Bornes stochastiques après génération partielle d’une chaîne de Markov

Hilal DJAFRI

LSV, ENS de Cachan

Journée CheckBound

16/10/2008

Sommaire

Sommaire

Introduction

- Chaînes de Markov : Formalisme simple, développements théoriques importants $\Rightarrow$ très utilisées en évaluation des performances.
- Espace d'état très grand :
  1. Problème d'espace de stockage.
  2. Problème du temps de calcul.
- $\Rightarrow$ Réduire l'espace d'états. *Censored Markov Chain*.
- Comparaison des chaines de Markov $\Rightarrow$ *Théorie des ordres stochastiques.*
- Construction de bornes : *algorithmes de bornes*

Introduction

- Chaînes de Markov : Formalisme simple, développements théoriques importants ⇒ très utilisées en évaluation des performances.
- Espace d'état très grand :
  1. Problème d'espace de stockage.
  2. Problème du temps de calcul.
- ⇒ Réduire l'espace d'états. *Censored Markov Chain*.
- Comparaison des chaines de Markov ⇒ *Théorie des ordres stochastiques.*
- Construction de bornes : *algorithmes de bornes*

Introduction

- Chaînes de Markov : Formalisme simple, développements théoriques importants $\Rightarrow$ très utilisées en évaluation des performances.
- Espace d'état très grand :
  1. Problème d'espace de stockage.
  2. Problème du temps de calcul.
- $\Rightarrow$ Réduire l'espace d'états. *Censored Markov Chain*.
- Comparaison des chaines de Markov $\Rightarrow$ *Théorie des ordres stochastiques.*
- Construction de bornes : *algorithmes de bornes*

Introduction

- Chaînes de Markov : Formalisme simple, développements théoriques importants $\Rightarrow$ très utilisées en évaluation des performances.
- Espace d'état très grand :
  1. Problème d'espace de stockage.
  2. Problème du temps de calcul.
- $\Rightarrow$ Réduire l'espace d'états. *Censored Markov Chain*.
- Comparaison des chaines de Markov $\Rightarrow$ *Théorie des ordres stochastiques.*
- Construction de bornes : *algorithmes de bornes*

Introduction

- Chaînes de Markov : Formalisme simple, développements théoriques importants $\Rightarrow$ très utilisées en évaluation des performances.
- Espace d'état très grand :
  1. Problème d'espace de stockage.
  2. Problème du temps de calcul.
- $\Rightarrow$ Réduire l'espace d'états. *Censored Markov Chain*.
- Comparaison des chaines de Markov $\Rightarrow$ *Théorie des ordres stochastiques.*
- Construction de bornes : *algorithmes de bornes*

Sommaire

Définition

- Relation d'ordre $(S, \preceq)$
  1. Reflexive : $x \preceq x \; \forall x \in S$
  2. Transitive : $x \preceq y$ et $y \preceq z \Rightarrow x \preceq z$
  3. Antisymetrique $x \preceq y$ et $y \preceq x \Rightarrow x = y$
- S ensemble de fonctions de répartitions $v.a \Rightarrow$ *Ordre stochastique*
- plusieurs ordres ont été développés
  1. L'ordre stochastique fort. *Strong stochatic order* $\preceq_{st}$.
  2. L'ordre au sens de Taux de Hasard. *Hazard Rate Order* $\preceq_{hr}$.
  3. L'ordre au sens du Rapport de Vraisemblance. *Likelihood Ratio Order* $\preceq_{lr}$
  4. Les ordres convexes $\preceq_{cx}, \preceq_{icx}, \preceq_{icv}$

Définition

- Relation d'ordre $(S, \preceq)$
  1. Reflexive : $x \preceq x \ \forall x \in S$
  2. Transitive : $x \preceq y$ et $y \preceq z \Rightarrow x \preceq z$
  3. Antisymetrique $x \preceq y$ et $y \preceq x \Rightarrow x = y$
- S ensemble de fonctions de répartitions $v.a \Rightarrow$ *Ordre stochastique*
- plusieurs ordres ont été développés
  1. L'ordre stochastique fort. *Strong stochatic order* $\preceq_{st}$.
  2. L'ordre au sens de Taux de Hasard. *Hazard Rate Order* $\preceq_{hr}$.
  3. L'ordre au sens du Rapport de Vraisemblance. *Likelihood Ratio Order* $\preceq_{lr}$
  4. Les ordres convexes $\preceq_{cx}, \preceq_{icx}, \preceq_{icv}$

Définition

- Relation d'ordre $(S, \preceq)$
  1. Reflexive : $x \preceq x \ \forall x \in S$
  2. Transitive : $x \preceq y$ et $y \preceq z \Rightarrow x \preceq z$
  3. Antisymetrique $x \preceq y$ et $y \preceq x \Rightarrow x = y$
- S ensemble de fonctions de répartitions $v.a \Rightarrow$ *Ordre stochastique*
- plusieurs ordres ont été développés
  1. L'ordre stochastique fort. *Strong stochatic order* $\preceq_{st}$.
  2. L'ordre au sens de Taux de Hasard. *Hazard Rate Order* $\preceq_{hr}$.
  3. L'ordre au sens du Rapport de Vraisemblance. *Likelihood Ratio Order* $\preceq_{lr}$
  4. Les ordres convexes $\preceq_{cx}$, $\preceq_{icx}$, $\preceq_{icv}$

Définition

- Relation d'ordre $(S, \preceq)$
  1. Reflexive : $x \preceq x \; \forall x \in S$
  2. Transitive : $x \preceq y$ et $y \preceq z \Rightarrow x \preceq z$
  3. Antisymetrique $x \preceq y$ et $y \preceq x \Rightarrow x = y$
- S ensemble de fonctions de répartitions $v.a \Rightarrow$ *Ordre stochastique*
- plusieurs ordres ont été développés
  1. L'ordre stochastique fort. *Strong stochatic order* $\preceq_{st}$.
  2. L'ordre au sens de Taux de Hasard. *Hazard Rate Order* $\preceq_{hr}$.
  3. L'ordre au sens du Rapport de Vraisemblance. *Likelihood Ratio Order* $\preceq_{lr}$
  4. Les ordres convexes $\preceq_{cx}, \preceq_{icx}, \preceq_{icv}$

Définition

- Relation d'ordre $(S, \preceq)$
  1. Reflexive : $x \preceq x \ \forall x \in S$
  2. Transitive : $x \preceq y$ et $y \preceq z \Rightarrow x \preceq z$
  3. Antisymetrique $x \preceq y$ et $y \preceq x \Rightarrow x = y$
- S ensemble de fonctions de répartitions $v.a \Rightarrow$ *Ordre stochastique*
- plusieurs ordres ont été développés
  1. L'ordre stochastique fort. *Strong stochatic order* $\preceq_{st}$.
  2. L'ordre au sens de Taux de Hasard. *Hazard Rate Order* $\preceq_{hr}$.
  3. L'ordre au sens du Rapport de Vraisemblance. *Likelihood Ratio Order* $\preceq_{lr}$
  4. Les ordres convexes $\preceq_{cx}, \preceq_{icx}, \preceq_{icv}$

