

Résolution Numérique de Chaînes de Markov

Jean-Michel Fourneau

PRiSM

Université de Versailles
45, Av. des Etats-Unis
F-78000 Versailles

email:jmf@prism.uvsq.fr

1

Hypothèses

Chaînes finies

Calcul de la distribution stationnaire π

Solution de :

$$\Pi Q = 0 \quad \Pi e = 1 \quad \text{En Temps Continu} \quad (1)$$

$$\Pi P = \Pi \quad \Pi e = 1 \quad \text{En Temps Discret} \quad (2)$$

On suppose que la solution existe

2

Plan

- Méthodes directes
- Méthodes itératives
- Méthodes récursives
- Méthodes par décomposition
- Lumpability (voir ailleurs, les SPA)
- Approche tensorielle (voir ailleurs, les SAN)

Uniformisation

On peut transformer les matrices pour passer d'un problème en temps continu en un problème en temps discret.

- Soit Q un générateur
- Posons $\delta = \max_{i=1..n} (|Q(i, i)|)$
- Soit $\epsilon \geq 0$
- Définissons $P_\epsilon = Id + Q(\delta + \epsilon)^{-1}$
- **Propriété 1** P_ϵ est une matrice stochastique qui a la même mesure invariante que Q .

Preuve :

1. P_ϵ admet Π comme mesure invariante

$$\Pi P_\epsilon = \Pi Id + \Pi Q(\delta + \epsilon)^{-1} = \Pi \quad (3)$$

2. Tous les éléments hors diagonaux de P_ϵ sont positifs car ils sont la somme d'éléments positifs.
3. Tous les éléments diagonaux de P_ϵ sont positifs : en effet,

$$-1 \leq \frac{Q(j, j)}{\epsilon + \max_i |Q(i, i)|} \leq 0 \quad (4)$$

4. Pour tout i :

$$\sum_j P_\epsilon(i, j) = \frac{1}{\epsilon + \delta} \sum_j Q(i, j) + \sum_j Id(i, j) = 0 + 1 = 1$$

Choix de ϵ

- Une valeur strictement positive de ϵ implique que P_ϵ a des éléments diagonaux non nuls; elle est donc apériodique.
- ϵ permet de changer le spectre de la matrice P_ϵ . Attention aux méthodes itératives
- Lorsque ϵ tend vers l'infini, toutes les valeurs propres de P_ϵ tendent vers 1 en module
- Prendre ϵ petit, heuristique de Wallace (solveur RQA)
 $\epsilon = \delta/100$.

Méthodes directes

- Résolution d'un problème linéaire
- Variante de l'élimination de Gauss
- Choisir la variante la plus précise, la complexité est toujours cubique pour des matrices pleines
- L'élimination a pour conséquence de remplir la matrice si elle était initialement creuse
- algorithme GTH : beaucoup plus stable numériquement car il n'y a ni nombres négatifs ni soustractions

7

GTH

- Du nom des ses auteurs : Grassman, Taksar et Heyman
- Idée : oter les sommets depuis le dernier
- Equilibre en n :

$$\Pi_n \left(\sum_{i=1}^{n-1} P(n, i) \right) = \sum_{j=1}^{n-1} P(j, n) \Pi_j \quad (5)$$

- Equation d'équilibre d'un sommet a successeur de n dans la chaine de Markov

$$\Pi_a \left(\sum_{i=1}^n P(a, i) \right) = \sum_{j=1}^{n-1} P(j, a) \Pi_j + P(n, a) \Pi_n \quad (6)$$

- Remplaçons Π_n par sa valeur.

