Complexité moyenne de la recherche de motifs fréquents et de traverses minimales d'hypergraphe

Loïck Lhote

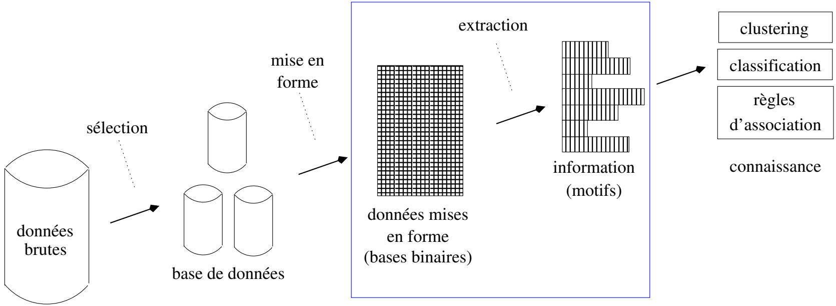
GREYC, ENSICAEN

13 mars 2012

Plan

Plan

Etapes successives en fouille de données



- cas particulier : bases de données binaires

Zoologie

Le domaine regorge de types de motifs

- motifs fréquents
- motifs fermés
- motifs libres
- motifs émergents
- motifs satisfaisant des mesures d'intérêt
- motifs de la bordure négative/positive
- ...

Le domaine regorge de types de bases

- bases binaires
- bases d'objets arborescents (ex : fichiers xml/pages web)
- bases de graphes (ex : molécules chimiques)
- bases de séquences d'éléments complexes (ex : on tient compte du temps)
- ...

Point de vue matriciel

Base de données

Attributs/items

Objets/Transactions/Itemsets

1	1	0	1	1
0	0	0	1	0
0	1	1	0	0
0	0	1	1	0
1	0	1	1	1

Exemple

- attribut = une réponse possible à une question posée
- objet = ensemble des réponses d'une personne

Plan

Motifs fréquents

Dimensions : $n = \text{nb. colonnes} \qquad m = \text{nb. lignes}$
Base de données : $\mathcal{B} = (b_{i,j})_{i=1\ldots m, j=1\ldots n}$
Seuil de fréquence : un entier γ

Définition : motifs fréquents

L'ensemble $X \subset \{1, \ldots, n\}$ est un *motif γ-fréquent* dans la base $\mathcal{B}$ ssi

$$\mathrm{card}\{i \in \{1, \ldots, m\} \ : \ \forall j \in X, \quad b_{i,j} = 1\} \geq \gamma.$$

	1	2	3	4	5
1	1	1	0	1	1
2	0	0	0	1	0
3	0	1	1	0	0
4	0	0	1	1	0
5	1	0	1	1	1

Motifs fréquents

Dimensions : $n = \text{nb. colonnes}$ $m = \text{nb. lignes}$
Base de données : $\mathcal{B} = (b_{i,j})_{i=1\ldots m, j=1\ldots n}$
Seuil de fréquence : un entier γ

Définition : motifs fréquents

L'ensemble $X \subset \{1, \ldots, n\}$ est un *motif γ-fréquent* dans la base $\mathcal{B}$ ssi

$$\mathrm{card}\{i \in \{1, \ldots, m\} \;:\; \forall j \in X, \quad b_{i,j} = 1\} \geq \gamma.$$

	1	2	3	4	5
1	1	1	0	1	1
2	0	0	0	1	0
3	0	1	1	0	0
4	0	0	1	1	0
5	1	0	1	1	1

Motifs fréquents

Dimensions : $n = \text{nb. colonnes}$ $m = \text{nb. lignes}$

Base de données : $\mathcal{B} = (b_{i,j})_{i=1\dots m, j=1\dots n}$

Seuil de fréquence : un entier γ

Définition : motifs fréquents

L'ensemble $X \subset \{1,\dots,n\}$ est un *motif γ-fréquent* dans la base $\mathcal{B}$ ssi

$$\text{card}\{i \in \{1,\dots,m\} \ : \ \forall j \in X, \quad b_{i,j} = 1\} \geq \gamma.$$

	1	2	3	4	5
1	1	1	0	1	1
2	0	0	0	1	0
3	0	1	1	0	0
4	0	0	1	1	0
5	1	0	1	1	1

- $\{3,4\}$ est 0,1,2-fréquent mais pas 3-fréquent

Motifs fréquents

Dimensions : $n = \text{nb. colonnes}$ $m = \text{nb. lignes}$

Base de données : $\mathcal{B} = (b_{i,j})_{i=1\ldots m, j=1\ldots n}$

Seuil de fréquence : un entier γ

Définition : motifs fréquents

L'ensemble $X \subset \{1, \ldots, n\}$ est un *motif γ-fréquent* dans la base $\mathcal{B}$ ssi

$$\text{card}\{i \in \{1, \ldots, m\} \ : \ \forall j \in X, \quad b_{i,j} = 1\} \geq \gamma.$$

	1	2	3	4	5
1	1	1	0	1	1
2	0	0	0	1	0
3	0	1	1	0	0
4	0	0	1	1	0
5	1	0	1	1	1

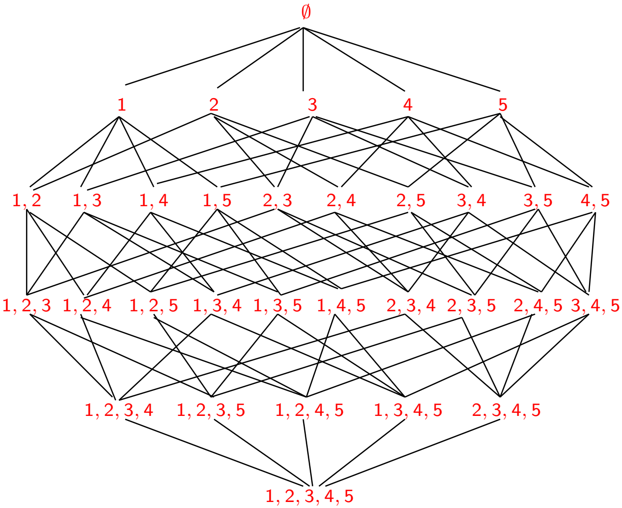
- $\{3, 4\}$ est 0,1,2-fréquent mais pas 3-fréquent

Support d'un motif : $\text{Supp}(X) = \{i \in \{1, \ldots, m\} \ : \ \forall j \in X, \quad b_{i,j} = 1\}$

