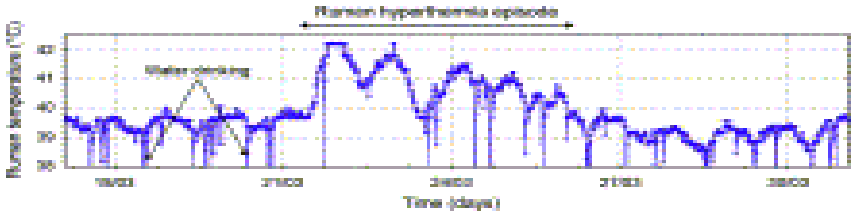


Figure : Temperature of cow : time series analysis

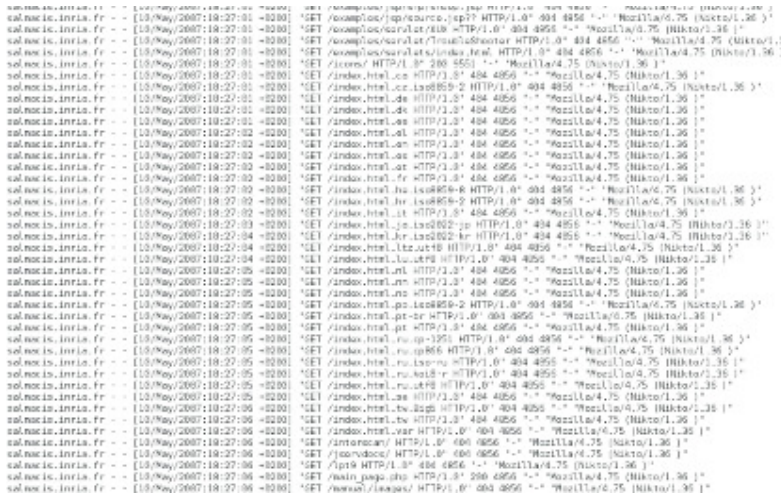
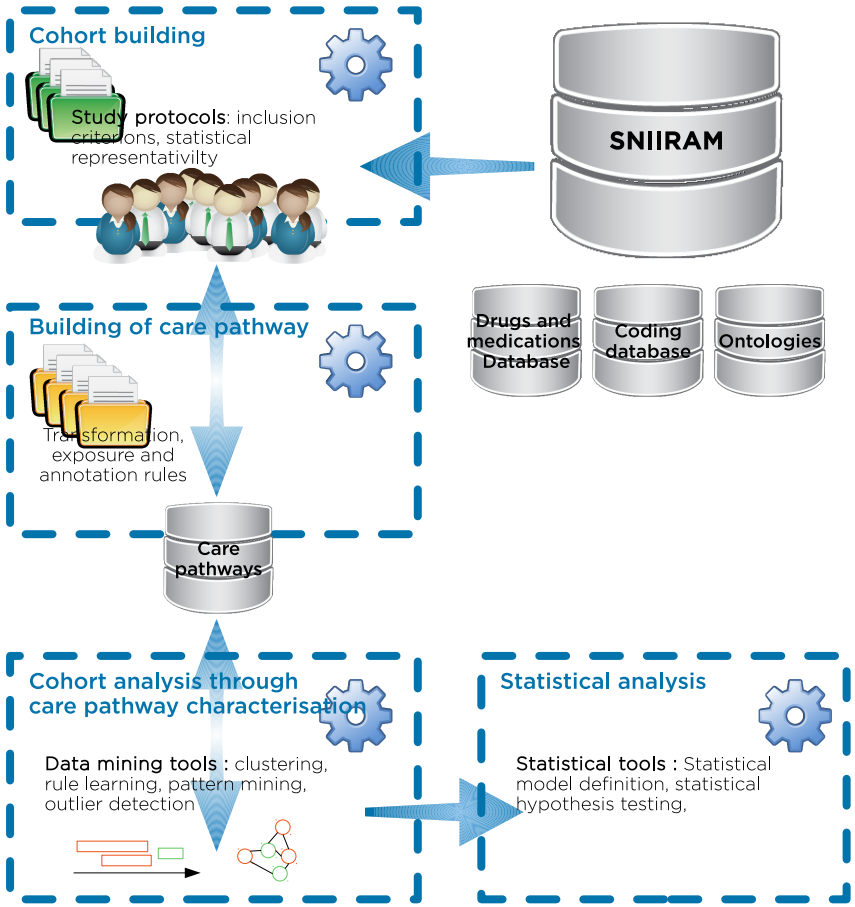


Figure : Web log : sequences of discrete events

Temporal data mining : PEPS project

Développement d'une plate-forme pour mener des études de pharmaco-épidémiologies à partir des données de l'Assurance Maladie



Temporal data mining : PEPS project

Développement d'une plate-forme pour mener des études de pharmaco-épidémiologies à partir des données de l'Assurance Maladie

Accélérer la phase initiale d'exploration des données Besoins d'outils pour manipuler facilement les parcours de soins

- Permettre l'expression de requêtes complexes
- Intégrer des outils de requête et de fouille
- Lier les outils aux méthodes de visualisation

Aider à la formulation de nouvelles hypothèses

- Besoin de dépasser le test statistique de co-occurrences
- Caractéristiques des parcours de soins : identification de séquences de consommations et d'effets
- Associations entre parcours et caractéristiques de patient

Focusing research on data mining

Les questions de fouille de données deviennent centrales dans nos activités, mais toujours dans une approche de l'Intelligence artificielle.