L'ordre stochastique fort

Définition

Soient F_X et F_Y deux fonctions de répartition sur $\mathbb{R}$. On a :
$F_X \preceq_{st} F_Y \Leftrightarrow F_X(x) \geq F_Y(x) \quad \forall x \in \mathbb{R}$

Définition

On dit que X est inférieur à Y au sens st noté $X \preceq_{st} Y$
Si on a $F_X \preceq_{st} F_Y$

Théorème

Soient X et Y deux v.a. à valeurs dans S. Les deux propositions suivantes sont équivalentes :

(i) $X \preceq_{st} Y$

(ii) $E[f(X)] \leq E[f(Y)]$ *pour toute fonction croissante* $f: \quad S \to \mathbb{R}$

L'ordre stochastique fort

Définition

Soient F_X et F_Y deux fonctions de répartition sur $\mathbb{R}$. On a :
$F_X \preceq_{st} F_Y \Leftrightarrow F_X(x) \geq F_Y(x) \ \ \forall x \in \mathbb{R}$

Définition

On dit que X est inférieur à Y au sens st noté $X \preceq_{st} Y$
Si on a $F_X \preceq_{st} F_Y$

Théorème

Soient X et Y deux v.a. à valeurs dans S. Les deux propositions suivantes sont équivalentes :

(i) $X \preceq_{st} Y$

(ii) $E[f(X)] \leq E[f(Y)]$ *pour toute fonction croissante* $f: \quad S \rightarrow \mathbb{R}$

L'ordre stochastique fort(suite)

Cas discret

Propriété

Soient X et Y deux variables aléatoires définies sur l'ensemble {1,...,n}. On note par p et q leurs distributions de probabilité respectives. On a alors :

$$X \preceq_{st} Y \Leftrightarrow \sum_{j=i}^{n} p[j] \leq \sum_{j=i}^{n} q[j] \ \ \forall i = \overline{n,1} \tag{1}$$

$$\begin{cases} p(n) & \leq & q(n) \\ p(n-1)+p(n) & \leq & q(n-1)+q(n) \\ & \vdots & \\ p(1)+\cdots+p(n) & \leq & q(1)+\cdots+q(n) \end{cases}$$

L'ordre stochastique fort(suite)

Cas discret

Propriété

Soient X et Y deux variables aléatoires définies sur l'ensemble {1,...,n}. On note par p et q leurs distributions de probabilité respectives. On a alors :

$$X \preceq_{st} Y \Leftrightarrow \sum_{j=i}^{n} p[j] \leq \sum_{j=i}^{n} q[j] \;\; \forall i = \overline{n,1} \tag{1}$$

$$\left\{ \begin{array}{ccc} p(n) & \leq & q(n) \\ p(n-1)+p(n) & \leq & q(n-1)+q(n) \\ & \vdots & \\ p(1)+\cdots+p(n) & \leq & q(1)+\cdots+q(n) \end{array} \right.$$

L'ordre stochastique fort(suite)

Comparaison matrices stochastiques

Définition

Soient P et Q deux matrices stochastiques :

$$P \preceq_{st} Q \Leftrightarrow P_{i,*} \preceq_{st} Q_{i,*} \quad \forall i \tag{2}$$

Monotonie stochastiques

Définition

Soit P une matrice stochastique. P est dite stochastiquement st-monotone si pour tout vecteurs de probabilités p et q on a :

$$p \preceq_{st} q \Rightarrow p.P \preceq_{st} q.P \tag{3}$$

$\Rightarrow P_{i,*} \preceq_{st} P_{j,*} \quad \forall i < j$

L'ordre stochastique fort(suite)

Comparaison matrices stochastiques

Définition

Soient P et Q deux matrices stochastiques :

$$P \preceq_{st} Q \Leftrightarrow P_{i,*} \preceq_{st} Q_{i,*} \quad \forall i \tag{2}$$

Monotonie stochastiques

Définition

Soit P une matrice stochastique. P est dite stochastiquement st-monotone si pour tout vecteurs de probabilités p et q on a :

$$p \preceq_{st} q \Rightarrow p.P \preceq_{st} q.P \tag{3}$$

$\Rightarrow P_{i,*} \preceq_{st} P_{j,*} \quad \forall i < j$

L'ordre stochastique fort(suite)

Pourquoi la monotonie ?

Proposition

Soient A, B, C et D des matrices stochastiques.
Si on a $A \preceq_{st} B$, $C \preceq_{st} D$ et D monotone alors $A.C \preceq_{st} B.D$

Théorème

Soit P (resp. Q) la matrice de transition d'une chaîne de Markov homogène $\{X_t, t \geq 0\}$ (resp. $\{Y_t, t \geq 0\}$). Si on a :

- $X_0 \preceq_{st} Y_0$
- *Au moins l'une des matrices de transition est st-monotone.*
- $P \preceq_{st} Q$

$$X_t \preceq_{st} Y_t \quad \forall t$$

L'ordre stochastique fort(suite)

Pourquoi la monotonie ?

Proposition

Soient A, B, C et D des matrices stochastiques.
Si on a $A \preceq_{st} B$, $C \preceq_{st} D$ et D monotone alors $A.C \preceq_{st} B.D$

Théorème

Soit P (resp. Q) la matrice de transition d'une chaîne de Markov homogène $\{X_t, t \geq 0\}$ (resp. $\{Y_t, t \geq 0\}$). Si on a :

- $X_0 \preceq_{st} Y_0$
- *Au moins l'une des matrices de transition est st-monotone.*
- $P \preceq_{st} Q$

$$X_t \preceq_{st} Y_t \;\; \forall t$$

Sommaire

Chaînes de Markov censorées

Définition

Considérons une chaîne de Markov à temps discret $\{X_t, t = 1, 2, ...\}$ *et à espace d'état fini S. Considérons une partition de* $S = E \cup E^c$ *et* $E \cap E^c = \phi$. *Supposons que les visites successives de E par* X_t *se produisent aux instants* $0 < t_1 < t_2 < ....$ *La chaîne* $\{X_u^E = X_{t_u}, u = 1, 2, ...\}$ *est appelée chaîne de Markov censorée.*

$$Q = \begin{pmatrix} Q_E & Q_{EE^c} \\ Q_{E^cE} & Q_{E^c} \end{pmatrix} \tag{4}$$

Chaînes de Markov censorées(suite)

Lemme

Soit Q la matrice de transition associée à une chaîne de Markov à temps discret $(X_t)_{t\geq 0}$. *Considérons une partition de l'espace d'état S en* (E, E^c). *La matrice de transition* S_E *associée à la chaîne de Markov censorée est donnée par :*

$$S_E = Q_E + Q_{EE^c} \left(\sum_{i=0}^{\infty} (Q_{E^c})^i \right) Q_{E^cE} \tag{5}$$

$$\sum_{i=0}^{\infty} (Q_{E^c})^i = (I - Q_{E^c})^{-1} \tag{6}$$

Chaînes de Markov censorées(suite)

Lemme

Soit Q la matrice de transition associée à une chaîne de Markov à temps discret $(X_t)_{t\geq 0}$. Considérons une partition de l'espace d'état S en (E, E^c). La matrice de transition S_E associée à la chaîne de Markov censorée est donnée par :

$$S_E = Q_E + Q_{EE^c}\left(\sum_{i=0}^{\infty}(Q_{E^c})^i\right)Q_{E^cE} \tag{5}$$

$$\sum_{i=0}^{\infty}(Q_{E^c})^i = (I - Q_{E^c})^{-1} \tag{6}$$

Chaînes de Markov censorées(suite)