$$\text{Supp}(\{3, 4\}) = \{4, 5\}$$

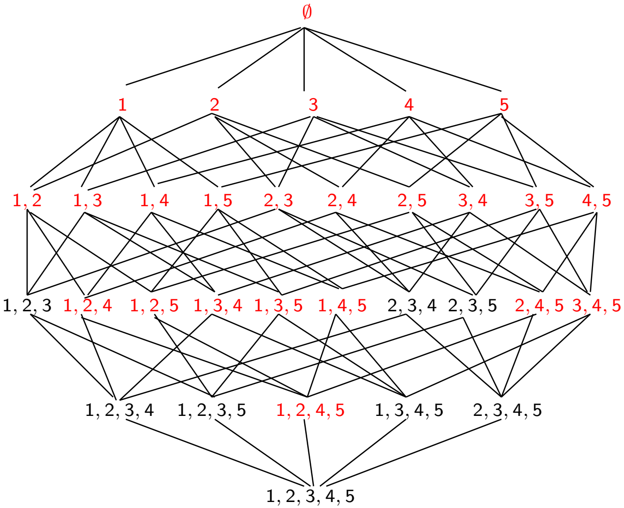
Problème

Motifs 0-fréquents



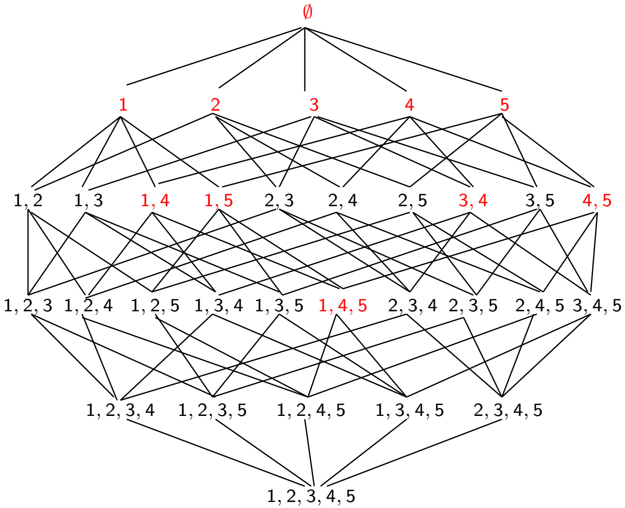
Problème

Motifs 1-fréquents



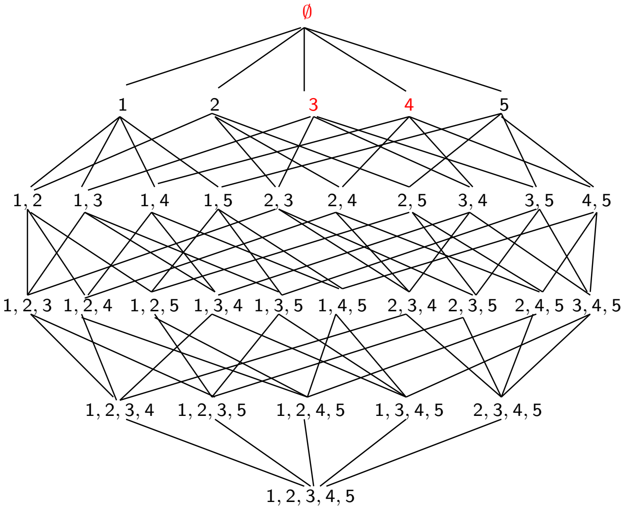
Problème

Motifs 2-fréquents



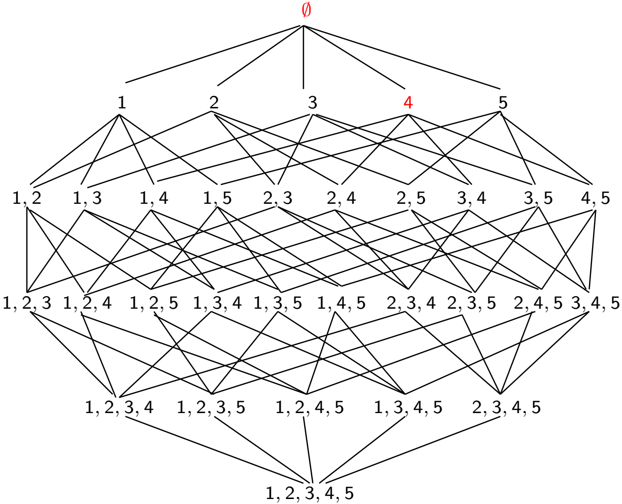
Problème

Motifs 3-fréquents



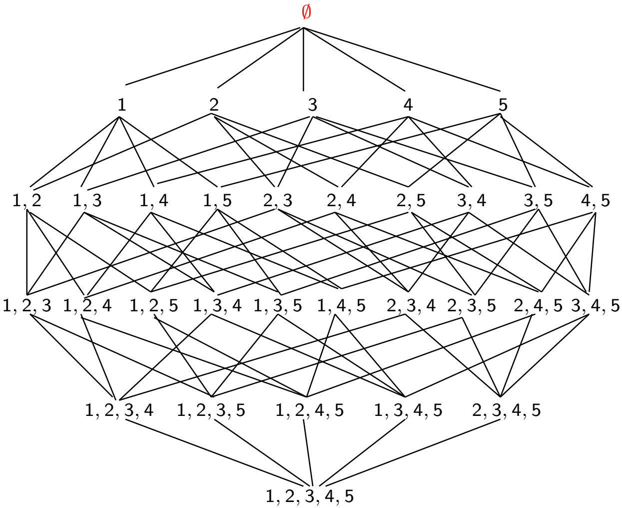
Problème

Motifs 4-fréquents



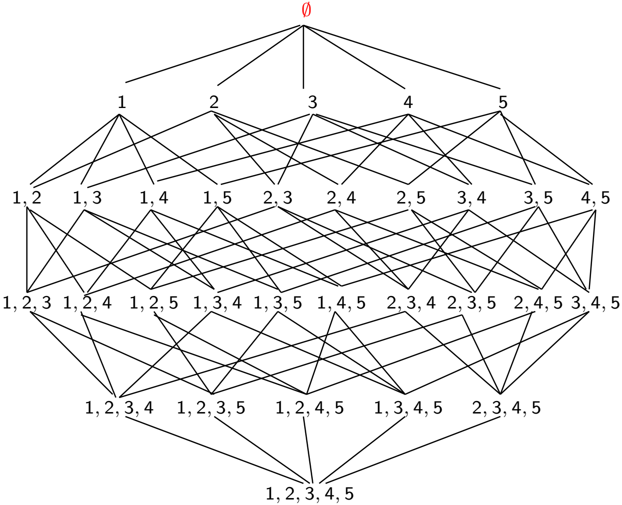
Problème

Motifs 5-fréquents



Problème

Motifs 5-fréquents



Problème : estimer le nombre de motifs γ-fréquents en fonction de γ ?

Pire et meilleur des cas

$$\begin{pmatrix} & & \\ & 0 & \\ & & \end{pmatrix} \qquad \begin{pmatrix} & & \\ & 1 & \\ & & \end{pmatrix}$$

$$O(1) \qquad\qquad O(2^n)$$

Pire des cas : exponentiel
Meilleur des cas : constant

Pire et meilleur des cas

$$\begin{pmatrix} & & \\ & 0 & \\ & & \end{pmatrix} \qquad \begin{pmatrix} & & \\ & 1 & \\ & & \end{pmatrix}$$

$$O(1) \qquad\qquad O(2^n)$$

Pire des cas : exponentiel
Meilleur des cas : constant

Étonnamment :

- ce sont les seuls ordres de grandeur connus (à l'époque !),
- les algorithmes sont efficaces pour γ *raisonnable*.

Pire et meilleur des cas

$$\begin{pmatrix} & & \\ & 0 & \\ & & \end{pmatrix} \qquad \begin{pmatrix} & & \\ & 1 & \\ & & \end{pmatrix}$$

$$O(1) \qquad\qquad O(2^n)$$

Pire des cas : exponentiel
Meilleur des cas : constant

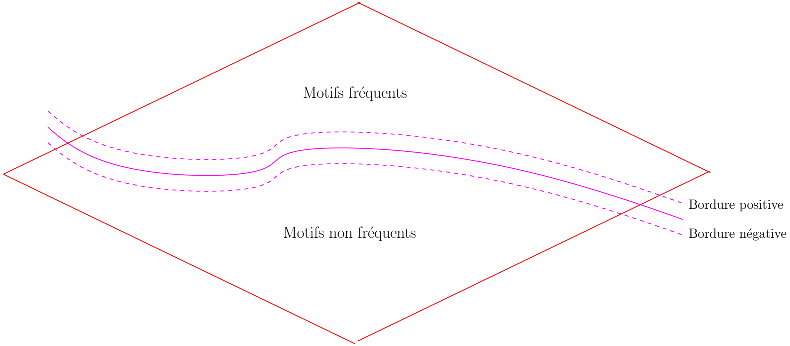
Étonnamment :

- ce sont les seuls ordres de grandeur connus (à l'époque !),
- les algorithmes sont efficaces pour γ *raisonnable*.