Vers une nouvelle équipe de recherche portée par A. Termier.

Thèmes d'intérêt (*non-contractuel ;)*)

- Automating the exploration of the KDD search space
- Collaborative knowledge and feedback management
- Scaling up through in-memory approaches
- User/system interactions

$\Rightarrow$ Large Collaborative Data Mining

Outline

Présentation d'une approche exemple

On présente ici une approche déclarative de l'extraction de motifs

- rapprochement vers la résolution de tâches formelles (telles que la planification ou autre ?)
- premier pas de rapprochement vers l'IA : utilisation d'outils de raisonnement automatique (*answer set programming*) pour résoudre la tâche
- perspective d'intégration de connaissances et d'heuristiques dans la résolution

Frequent pattern mining

- A sub-task of the data mining field
- The task
  - Patterns : itemsets (a set of items), sequences, graphs
  - Data : database of structured transactions, *e.g.* itemsets, sequences
  - Goal : **extracting all the frequent patterns in a database**.
- Frequent ? The number of transaction that support the pattern is above a given threshold σ
- Classical algorithms
  - Itemsets : Apriori, FP-Growth, LCM, ...
  - Sequential patterns : GSP, PrefixSpan, SPADE, BiDE, ...
  - Graphs : gSpan

Frequent patterns : itemsets

TID	itemset
10	a,b,c
20	a,c,d
30	a,d
40	b,e,f

- frequency threshold : 2
- set of frequent patterns : $\{a, b, c, d, ac, ad\}$

Frequent pattern mining

- A sub-task of the data mining field
- The task
  - Patterns : itemsets (a set of items), sequences, graphs
  - Data : database of structured transactions, *e.g.* itemsets, sequences
  - Goal : **extracting all the frequent patterns in a database**.
- Frequent ? The number of transaction that support the pattern is above a given threshold σ
- Classical algorithms
  - Itemsets : Apriori, FP-Growth, LCM, ...
  - Sequential patterns : GSP, PrefixSpan, SPADE, BiDE, ...
  - Graphs : gSpan

Frequent patterns : sequences

SID	sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

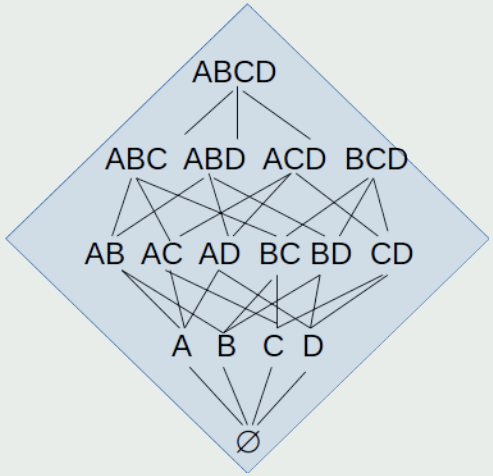
- frequency threshold : 2
- set of frequent patterns : $\{a, ab, ac, ad, af, (ab), abc, \ldots\}$

Mining frequent patterns : algorithms principles I

Search space of patterns

- patterns are ordered according to a partial order (inclusion relation, $\subset$)

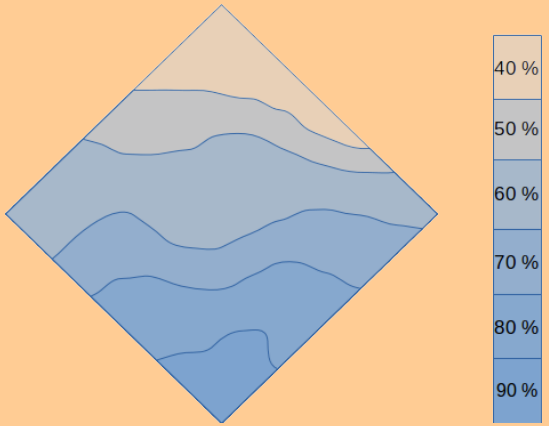
$\Rightarrow$ lattice of patterns



Anti-monotony property of the frequency

- Let P be a pattern and $supp(P)$ its support. $\forall Q \subset P$, we have $supp(Q) \geq supp(P)$.