Théorème

Si Q est irréductible, avec la distribution stationnaire $\Pi_Q = (\Pi_E, \Pi_{E^c})$ *et la distribution transitoire à l'instant t,* $\Pi_Q^t = (\Pi_E^t, \Pi_{E^c}^t)$*, alors, la distribution stationnaire* Π_{S_E} *et transitoire* $\Pi_{S_E}^t$ *à l'instant t sont données par :*

$$\Pi_{S_E} = \frac{\Pi_E}{\sum_{i \in E} \Pi_E(i)} \quad \text{et} \quad \Pi_{S_E}^t = \frac{\Pi_E^t}{\sum_{i \in E} \Pi_E^t(i)} \tag{7}$$

Sommaire

Quelques opérateurs

Définition

Soit P une matrice stochastique d'ordre $n \times n$

- *L'opérateur de sommation r est donné par* $r(P)[i,j] = \sum_{k=j}^{n} P[i,k]$
- *Son inverse r^{-1} est défini par*

$$r^{-1}(P)[i,j] = \begin{cases} P[i,n] & Si \quad j = n \\ P[i,j] - P[i,j+1] & Si \quad j < n \end{cases}$$

- *L'opérateur v est donné par* $v(P)[i,j] = max_{m \leq i}(\sum_{k \geq j} P[m,k]) = max_{m \geq i} r(P)[m,j]$

Quelques opérateurs

Définition

Soit P une matrice stochastique d'ordre $n \times n$

- *L'opérateur de sommation r est donné par* $r(P)[i,j] = \sum_{k=j}^{n} P[i,k]$
- *Son inverse* r^{-1} *est défini par*

$$r^{-1}(P)[i,j] = \begin{cases} P[i,n] & Si \quad j = n \\ P[i,j] - P[i,j+1] & Si \quad j < n \end{cases}$$

- *L'opérateur v est donné par* $v(P)[i,j] = max_{m \leq i}(\sum_{k \geq j} P[m,k]) = max_{m \geq i} r(P)[m,j]$

Quelques opérateurs

Définition

Soit P une matrice stochastique d'ordre $n \times n$

- *L'opérateur de sommation r est donné par* $r(P)[i,j] = \sum_{k=j}^{n} P[i,k]$
- *Son inverse r^{-1} est défini par*

$$r^{-1}(P)[i,j] = \begin{cases} P[i,n] & Si \quad j = n \\ P[i,j] - P[i,j+1] & Si \quad j < n \end{cases}$$

- *L'opérateur v est donné par* $v(P)[i,j] = max_{m \leq i}(\sum_{k \geq j} P[m,k]) = max_{m \geq i} r(P)[m,j]$

Algorithme de Truffet

Définition

Soit P une matrice sous stochastique. Posons $\beta_i = 1 - \sum_{j=1}^{n} P[i,j] \ \forall i$.

$$\theta(P)[i,j] = \begin{cases} P[i,j] & Si \quad j < n \\ P[i,j] + \beta_i & Si \quad j = n \end{cases}$$

Algorithme de Truffet

- Application de θ sur Q_E

Algorithme de Truffet

Définition

Soit P une matrice sous stochastique. Posons $\beta_i = 1 - \sum_{j=1}^{n} P[i,j] \ \forall i$.

$$\theta(P)[i,j] = \begin{cases} P[i,j] & Si \quad j < n \\ P[i,j] + \beta_i & Si \quad j = n \end{cases}$$

Algorithme de Truffet

- Application de θ sur Q_E

Algorithme de Vincent

- Déterminer Q telle que $P \preceq_{st} Q$ et Q $st - monotone$
- $$\begin{cases} \sum_{k=j}^{p} P[i,k] \leq \sum_{k=j}^{p} Q[i,k] & \forall i, \forall j \\ \sum_{k=j}^{p} Q[i-1,k] \leq \sum_{k=j}^{p} Q[i,k] & \forall i \geq 2, \forall j \end{cases}$$
- Remplacer les inégalités par des égalités.
- $\Rightarrow$ Algorithme de Vincent : $r^{-1}v(P)$

Algorithme de Vincent

- Déterminer Q telle que $P \preceq_{st} Q$ et Q $st-monotone$
-
$$\begin{cases} \sum_{k=j}^{p} P[i,k] \leq \sum_{k=j}^{p} Q[i,k] & \forall i, \forall j \\ \sum_{k=j}^{p} Q[i-1,k] \leq \sum_{k=j}^{p} Q[i,k] & \forall i \geq 2, \forall j \end{cases}$$
- Remplacer les inégalités par des égalités.
- $\Rightarrow$ Algorithme de Vincent : $r^{-1}v(P)$

Algorithme de Vincent

- Déterminer Q telle que $P \preceq_{st} Q$ et Q $st-monotone$
- $$\begin{cases} \sum_{k=j}^{p} P[i,k] \leq \sum_{k=j}^{p} Q[i,k] & \forall i, \forall j \\ \sum_{k=j}^{p} Q[i-1,k] \leq \sum_{k=j}^{p} Q[i,k] & \forall i \geq 2, \forall j \end{cases}$$
- Remplacer les inégalités par des égalités.
- $\Rightarrow$ Algorithme de Vincent : $r^{-1}v(P)$

Algorithme de Vincent

- Déterminer Q telle que $P \preceq_{st} Q$ et Q $st - monotone$
-
$$\begin{cases} \sum_{k=j}^{p} P[i,k] \leq \sum_{k=j}^{p} Q[i,k] & \forall i, \forall j \\ \sum_{k=j}^{p} Q[i-1,k] \leq \sum_{k=j}^{p} Q[i,k] & \forall i \geq 2, \forall j \end{cases}$$
- Remplacer les inégalités par des égalités.
- $\Rightarrow$ Algorithme de Vincent : $r^{-1}v(P)$

Algorithmes de graphes

- Travailler sur le graphe associé à la chaîne de Markov.
- Supprimer les arcs de type $(i, j) \in E \times E$.
- Chercher les chemins les plus probables entre chaque couple de sommets $i, j \in E$.
- Limiter le nombre d'arcs constituant les chemins $\leq \Delta$
- Algorithme de DIJKSTRA : Un chemin vers chaque destination.
- Parcours en Largeur BFS : Un arbre depuis un état racine.
- Nécessitent des éléments de Q_{E^c}, Q_{EE^c} et Q_{E^cE}

Algorithmes de graphes

- Travailler sur le graphe associé à la chaîne de Markov.
- Supprimer les arcs de type $(i, j) \in E \times E$.
- Chercher les chemins les plus probables entre chaque couple de sommets $i, j \in E$.
- Limiter le nombre d'arcs constituant les chemins $\leq \Delta$
- Algorithme de DIJKSTRA : Un chemin vers chaque destination.
- Parcours en Largeur BFS : Un arbre depuis un état racine.
- Nécessitent des éléments de Q_{E^c}, Q_{EE^c} et Q_{E^cE}