8

$$\Pi_a \left(\sum_{i=1}^n P(a, i) \right) = \sum_{j=1}^{n-1} P(j, a) \Pi_j + \frac{\sum_{j=1}^{n-1} P(j, n) P(n, a) \Pi_j}{\sum_{i=1}^{n-1} P(n, i)} \quad (7)$$

- Posons $R(j, a) = P(j, a) + \frac{P(j, n) P(n, a)}{\sum_{i=1}^{n-1} P(n, i)}$.
- R est une matrice stochastique de taille $n - 1$
- Il est possible que $P(j, a)$ soit nul et que $R(j, a)$ ne le soit pas. Il suffit qu'il y ait, dans la chaîne de Markov, une transition de j à n et une autre de n à a .
- R est moins dense que la matrice P et on doit modifier la structure des éléments non nuls de P pour construire R .
- Le choix du sommet à éliminer est important pour minimiser le remplissage de la matrice

Code de GTH

```

program gth(input, output);
begin
  for n:=nr downto 1 do begin
    S:=0;
    for j:=0 to n-1 do S:=S+P[n, j];
    for i:=0 to n-1 do P[i, n]:=P[i, n]/S;
    for i:=0 to n-1 do for j:=0 to n-1 do
      P[i, j]:=P[i, j]+P[i, n]*P[n, j];
      if S<=0.0 then writeln('erreur', n) else writeln(n);
    end;
    Tot:=1;
    pi[0]:=1;
    for j:=1 to nr do begin
      pi[j]:=P[0, j];
      for k:=1 to j-1 do pi[j]:=pi[j]+pi[k]*P[k, j];
      Tot:=Tot+pi[j];
    end;
    for j:=0 to nr do pi[j]:=pi[j]/Tot;
  end gth.

```

Comparaison GTH-QR

- En simple précision...
- Le résultat "exact" est fourni par un algorithme en double précision
- On calcule les erreurs absolues sur GTH et les erreurs absolues min et max pour QR (ça dépend de la colonne enlevée)

Exemple : matrice petite, probabilités fortes

$$P = \begin{pmatrix} 0.2 & 0 & 0 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0.2 \\ 0 & 0.1 & 0 & 0 & 0.6 & 0 & 0.3 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0.1 \\ 0 & 0 & 0.6 & 0 & 0.3 & 0 & 0 & 0 & 0 & 0.1 \\ 0 & 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0.3 & 0 \\ 0 & 0 & 0 & 0 & 0.7 & 0 & 0.2 & 0 & 0 & 0.1 \\ 0.1 & 0 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0 & 0.1 \\ 0 & 0.4 & 0 & 0 & 0 & 0.4 & 0 & 0 & 0 & 0.2 \end{pmatrix} \quad (8)$$

erreur-GTH	erreur-min-QR	erreur-max-QR
4.510^{-8}	6.910^{-8}	3.710^{-6}

Autre exemple : petite matrice mais probabilités assez faibles...

$$P = \begin{pmatrix} 1 - 10^{-6} & 10^{-7} & 2 \cdot 10^{-7} & 3 \cdot 10^{-7} & 4 \cdot 10^{-7} \\ 0.4 & 0.3 & 0 & 0 & 0.3 \\ 5 \cdot 10^{-7} & 0 & 1 - 10^{-6} & 0 & 5 \cdot 10^{-7} \\ 5 \cdot 10^{-7} & 0 & 0 & 1 - 10^{-6} & 5 \cdot 10^{-7} \\ 2 \cdot 10^{-7} & 3 \cdot 10^{-7} & 10^{-7} & 4 \cdot 10^{-7} & 1 - 10^{-6} \end{pmatrix} \quad (9)$$

erreur-GTH	erreur-min-QR	erreur-max-QR
$3.1 \cdot 10^{-8}$	$6.12 \cdot 10^{-3}$	$3.89 \cdot 10^{-2}$

Conclusions-méthodesdirectes

- Avantages
 - Complexité fixe indépendante des données
 - Stabilité numérique (GTH)
 - Facile à implémenter
- Inconvénients
 - Complexité cubique en temps, quadratique en espace pour les matrices pleines
 - Peu adaptées aux structures creuses
- Pour des matrices de taille inférieure à 1000

Méthodes itératives

- Résolution d'un problème de vecteur propre
- Variantes de la méthode des puissances

$$x^{(k+1)} = \frac{1}{\lambda_1} x^{(k)} A \quad (10)$$

Méthode des Puissances

- Soit $(\lambda_1, \lambda_2, \dots, \lambda_n)$ les vecteurs propres à gauche de la matrice A , ordonnées par ordre décroissant en module
- Supposons $|\lambda_1| > |\lambda_2|$
- Les vecteurs propres (y_1, \dots, y_n) forment une base; exprimons $x^{(0)}$ dans cette base.

$$x^{(0)} = \sum_{i=1}^n \alpha_i y_i \quad (11)$$

On suppose que $\alpha_1 \neq 0$.