$\Rightarrow$ the frequency of a pattern decrease with its depth in the lattice



Mining frequent patterns : algorithms principles II

Frequent pattern mining

Mining all the pattern whose support is above a given threshold σ.

- support : number of transaction in the database that support the pattern
- a transaction supports a pattern if it "contains" the pattern
  - itemsets : subset of a transaction
  - sequences : sub-sequence of a transaction (sequence) with gaps !

Mining frequent patterns : algorithms principles III

Two most important issues of the pattern mining algorithms

1. search space traversal
   - ⇒ use the anti-monotony property to cut the search as soon as possible
   - ⇒ the browsing must be efficient and it avoids redundant evaluation of pattern
2. support evaluation
   - ⇒ reduce the number of access to the database
   - ⇒ split the databases in sub-databases (enable efficient parallelism)
   - ⇒ pattern-transaction matching algorithm (simple for itemsets, but may be harder, *e.g.* graphs)

Pattern mining and Declarative Programming

"Main" principles of Declarative Programming

1. a set of "generators" that generates potential answers ← **traversal of the search space**
2. a set of "constraints" that defines what is a "valid" model and prunes the others ← **support evaluation**
3. optimization criteria ← **pattern selection**

- Constraint programming approaches []
- Answer Set Programming [] : appropriate to integrate knowledge

Our objectives

- finding efficient declarative encoding of the pattern mining tasks
- integrating reasoning into the mining process

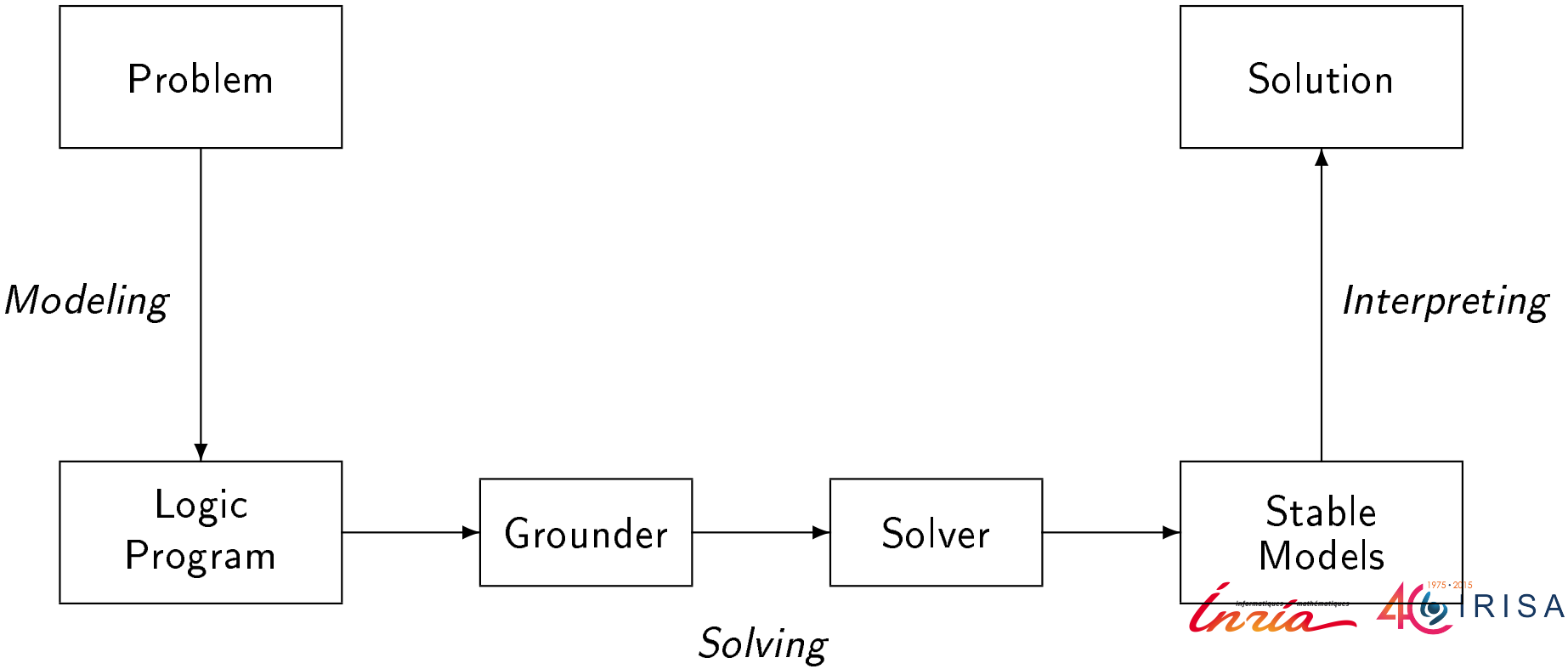
⇒ especially applied on the **sequential pattern mining** issue