Algorithmes de graphes

- Travailler sur le graphe associé à la chaîne de Markov.
- Supprimer les arcs de type $(i, j) \in E \times E$.
- Chercher les chemins les plus probables entre chaque couple de sommets $i, j \in E$.
- Limiter le nombre d'arcs constituant les chemins $\leq \Delta$
- Algorithme de DIJKSTRA : Un chemin vers chaque destination.
- Parcours en Largeur BFS : Un arbre depuis un état racine.
- Nécessitent des éléments de Q_{E^c}, Q_{EE^c} et Q_{E^cE}

Algorithmes de graphes

- Travailler sur le graphe associé à la chaîne de Markov.
- Supprimer les arcs de type $(i, j) \in E \times E$.
- Chercher les chemins les plus probables entre chaque couple de sommets $i, j \in E$.
- Limiter le nombre d'arcs constituant les chemins $\leq \Delta$
- Algorithme de DIJKSTRA : Un chemin vers chaque destination.
- Parcours en Largeur BFS : Un arbre depuis un état racine.
- Nécessitent des éléments de Q_{E^c}, Q_{EE^c} et Q_{E^cE}

Algorithmes de graphes

- Travailler sur le graphe associé à la chaîne de Markov.
- Supprimer les arcs de type $(i, j) \in E \times E$.
- Chercher les chemins les plus probables entre chaque couple de sommets $i, j \in E$.
- Limiter le nombre d'arcs constituant les chemins $\leq \Delta$
- Algorithme de DIJKSTRA : Un chemin vers chaque destination.
- Parcours en Largeur BFS : Un arbre depuis un état racine.
- Nécessitent des éléments de Q_{E^c}, Q_{EE^c} et Q_{E^cE}

Algorithmes de graphes

- Travailler sur le graphe associé à la chaîne de Markov.
- Supprimer les arcs de type $(i, j) \in E \times E$.
- Chercher les chemins les plus probables entre chaque couple de sommets $i, j \in E$.
- Limiter le nombre d'arcs constituant les chemins $\leq \Delta$
- Algorithme de DIJKSTRA : Un chemin vers chaque destination.
- Parcours en Largeur BFS : Un arbre depuis un état racine.
- Nécessitent des éléments de Q_{E^c}, Q_{EE^c} et Q_{E^cE}

Algorithmes de graphes

- Travailler sur le graphe associé à la chaîne de Markov.
- Supprimer les arcs de type $(i, j) \in E \times E$.
- Chercher les chemins les plus probables entre chaque couple de sommets $i, j \in E$.
- Limiter le nombre d'arcs constituant les chemins $\leq \Delta$
- Algorithme de DIJKSTRA : Un chemin vers chaque destination.
- Parcours en Largeur BFS : Un arbre depuis un état racine.
- Nécessitent des éléments de Q_{E^c}, Q_{EE^c} et Q_{E^cE}

Algorithme DPY

- Normaliser la matrice $Q_{E^cE} \rightarrow G$
- Appliquer l'algorithme de Vincent sur G. $F \leftarrow r^{-1}v(G)$
- Choisir la dernière ligne de F. $V \leftarrow max_i F[i,*]$
- $U[i,j] = \beta[i] \times V[j]$

Théorème

$$S_E \preceq_{st} A + U \quad (8)$$

Algorithme DPY

- Normaliser la matrice $Q_{E^cE} \rightarrow G$
- Appliquer l'algorithme de Vincent sur G. $F \leftarrow r^{-1}v(G)$
- Choisir la dernière ligne de F. $V \leftarrow max_i F[i,*]$
- $U[i,j] = \beta[i] \times V[j]$

Théorème

$$S_E \preceq_{st} A + U \tag{8}$$

Algorithme DPY

- Normaliser la matrice $Q_{E^cE} \rightarrow G$
- Appliquer l'algorithme de Vincent sur G. $F \leftarrow r^{-1}v(G)$
- Choisir la dernière ligne de F. $V \leftarrow max_i F[i,*]$
- $U[i,j] = \beta[i] \times V[j]$

Théorème

$$S_E \preceq_{st} A + U \qquad (8)$$

Algorithme DPY

- Normaliser la matrice $Q_{E^cE} \rightarrow G$
- Appliquer l'algorithme de Vincent sur G. $F \leftarrow r^{-1}v(G)$
- Choisir la dernière ligne de F. $V \leftarrow max_i F[i, *]$
- $U[i, j] = \beta[i] \times V[j]$

Théorème

$$S_E \preceq_{st} A + U \tag{8}$$

Algorithme de division euclidienne

Définition

Soient V et W deux vecteurs de mêmes tailles dont les éléments sont positifs. On défini la division euclidienne de W par V comme suit :

$$W = qV + R \tag{9}$$

$$q = min_i(\frac{W(i)}{V(i)}) \tag{10}$$

Théorème

Notons par $\sigma[i] = \sum_{j=1}^{p} Q_{E^cE}[i,j]$, $H = Q_{EE^c}(I - Q_{E^c})^{-1} Q_{E^cE}$ et $\beta[i] = \sum_{j=1}^{p} H[i,j]$. Effectuons la division euclidienne de σ par une colonne arbitraire k de Q_{E^cE} pour obtenir q_k et R_k. La colonne k de H est alors bornée supérieurement par $\frac{\beta}{q_k}$

Algorithme de division euclidienne

Définition

Soient V et W deux vecteurs de mêmes tailles dont les éléments sont positifs. On défini la division euclidienne de W par V comme suit :

$$W = qV + R \tag{9}$$

$$q = min_i(\frac{W(i)}{V(i)}) \tag{10}$$

Théorème

Notons par $\sigma[i] = \sum_{j=1}^{p} Q_{E^cE}[i,j]$, $H = Q_{EE^c}(I - Q_{E^c})^{-1}Q_{E^cE}$ et $\beta[i] = \sum_{j=1}^{p} H[i,j]$. Effectuons la division euclidienne de σ par une colonne arbitraire k de Q_{E^cE} pour obtenir q_k et R_k. La colonne k de H est alors bornée supérieurement par $\frac{\beta}{q_k}$

Algorithme de Haddad et Moreaux

- Incertitudes sur les probabilités de transition.
- Bornes élément par élément : $0 \preceq_{el} L \preceq_{el} P \preceq_{el} U \preceq_{el} 1$.
- $\Rightarrow$ Algorithme de Haddad et Moreaux : Déterminer Q telle que $P \preceq_{st} Q$.

Algorithme 15 Algorithme de Haddad et Moreaux pour borne supérieure

Entrées: P, L et U
Sorties: A

```
pour i = 1 to n faire
    pour j = n downto 2 faire
        C_{i,j} ← min{1 − Σ_{k=1}^{j−1} L_{i,k}, Σ_{k=j}^{n} U_{i,k}}
    fin pour
    C_{i,1} ← 1, A_{i,n} ← C_{i,n}
    pour j = n − 1 downto 1  faire
        A_{i,j} ← C_{i,j} − C_{i,j+1}
    fin pour
fin pour
```

Algorithme de Haddad et Moreaux

- Incertitudes sur les probabilités de transition.
- Bornes élément par élément : $0 \preceq_{el} L \preceq_{el} P \preceq_{el} U \preceq_{el} 1$.
- $\Rightarrow$ Algorithme de Haddad et Moreaux : Déterminer Q telle que $P \preceq_{st} Q$.