- puisque λ_i est valeur propre associée à y_i , on a :

$$x^{(1)} = \frac{1}{\lambda_1} x^{(0)} A = \frac{1}{\lambda_1} \sum_{i=1}^n \alpha_i y_i A = \frac{1}{\lambda_1} \sum_{i=1}^n \alpha_i \lambda_i y_i \quad (12)$$

Plus généralement :

$$x^{(k+1)} = \left(\frac{1}{\lambda_1}\right)^k x^{(0)} A^k = \sum_{i=1}^n \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^k y_i \quad (13)$$

On sépare le premier terme :

$$x^{(k+1)} = \alpha_1 y_1 + \left(\sum_{i=2}^n \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^k y_i\right) \quad (14)$$

- Puisque toutes les valeurs propres à partir de la seconde sont strictement inférieures à λ_1 , tous les termes de la somme tendent vers 0.
- L'itération a pour limite un vecteur propre de A associé à λ_1 .
- Mais la vitesse de convergence dépend des autres valeurs propres et surtout de la seconde.

Application aux chaînes de Markov

- Itérer :

$$\Pi^{(n+1)} = \Pi^{(n)} P \quad (15)$$

- Inutile de normaliser car $\lambda_1 = 1$
- temps de convergence trop long : améliorations nécessaires

Améliorer

- pour résoudre $xA = b$ par des itérations successives,
- si on peut trouver une décomposition de A en $M - N$ où M est facilement inversible, alors

$$xM = xN + b \quad (16)$$

et grâce à l'inversibilité de M ,

$$x = xNM^{-1} + bM^{-1} \quad (17)$$

- Ce qui conduit au schéma itératif simple :

$$x^{(n+1)} = x^{(n)}NM^{-1} + bM^{-1}$$

- La matrice $H = NM^{-1}$ est la matrice de l'itération.

- Pour la suite, on cherche à résoudre le problème transposé et en temps continu, c'est à dire :

$$Q^t \Pi^t = 0 \quad (18)$$

- On suppose que le générateur Q^t est décomposé en une matrice diagonale D , une matrice triangulaire inférieure L et une matrice triangulaire supérieure U .

$$Q^t = D - (L + U) \quad (19)$$

Jacobi

- $M = D$ et $N = L + U$
- $H = D^{-1}(L + U)$
- Sous forme scalaire:

$$x_i^{(n+1)} = \frac{1}{d_{i,i}} \left\{ \sum_{j \neq i} (l_{i,j} + u_{i,j}) x_j^{(n)} \right\} \quad (20)$$

Gauss-Seidel

- Les valeurs de la nouvelle itération $x_j^{(n+1)}$ pour $j < i$ peuvent être utilisés à la place des valeurs de l'itération (n) , pour calculer $x_i^{(n+1)}$.
- Opérateur $H = (D - L)^{-1}U$
- Iteration

$$x_i^{(n+1)} = \frac{1}{d_{i,i}} \left\{ \sum_{j < i} l_{i,j} x_j^{(n+1)} + \sum_{j > i} u_{i,j} x_j^{(n)} \right\} \quad (21)$$

- Gauss Seidel ascendante (ci-dessus) ou descendante (H vaut $(D - U)^{-1}L$)
- La vitesse de convergence peut dépendre de l'ordre des variables.