$\Rightarrow$ l'analyse en moyenne peut éclaircir ce phénomène

Plan

Bordure négative

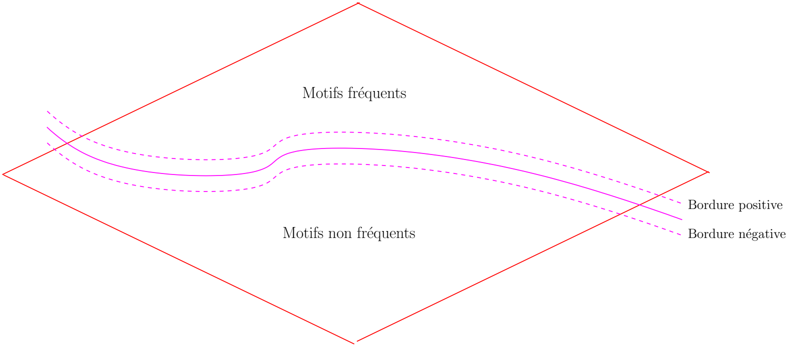


Intérêts des bordures :

- représentation condensée des motifs fréquents
- Complexité algorithmes par niveaux

Nb de motifs fréquents + taille de la bordure négative

Bordure négative



Intérêts des bordures :

- représentation condensée des motifs fréquents
- Complexité algorithmes par niveaux

Nb de motifs fréquents + taille de la bordure négative

Lien avec les hypergraphes

Calculer la bordure négative

=

Calculer les traverses minimales d'un hypergraphe

Hypergraphe

Représentation matricielle

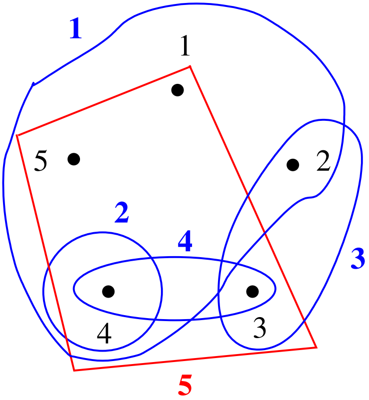
	1	2	3	4	5
1	1	1	0	1	1
2	0	0	0	1	0
3	0	1	1	0	0
4	0	0	1	1	0
5	1	0	1	1	1

Hypergraphe

Représentation matricielle

	1	2	3	4	5
1	1	1	0	1	1
2	0	0	0	1	0
3	0	1	1	0	0
4	0	0	1	1	0
5	1	0	1	1	1

Représentation graphique

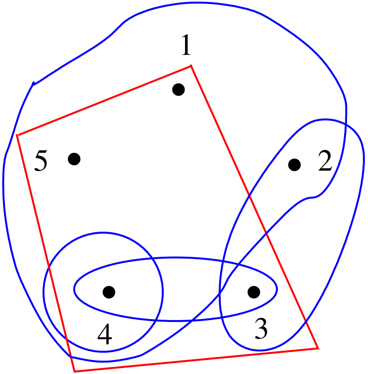


Traverses et traverses minimales

Définition : Une traverse est un ensemble de sommets qui intersecte toutes les hyperarêtes.

Définition : Une traverse est dite minimale si aucun de ses sous-ensembles stricts n'est une traverse.

Représentation graphique

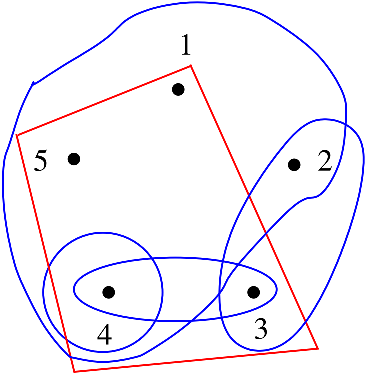


Traverses et traverses minimales

Définition : Une traverse est un ensemble de sommets qui intersecte toutes les hyperarêtes.

Définition : Une traverse est dite minimale si aucun de ses sous-ensembles stricts n'est une traverse.

Représentation graphique



Les traverses (minimales) sont :

$$\{2,4\},\ \{3,4\},$$

$$\{1,2,4\},\ \{1,3,4\},\ \{2,3,4\},\ \{2,4,5\},\ \{3,4,5\},$$

$$\{1,2,3,4\},\ \{1,2,4,5\},\ \{1,3,4,5\},\ \{2,3,4,5\},$$

$$\{1,2,3,4,5\}$$

Bordure négative et traverses minimales

Base

1	2	3	4	5	6	7
1	1	1	0	0	0	0
0	1	1	1	1	0	0
0	0	0	0	1	1	1

$--\!>$

Base opposée

1	2	3	4	5	6	7
0	0	0	1	1	1	1
1	0	0	0	0	1	1
1	1	1	1	0	0	0

$=$

Hypergraphe

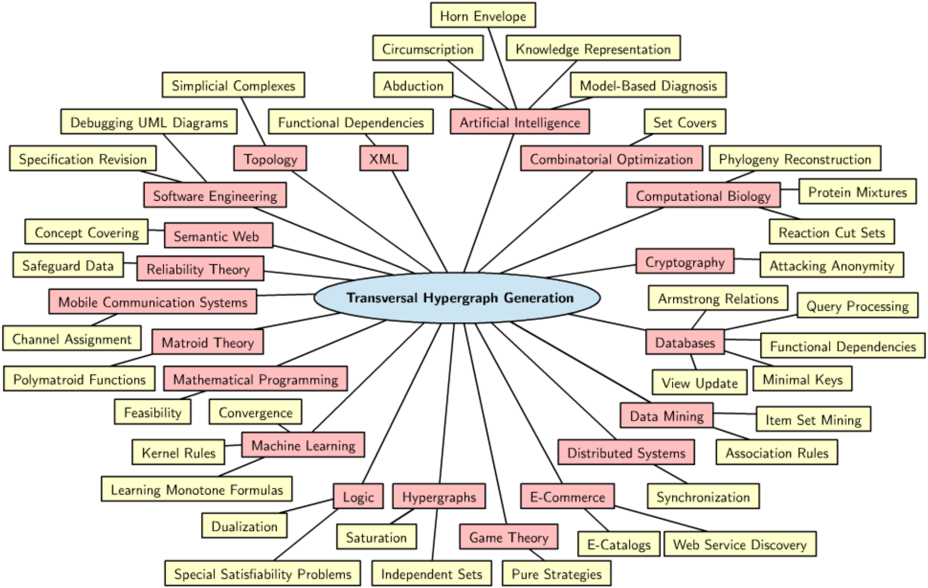
1	2	3	4	5	6	7
0	0	0	1	1	1	1
1	0	0	0	0	1	1
1	1	1	1	0	0	0

$<\!--$

Les traverses minimales de l'hypergraphe
sont les motifs de
la bordure négative

Exemple :
$\{1,4\}$ est une traverse minimale $\Rightarrow$ $\{1,4\}$ est un élément de la bordure négative ($\gamma = 1$).

Autres applications (Mathias Hagen)



Problèmes

Problème THG (Transversal Hypergraph Generation)

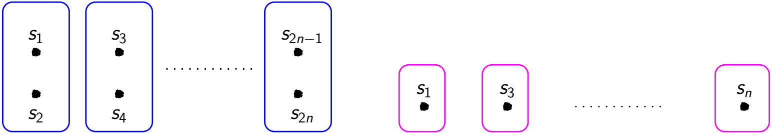
Étant donné un hypergraphe $\mathcal{H}$, calculer l'ensemble des traverses minimales $\mathcal{T}(\mathcal{H})$ de $\mathcal{H}$

Remarque : l'ensemble des traverses minimales forme un hypergraphe appelé hypergraphe transversal.

Problème THD (Transversal Hypergraph Decision)

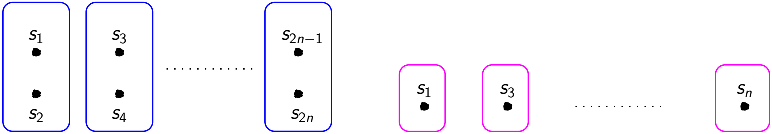
Étant donnés deux hypergraphes $\mathcal{H}_1$ et $\mathcal{H}_2$, a-t-on $\mathcal{T}(\mathcal{H}_1) = \mathcal{H}_2$?

Pire et meilleur des cas



$|\mathcal{T}(\mathcal{H})| = 2^n$ $\qquad$ $|\mathcal{T}(\mathcal{H})| = 1$

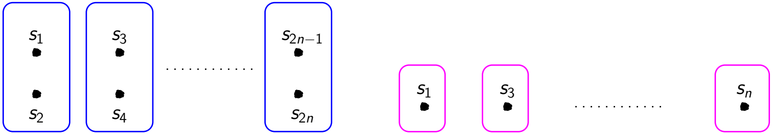
Pire et meilleur des cas



$|\mathcal{T}(\mathcal{H})| = 2^n$ $\qquad$ $|\mathcal{T}(\mathcal{H})| = 1$

Une bonne notion de complexité pour ce problème fait intervenir la taille de la sortie.