Algorithme 15 Algorithme de Haddad et Moreaux pour borne supérieure

Entrées: P, L et U
Sorties: A

```
pour i = 1 to n faire
    pour j = n downto 2 faire
        C_{i,j} ← min{1 − Σ_{k=1}^{j−1} L_{i,k}, Σ_{k=j}^{n} U_{i,k}}
    fin pour
    C_{i,1} ← 1, A_{i,n} ← C_{i,n}
    pour j = n − 1 downto 1 faire
        A_{i,j} ← C_{i,j} − C_{i,j+1}
    fin pour
fin pour
```

Algorithme de Haddad et Moreaux

- Incertitudes sur les probabilités de transition.
- Bornes élément par élément : $0 \preceq_{el} L \preceq_{el} P \preceq_{el} U \preceq_{el} 1$.
- $\Rightarrow$ Algorithme de Haddad et Moreaux : Déterminer Q telle que $P \preceq_{st} Q$.

Algorithme 15 Algorithme de Haddad et Moreaux pour borne supérieure

Entrées: P, L et U
Sorties: A
  pour $i = 1$ **to** n **faire**
    pour $j = n$ **downto** 2 **faire**
      $C_{i,j} \leftarrow min\{1 - \sum_{k=1}^{j-1} L_{i,k}, \sum_{k=j}^{n} U_{i,k}\}$
    fin pour
    $C_{i,1} \leftarrow 1$, $A_{i,n} \leftarrow C_{i,n}$
    pour $j = n - 1$ **downto** 1 **faire**
      $A_{i,j} \leftarrow C_{i,j} - C_{i,j+1}$
    fin pour
  fin pour

Algorithme de Haddad et Moreaux

- Incertitudes sur les probabilités de transition.
- Bornes élément par élément : $0 \preceq_{el} L \preceq_{el} P \preceq_{el} U \preceq_{el} 1$.
- $\Rightarrow$ Algorithme de Haddad et Moreaux : Déterminer Q telle que $P \preceq_{st} Q$.

Algorithme 15 Algorithme de Haddad et Moreaux pour borne supérieure

Entrées: P, L et U
Sorties: A
  pour $i = 1$ **to** n **faire**
    pour $j = n$ **downto** 2 **faire**
      $C_{i,j} \leftarrow min\{1 - \sum_{k=1}^{j-1} L_{i,k}, \sum_{k=j}^{n} U_{i,k}\}$
    fin pour
    $C_{i,1} \leftarrow 1$, $A_{i,n} \leftarrow C_{i,n}$
    pour $j = n - 1$ **downto** 1 **faire**
      $A_{i,j} \leftarrow C_{i,j} - C_{i,j+1}$
    fin pour
  fin pour

Sommaire

Construction d'une Borne Supérieure

- Idéee de départ : décomposer $S_E = A + B.(I - D)^{-1}.W.W^{-1}.C$
- $W = diag(1 - s_1, ..., 1 - s_q)$ avec $s_i = \sum_{j=1}^{q} D_{ij} = 1 - \sum_{j=1}^{p} C_{ij}$.
- $\Rightarrow W^{-1}.C$ est stochastique

Lemme

La matrice $(I - D)^{-1}.W$ est stochastique.

- Mais D est inconnue.
- $\Rightarrow$ Remplacer D par $D' = \begin{pmatrix} 0 & \cdots & 0 & s_1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & s_q \end{pmatrix}$

Construction d'une Borne Supérieure

- Idéee de départ : décomposer $S_E = A + B.(I - D)^{-1}.W.W^{-1}.C$
- $W = diag(1 - s_1, ..., 1 - s_q)$ avec $s_i = \sum_{j=1}^{q} D_{ij} = 1 - \sum_{j=1}^{p} C_{ij}$.
- $\Rightarrow W^{-1}.C$ est stochastique

Lemme

La matrice $(I - D)^{-1}.W$ est stochastique.

- Mais D est inconnue.
- $\Rightarrow$ Remplacer D par $D' = \begin{pmatrix} 0 & \cdots & 0 & s_1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & s_q \end{pmatrix}$

Construction d'une Borne Supérieure

- Idéee de départ : décomposer $S_E = A + B.(I - D)^{-1}.W.W^{-1}.C$
- $W = diag(1 - s_1, ..., 1 - s_q)$ avec $s_i = \sum_{j=1}^{q} D_{ij} = 1 - \sum_{j=1}^{p} C_{ij}$.
- $\Rightarrow W^{-1}.C$ est stochastique

Lemme

La matrice $(I - D)^{-1}.W$ est stochastique.

- Mais D est inconnue.
- $\Rightarrow$ Remplacer D par $D' = \begin{pmatrix} 0 & \cdots & 0 & s_1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & s_q \end{pmatrix}$

Construction d'une Borne Supérieure

- Idéee de départ : décomposer $S_E = A + B.(I - D)^{-1}.W.W^{-1}.C$
- $W = diag(1 - s_1, ..., 1 - s_q)$ avec $s_i = \sum_{j=1}^{q} D_{ij} = 1 - \sum_{j=1}^{p} C_{ij}$.
- $\Rightarrow W^{-1}.C$ est stochastique

Lemme

La matrice $(I - D)^{-1}.W$ est stochastique.

- Mais D est inconnue.
- $\Rightarrow$ Remplacer D par $D' = \begin{pmatrix} 0 & \cdots & 0 & s_1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & s_q \end{pmatrix}$

Construction d'une Borne Supérieure(Suite)

Lemme

$$(I - D')^{-1}.W = \begin{pmatrix} 1 - s_1 & \cdots & 0 & s_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 - s_{q-1} & s_{q-1} \\ 0 & \cdots & 0 & 1 \end{pmatrix} = H$$

$\Rightarrow$ H est aussi stochastique.

Lemme

$$(I - D)^{-1}.W \preceq_{st} H \tag{11}$$

Application de l'algorithme de Haddad et Moreaux avec $L = W$ et $U = 1$

Construction d'une Borne Supérieure(Suite)

Lemme

$$(I - D')^{-1}.W = \begin{pmatrix} 1 - s_1 & \cdots & 0 & s_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 - s_{q-1} & s_{q-1} \\ 0 & \cdots & 0 & 1 \end{pmatrix} = H$$

$\Rightarrow$ H est aussi stochastique.

Lemme

$$(I - D)^{-1}.W \preceq_{st} H \tag{11}$$

Application de l'algorithme de Haddad et Moreaux avec $L = W$ et $U = 1$

Construction d'une Borne Supérieure(Suite)

Lemme

$$(I - D')^{-1}.W = \begin{pmatrix} 1-s_1 & \cdots & 0 & s_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1-s_{q-1} & s_{q-1} \\ 0 & \cdots & 0 & 1 \end{pmatrix} = H$$

$\Rightarrow$ H est aussi stochastique.