RISA

Answer Set Programming

Programme ASP

- idéal de la programmation déclarative : *le problème est le programme*
- la solution est un ensemble de modèles (ensembles réponses)
- utilisation de contraintes sur des ensembles



Programme ASP

Principe "général" de programmation

1. ensemble de règles pour "générer" des modèles (réponses potentielles)
2. ensemble de contraintes : éliminent les modèles non-souhaités
3. directive d'optimisation : #minimize, #maximize
4. directive d'affichage : #show

$\Rightarrow$ syntaxe proche du prolog (symboles de premier ordre) faciles à écrire

Coloration de graphe

```
node(1). node(2). node(3). node(4). node(5). node(6).
edge(1,2). edge(1,3). edge(1,4). edge(2,4). edge(2,5). edge(2,6).
edge(3,1). edge(3,4). edge(3,5). edge(4,1). edge(4,2). edge(5,3).
edge(5,4). edge(5,6). edge(6,2). edge(6,3). edge(6,5).
col(r). col(b). col(g).

1 { color(X,C) : col(C) } 1 :- node(X).
:- edge(X,Y), color(X,C), color(Y,C).

#show color/2.
```

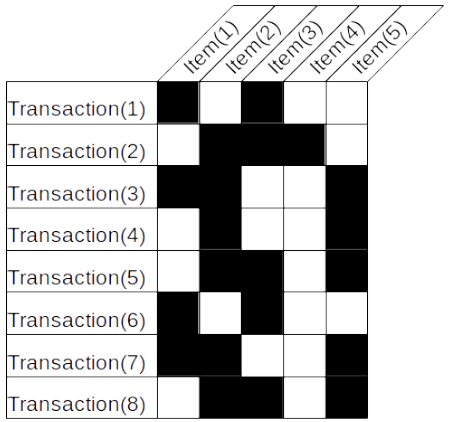
Fouille d'itemsets avec ASP

Programme ASP proposé par Järvisalo [Jär11]

```
item(I) :- db(_,I). % ensemble des items de la base de transactions
transaction(T) :- db(T,_). % ensemble des transactions de la base

{ in_itemset(I) } :- item(I). % generation d'un motif par AS
in_support(T) :- { conflict_at(T,I) : item(I) } 0, transaction(T).
conflict_at(T,I) :- not db(T,I), in_itemset(I), transaction(T).
:- { in_support(T) } N-1, threshold(N).
```

- **item**/1 : liste d'items
- **transaction**/1 : liste des transaction
- **db**/2 : contenu de la matrice



Fouille d'itemsets avec ASP

Programme ASP proposé par Järvisalo [Jär11]

```
item(I) :- db(_,I). % ensemble des items de la base de transactions
transaction(T) :- db(T,_). % ensemble des transactions de la base

{ in_itemset(I) } :- item(I). % generation d'un motif par AS
in_support(T) :- { conflict_at(T,I) : item(I) } 0, transaction(T).
conflict_at(T,I) :- not db(T,I), in_itemset(I), transaction(T).
:- { in_support(T) } N-1, threshold(N).
```

- il propose comme principe : 1 AS = 1 motif
- il reprend la modélisation basée sur l'utilisation de conflit de [GNR11] : `conflict_at/2` indique qu'un motif n'est pas supporté par une transaction `T`

Fouille d'itemsets avec ASP

Programme ASP proposé par Järvisalo [Jär11]

```
item(I) :- db(_,I). % ensemble des items de la base de transactions
transaction(T) :- db(T,_). % ensemble des transactions de la base

{ in_itemset(I) } :- item(I). % generation d'un motif par AS
in_support(T) :- { conflict_at(T,I) : item(I) } 0, transaction(T).
conflict_at(T,I) :- not db(T,I), in_itemset(I), transaction(T).
:- { in_support(T) } N-1, threshold(N).
```

Quid de la propriété d'anti-monotonie ?

- Le solveur "apprend" la relation d'anti-monotonie
- Expérimentations d'alternatives peu concluantes (approche incrémentale, modification des heuristiques de recherche)

$\Rightarrow$ déterminer une bonne heuristique de parcours de l'espace de recherche des motifs

ASP : un outil pour mixer fouille de données et intelligence artificielle

- le programme précédent résoud la tâche de fouille d'itemsets (relativement efficacement)
- il est très aisé d'ajouter des connaissances sur les motifs souhaités
  - contraintes sur les items : présence d'un item, obliger/interdire deux items à être ensemble,
  - contraintes sur l'ensemble de motif : difficile avec cette solution.
- connaissances modélisant des informations complémentaire sur les items (*e.g.* le prix de chaque item)

  $\Rightarrow$ permet de définir des contraintes sémantiquement très riches
- il est possible de manipuler l'heuristique de résolution

  $\Rightarrow$ sortir les motifs "les plus intéressants" en premier

```
cost(1,12).cost(2,23).cost(3,56). ...
:- L=#sum{C : cost(I,C), in_itemset(I)}, L>100.
```

Motifs séquentiels

Extraction de motifs séquentiels

La fouille de séquences est plus intéressantes algorithmiquement parlant et facilite l'expression d'information riches !