Pire et meilleur des cas



$|\mathcal{T}(\mathcal{H})| = 2^n$ $\qquad$ $|\mathcal{T}(\mathcal{H})| = 1$

Une bonne notion de complexité pour ce problème fait intervenir la taille de la sortie.

Problème

Le problème THG est-il output–polynomial ?

Autrement, existe-t-il un algorithme en temps polynômial en la taille de l'entrée plus celle de la sortie ?

Résultats de complexité

- Calculer une traverse minimale est dans **P**
- Calculer une traverse minimale de plus petit cardinal est **NP-Hard** (Minimum Vertex Cover est déjà **NP-Hard** pour les graphes)
- [Eiter, 94] Trouver la première (dernière) traverse minimale selon l'ordre lexicographique est **NP-Hard**
- Conséquence : un algorithme "polynomial delay" n'existe pas pour le problème Lexicographic-THG sauf si **P**=**NP**
- [Eiter-Gotlob,02] Il existe un algorithme output–polynomial pour THG ssi THD est dans **P**.
  [Bioch-Ibaraki,93] Il existe un algorithme "incrementally polynomial" pour THG ssi THD est dans **P**.
- THD est dans **co-NP** ("clearly").
  Mais on ne sait pas si THD est **co-NP**-complet.
- si THD est **co-NP**-complet, alors THG n'est probablement pas output–polynomial sinon **P**=**NP**.

Résultats de complexité

- Calculer une traverse minimale est dans **P**
- Calculer une traverse minimale de plus petit cardinal est **NP-Hard** (Minimum Vertex Cover est déjà **NP-Hard** pour les graphes)
- [Eiter, 94] Trouver la première (dernière) traverse minimale selon l'ordre lexicographique est **NP-Hard**
- Conséquence : un algorithme "polynomial delay" n'existe pas pour le problème Lexicographic-THG sauf si **P**=**NP**
- [Eiter-Gotlob,02] Il existe un algorithme output–polynomial pour THG ssi THD est dans **P**.
  [Bioch-Ibaraki,93] Il existe un algorithme "incrementally polynomial" pour THG ssi THD est dans **P**.
- THD est dans **co-NP** ("clearly").
  Mais on ne sait pas si THD est **co-NP**-complet.
- si THD est **co-NP**-complet, alors THG n'est probablement pas output–polynomial sinon **P**=**NP**.

Résultats de complexité

- Calculer une traverse minimale est dans **P**
- Calculer une traverse minimale de plus petit cardinal est **NP-Hard** (Minimum Vertex Cover est déjà **NP-Hard** pour les graphes)
- [Eiter, 94] Trouver la première (dernière) traverse minimale selon l'ordre lexicographique est **NP-Hard**
- Conséquence : un algorithme "polynomial delay" n'existe pas pour le problème Lexicographic-THG sauf si **P**=**NP**
- [Eiter-Gotlob,02] Il existe un algorithme output–polynomial pour THG ssi THD est dans **P**.
  [Bioch-Ibaraki,93] Il existe un algorithme "incrementally polynomial" pour THG ssi THD est dans **P**.
- THD est dans **co-NP** ("clearly").
  Mais on ne sait pas si THD est **co-NP**-complet.
- si THD est **co-NP**-complet, alors THG n'est probablement pas output–polynomial sinon **P**=**NP**.

Résultats de complexité

- Calculer une traverse minimale est dans **P**
- Calculer une traverse minimale de plus petit cardinal est **NP-Hard** (Minimum Vertex Cover est déjà **NP-Hard** pour les graphes)
- [Eiter, 94] Trouver la première (dernière) traverse minimale selon l'ordre lexicographique est **NP-Hard**
- Conséquence : un algorithme "polynomial delay" n'existe pas pour le problème Lexicographic-THG sauf si **P**=**NP**
- [Eiter-Gotlob,02] Il existe un algorithme output–polynomial pour THG ssi THD est dans **P**.
  [Bioch-Ibaraki,93] Il existe un algorithme "incrementally polynomial" pour THG ssi THD est dans **P**.
- THD est dans **co-NP** ("clearly").
  Mais on ne sait pas si THD est **co-NP**-complet.
- si THD est **co-NP**-complet, alors THG n'est probablement pas output–polynomial sinon **P**=**NP**.

Résultats de complexité

- Calculer une traverse minimale est dans **P**
- Calculer une traverse minimale de plus petit cardinal est **NP-Hard** (Minimum Vertex Cover est déjà **NP-Hard** pour les graphes)
- [Eiter, 94] Trouver la première (dernière) traverse minimale selon l'ordre lexicographique est **NP-Hard**
- Conséquence : un algorithme “polynomial delay” n'existe pas pour le problème Lexicographic-THG sauf si **P**=**NP**
- [Eiter-Gotlob,02] Il existe un algorithme output–polynomial pour THG ssi THD est dans **P**.
  [Bioch-Ibaraki,93] Il existe un algorithme “incrementally polynomial” pour THG ssi THD est dans **P**.
- THD est dans **co-NP** (“clearly”).
  Mais on ne sait pas si THD est **co-NP**-complet.
- si THD est **co-NP**-complet, alors THG n'est probablement pas output–polynomial sinon **P**=**NP**.

Résultats de complexité

- Calculer une traverse minimale est dans **P**
- Calculer une traverse minimale de plus petit cardinal est **NP-Hard** (Minimum Vertex Cover est déjà **NP-Hard** pour les graphes)
- [Eiter, 94] Trouver la première (dernière) traverse minimale selon l'ordre lexicographique est **NP-Hard**
- Conséquence : un algorithme "polynomial delay" n'existe pas pour le problème Lexicographic-THG sauf si **P**=**NP**
- [Eiter-Gotlob,02] Il existe un algorithme output–polynomial pour THG ssi THD est dans **P**.
  [Bioch-Ibaraki,93] Il existe un algorithme "incrementally polynomial" pour THG ssi THD est dans **P**.
- THD est dans **co-NP** ("clearly").
  Mais on ne sait pas si THD est **co-NP**-complet.
- si THD est **co-NP**-complet, alors THG n'est probablement pas output–polynomial sinon **P**=**NP**.

Résultats de complexité

- Calculer une traverse minimale est dans **P**
- Calculer une traverse minimale de plus petit cardinal est **NP-Hard** (Minimum Vertex Cover est déjà **NP-Hard** pour les graphes)
- [Eiter, 94] Trouver la première (dernière) traverse minimale selon l'ordre lexicographique est **NP-Hard**
- Conséquence : un algorithme "polynomial delay" n'existe pas pour le problème Lexicographic-THG sauf si **P**=**NP**
- [Eiter-Gotlob,02] Il existe un algorithme output–polynomial pour THG ssi THD est dans **P**.
  [Bioch-Ibaraki,93] Il existe un algorithme "incrementally polynomial" pour THG ssi THD est dans **P**.
- THD est dans **co-NP** ("clearly").
  Mais on ne sait pas si THD est **co-NP**-complet.
- si THD est **co-NP**-complet, alors THG n'est probablement pas output–polynomial sinon **P**=**NP**.