Lemme

$$(I - D)^{-1}.W \preceq_{st} H \tag{11}$$

Application de l'algorithme de Haddad et Moreaux avec $L = W$ et $U = 1$

Construction d'une Borne Supérieure(Suite)

Théorème

$$S_E \preceq_{st} A + B.r^{-1}v(H).r^{-1}v(W^{-1}.C) = \overline{S}_E \quad (12)$$

- $W^{-1}.C \preceq_{st} r^{-1}v(W^{-1}.C)$
- $(I-D)^{-1}.W \preceq_{st} H \preceq_{st} r^{-1}v(H)$
- $r^{-1}v(W^{-1}.C)$ est monotone.
- $(I-D)^{-1}.W.W^{-1}.C \preceq_{st} r^{-1}v(H).r^{-1}v(W^{-1}.C)$
- $r^{-1}v(H).r^{-1}v(W^{-1}.C)$ monotone
- $S_E = A + B.(I-D)^{-1}.C \preceq_{st} A + B.r^{-1}v(H).r^{-1}v(W^{-1}.C)$

Construction d'une Borne Supérieure(Suite)

Théorème

$$S_E \preceq_{st} A + B.r^{-1}v(H).r^{-1}v(W^{-1}.C) = \overline{S}_E \tag{12}$$

- $W^{-1}.C \preceq_{st} r^{-1}v(W^{-1}.C)$
- $(I - D)^{-1}.W \preceq_{st} H \preceq_{st} r^{-1}v(H)$
- $r^{-1}v(W^{-1}.C)$ est monotone.
- $(I - D)^{-1}.W.W^{-1}.C \preceq_{st} r^{-1}v(H).r^{-1}v(W^{-1}.C)$
- $r^{-1}v(H).r^{-1}v(W^{-1}.C)$ monotone
- $S_E = A + B.(I - D)^{-1}.C \preceq_{st} A + B.r^{-1}v(H).r^{-1}v(W^{-1}.C)$

Construction d'une Borne Supérieure(Suite)

Théorème

$$S_E \preceq_{st} A + B.r^{-1}v(H).r^{-1}v(W^{-1}.C) = \overline{S}_E \tag{12}$$

- $W^{-1}.C \preceq_{st} r^{-1}v(W^{-1}.C)$
- $(I - D)^{-1}.W \preceq_{st} H \preceq_{st} r^{-1}v(H)$
- $r^{-1}v(W^{-1}.C)$ est monotone.
- $(I - D)^{-1}.W.W^{-1}.C \preceq_{st} r^{-1}v(H).r^{-1}v(W^{-1}.C)$
- $r^{-1}v(H).r^{-1}v(W^{-1}.C)$ monotone
- $S_E = A + B.(I - D)^{-1}.C \preceq_{st} A + B.r^{-1}v(H).r^{-1}v(W^{-1}.C)$

Construction d'une Borne Supérieure(Suite)

Algorithme 16 Algorithme pour la borne supérieure

Entrées: A, B et C
Sorties: $\overline{S}_E$

pour $i = 1$ to q **faire**
  $W[i] \leftarrow \sum_{j=1}^{p} C[i,j]$
fin pour
pour tout i et j **faire**
  $C[i,j] \leftarrow \frac{C[i,j]}{W[i]}$
fin pour
$C \leftarrow r^{-1}v(C)$
pour $i = 2$ to n **faire**
  si $W[i-1] < W[i]$ **alors**
    $W[i] \leftarrow W[i-1]$
  finsi
fin pour
pour $i = 1$ to $q-1$ **faire**
  pour $j = 1$ to p **faire**
    $C[i,j] \leftarrow W[i].C[i,j] + (1 - W[i]).C[q,j]$
  fin pour
fin pour
$\overline{S}_E \leftarrow A + B.C$

Comparaison avec l'algorithme de Truffet

- Seule A est connue
- Remplacer B dans $\overline{S}_E$ par $\begin{pmatrix} 0 & \cdots & 0 & b_1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & b_p \end{pmatrix}$ où $b_i = \sum_{j=1}^{q} B_{ij} = 1 - \sum_{j=1}^{p} A_{ij}$
- Remplacer $r^{-1}v(H).r^{-1}v(W^{-1}.C)$ par $\begin{pmatrix} 0 & \cdots & 0 & 1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}$
- $\overline{S}_E = A + \begin{pmatrix} 0 & \cdots & 0 & b_1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & b_p \end{pmatrix} . \begin{pmatrix} 0 & \cdots & 0 & 1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}$

Comparaison avec l'algorithme de Truffet

- Seule A est connue
- Remplacer B dans $\overline{S}_E$ par $\begin{pmatrix} 0 & \cdots & 0 & b_1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & b_p \end{pmatrix}$ où $b_i = \sum_{j=1}^{q} B_{ij} = 1 - \sum_{j=1}^{p} A_{ij}$
- Remplacer $r^{-1}v(H).r^{-1}v(W^{-1}.C)$ par $\begin{pmatrix} 0 & \cdots & 0 & 1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}$
- $\overline{S}_E = A + \begin{pmatrix} 0 & \cdots & 0 & b_1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & b_p \end{pmatrix} . \begin{pmatrix} 0 & \cdots & 0 & 1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}$

Comparaison avec l'algorithme de Truffet

- Seule A est connue
- Remplacer B dans $\overline{S}_E$ par $\begin{pmatrix} 0 & \cdots & 0 & b_1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & b_p \end{pmatrix}$ où

  $b_i = \sum_{j=1}^{q} B_{ij} = 1 - \sum_{j=1}^{p} A_{ij}$
- Remplacer $r^{-1}v(H).r^{-1}v(W^{-1}.C)$ par $\begin{pmatrix} 0 & \cdots & 0 & 1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}$
- $\overline{S}_E = A + \begin{pmatrix} 0 & \cdots & 0 & b_1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & b_p \end{pmatrix} . \begin{pmatrix} 0 & \cdots & 0 & 1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}$

Comparaison avec l'algorithme de Truffet

- $\overline{S}_E = A + \begin{pmatrix} 0 & \cdots & 0 & b_1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & b_p \end{pmatrix} \rightarrow$ Borne de l'algoritme de Truffet

Cas où A, B connues ; C et D inconnues

Supposons que B est connue et que C et D sont inconnues.

- On remplace comme précédemment $r^{-1}v(H).r^{-1}v(W^{-1}.C)$
- $\overline{S}_E = A + \begin{pmatrix} b_{11} & \cdots & b_{1q} \\ \vdots & & \vdots \\ b_{p1} & \cdots & b_{pq} \end{pmatrix} \cdot \begin{pmatrix} 0 & \cdots & 0 & 1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}$
- $\overline{S}_E = A + \begin{pmatrix} 0 & \cdots & 0 & b_1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & b_p \end{pmatrix}$ Encore Borne de l'algorithme de Truffet.

Cas où A, B connues ; C et D inconnues

Supposons que B est connue et que C et D sont inconnues.

- On remplace comme précédemment $r^{-1}v(H).r^{-1}v(W^{-1}.C)$
- $\overline{S}_E = A + \begin{pmatrix} b_{11} & \cdots & b_{1q} \\ \vdots & & \vdots \\ b_{p1} & \cdots & b_{pq} \end{pmatrix} . \begin{pmatrix} 0 & \cdots & 0 & 1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}$
- $\overline{S}_E = A + \begin{pmatrix} 0 & \cdots & 0 & b_1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & b_p \end{pmatrix}$ Encore Borne de l'algorithme de Truffet.