- La méthode de relaxation (SOR) consiste à effectuer, pour l'itération $(n + 1)$, une moyenne pondérée entre le résultat de l'itération (n) et l'application de l'itération de Gauss Seidel
- Itération :

$$x_i^{(n+1)} = (1 - \omega)x_i^{(n)} + \omega \left[\frac{1}{d_{i,i}} \left\{ \sum_{j < i} l_{i,j} x_j^{(n+1)} + \sum_{j > i} u_{i,j} x_j^{(n)} \right\} \right] \quad (22)$$

- Opérateur : $H_w = (D - \omega L)^{-1}((1 - \omega)D + \omega U)$
- Choisir une valeur correcte de ω est un problème difficile.
- ω peut modifier la valeur propre dominante de l'opérateur H_w

```

program GaussSeidel(input,output);
type element = record ori : integer; prob : real; end;
var nz,n,i,c,j, degre,iter : integer; diff,som,y : real;
p,op : array[0..nodemax] of real;
debut : array[0..nodemax] of integer;
arc : array[0..arcmax] of element;
begin
  readln(nz); readln(n); c:=0;
  for i:=0 to n-1 do begin
    read(degre); debut[i]:=degre;
    for j:=1 to degre do begin
      read(arc[c].prob); read(arc[c].ori); c:=c+1;
    end;
    readln; op[i]:=1;
  end;
  iter:=0; som:=0.0; for i:=0 to n-1 do som:=som+op[i];
  for i:=0 to n-1 do op[i]:=op[i]/som;
  repeat
    iter:=iter+1; c:=0;
    for i:=0 to n-1 do begin
      som:=0.0;
      for j:=c to c+debut[i]-1 do begin
        if arc[j].ori > i then y:= op[arc[j].ori] else y:= p[arc[j].ori];
        som:=som+op[arc[j].ori]*arc[j].prob;
      end;
      p[i]:=som; c:=c+debut[i];
    end;
    som:=0.0; for i:=0 to n-1 do som:=som+p[i]; for i:=0 to n-1 do p[i]:=p[i]/som;
    diff:=0.0; for i:=0 to n-1 do diff:=diff+abs(p[i]-op[i]);
    op:=p;
  until (iter=maxiter) or (diff<maxdiff);
  for i:=0 to n-1 do writeln(p[i]);
end.

```

Comparaisons

- le coût d'une iteration est à peu près le même
- le nombre d'itérations dépend du spectre de H mais on ne sait pas comment il a été changé.

25

Exemple 1 : Pb de Courtois

$$P = \begin{bmatrix} 0.85 & 0 & 0.149 & 9 \cdot 10^{-4} & 0 & 5 \cdot 10^{-5} & 0 & 5 \cdot 10^{-5} \\ 0.1 & 0.65 & 0.249 & 0 & 9 \cdot 10^{-4} & 5 \cdot 10^{-5} & 0 & 5 \cdot 10^{-5} \\ 0.1 & 0.8 & 0.0996 & 3 \cdot 10^{-4} & 0 & 0 & 10^{-4} & 0 \\ \hline 0 & 4 \cdot 10^{-4} & 0 & 0.7 & 0.2995 & 0 & 10^{-4} & 0 \\ 5 \cdot 10^{-4} & 0 & 4 \cdot 10^{-4} & 0.39 & 0.6 & 10^{-4} & 0 & 0 \\ \hline 0 & 5 \cdot 10^{-5} & 0 & 0 & 5 \cdot 10^{-5} & 0.6 & 0.2499 & 0.15 \\ 3 \cdot 10^{-5} & 0 & 3 \cdot 10^{-5} & 4 \cdot 10^{-5} & 0 & 0.1 & 0.8 & 0.0999 \\ 0 & 5 \cdot 10^{-5} & 0 & 0 & 5 \cdot 10^{-5} & 0.1999 & 0.25 & 0.55 \end{bmatrix}$$

Méthode	Itérations pour 10^{-6}
Puissance	69000
Jacobi	23000
Gauss Seidel	11000
SOR opt	3500

C'est un problème NCD...

26

$$\lambda_1 = 0.2, \lambda_2 = 30.0, \mu_1 = 0.5, \mu_2 = 60.0$$

Méthode	2ème valeur propre	Itérations pour 10^{-6}
Puissance	0.9942	2375
Jacobi	1.0	Diverge
Gauss Seidel	0.9827	790
SOR opt	0.7677	53

C'est un problème NCD...