- défis de modélisations ASP ouvrant vers la résolution de tâches difficiles de fouille
- démonstratif de l'approche déclarative

Encodage de la base de séquences

- seq(T,D,I) : transaction T, at date D, the item I

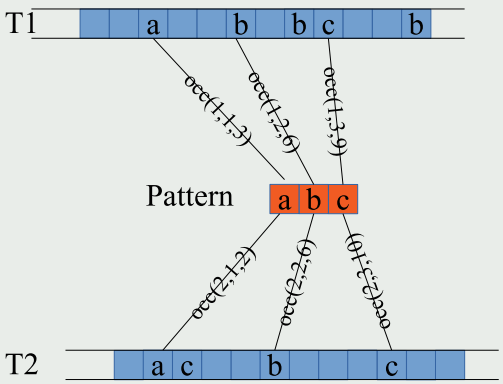
```
seq(0,1,20). seq(0,2,19). seq(0,3,2). seq(0,4,18).
seq(1,1,5). seq(1,2,6). seq(1,3,6). seq(1,4,14).
seq(2,1,13). seq(2,2,12). seq(2,3,9). seq(2,4,19).
seq(3,1,15). seq(3,2,7). seq(3,3,17). seq(3,4,17).
seq(4,1,11). seq(4,2,2). seq(4,3,2). seq(4,4,13).
seq(5,1,14). seq(5,2,8). seq(5,3,13). seq(5,4,1).
```

Motifs séquentiels I

Version d'un encodage inspirée de l'approche Järvisalo.

- **patlen**/1 donne la longueur des motifs (fixé ici)
- **pat**/2 décrit le motif (1 motif/AS)
- **occ**/3 décrit les occurrences du motif
- **covered_transaction**/1 décrit les transactions couvertes par le motif

Exemple : description des occurrences d'une transaction




```
pat(a). pat(b). pat(c).
occ(1,1,3). occ(1,2,6), occ(1,3,9).
occ(2,1,2). occ(2,2,6), occ(2,3,10).
```

Motifs séquentiels II

```
1  transaction(T) :- seq(T,_,_).
2  symb(S) :- seq(_,_,S).
3  1 { patlen(1..maxlen) } 1.
4  pat(1..L, C): symb(C) :- patlen(L).
5
6  th { covered_transaction(T) : transaction(T) } th.
7
8  % I / T is the instance id, sequence id
9  % L est la longueur du pattern, P position in the sequence,
10 % C: the item
11 occ(T, 1..L, P): seq(T,P,C) :- patlen(L), covered_transaction(T).
12 :- occ(I,N,P), occ(I,N,Q), P<Q.
13 :- occ(I,N,P), pat(N,C), not seq(I,P,C) .
14
15 % respect de l'ordre dans les patterns :
16 :- occ(I, N, P), occ(I, N2, P2), N<N2, P>=P2.
```

Expérimentation en cours I

Cas d'application : fouille de parcours de soins

- Base de données de l'assurance maladie
  - information détaillée sur les remboursements de l'ensemble des assurés sociaux
    - médicaments (date de délivrance, quantité, etc.)
    - passage à l'hôpital (identification de pathologies)
  - les remboursements sont codifiés (taxonomie des médicaments)
- Données de l'étude GenEpi : étude des parcours de soin de patient épileptiques avant une hospitalisation pour crise convulsive
- Objectif : identification de parcours "expliquant" des crises

Données

- Nombres de codes CIPs : 1208
- Nombre de transaction : 100
- nb items/transaction : 243.16
- nb items différents/transaction : 48.24