Cas où A, B connues ; C et D inconnues

Supposons que B est connue et que C et D sont inconnues.

- On remplace comme précédemment $r^{-1}v(H).r^{-1}v(W^{-1}.C)$
- $\overline{S}_E = A + \begin{pmatrix} b_{11} & \cdots & b_{1q} \\ \vdots & & \vdots \\ b_{p1} & \cdots & b_{pq} \end{pmatrix} . \begin{pmatrix} 0 & \cdots & 0 & 1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}$
- $\overline{S}_E = A + \begin{pmatrix} 0 & \cdots & 0 & b_1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & b_p \end{pmatrix}$ Encore Borne de l'algorithme de Truffet.

Comparaison avec DPY

- Remplacer B par $[0_{p\times(q-1)}|b_{p\times 1}]$ où $b_i = \sum_{j=1}^{q} B_{i,j} = 1 - \sum_{j=1}^{p} A_{i,j}$
- $\overline{S}_E = A + [0_{p\times(q-1)}|b_{p\times 1}].r^{-1}v((I-D')^{-1}W).r^{-1}v(W^{-1}C)$
- La dernière ligne de $r^{-1}v((I-D')^{-1}W)$ vaut $(0, ..., 0, 1)$
- Donc $[0_{p\times(q-1)}|b_{p\times 1}].r^{-1}v((I-D')^{-1}W) = [0_{p\times(q-1)}|b_{p\times 1}]$
- $\overline{S}_E = A + [0_{p\times(q-1)}|b_{p\times 1}].r^{-1}v(W^{-1}C)$
- La dernière ligne de $r^{-1}v(W^{-1}C)$ peut être exprimée par $max_{i=1,q}r^{-1}v(W^{-1}C)[i,*]$
- $\overline{S}_E = A + b.\left(max_{i=1,q}r^{-1}v(W^{-1}C)[i,*]\right)^t$ qui coïncide avec la borne fournie par DPY.

Comparaison avec DPY

- Remplacer B par $[0_{p\times(q-1)}|b_{p\times 1}]$ où $b_i = \sum_{j=1}^{q} B_{i,j} = 1 - \sum_{j=1}^{p} A_{i,j}$
- $\overline{S}_E = A + [0_{p\times(q-1)}|b_{p\times 1}].r^{-1}v((I - D')^{-1}W).r^{-1}v(W^{-1}C)$
- La dernière ligne de $r^{-1}v((I - D')^{-1}W)$ vaut $(0, ..., 0, 1)$
- Donc $[0_{p\times(q-1)}|b_{p\times 1}].r^{-1}v((I - D')^{-1}W) = [0_{p\times(q-1)}|b_{p\times 1}]$
- $\overline{S}_E = A + [0_{p\times(q-1)}|b_{p\times 1}].r^{-1}v(W^{-1}C)$
- La dernière ligne de $r^{-1}v(W^{-1}C)$ peut être exprimée par $max_{i=1,q}r^{-1}v(W^{-1}C)[i,*]$
- $\overline{S}_E = A + b.\left(max_{i=1,q}r^{-1}v(W^{-1}C)[i,*]\right)^t$ qui coïncide avec la borne fournie par DPY.

Comparaison avec DPY

- Remplacer B par $[0_{p\times(q-1)}|b_{p\times 1}]$ où $b_i = \sum_{j=1}^{q} B_{i,j} = 1 - \sum_{j=1}^{p} A_{i,j}$
- $\overline{S}_E = A + [0_{p\times(q-1)}|b_{p\times 1}].r^{-1}v((I-D')^{-1}W).r^{-1}v(W^{-1}C)$
- La dernière ligne de $r^{-1}v((I-D')^{-1}W)$ vaut $(0, ..., 0, 1)$
- Donc $[0_{p\times(q-1)}|b_{p\times 1}].r^{-1}v((I-D')^{-1}W) = [0_{p\times(q-1)}|b_{p\times 1}]$
- $\overline{S}_E = A + [0_{p\times(q-1)}|b_{p\times 1}].r^{-1}v(W^{-1}C)$
- La dernière ligne de $r^{-1}v(W^{-1}C)$ peut être exprimée par $max_{i=1,q}r^{-1}v(W^{-1}C)[i,*]$
- $\overline{S}_E = A + b.\left(max_{i=1,q}r^{-1}v(W^{-1}C)[i,*]\right)^t$ qui coïncide avec la borne fournie par DPY.

Comparaison avec DPY

- Remplacer B par $[0_{p\times(q-1)}|b_{p\times 1}]$ où $b_i = \sum_{j=1}^{q} B_{i,j} = 1 - \sum_{j=1}^{p} A_{i,j}$
- $\overline{S}_E = A + [0_{p\times(q-1)}|b_{p\times 1}].r^{-1}v((I-D')^{-1}W).r^{-1}v(W^{-1}C)$
- La dernière ligne de $r^{-1}v((I-D')^{-1}W)$ vaut $(0, ..., 0, 1)$
- Donc $[0_{p\times(q-1)}|b_{p\times 1}].r^{-1}v((I-D')^{-1}W) = [0_{p\times(q-1)}|b_{p\times 1}]$
- $\overline{S}_E = A + [0_{p\times(q-1)}|b_{p\times 1}].r^{-1}v(W^{-1}C)$
- La dernière ligne de $r^{-1}v(W^{-1}C)$ peut être exprimée par $max_{i=1,q}r^{-1}v(W^{-1}C)[i,*]$
- $\overline{S}_E = A + b.\left(max_{i=1,q}r^{-1}v(W^{-1}C)[i,*]\right)^t$ qui coïncide avec la borne fournie par DPY.

Exemple

- $A = \begin{pmatrix} 0.1762 & 0.0825 & 0.0761 & 0.1120 \\ 0.0434 & 0.1154 & 0.1676 & 0.0511 \\ 0.1642 & 0.2143 & 0.0157 & 0.0538 \\ 0.1043 & 0.1979 & 0.0758 & 0.0033 \end{pmatrix}$
- $B = \begin{pmatrix} 0.1569 & 0.0932 & 0.0280 & 0.1196 & 0.1555 \\ 0.0985 & 0.1331 & 0.1309 & 0.1534 & 0.1066 \\ 0.0548 & 0.1160 & 0.1024 & 0.1786 & 0.1002 \\ 0.1443 & 0.0654 & 0.1847 & 0.0734 & 0.1509 \end{pmatrix}$
- $C = \begin{pmatrix} 0.1509 & 0.1250 & 0.1377 & 0.1264 \\ 0.2552 & 0.0590 & 0.0033 & 0.1491 \\ 0.0909 & 0.0808 & 0.0277 & 0.1856 \\ 0.0041 & 0.2055 & 0.0445 & 0.1023 \\ 0.1453 & 0.1621 & 0.0351 & 0.0740 \end{pmatrix}$

Exemple(suite)