Initialisation

- Ne pas initialiser le vecteur de probabilités par un 1.0 pour le plus petit état dans une méthode de Gauss Seidel ascendante.
- Employer une approximation pour choisir un vecteur initial proche de la solution

Conclusions-méthodes itératives

- Avantages
 - La complexité d'une itération est le nombre d'élément non nuls de P
 - Stabilité numérique (on ne fait que des additions)
 - adaptées aux matrices creuses
- Inconvénients
 - La complexité en nombre d'itérations dépend du spectre (et donc des données)
- Pour des matrices creuses de grande taille dont la seconde valeur propre est faible

Méthode par Récursion

- Construire un système récursif sur les probabilités de manière à les calculer à un facteur multiplicatif près qui est obtenu par normalisation.
- Dépend de la structure de la matrice, pas des valeurs numériques distinctes de 0.

Matrices Hessenberg

- Hessenberg Supérieure : triangle supérieur + sous diagonale principale; Chaîne Inclusive aux instants de départ de la G/M/1

$$\begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot \end{bmatrix}$$

- Hessenberg Inférieure : triangle inférieur + sur diagonale principale; Chaîne Inclusive aux instants d'arrivées de la M/G/1

$$\begin{bmatrix} \cdot & \cdot & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

Principe pour une Hessenber sup

- Equilibre en j :

$$\pi_j = \sum_{i=0}^{j+1} \pi_i P_{i,j}$$

- On en déduit π_{j+1}

$$\pi_{j+1} = \frac{1}{P_{j+1,j}} \left[\pi_j (1 - P_{j,j}) - \sum_{i=0}^{j-1} \pi_i P_{i,j} \right]$$

Si $P_{j+1,j}$ est non nul.

- Cas fini : on pose $\pi_0 = 1$, on effectue la récurrence, on somme toutes les probabilités et on normalise.
- Cas Infini : il faut que P soit régulière.

Matrices Block Hessenberg

- Même structure que les matrices Hessenberg mais par bloc.

$$Q = \begin{bmatrix} Q_{0,0} & Q_{0,1} & \cdot & \cdot & \cdot \\ Q_{1,0} & Q_{1,1} & Q_{1,2} & \cdot & \cdot \\ 0 & Q_{2,1} & Q_{2,2} & \cdot & \cdot \\ 0 & 0 & Q_{3,2} & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot \end{bmatrix}$$

- $Q_{i,j}$ blocs de taille $k \times k$
- π divisé en sous vecteurs de taille k

35

Algorithme

- Equilibre pour le bloc 0

$$\pi_0 Q_{0,0} + \pi_1 Q_{1,0} = 0$$

- Si $Q_{1,0}$ est inversible

$$\pi_1 = -\pi_0 Q_{0,0} Q_{1,0}^{-1}$$

- Equilibre pour le bloc 1

$$\pi_0 Q_{1,0} + \pi_1 Q_{1,1} + \pi_2 Q_{2,1} = 0$$

- Si $Q_{2,1}$ est inversible

$$\pi_2 = (\pi_0 Q_{1,0} + \pi_1 Q_{1,1}) Q_{2,1}^{-1}$$

36

- Après simplification

$$\pi_2 = \pi_0(Q_{1,0} + Q_{0,0}Q_{1,0}^{-1}Q_{1,1})Q_{2,1}^{-1}$$

- On peut continuer... si $Q_{i+1,i}$ est inversible (récursion descendante)
- Si lower block Hessenberg et $Q_{i,i+1}$ inversible, alors récursion ascendante
- Si bloc tridiagonal, on peut essayer les deux sens de récursion.

Exemple : M/C₂/1/N

- Arrivées Poisson λ , Service Cox à deux phases (μ_1, μ_2, a)
- Etats (x, y) , x : nombre de clients, y : phase du service.
- $(0, 2)$ état transitoire. N places dans la file

*	0	λ	0			
0	*	0	λ			
$(1-a)\mu_1$	0	*	$a\mu_1$	λ	0	
μ_2	0	0	*	0	λ	
		$(1-a)\mu_1$	0	*	$a\mu_1$	λ
		μ_2	0	0	*	λ
				$(1-a)\mu_1$	0	*
				μ_2	0	*

- Blocs sousdiagonaux non inversibles,
- Blocs surdiagonaux inversibles
- Analyse ascendante

Calcul de $\pi(0, 1)$

$\lambda = 0.5, \mu_1 = 1.0, \mu_2 = 10.0$

N	Calcul Récursif	GTH
2	0.0007158	idem
5	0.0006958	idem
10	0.0006955	idem
16	0.0006808	0.0006955
20	-0.1973	0.0006955

Une probabilité négative ????