Résultats de complexité en moyenne

- Les auteurs de [1] considèrent la distribution uniforme sur les hypergraphes (simples) à n sommets
- pour presque tous les hypergraphes à n sommets [2] :
  - les traverses minimales ont une taille comprise entre $\frac{n}{2} - 2$ et $\frac{n}{2} + 2$
  - tout ensemble de sommets de taille $> \frac{n}{2} + 2$ est une traverse
- Il existe un algorithme polynomial pour THG dans cette situation [3]

Références :

[1] Shmulevich Ilya, Korshunov Aleksey D., Astola Jaakko
Almost all monotone Boolean functions are polynomially learnable using membership queries,
Inf. Process. Lett., Elsevier North-Holland, Inc., 79 (5),pp211–213, 2001

Résultats de complexité en moyenne

- Les auteurs de [1] considèrent la distribution uniforme sur les hypergraphes (simples) à n sommets
- pour presque tous les hypergraphes à n sommets [2] :
  - les traverses minimales ont une taille comprise entre $\frac{n}{2} - 2$ et $\frac{n}{2} + 2$
  - tout ensemble de sommets de taille $> \frac{n}{2} + 2$ est une traverse
- Il existe un algorithme polynomial pour THG dans cette situation [3]

Références :

[1] Shmulevich Ilya, Korshunov Aleksey D., Astola Jaakko
Almost all monotone Boolean functions are polynomially learnable using membership queries,
Inf. Process. Lett., Elsevier North-Holland, Inc., 79 (5),pp211–213, 2001

[2] A.D. Korshunov,
On the number of monotone Boolean functions,
Problemy Kibernetiki, 38, pp5–108,1981

Résultats de complexité en moyenne

- Les auteurs de [1] considèrent la distribution uniforme sur les hypergraphes (simples) à n sommets
- pour presque tous les hypergraphes à n sommets [2] :
  - les traverses minimales ont une taille comprise entre $\frac{n}{2} - 2$ et $\frac{n}{2} + 2$
  - tout ensemble de sommets de taille $> \frac{n}{2} + 2$ est une traverse
- Il existe un algorithme polynomial pour THG dans cette situation [3]

Références :

[1] Shmulevich Ilya, Korshunov Aleksey D., Astola Jaakko
Almost all monotone Boolean functions are polynomially learnable using membership queries,
Inf. Process. Lett., Elsevier North-Holland, Inc., 79 (5),pp211–213, 2001

[2] A.D. Korshunov,
On the number of monotone Boolean functions,
Problemy Kibernetiki, 38, pp5–108,1981

[3] K. Makino, T. Ibaraki,
The maximum latency and identification of positive Boolean functions,
SIAM J. Comput. 26 (5), pp1363-1383, 1997

Résultats de complexité en moyenne

- Les auteurs de [1] considèrent la distribution uniforme sur les hypergraphes (simples) à n sommets
- pour presque tous les hypergraphes à n sommets [2] :
  - les traverses minimales ont une taille comprise entre $\frac{n}{2} - 2$ et $\frac{n}{2} + 2$
  - tout ensemble de sommets de taille $> \frac{n}{2} + 2$ est une traverse
- Il existe un algorithme polynomial pour THG dans cette situation [3]

Références :

[1] Shmulevich Ilya, Korshunov Aleksey D., Astola Jaakko
Almost all monotone Boolean functions are polynomially learnable using membership queries,
Inf. Process. Lett., Elsevier North-Holland, Inc., 79 (5),pp211–213, 2001

[2] A.D. Korshunov,
On the number of monotone Boolean functions,
Problemy Kibernetiki, 38, pp5–108,1981

[3] K. Makino, T. Ibaraki,
The maximum latency and identification of positive Boolean functions,
SIAM J. Comput. 26 (5), pp1363-1383, 1997

[2] $\Rightarrow$ La taille de la sortie est exponentielle en n
$\Rightarrow$ "peu intéressant" d'un point de vue complexité
$\Rightarrow$ il faut préciser le modèle aléatoire

Plan

Modèle aléatoire pour les BDD

	Attributs				
Objets	1	1	0	1	1
	0	0	0	1	0
	0	1	1	0	0
	0	0	1	1	0
	1	0	1	1	1

Chaque ligne est produit par un processus probabiliste $\mathcal{S}$, le processus étant identique pour toutes les lignes.

L'ensemble des lignes forment une famille indépendante de variables aléatoires.

Le nombre de lignes et le nombre de colonnes sont reliés de manière polynomial

$$\log m \sim c \cdot \log n, \quad n = \text{nb. colonnes}, \quad m = \text{nb. lignes}$$

Premier résultat : $\gamma = r \cdot m$

Seuil de fréquence :

- seuil proportionnel : $\gamma = r \cdot m$, $r \in]0, 1[$

Notations :

- X un motif
- $p_X =$ probabilité qu'un mot "contient" le motif X

Premier résultat : $\gamma = r \cdot m$

Seuil de fréquence :

- seuil proportionnel : $\gamma = r \cdot m$, $r \in]0,1[$

Notations :

- X un motif
- p_X= probabilité qu'un mot "contient" le motif X

Condition 1

Il existe $M > 0$ et $\theta \in]0,1[$ tels pour tout motif X,

$$p_X \leq M \cdot \theta^{|X|}.$$

Premier résultat : $\gamma = r \cdot m$

Seuil de fréquence :

- seuil proportionnel : $\gamma = r \cdot m$, $r \in]0,1[$

Notations :

- X un motif
- p_X= probabilité qu'un mot "contient" le motif X

Condition 1

Il existe $M > 0$ et $\theta \in]0,1[$ tels pour tout motif X,

$$p_X \leq M \cdot \theta^{|X|}.$$

Seuil proportionnel [L., Rioult, Soulet]

Pour un seuil de fréquence proportionnel $\gamma = r \cdot n$ avec $r \in]0,1[$ et un processus satisfaisant la condition 1, le nombre de motifs γ-fréquents est **polynomial en le nombre de colonnes (attributs)**.

Deuxième résultat : seuil constant

- seuil constant : γ constant

Deuxième résultat : seuil constant

- seuil constant : γ constant

Notations :

$$S_{\gamma,n} = \sum_{x \subset \{1,\dots,n\}} p_X^{\gamma}.$$

Condition 2

Il existe trois constantes $\lambda_\gamma > 1$, $\kappa_\gamma > 0$ et $\theta_\gamma \in]0,1[$ telles que,

$$S_{\gamma,n} = \kappa_\gamma \cdot \lambda_\gamma^n (1 + O(\theta_\gamma^n)), \qquad \lambda_{\gamma+1} < \lambda_\gamma.$$

Deuxième résultat : seuil constant

- seuil constant : γ constant

Notations :

$$S_{\gamma,n} = \sum_{x \subset \{1,\ldots,n\}} p_X^{\gamma}.$$

Condition 2

Il existe trois constantes $\lambda_\gamma > 1$, $\kappa_\gamma > 0$ et $\theta_\gamma \in]0,1[$ telles que,

$$S_{\gamma,n} = \kappa_\gamma \cdot \lambda_\gamma^n (1 + O(\theta_\gamma^n)), \qquad \lambda_{\gamma+1} < \lambda_\gamma.$$

Seuil constant [L., Rioult, Soulet]

Pour un seuil de fréquence γ constant et un processus satisfaisant la condition 2, le nombre de motifs γ-fréquents est **exponentiel en le nombre de colonnes et polynomial en le nombre de lignes**

Deuxième résultat : seuil constant

- seuil constant : γ constant

Notations :

$$S_{\gamma,n} = \sum_{x \subset \{1,\ldots,n\}} p_X^{\gamma}.$$

Condition 2

Il existe trois constantes $\lambda_\gamma > 1$, $\kappa_\gamma > 0$ et $\theta_\gamma \in]0,1[$ telles que,

$$S_{\gamma,n} = \kappa_\gamma \cdot \lambda_\gamma^n (1 + O(\theta_\gamma^n)), \qquad \lambda_{\gamma+1} < \lambda_\gamma.$$

Seuil constant [L., Rioult, Soulet]

Pour un seuil de fréquence γ constant et un processus satisfaisant la condition 2, le nombre de motifs γ-fréquents est **exponentiel en le nombre de colonnes et polynomial en le nombre de lignes**

Asymptotique :

$$Fr_\gamma \sim \binom{m}{\gamma} S_{\gamma,n} \sim \kappa_\gamma \binom{m}{\gamma} \lambda_\gamma^n.$$

Modèles *sans mémoire*

Modèle de Bernoulli simple :

- la source est sans-mémoire
- émet un 1 avec une probabilité $p \in]0,1[$
- tient compte de la proportion de 1 dans la base

$\Rightarrow$ satisfait les 2 conditions

Modèles *sans mémoire*

Modèle de Bernoulli simple :

- la source est sans-mémoire
- émet un 1 avec une probabilité $p \in]0, 1[$
- tient compte de la proportion de 1 dans la base