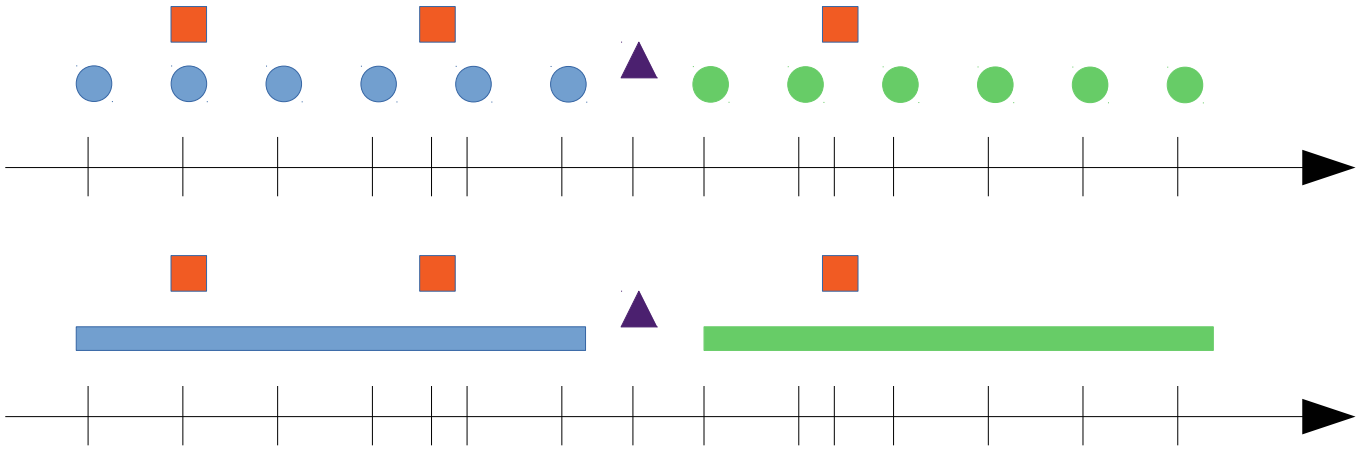
IRISA

Expérimentation en cours II

Les défis

- modélisation d'une taxonomie riche
- inclusion d'étapes de pré-traitements : regroupement des délivrances régulières de médicaments

⇒ approches encore en cours de développement



Outline

Espaces de recherche en extraction de motifs

- le problème d'extraction de motifs est-il semblable à un problème de planification ?

- l'intérêt de répondre positivement à cette question serait de bénéficier des travaux en planification pour appliquer des heuristiques (complètes ou non) à l'extraction de motifs
  - $\Rightarrow$ planification sous contraintes = motifs contraints ?

Espaces de recherche en extraction de motifs

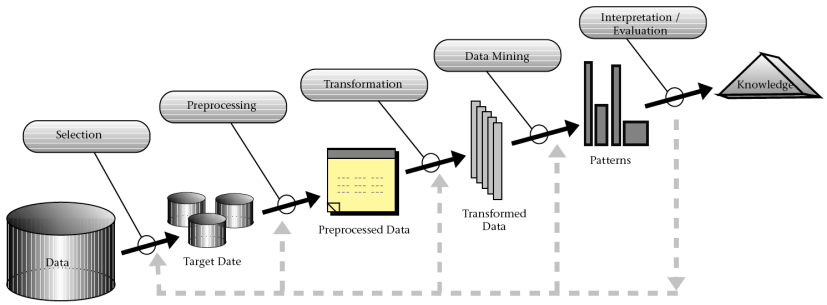
- le problème d'extraction de motifs est-il semblable à un problème de planification ?
  - Un motif est-il un "plan" ? (séquence $\rightarrow$ planification séquentielle ?)
  - Un motif est-il une "allocation de ressource" ?
  - Les contraintes sur les "plans" sont liées à la base de données !

  $\Rightarrow$ Comment considérer la contrainte forte de limitation d'un accès aux données
- l'intérêt de répondre positivement à cette question serait de bénéficier des travaux en planification pour appliquer des heuristiques (complètes ou non) à l'extraction de motifs

  $\Rightarrow$ planification sous contraintes = motifs contraints ?

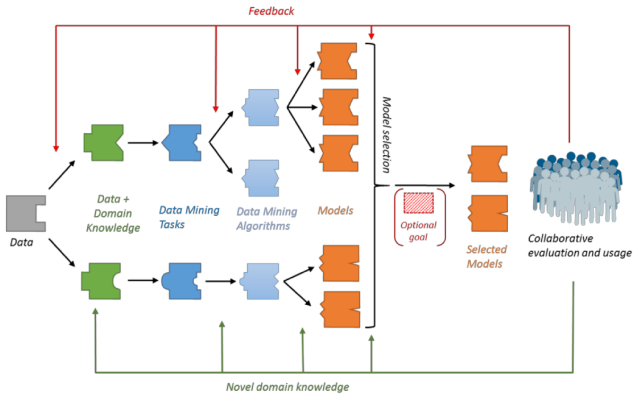
Composer des chaînes d'extraction de connaissances

- les outils de patterns mining ne sont qu'une étape du processus de fouille de données
- beaucoup d'autres méthodes doivent être mise en oeuvre pour extraire de l'information des données brutes
- le schéma de Fayyad est très linéaire et ne montre pas la difficulté de l'utilisateur à concevoir et interagir avec cette chaîne de traitements



Composer des chaînes d'extraction de connaissances

- les outils de patterns mining ne sont qu'une étape du processus de fouille de données
- beaucoup d'autres méthodes doivent être mise en oeuvre pour extraire de l'information des données brutes
- le schéma de Fayyad est très linéaire et ne montre pas la difficulté de l'utilisateur à concevoir et interagir avec cette chaîne de traitements
- l'ensemble des outils d'analyse possibles et leurs paramétrisation fait de la question de la conception d'une chaîne de traitement un problème hautement combinatoire