Résidus πQ

N	Résidus
2	10^{-16}
3	10^{-15}
5	10^{-13}
10	10^{-9}
12	10^{-7}
16	10^{-3}
20	50.9

Croissance géométrique avec N des erreurs

Raison de l'instabilité

- $\lambda < \mu_1$ et $\lambda \ll \mu_2$
- Les blocs $Q_{i,i} \begin{bmatrix} 1/\lambda & 0 \\ 0 & 1/\lambda \end{bmatrix}$ et $Q_{i-1,i} \begin{bmatrix} 1/\lambda & 0 \\ 0 & 1/\lambda \end{bmatrix}$ ont une norme très grande
- A chaque itération, les erreurs sont multipliées par cette norme.

Conclusions-méthodes récursives

- Avantages
 - Utilisation de la structure de la matrice
 - Problèmes de très grande taille
 - associée à des files usuelles
- Inconvénients
 - Problème de stabilité numérique
- Pour des matrices creuses de grande taille quasi Hessenberg

Méthodes de Décomposition

Définition 1 *une matrice stochastique P est presque complètement décomposable (NCD) ssi on peut décomposer la matrice en blocs $P_{i,j}$ tels que :*

1. $\|P_{i,i}\|_1 = O(1), \forall i$
2. $\|P_{i,j}\|_1 = O(\epsilon), \forall i, j, i \neq j$
3. $\epsilon \ll 1$

$$P = \begin{pmatrix} P_{1,1} & P_{1,2} & P_{1,3} & P_{1,4} & P_{1,5} \\ P_{2,1} & P_{2,2} & P_{2,3} & P_{2,4} & P_{2,5} \\ P_{3,1} & P_{3,2} & P_{3,3} & P_{3,4} & P_{3,5} \\ P_{4,1} & P_{4,2} & P_{4,3} & P_{4,4} & P_{4,5} \\ P_{5,1} & P_{5,2} & P_{5,3} & P_{5,4} & P_{5,5} \end{pmatrix} \quad (24)$$

Approche de Courtois

- Solutions locales
- Matrice de couplage
- Approximations

Solutions Locales de $P_{i,i}$

- Chercher u_i le vecteur propre normalisé à 1 et associé à la valeur propre dominante de $P_{i,i}$.

$$u_i P_{i,i} = \lambda_i u_i$$

- Cette valeur propre est inférieure à 1 mais très proche de 1.
- $u_i(j)$ est une approximation de la probabilité conditionnelle d'être dans le sommet d'indice j sachant que l'on est dans le bloc i .

Couplage

- A^* : Matrice de couplage entre bloc : probabilité d'aller du bloc i au bloc j
- Etape 1 : sommer les colonnes d'un bloc pour obtenir la probabilité d'aller d'un état vers un bloc : $P_{i,j}e$
- Etape 2 : pondérer les lignes par les poids des $u_i(k)$ (probabilité d'être dans l'état k quand on est dans le bloc i)

$$A^*(i, j) = u_i P_{i,j}e$$

- $\xi(i)$ est la probabilité stationnaire de la matrice A^*

$$\xi A^* = \xi$$

Approximation NCD

- l'état x correspond à l'état k de la composante i
- $u_i(k)$ approxime la probabilité d'être dans l'état k sachant que l'on est dans la composante i
- $\xi(i)$ est la probabilité d'être dans la composante i

$$\pi(x) = u_i(k)\xi(i)$$

Algorithme MKS

- Itérer l'approche de Courtois
- Mais agréger les probabilités π calculée par l'équation de décomposition redonne la quantité $u_i(\cdot)$ précédemment calculée.
- L'algorithme modifie π par une simple itération de la méthode de Gauss-Seidel
- Inutile de calculer les solutions locales, il suffit d'initialiser par une distribution uniforme

