Censored Markov Chains

J.M. Fourneau

Laboratoire PRiSM, CNRS UMR 8144

Université de Versailles St-Quentin and INRIA Mescal

[a] A partir des résultats de N. Pekergin, S. Younès, T. Dayar, L. Truffet beaucoup d'autres, et un peu moi

Un peu de vocabulaire

- Chaine de Markov en temps discret: DTMC = (une distribution initiale, et une matrice stochastique)
- Espace d'états (fini, infini)
- Matrice stochastique: matrice positive ($P[i,j] \geq 0$) avec $\sum_j P[i,j] = 1$ pour tout i.
- Type de pb: existence d'un équilibre unique, distribution à l'équilibre, existence de plusieurs régimes, temps avant absorption, probabilité d'absorption.
- Les réponses dépendent de propriétés structurelles (finitude, irréductibilité) ou numériques (sur la matrice et la distribution initiale).

Basic Ideas for Censoring

- Consider a DTMC X with stochastic matrix Q and state space $\mathcal{S}$.
- Consider a partition of the state space into (E, E^c) and the associated block representation for Q:

$$Q = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

- The censored Markov Chain (CMC) only matches the chain when its state is in E (also known as watched Markov chain (Levy 57)).
- Not proved (but helpful) the CMC is associated to the chain where the uncensored states have immediate transitions (we must prove that...).

Basic...

- Transition matrix of the CMC:

$$S_A = A + B\left(\sum_{i=0}^{\infty} D^i\right) C$$

- Not completely true.... OK if the chain is ergodic.
- If the chain is finite but not ergodic, all the states in E^c must be transient (no recurrent class or absorbing states).
- If the chain is infinite and not ergodic, some work is necessary.

Censored Chains and St-St Analysis

- The ste

Transient Problems

- Assumptions: the chain is finite and contains several absorbing states which are all censored. Let the initial distribution be π_0.
- Property: Assume that $\sum_{i \in E} \pi_0(i) = 1$. Assume also that the states which immediately precede absorbing states are also in E. The absorbing probabilities in the CMC are equal to the absorbing probabilities of the original chain.
- Proof: Algebraic. Remember that when we have a block decomposition of a transition matrix with absorbing states:
  $$\left[\begin{array}{c|c} Id & 0 \\ \hline F & H \end{array}\right],$$
  matrix $M = (Id - H)^{-1}$ exists and is called the fundamental matrix. The entry $[i, j]$ of the product matrix $M * F$ gives the absorption probability in j knowing that the initial state is i.

Proof

- Gather the absorbing states at the beginning of E.

$$\left[\begin{array}{c|c||c} Id & 0 & 0 \\ \hline R & A & B \\ \hline\hline 0 & C & D \end{array}\right]$$

- The transition matrix of the censored chain is:

$$\left[\begin{array}{c|c} Id & 0 \\ \hline R & A \end{array}\right] + \left[\begin{array}{c} 0 \\ \hline B \end{array}\right] \sum_i [D]^i \left[\begin{array}{c|c} 0 & C \end{array}\right]$$

which is finally equal to: $\left[\begin{array}{c|c} Id & 0 \\ \hline R & A + B\sum_i D^i C \end{array}\right]$.

- As D is transient, we have: $\sum_i D^i = (Id - D)^{-1}$. And the fundamental matrix of the censored chain is:

$$(Id - A - B(Id - D)^{-1}C)^{-1}.$$

Proof

- The fundamental matrix of the initial chain is:

$$M = \left[\begin{array}{c|c} Id - A & B \\ \hline C & Id - D \end{array}\right]^{-1}.$$

- To obtain the probability we must multiply by $\left[\begin{array}{c} R \\ \hline 0 \end{array}\right]$ and consider an initial state in E.
- Thus we only have to compute the upper-left block of F which is equal to:

$$(Id - A - B(Id - D)^{-1}C)^{-1}$$

if blocks $(Id - A)$ and its Schur complements are non singular.

- we have the same absorption probability in Q and in S_A.

Transient Problems II

- Same Assumptions.
- The expectation of the first passage time (or absorbing time) in CMC are smaller than the expectation of these times in the original chain.