Composer des chaînes d'extraction de connaissances

- les outils de patterns mining ne sont qu'une étape du processus de fouille de données
- beaucoup d'autres méthodes doivent être mise en oeuvre pour extraire de l'information des données brutes
- le schéma de Fayyad est très linéaire et ne montre pas la difficulté de l'utilisateur à concevoir et interagir avec cette chaîne de traitements
- l'ensemble des outils d'analyse possibles et leurs paramétrisation fait de la question de la conception d'une chaîne de traitement un problème hautement combinatoire
- l'utilisation de méthodes de planification permettrait d'automatiser en partie cette étape

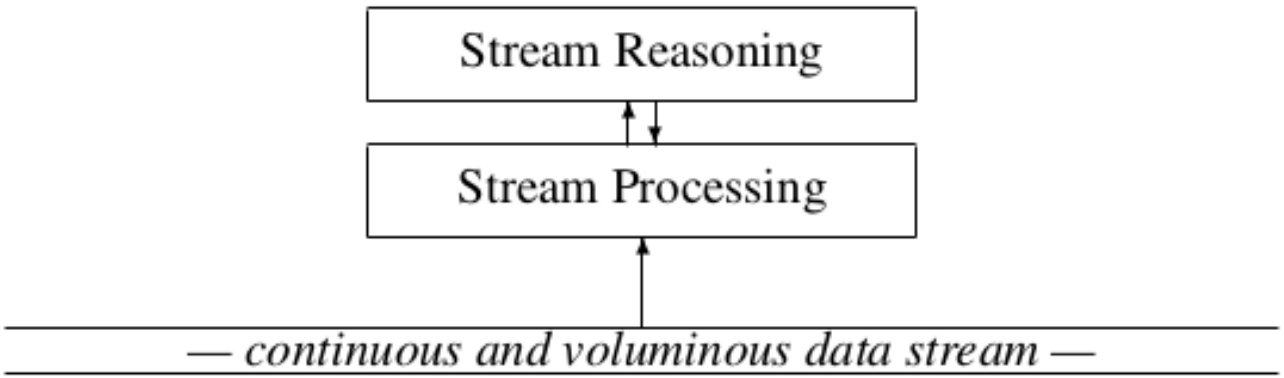
  - Automatisation de la construction d'un plan d'analyse de données guidée par un "but d'analyse"
  - Suggestions de paramétrisation
  - Apprentissage à partir de plans
  - Généralisation de plans d'analyse de données

Mieux choisir les motifs I

Constat

- les méthodes d'extraction de motifs génèrent un très grand nombre de motifs
- besoin d'affiner l'ensemble des motifs présentés à l'utilisateur
  - a priori : trouver des méthodes capables d'intégrer des connaissances utilisables pendant la fouille
  - a posteriori : filtrage de l'ensemble des motifs intéressants (toujours en utilisant des connaissances du domaine)
    - ⇒ bénéficier de l'efficacité des outils de fouille
    - ⇒ utiliser de la connaissance du domaine pour filtrer les motifs

⇒ mise en œuvre de raisonnements sur les motifs



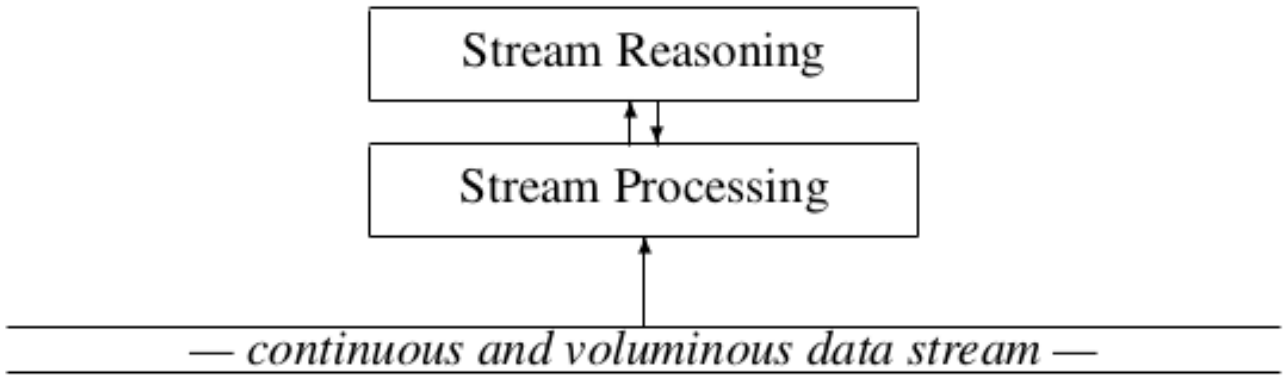
Mieux choisir les motifs II

Prendre des décisions sur les motifs

- Sachant un ensemble de motifs et des connaissances expertes,
  - quelles sont les motifs ou l'ensemble de motifs les plus "intéressants"
    exemple de critères : actionnabilité des actions, ...
  - comment les structurer (préférences)

⇒ comment comparer des motifs ?