1. Choisir une solution initiale $\pi^{(0)}(x, y)$. x désigne le numéro de l'agrégat, y le numéro de l'état dans l'agrégat. m désigne le compteur d'itérations. Il est initialisé à 0.
2. Calculer les probabilités conditionnelles $u_i(j)$ d'être dans l'état y sachant que l'on est dans l'agrégat x .

$$u_i(j)^{(m)} = \frac{\pi^{(m)}(i, j)}{\sum_k \pi^{(m)}(i, k)} \quad (25)$$

3. Construire la nouvelle matrice de couplage $A^{(m)}$:

$$A^{(m)}(i, j) = u_i^{(m)} P_{i,j} e \quad (26)$$

4. Calculer la mesure invariante $\xi^{(m)}$ de $A^{(m)}$.
5. Calculer la nouvelle approximation de la distribution stationnaire

$$z^{(m)}(x) = u_i(j)^{(m)} \xi(i)^{(m)} \quad \text{avec } x = (i, j) \quad (27)$$

6. Perturber $z^{(m)}$ pour obtenir $\pi^{(m)}$ par une itération de Gauss Seidel

$$\pi^{(m)}(k) = \pi^{(m)}(k) P_{k,k} + \sum_{j < k} \pi^{(m)}(j) P_{j,k} + \sum_{j > k} z^{(m)}(j) P_{j,k}$$

7. Vérifier la convergence et dans le cas négatif incrémenter m et recommencer à l'étape 2

Pb de Courtois

$$P = \left[\begin{array}{ccc|cc|ccc} 0.85 & 0 & 0.149 & 9 \cdot 10^{-4} & 0 & 5 \cdot 10^{-5} & 0 & 5 \cdot 10^{-5} \\ 0.1 & 0.65 & 0.249 & 0 & 9 \cdot 10^{-4} & 5 \cdot 10^{-5} & 0 & 5 \cdot 10^{-5} \\ 0.1 & 0.8 & 0.0996 & 3 \cdot 10^{-4} & 0 & 0 & 10^{-4} & 0 \\ \hline 0 & 4 \cdot 10^{-4} & 0 & 0.7 & 0.2995 & 0 & 10^{-4} & 0 \\ 5 \cdot 10^{-4} & 0 & 4 \cdot 10^{-4} & 0.39 & 0.6 & 10^{-4} & 0 & 0 \\ \hline 0 & 5 \cdot 10^{-5} & 0 & 0 & 5 \cdot 10^{-5} & 0.6 & 0.2499 & 0.15 \\ 3 \cdot 10^{-5} & 0 & 3 \cdot 10^{-5} & 4 \cdot 10^{-5} & 0 & 0.1 & 0.8 & 0.0999 \\ 0 & 5 \cdot 10^{-5} & 0 & 0 & 5 \cdot 10^{-5} & 0.1999 & 0.25 & 0.55 \end{array} \right]$$

Approximation de Courtois : l'approximation est de l'ordre de 10^{-4} .

$$u_1 = (0.401, 0.416, 0.182), u_2 = (0.5714, 0.4286),$$

$$u_3 = (0.240, 0.555, 0.204)$$

$$A^* = \begin{bmatrix} 0.99911 & 0.00079 & 0.0001 \\ 0.00061 & 0.99929 & 0.0001 \\ 0.00006 & 0.00004 & 0.9999 \end{bmatrix}$$

Algorithme MKS : en 4 itérations les résidus sont de l'ordre de 10^{-16} .

Conclusions-méthodes de décomposition

- Avantages
 - Traitement efficaces de problèmes de grande taille
 - Adaptées aux matrices NCD
 - Problèmes réalistes (plusieurs échelles de temps)
- Inconvénients
 - Détection de la structure NCD (algorithme de connexité)
- Pour des matrices creuses de grande taille NCD

Bibliographie

- Introduction to numerical solution of Markov chains, W.J. Stewart, Princeton University Press, 1994