$\Rightarrow$ satisfait les 2 conditions

Modèle de Bernoulli par groupe :

- la source est sans-mémoire
- émet des paquets de l'alphabet $\{100\ldots0, 010\ldots0, \ldots\}$ avec une probabilité uniforme (question à choix multiple)
- pour tenir compte des attributs continus qui sont scindés

$\Rightarrow$ satisfait les 2 conditions

Modèles *sans mémoire*

Modèle de Bernoulli simple :

- la source est sans-mémoire
- émet un 1 avec une probabilité $p \in]0,1[$
- tient compte de la proportion de 1 dans la base

$\Rightarrow$ satisfait les 2 conditions

Modèle de Bernoulli par groupe :

- la source est sans-mémoire
- émet des paquets de l'alphabet $\{100\ldots0, 010\ldots0, \ldots\}$ avec une probabilité uniforme (question à choix multiple)
- pour tenir compte des attributs continus qui sont scindés

$\Rightarrow$ satisfait les 2 conditions

Modèle de Bernoulli évolué :

- la source est sans-mémoire
- la i^e lettre est un 1 avec une probabilité p_i
- tient compte de la proportion de 1 dans chaque colonne

$\Rightarrow$ satisfait la première condition et pour un seuil constant :

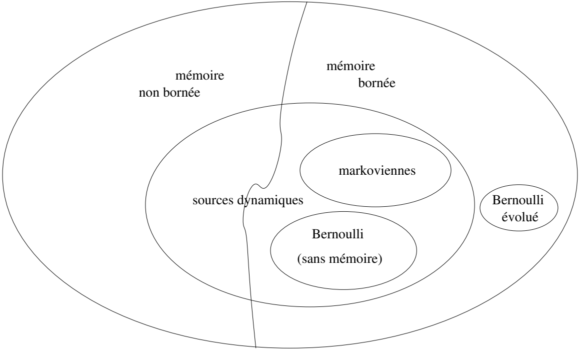
$$Fr_\gamma \sim \binom{m}{\gamma} \prod_{i=1}^{n} (1 + p_i^\gamma).$$

Processus avec corrélations

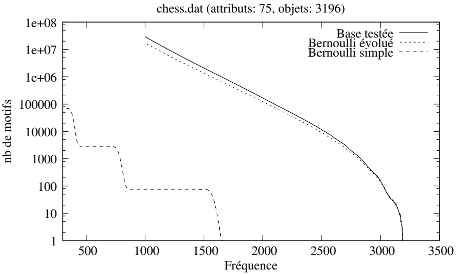
Chaîne de Markov :

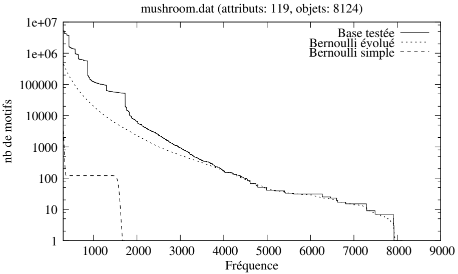
- chaîne finie irréductible et apériodique

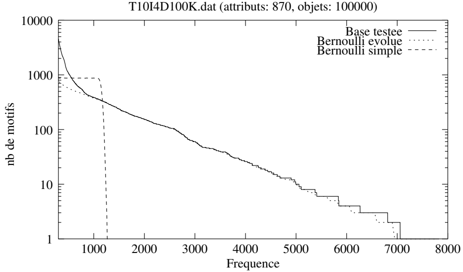
Sources dynamiques :



Expériences : modèles de Bernoulli







Conclusion pour les motifs fréquents

Résultats :

- seuil linéaire : polynomial
- seuil constant : exponentiel

Modèles :

- Bernoulli simple (analyses théoriques)
- Bernoulli évolué (bons résultats en pratique)
- Chaînes de Markov
- Sources dynamiques

Perspectives :

- quid des autres motifs
- taille du plus grand motif fréquent, profil du treillis, etc.
- complexités des algorithmes en profondeur, en largeur
- ...

Plan

Travaux sur les traverses minimales

Work in progress ! ! !

- Modèle aléatoire très limité (futur travail avec Julien)

Modèle aléatoire

Les paramètres du modèle aléatoire sont :

- Le nombre de sommets est noté n (nombre de colonnes).
- Le nombre d'hyperarêtes est noté m (nombre de lignes).
- Un hypergraphe est identifié à sa représentation matricielle.

Contraintes fortes du modèle

- on supposera qu'il existe une constante c telle que $m \sim c \cdot n$.
- On se fixe un réel $p \in]0, 1[$ qui ne dépend pas de m et n. $(q = 1 - p)$

Modèle aléatoire

Les paramètres du modèle aléatoire sont :

- Le nombre de sommets est noté n (nombre de colonnes).
- Le nombre d'hyperarêtes est noté m (nombre de lignes).
- Un hypergraphe est identifié à sa représentation matricielle.

Contraintes fortes du modèle

- on supposera qu'il existe une constante c telle que $m \sim c \cdot n$.
- On se fixe un réel $p \in]0, 1[$ qui ne dépend pas de m et n. $(q = 1 - p)$

Un hypergraphe aléatoire à n sommets et m hyperarêtes est obtenu en tirant aléatoirement une matrice binaire $m \times n$ telle que

- chaque coefficient suit une loi de Bernoulli de paramètre p,
- les cases forment une famille indépendante de variables aléatoires

Premier résultat

Espérance du nombre moyen de traverses minimales

Le nombre moyen de traverses minimales est superpolynômial et satisfait

$$\mathrm{E}_n[M] \quad = \quad \exp\left(\frac{(\log n)^2}{|\log q|} - \frac{\log\log n}{|\log q|}\log n + O(\log n)\right).$$

Remarque : Pourtant, la situation est plutôt favorable car les traverses sont courtes (de l'ordre de $\Theta(\log n)$).

Second résultat

Est-il très probable d'avoir peu de traverses minimales ?

Borne inférieure fortement probable

De plus, pour tout ϵ avec $\frac{2 \log \log n}{\log n} < \epsilon < 1$,

$$\mathrm{P}_n \left[M > \mathrm{E}_n[M]^{1-\epsilon} \right] = 1 + O(r_n(\epsilon))$$

avec $r_n(\epsilon) \to 0$ explicite.

Autrement dit, avec une probabilité très proche de 1, il y a un nombre superpolynômial de traverses minimales.

Troisième résultat

Le troisième résultat concerne la complexité de l'algorithme MTMiner (ou HBC-algorithm)

Définition : Soit $\mathcal{H} = (\mathbf{V}, \mathbf{E})$ et $X \subset \mathbf{V}$. Le support de X (noté $Supp(X)$) est le nombre d'hyperarêtes qu'intersecte X.

Troisième résultat

Le troisième résultat concerne la complexité de l'algorithme MTMiner (ou HBC-algorithm)

Définition : Soit $\mathcal{H} = (\mathbf{V}, \mathbf{E})$ et $X \subset \mathbf{V}$. Le support de X (noté $Supp(X)$) est le nombre d'hyperarêtes qu'intersecte X.

Définition : Un ensemble de sommets X est dit clé si tout sous-ensemble strict de X a un support strictement plus petit que celui de X.

Troisième résultat

Le troisième résultat concerne la complexité de l'algorithme MTMiner (ou HBC-algorithm)

Définition : Soit $\mathcal{H} = (\mathbf{V}, \mathbf{E})$ et $X \subset \mathbf{V}$. Le support de X (noté $Supp(X)$) est le nombre d'hyperarêtes qu'intersecte X.

Définition : Un ensemble de sommets X est dit clé si tout sous-ensemble strict de X a un support strictement plus petit que celui de X.

Étant donné l'hypergraphe $\mathcal{H} = (\mathbf{V}, \mathbf{E})$ avec $\mathbf{E} = \{\{1,3\},\{2,3\},\{4,5\},\{3,5\}\}$,

Troisième résultat

Le troisième résultat concerne la complexité de l'algorithme MTMiner (ou HBC-algorithm)

Définition : Soit $\mathcal{H} = (\mathbf{V}, \mathbf{E})$ et $X \subset \mathbf{V}$. Le support de X (noté $Supp(X)$) est le nombre d'hyperarêtes qu'intersecte X.