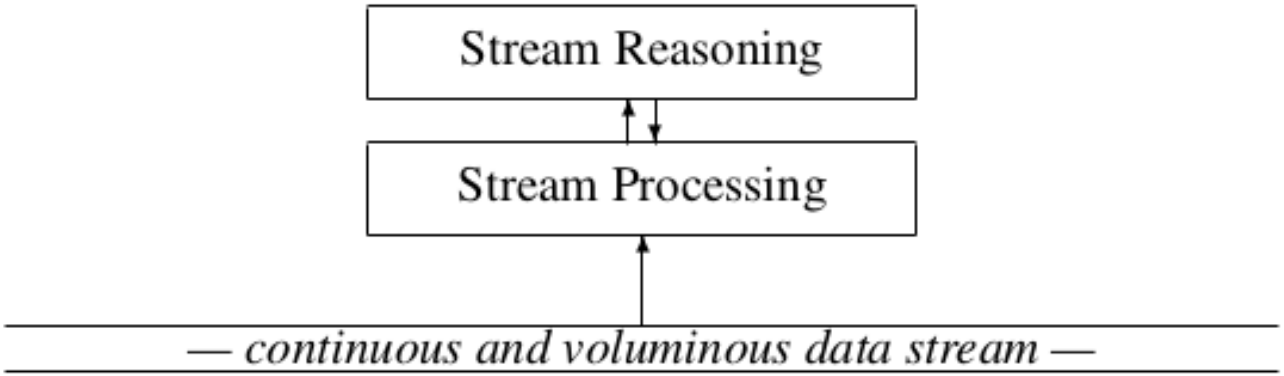
⇒ vers des réflexions sur l'utilisation de méthodes d'argumentation : pourquoi un motif est-il plus intéressant ?



Mieux choisir les motifs III

Prendre des décisions avec des motifs

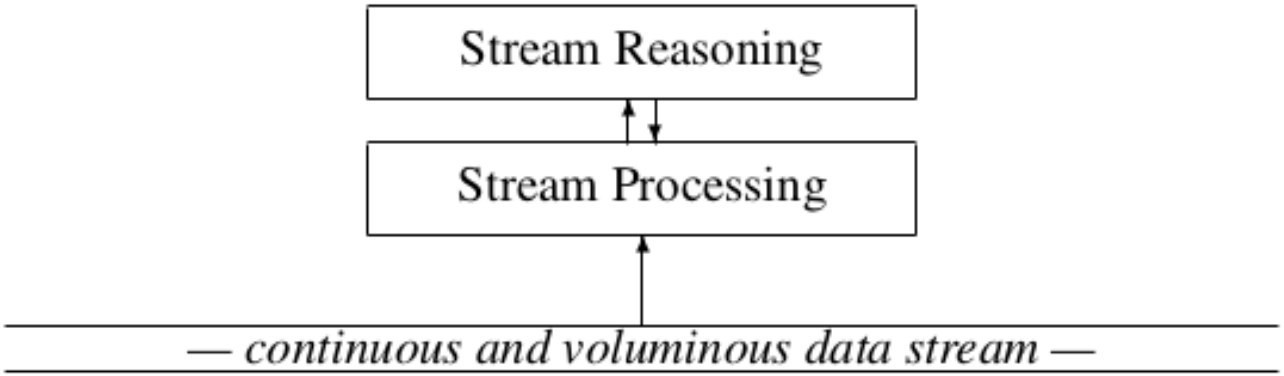
- comment un ensemble de motifs peut être utilisé pour prendre des décisions en ligne ?
- problème : quelle est la pertinence des motifs fréquents pour prendre des décision ?



Mieux choisir les motifs IV

Prise de décision d'ordre 2

Décider de quand et sur quoi renouveller l'ensemble de motifs (en cas *concept drift*)



Conclusion

⇒ L'extraction de motifs dans des bases de données ouvrent des questions intéressantes pour des méthodes d'intelligence artificielle

⇒ Nos activités nous conduisent à traiter de plus en plus de données ... un des défis majeurs sera faire passer les méthodes d'IA à l'échelle de ces données !

Questions ?

References I

Tias Guns, Siegfried Nijssen, and Luc De Raedt, *Itemset mining : A constraint programming perspective*, Artificial Intelligence 175 (2011), no. 12-13, 1951–1983.

Matti Järvisalo, *Itemset mining as a challenge application for answer set enumeration*, Logic Programming and Nonmonotonic Reasoning, Springer, 2011, pp. 304–310.

Simon Malinowski, Thomas Guyet, and René Quiniou, Proceedings of the Intelligence Data Analysis, 2013.