  Proof: Algebraic. Same proof. Remember that the average number of visits in j when the initial state is i is entry $[i, j]$ of the fundamental matrix.
- Conjecture: the first passage time (or absorbing time) in CMC are stochastically smaller than these times in the original chain. [A direct consequence in the model with 0-time delays for uncensored states]

Truncated Solution

- Truncated st-st solution: the st-st distribution for the censored process is the initial solution with an appropriate normalization (see Kelly for truncation of reversible processes).
- **Theorem 1** *the CMC has a truncated st-st solution.*
- Proof: algebraic.
- Does not change the structure of the solution: Product form for the DTMC implies Product Form for the CMC (false in continuous-time).
- If the reward is the ratio of two homogenous polynomials of degree k on the steady-state distribution of states in E, the reward has the same value on the DTMC and on the CMC.

Numerical Computation

- The augmentation problem for infinite MC: adding appropriate probabilities to A such that the st-st distribution of the augmented chain converges to the original one (Seneta 67, Wolf, Heyman, Freedman)
- Censoring is the best method to approximate an infinite MC (in some sense) (Zhao, Liu).
- $(B(I-D)^{-1}C)(i,j)$ is the taboo probability of the paths from i in E to j in E which are not allowed to visit E in between.

Numerical Computation of Bounds

- Computing bounds rather than exact results.
- Stochastic bounds (not component-wise bounds).
- With complete state space, we use lumpability to reduce the state-space (Truffet) or Patterns to simplify the structure of the chain (Busic).
- With censoring, we compute bounds with only a small part of the state space.

Comparison for Markov Chains

- Monotonicity and comparability of the transition probability matrices yield su□cient conditions for the stochastic comparison of MC.
- $P_{i,*}$ is row i of P.
- **Definition 1 (st-Comparison of Stochastic Matrices)** *Let P and Q be two stochastic matrices. $P \leq_{st} Q$ if and only if $P_{i,*} \leq_{st} Q_{i,*}$ for all i.*
- **Definition 2** *Let P be a stochastic matrix, P is st-monotone if and only if for all i, $j > i$, we have $P_{i,*} \leq_{st} P_{j,*}$*

Examples

- $\begin{bmatrix} 0.1 & 0.2 & 0.6 & 0.1 \\ 0.1 & 0.1 & 0.2 & 0.6 \\ 0.0 & 0.1 & 0.3 & 0.6 \\ 0.0 & 0.0 & 0.1 & 0.9 \end{bmatrix}$ is monotone.
- $\begin{bmatrix} 0.1 & 0.2 & 0.6 & 0.1 \\ 0.2 & 0.1 & 0.1 & 0.6 \\ 0.0 & 0.1 & 0.3 & 0.6 \\ 0.1 & 0.0 & 0.1 & 0.8 \end{bmatrix}$ is not monotone.

Vincent's Algorithm

- It is possible to use a set of equalities, instead of inequalities:

$$\begin{cases} \sum_{k=j}^{n} Q_{1,k} & = \sum_{k=j}^{n} P_{1,k} \\ \sum_{k=j}^{n} Q_{i+1,k} & = max(\sum_{k=j}^{n} Q_{i,k}, \sum_{k=j}^{n} P_{i+1,k}) \quad \forall\ i, j \end{cases}$$

- Properly ordered (in increasing order for i and in decreasing order for j in previous system), a constructive way to obtain a stochastic bound (Vincent's algorithm).
- Written as $V = r\bar{t}^{1}$

An example

$$P1 = \begin{bmatrix} 0.5 & 0.2 & 0.1 & 0.2 & 0.0 \\ 0.1 & 0.7 & 0.1 & 0.0 & 0.1 \\ 0.2 & 0.1 & 0.5 & 0.2 & 0.0 \\ 0.1 & 0.0 & 0.1 & 0.7 & 0.1 \\ 0.0 & 0.2 & 0.2 & 0.1 & 0.5 \end{bmatrix}$$

$$V(P1) = \begin{bmatrix} 0.5 & 0.2 & 0.1 & 0.2 & 0.0 \\ 0.1 & 0.6 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.2 & 0.5 & 0.1 & 0.1 \\ 0.1 & 0.0 & 0.1 & 0.7 & 0.1 \\ 0.0 & 0.1 & 0.1 & 0.3 & 0.5 \end{bmatrix}.$$

Bounds for the CMC

- Avoid to build the whole chain.
- Assume that we build block A by a BFS from an initial state 00.
- Possible to find monotone upper and lower bound for S_A (Tru�et). Proved optimal if we only know A.
- Improve Tru�et's bound if we build A and C (Dayar, Pekergin, Younnes). Conjectured to be optimal if we know A and C
- Improve Tru�et's bound if we build A and some columns of C (less accurate than DPY but needs less informat

Truffet's Algorithm to bound S_A

- Only use block A.
- 2 steps:
  - Compute of a stochastic upper bound of S_A (operator $T()$): add the slack probability in the last column of A.
  - Make it st-monotone (Vincent's algorithm) (operator $V()$).
- Simple, but needs to obtain something more accurate.
- A lower bound is obtained when we add the slack probability to the first column of A.