- $D = \begin{pmatrix} 0.1419 & 0.0321 & 0.1445 & 0.0490 & 0.0925 \\ 0.0066 & 0.0648 & 0.1988 & 0.1143 & 0.1490 \\ 0.1357 & 0.1359 & 0.0989 & 0.1064 & 0.1383 \\ 0.0833 & 0.0665 & 0.1976 & 0.1597 & 0.1365 \\ 0.1471 & 0.0958 & 0.1453 & 0.0547 & 0.1406 \end{pmatrix}$
- $S_E = \begin{pmatrix} 0.3378 & 0.2492 & 0.1456 & 0.2675 \\ 0.2235 & 0.3003 & 0.2369 & 0.2394 \\ 0.3197 & 0.3848 & 0.0748 & 0.2207 \\ 0.2843 & 0.3758 & 0.1508 & 0.1891 \end{pmatrix}$

Exemple(suite)

- $\overline{T} = \begin{pmatrix} 0.1762 & 0.0825 & 0.0761 & 0.6652 \\ 0.0434 & 0.1154 & 0.1676 & 0.6736 \\ 0.1642 & 0.2143 & 0.0157 & 0.6059 \\ 0.1043 & 0.1979 & 0.0758 & 0.6220 \end{pmatrix}$
- $\overline{DPY} = \begin{pmatrix} 0.1825 & 0.3229 & 0.1158 & 0.3787 \\ 0.0505 & 0.3860 & 0.2124 & 0.3512 \\ 0.1705 & 0.4542 & 0.0553 & 0.3199 \\ 0.1114 & 0.4668 & 0.1202 & 0.3015 \end{pmatrix}$
- $\overline{S} = \begin{pmatrix} 0.2193 & 0.2945 & 0.1356 & 0.3506 \\ 0.0927 & 0.3513 & 0.2282 & 0.3279 \\ 0.2018 & 0.4284 & 0.0660 & 0.3038 \\ 0.1564 & 0.4288 & 0.1375 & 0.2773 \end{pmatrix}$

Exemple(suite)

- $r(\overline{T}) = \begin{pmatrix} 1.0000 & 0.8238 & 0.7413 & 0.6652 \\ 1.0000 & 0.9566 & 0.8412 & 0.6736 \\ 1.0000 & 0.8358 & 0.6215 & 0.6059 \\ 1.0000 & 0.8957 & 0.6977 & 0.6220 \end{pmatrix}$
- $r(\overline{DPY}) = \begin{pmatrix} 1.0000 & 0.8175 & 0.4945 & 0.3787 \\ 1.0000 & 0.9495 & 0.5635 & 0.3512 \\ 1.0000 & 0.8295 & 0.3753 & 0.3199 \\ 1.0000 & 0.8886 & 0.4218 & 0.3015 \end{pmatrix}$
- $r(\overline{S}) = \begin{pmatrix} 1.0000 & 0.7807 & 0.4862 & 0.3506 \\ 1.0000 & 0.9073 & 0.5560 & 0.3279 \\ 1.0000 & 0.7982 & 0.3699 & 0.3038 \\ 1.0000 & 0.8436 & 0.4147 & 0.2773 \end{pmatrix}$

Connaissance partielle de la matrice D

- $$D^0[i,j] = \begin{cases} D[i,j] & \textit{Si} \quad D[i,j] \text{ est connu} \\ 0 & \textit{Sinon} \end{cases}$$
- $0 \preceq_{el} D^0 \preceq_{el} D$.

Théorème

Supposons que l'on connaisse une matrice D^0 telle que $0 \preceq_{el} D^0 \preceq_{el} D$. On a alors

$$S_E \preceq_{st} A + B.r^{-1}v(H).r^{-1}v(W^{-1}C) \tag{13}$$

Où H est la matrice obtenue par l'algorithme de Haddad et Moreaux avec $L = (I - D^0)^{-1}.W$ et $U = 1_{q \times q}$.

Connaissance partielle de D

Algorithme 18 Algorithme pour la borne supérieure avec connaissance partielle de D

Entrées: A, B, C et D^0
Sorties: $\overline{S}_E$
  pour $i = 1$ to q **faire**
    $w[i] \leftarrow \sum_{j=1}^{p} C[i,j]$
  fin pour
  pour tout i et j **faire**
    $C[i,j] \leftarrow \frac{C[i,j]}{W[i]}$
  fin pour
  $C \leftarrow r^{-1}v(C)$
  $H \leftarrow (I - D^0)^{-1}$
  pour tout i et j **faire**
    $H[i,j] \leftarrow H[i,j].W[j]$
  fin pour
  $H \leftarrow algo15(H, 1_{q \times q})$
  $H \leftarrow r^{-1}v(H)$
  $\overline{S}_E \leftarrow A + B.H.C$

- Plus coûteux en temps de calcul que le précédent : $(I - D^0)^{-1}$, Algorithme de Haddad, algorithme de Vincent.
- $(I - D^0)^{-1} = I + \sum_{k=1}^{\Delta} (D^0)^k \Rightarrow$ algorithmes de graphes

Connaissance partielle de D

Algorithme 18 Algorithme pour la borne supérieure avec connaissance partielle de D

Entrées: A, B, C et D^0
Sorties: $\overline{S}_E$
  pour $i = 1$ **to** q **faire**
    $w[i] \leftarrow \sum_{j=1}^{p} C[i,j]$
  fin pour
  pour tout i et j **faire**
    $C[i,j] \leftarrow \frac{C[i,j]}{W[i]}$
  fin pour
  $C \leftarrow r^{-1}v(C)$
  $H \leftarrow (I - D^0)^{-1}$
  pour tout i et j **faire**
    $H[i,j] \leftarrow H[i,j].W[j]$
  fin pour
  $H \leftarrow algo15(H, 1_{q \times q})$
  $H \leftarrow r^{-1}v(H)$
  $\overline{S}_E \leftarrow A + B.H.C$

- Plus coûteux en temps de calcul que le précédent : $(I - D^0)^{-1}$, Algorithme de Haddad, algorithme de Vincent.
- $(I - D^0)^{-1} = I + \sum_{k=1}^{\Delta} (D^0)^k \Rightarrow$ algorithmes de graphes

Connaissance partielle de C

- construire C^0 comme on l'a fait avec D^0.

Théorème

Considérons les deux matrices C^0 et D^0 vérifiant $0 \preceq_{el} C^0 \preceq_{el} C$ et $0 \preceq_{el} D^0 \preceq_{el} D$. On a alors

$$S_E \preceq_{st} A + B.r^{-1}v(H) \tag{14}$$

Où H est la matrice obtenue par l'algorithme de Haddad avec $L = (I - D^0)^{-1}.C^0$ et $U = 1_{q \times p}$.

Connaissance partielle de C

- construire C^0 comme on l'a fait avec D^0.

Théorème

Considérons les deux matrices C^0 et D^0 vérifiant $0 \preceq_{el} C^0 \preceq_{el} C$ et $0 \preceq_{el} D^0 \preceq_{el} D$. On a alors

$$S_E \preceq_{st} A + B.r^{-1} v(H) \tag{14}$$

Où H est la matrice obtenue par l'algorithme de Haddad avec $L = (I - D^0)^{-1}.C^0$ et $U = 1_{q \times p}$.

Conclusion

- Schéma général de borne.
- Déduction de plusieurs algoritmes (Truffet, DPY, algorithmes de graphes)
- Possibilité de considérer d'autre cas particuliers
- outils principaux : les opérateurs : r, r^{-1} et v. les algorithmes de Vincent, Haddad et Moreaux