Définition : Un ensemble de sommets X est dit clé si tout sous-ensemble strict de X a un support strictement plus petit que celui de X.

Étant donné l'hypergraphe $\mathcal{H} = (\mathbf{V}, \mathbf{E})$ avec $\mathbf{E} = \{\{1,3\},\{2,3\},\{4,5\},\{3,5\}\}$,

- $Supp(\{3\}) = 3$, $Supp(\{4\}) = 1$, $Supp(\{2,3\}) = 3$, $Supp(\{3,4\}) = 4$
- $\{3\}$ et $\{3,4\}$ sont des ensembles clés
- $\{2,3\}$ n'est pas un ensemble clé car $Supp(\{2,3\}) = Supp(\{3\})$

Troisième résultat

Le troisième résultat concerne la complexité de l'algorithme MTMiner (ou HBC-algorithm)

Définition : Soit $\mathcal{H} = (\mathbf{V}, \mathbf{E})$ et $X \subset \mathbf{V}$. Le support de X (noté $Supp(X)$) est le nombre d'hyperarêtes qu'intersecte X.

Définition : Un ensemble de sommets X est dit clé si tout sous-ensemble strict de X a un support strictement plus petit que celui de X.

Étant donné l'hypergraphe $\mathcal{H} = (\mathbf{V}, \mathbf{E})$ avec $\mathbf{E} = \{\{1,3\}, \{2,3\}, \{4,5\}, \{3,5\}\}$,

- $Supp(\{3\}) = 3$, $Supp(\{4\}) = 1$, $Supp(\{2,3\}) = 3$, $Supp(\{3,4\}) = 4$
- $\{3\}$ et $\{3,4\}$ sont des ensembles clés
- $\{2,3\}$ n'est pas un ensemble clé car $Supp(\{2,3\}) = Supp(\{3\})$

Les traverses minimales sont des ensembles clés.

MTMiner (ou HBC-algorithm)

Principes de l'algorithme

- MTMiner suit une stratégie par niveau (parcours en largeur du treillis)
- MTMiner construit les ensembles clés de cardinal $i+1$ à partir des ensembles clés de cardinal i
- Notons K_i l'ensemble des ensembles clés de cardinal i. K_{i+1} peut être construit à partir de K_i en temps $O((m+n)^3|K_i|)$ (avec une structure d'arbre préfixe par exemple).
- La complexité de MTMiner est donc de l'ordre $O(Poly(n,m) \cdot K)$ où K est le nombre total d'ensembles clés.

MTMiner (ou HBC-algorithm)

Principes de l'algorithme

- MTMiner suit une stratégie par niveau (parcours en largeur du treillis)
- MTMiner construit les ensembles clés de cardinal $i+1$ à partir des ensembles clés de cardinal i
- Notons K_i l'ensemble des ensembles clés de cardinal i. K_{i+1} peut être construit à partir de K_i en temps $O((m+n)^3|K_i|)$ (avec une structure d'arbre préfixe par exemple).
- La complexité de MTMiner est donc de l'ordre $O(Poly(n,m) \cdot K)$ où K est le nombre total d'ensembles clés.

MTMiner (ou HBC-algorithm)

Principes de l'algorithme

- MTMiner suit une stratégie par niveau (parcours en largeur du treillis)
- MTMiner construit les ensembles clés de cardinal $i+1$ à partir des ensembles clés de cardinal i
- Notons K_i l'ensemble des ensembles clés de cardinal i. K_{i+1} peut être construit à partir de K_i en temps $O((m+n)^3|K_i|)$ (avec une structure d'arbre préfixe par exemple).
- La complexité de MTMiner est donc de l'ordre $O(Poly(n,m) \cdot K)$ où K est le nombre total d'ensembles clés.

MTMiner (ou HBC-algorithm)

Principes de l'algorithme

- MTMiner suit une stratégie par niveau (parcours en largeur du treillis)
- MTMiner construit les ensembles clés de cardinal $i+1$ à partir des ensembles clés de cardinal i
- Notons K_i l'ensemble des ensembles clés de cardinal i. K_{i+1} peut être construit à partir de K_i en temps $O((m+n)^3|K_i|)$ (avec une structure d'arbre préfixe par exemple).
- La complexité de MTMiner est donc de l'ordre $O(Poly(n,m) \cdot K)$ où K est le nombre total d'ensembles clés.

MTMiner (ou HBC-algorithm)

Input : an hypergraph $\mathcal{H}(\mathbf{V},\mathbf{E})$ with n hyperedges
Output : the minimal tranversals of $\mathcal{H}$

$MT := \{\{v\} |\ v \in \mathbf{V},\ Supp(\{v\}) = n\}$
$K_1 := \{\{v\} |\ v \in \mathbf{V},\ n > Supp(\{v\}) \neq 0\}$
$j = 1$
While $K_j \neq \emptyset$ **do**
  for all prefix V with $V \cup \{v_1\}$ and $V \cup \{v_2\}$ in $K_j \times K_j$ **do**
    $W = V \cup \{v_1\} \cup \{v_2\}$
    if W is a key set **then**
      if $Supp(W) = n$ **then** add W to MT
      else add W to K_{j+1} **end if**
    end if
  end for
  j=j+1
end While
return MT.

Troisième résultat

Complexité de MTMiner

Le nombre moyen d'ensembles clés satisfait

$$\mathrm{E}_n[K] \underset{n\to\infty}{\sim} \mathrm{E}_n[M] = \exp\left(\frac{(\log n)^2}{|\log q|} - \frac{\log n}{|\log q|}\log\log n + O(\log n)\right)$$

La complexité moyenne de MTMiner vérifie alors

$$\mathrm{E}_n[C] = \exp\left(\frac{(\log n)^2}{|\log q|} - \frac{\log n}{|\log q|}\log\log n + O(\log n)\right)$$

Remarque : le rapport entre la complexité moyenne de MTMiner et la taille de la sortie (i.e., le nombre de traverses minimales) suggère que MTMiner output–quasi–linéaire.

$$\frac{\mathrm{E}_n[C]}{\mathrm{E}_n[M]} = \exp\left(O(\log n)\right).$$

Quatrième résultat

Complexité moyenne du problème THG

Résoudre le problème THG peut se faire presque sûrement en temps output–quasi–linéaire. Précisément pour tout $\epsilon > 0$,

$$\mathrm{P}[C \leq M^{1+\epsilon}] = 1 + O\left(\tilde{r}_n(\epsilon)\right)$$

avec $\tilde{r}_n(\epsilon) \to 0$ explicite.

Eléments de preuve

Quatrième résultat

Il découle des trois premiers résultats et des inégalités de Markov et Bienaymé-Tchebychev.

Premier et troisième résultats

Par des méthodes combinatoires de niveau L2, on obtient les formules suivantes pour les espérances :

$$\mathrm{E}_n[X] = \sum_{j=1}^{m} \binom{m}{j} \sum_{\substack{k_1 \geq 1, \ldots, k_j \geq 1 \\ k_1 + \ldots + k_j \leq n}} \binom{n}{k_1, \ldots, k_j} A_j^{k_1+\ldots+k_j} \cdot B_j^{n-(k_1+\ldots+k_j)}$$

avec $A_j = pq^{j-1}$ et

$$B_j = \begin{cases} 1 - q^j - jpq^{j-1} & \text{si } X = M \\ 1 - jpq^{j-1} & \text{si } X = K \end{cases}$$

Eléments de preuve (suite)

Premier et troisième résultats (suite)

Ensuite, la somme multiple est transformée en une intégrale multiple,

$$\mathrm{E}_n[X] = \sum_{j=1}^{m} \binom{m}{j} \frac{n!}{(n-j)!} \int_{[0,A_j]^j} (B_j + t_1 + \ldots + t_j)^{n-j} dt_1 \ldots dt_j.$$

- pour j "petit", une majoration de l'intégrande suffit
- pour j "grand", l'intégrande est asymptotiquement proche de 1
- pour j intermédiaire, un développement limité à l'ordre 1 suffit et l'intégrale peut être calculée

Ensuite, ce sont des calculs asymptotiques.