Example

$$Q = \left[\begin{array}{ccc|cc} 0.2 & 0.3 & 0.2 & 0.2 & 0.1 \\ 0.4 & 0.2 & 0.2 & 0.0 & 0.2 \\ 0.2 & 0.3 & 0.3 & 0.1 & 0.1 \\ \hline 0.1 & 0.2 & 0.2 & 0.3 & 0.2 \\ 0.0 & 0.3 & 0.3 & 0.3 & 0.1 \end{array}\right] \quad SlackProbability = \left[\begin{array}{c} 0.3 \\ 0.2 \\ 0.2 \end{array}\right]$$

$$T(A) = \left[\begin{array}{ccc} 0.2 & 0.3 & \mathbf{0.5} \\ 0.4 & 0.2 & \mathbf{0.4} \\ 0.2 & 0.3 & \mathbf{0.5} \end{array}\right] \quad V(T(A)) = \left[\begin{array}{ccc} 0.2 & 0.3 & \mathbf{0.5} \\ 0.2 & 0.3 & \mathbf{0.5} \\ 0.2 & 0.3 & \mathbf{0.5} \end{array}\right]$$

$$S_A = \left[\begin{array}{ccc} 0.23 & 0.43 & 0.33 \\ 0.41 & 0.29 & 0.29 \\ 0.22 & 0.38 & 0.38 \end{array}\right]$$

$$S_A \leq_{st} T(A) \leq_{st} V(T(A))$$

DPY

- My own presentation...
- Assume that one must compute a matrix M such that $M1 \leq_{st} M$ and $M2 \leq_{st} M$.
- $M1 \leq_{st} M2$ is equivalent to $r(M1) \leq_{el} r(M)$. And we also have: $r(M2) \leq_{el} r(M)$.
- Thus $max(r(M1), r(M2)) \leq_{el} r(M)$. Or $r^{-1}(max(r(M1), r(M2))) \leq_{st} M$
- Easily generalized to n matrices = StMax(M1,M2,...Mn)

DPY

- Turn back to the CMC and its matrix $S_A = A + Z$, $Z = B(I-D)^{-1}C$.
- Define G as $G(i,j) = C(i,j)/\sum_k C(i,k)$: normalization of C
- Define G_k as matrix whose rows are all equal to row k of G
- DPY: Define $U = \vec{\beta} StMax(G_1, G_2, ..., G_n)$,
- **Theorem 2** *$A + U$ is a st-bound of S_A.*
- U has rank 1.

Examples

$$A = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.1 \\ 0.1 & 0.4 & 0.2 & 0.0 \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.2 & 0.0 & 0.4 & 0.0 \end{bmatrix}, \vec{\beta} = \begin{bmatrix} 0.3 \\ 0.3 \\ 0.0 \\ 0.4 \end{bmatrix}, C = \begin{bmatrix} 0.1 & 0.0 & 0.1 & 0.2 \\ 0.0 & 0.1 & 0.0 & 0.0 \\ 0.2 & 0.2 & 0.1 & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.1 & 0.0 & 0.0 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix}.$$

Normalized Matrix :

$$G = \begin{bmatrix} 0.25 & 0.0 & 0.25 & 0.5 \\ 0.0 & 1 & 0.0 & 0.0 \\ 1/3 & 1/3 & 1/6 & 1/6 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1 & 0.0 & 0.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}.$$

And finally U is $\vec{\beta}(0, 0.25, 0.25, 0.5)$.

An algorithm based on Euclidean Division

- **Definition 3 (Euclidean Division)** *Let V and W be two columns vector of the same size whose elements are non negative. We define the Euclidean division of W as follows:*

$$W = qV + R$$

where R is a vector and q is the maximum positive real such that all components of R are non negative.

- **Property 1** *We compute q and R as follows:*

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

where the min is computed on the values of i such that $V(i)$ is positive.

- Let us denote $\vec{\sigma}(i) = \sum_j C(i,j)$
- It can be obtained from a high level specification of the model.