Eléments de preuve (suite)

Deuxième résultat

Il s'agit de la borne inférieure de probabilité pour le nombre de traverses minimales.

- Première idée : calculer la variance de M
  $\rightarrow$ difficile car il y a beaucoup de cas à traiter
- Deuxième idée :
  - On a les majorations suivantes :
  $$M \geq M_j, \qquad \text{et} \qquad T_j \geq M_j \geq T_j - (m-j)T_{j-1},$$
  avec M_j (resp. T_j) le nombre de traverses minimales (resp. traverses) de cardinal j
  - [Ss-Res. 1] pour j assez petit, essentiellement toutes les traverses sont minimales
  $$\mathbb{E}_n[T_j - (m-j)T_{j-1}] \sim \mathbb{E}_n[T_j] \sim \mathbb{E}_n[M_j]$$
  - [Ss-Res. 2] pour j assez grand, T_j est superpolynomial
  - [Ss-Res. 3] T_j est concentré autour de sa moyenne (calcul de variance)
  - [Conclusion] Par l'encadrement, M_j est avec grande probabilité égal à sa moyenne qui est superexponentielle

Eléments de preuve (suite)

Deuxième résultat

Il s'agit de la borne inférieure de probabilité pour le nombre de traverses minimales.

- Deuxième idée :
  - On a les majorations suivantes :

    $$M \geq M_j, \qquad \text{et} \qquad T_j \geq M_j \geq T_j - (m-j)T_{j-1}.$$

    avec M_j (resp. T_j) le nombre de traverses minimales (resp. traverses) de cardinal j
  - [Ss-Res. 1] pour j assez petit, essentiellement toutes les traverses sont minimales

    $$\mathrm{E}_n[T_j - (m-j)T_{j-1}] \sim \mathrm{E}_n[T_j] \sim \mathrm{E}_n[M_j]$$
  - [Ss-Res. 2] pour j assez grand, T_j est superpolynômial
  - [Ss-Res. 3] T_j est concentré autour de sa moyenne (calcul de variance)
  - [Conclusion] Par l'encadrement, M_j est avec grande probabilité proche de sa moyenne qui est superexponentielle

Eléments de preuve (suite)

Deuxième résultat

Il s'agit de la borne inférieure de probabilité pour le nombre de traverses minimales.

- Deuxième idée :
  - On a les majorations suivantes :

$$M \geq M_j, \qquad \text{et} \qquad T_j \geq M_j \geq T_j - (m-j)T_{j-1}.$$

avec M_j (resp. T_j) le nombre de traverses minimales (resp. traverses) de cardinal j

  - [Ss-Res. 1] pour j assez petit, essentiellement toutes les traverses sont minimales

$$\mathrm{E}_n[T_j - (m-j)T_{j-1}] \sim \mathrm{E}_n[T_j] \sim \mathrm{E}_n[M_j]$$

  - [Ss-Res. 2] pour j assez grand, T_j est superpolynômial
  - [Ss-Res. 3] T_j est concentré autour de sa moyenne (calcul de variance)
  - [Conclusion] Par l'encadrement, M_j est avec grande probabilité proche de sa moyenne qui est superexponentielle

Eléments de preuve (suite)

Deuxième résultat

Il s'agit de la borne inférieure de probabilité pour le nombre de traverses minimales.

- Deuxième idée :
  - On a les majorations suivantes :

$$M \geq M_j, \qquad \text{et} \qquad T_j \geq M_j \geq T_j - (m-j)T_{j-1}.$$

    avec M_j (resp. T_j) le nombre de traverses minimales (resp. traverses) de cardinal j
  - [Ss-Res. 1] pour j assez petit, essentiellement toutes les traverses sont minimales

$$\mathrm{E}_n[T_j - (m-j)T_{j-1}] \sim \mathrm{E}_n[T_j] \sim \mathrm{E}_n[M_j]$$

  - [Ss-Res. 2] pour j assez grand, T_j est superpolynômial
  - [Ss-Res. 3] T_j est concentré autour de sa moyenne (calcul de variance)
  - [Conclusion] Par l'encadrement, M_j est avec grande probabilité proche de sa moyenne qui est superexponentielle

Eléments de preuve (suite)

Deuxième résultat

Il s'agit de la borne inférieure de probabilité pour le nombre de traverses minimales.

- Deuxième idée :
  - On a les majorations suivantes :

$$M \geq M_j, \qquad \text{et} \qquad T_j \geq M_j \geq T_j - (m-j)T_{j-1}.$$

    avec M_j (resp. T_j) le nombre de traverses minimales (resp. traverses) de cardinal j
  - [Ss-Res. 1] pour j assez petit, essentiellement toutes les traverses sont minimales

$$\mathrm{E}_n[T_j - (m-j)T_{j-1}] \sim \mathrm{E}_n[T_j] \sim \mathrm{E}_n[M_j]$$

  - [Ss-Res. 2] pour j assez grand, T_j est superpolynômial
  - [Ss-Res. 3] T_j est concentré autour de sa moyenne (calcul de variance)
  - [Conclusion] Par l'encadrement, M_j est avec grande probabilité proche de sa moyenne qui est superexponentielle

Eléments de preuve (suite)

Deuxième résultat

Il s'agit de la borne inférieure de probabilité pour le nombre de traverses minimales.

- Deuxième idée :
  - On a les majorations suivantes :

$$M \geq M_j, \qquad \text{et} \qquad T_j \geq M_j \geq T_j - (m-j)T_{j-1}.$$

    avec M_j (resp. T_j) le nombre de traverses minimales (resp. traverses) de cardinal j
  - [Ss-Res. 1] pour j assez petit, essentiellement toutes les traverses sont minimales

$$\mathrm{E}_n[T_j - (m-j)T_{j-1}] \sim \mathrm{E}_n[T_j] \sim \mathrm{E}_n[M_j]$$

  - [Ss-Res. 2] pour j assez grand, T_j est superpolynômial
  - [Ss-Res. 3] T_j est concentré autour de sa moyenne (calcul de variance)
  - [Conclusion] Par l'encadrement, M_j est avec grande probabilité proche de sa moyenne qui est superexponentielle

Eléments de preuve (suite)

Deuxième résultat

Il s'agit de la borne inférieure de probabilité pour le nombre de traverses minimales.

- Deuxième idée :
  - On a les majorations suivantes :

    $$M \geq M_j, \qquad \text{et} \qquad T_j \geq M_j \geq T_j - (m-j)T_{j-1}.$$

    avec M_j (resp. T_j) le nombre de traverses minimales (resp. traverses) de cardinal j
  - [Ss-Res. 1] pour j assez petit, essentiellement toutes les traverses sont minimales

    $$\mathrm{E}_n[T_j - (m-j)T_{j-1}] \sim \mathrm{E}_n[T_j] \sim \mathrm{E}_n[M_j]$$
  - [Ss-Res. 2] pour j assez grand, T_j est superpolynômial
  - [Ss-Res. 3] T_j est concentré autour de sa moyenne (calcul de variance)
  - [Conclusion] Par l'encadrement, M_j est avec grande probabilité proche de sa moyenne qui est superexponentielle

Conclusion (partie traverses)

- On a apporté un nouvel éclairage sur la complexité du problème THG : celui de la complexité en moyenne
- Dans le modèle aléatoire choisi, THG est avec une grande probabilité output–quasi–linéaire
- Le modèle aléatoire, avantage beaucoup MTMiner qui est efficace avec des traverses minimales courtes
- Pour être plus complet, il faudrait
  - proposer d'autres modèles probabilistes ($p \to 0$, $m = n^{\alpha}$)
  - analyser les autres algorithmes

Plan

Conclusion générale

La fouille de données est un domaine *relativement vierge* pour l'analyse (en moyenne) d'algorithmes.

- La complexité dans le pire des cas ne permet pas d'expliquer certains phénomènes
- difficulté principale : concevoir des modèles aléatoires *raisonnables*
- difficulté secondaire : convaincre le domaine de l'utilité de l'analyse en moyenne