Theory

- The bounding algorithm is based on the Euclidean division of $\vec{\sigma}$ by each column vector of C.
- **Theorem 3** *Consider $\vec{\sigma}$ and an arbitrary column index k. Let Z be $B(I-D)^{-1}C$. Perform the Euclidean division of $\vec{\sigma}$ by $C(*,k)$ to obtain q_k and R_k. Column k of Z is upper bounded by $\frac{\vec{\beta}}{q_k}$.*
- Proof: Algebraic

$$Z = B(I-D)^{-1}C \;\; and \sum_j Z_{i,j} = \vec{\beta}(i)$$

After some algebra: $\vec{\beta} = B(I-D)^{-1}C\vec{\sigma}$
Thus:$\vec{\beta} = B(I-D)^{-1}(q_k C_{*,k} + R_k)$
R_k, B and $(I-D)^{-1}$ are positive. Therefore:

$$B(I-D)^{-1}q_k C_{*,k} \leq_{el} \vec{\beta}$$

Examples

$$A = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.1 \\ 0.1 & 0.4 & 0.2 & 0.0 \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.2 & 0.0 & 0.4 & 0.0 \end{bmatrix}, \vec{\beta} = \begin{bmatrix} 0.3 \\ 0.3 \\ 0.0 \\ 0.4 \end{bmatrix}, C = \begin{bmatrix} 0.1 & 0.0 & 0.1 & 0.2 \\ 0.0 & 0.1 & 0.0 & 0.0 \\ 0.2 & 0.2 & 0.1 & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.1 & 0.0 & 0.0 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix}, \vec{\sigma} = \begin{bmatrix} 0.4 \\ 0.1 \\ 0.6 \\ 0.0 \\ 0.1 \\ 0.4 \end{bmatrix}.$$

Euclidean divisions:

$$q_1 = 3 \ \ and \ \ R_1^t = \begin{bmatrix} 0.1 & 0.1 & 0.0 & 0.0 & 0.1 & 0.1 \end{bmatrix}$$

$$q_2 = 1 \ \ and \ \ R_2^t = \begin{bmatrix} 0.4 & 0.0 & 0.4 & 0.0 & 0.0 & 0.3 \end{bmatrix}$$

$$q_3 = 4 \ \ and \ \ R_3^t = \begin{bmatrix} 0.0 & 0.1 & 0.2 & 0.0 & 0.1 & 0.0 \end{bmatrix}$$

$$q_4 = 2 \ \ and \ \ R_4^t = \begin{bmatrix} 0.0 & 0.1 & 0.4 & 0.0 & 0.1 & 0.2 \end{bmatrix}$$

And finally the bounding matrix is $\vec{\beta}(1/3, 1, 1/4, 1/2)$ and the st bound is $\vec{\beta}(0, 1/4, 1/4, 1/2)$.

Theory again

- It is even possible to find a bound if we are not able to compute exactly $\vec{\sigma}$.
- Assume that we are able to compute $\vec{\delta}$ such that $\vec{\delta} \leq_{el} \vec{\sigma}$.

  Theorem 4 *Consider $\vec{\delta}$ and an arbitrary column index k. Perform the Euclidean division of $\vec{\delta}$ by column k of C to obtain q'_k and R_k. If $q'_k \geq 1$, column k of Z is upper bounded by $\frac{\vec{\beta}}{q'_k}$.*

- Proof: As $\vec{\delta} \leq_{el} \vec{\sigma}$, we have $q'_k \leq q_k$ and we apply the former theorem.

Examples

- Assume now that we are not able to compute the second column of C. We have: $\vec{\delta}^t = \begin{bmatrix} 0.4 & 0.0 & 0.4 & 0.0 & 0.0 & 0.3 \end{bmatrix}$.
- Then we perform the Euclidean divisions.

$$q_1 = 2 \quad and \quad R_1 = \begin{bmatrix} 0.2 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.1 \end{bmatrix} \quad q_3 = 3 \quad and \quad R_3 = \begin{bmatrix} 0.1 \\ 0.0 \\ 0.1 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} \quad q_4 = 2 \quad and \quad R_4 = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.1 \end{bmatrix}$$

- As we cannot compute another bound than 1 for the second column, the bound is: $\vec{\beta}(1/2, 1, 1/3, 1/2)$ and the st bound is $\vec{\beta}(0, 1/6, 1/3, 1/2)$.

Paths and Graphs: Theoretical Results

- **Theorem 5** *Let L such that $A \leq_{el} L \leq_{el} S_A$. Then*

$$S_A \leq_{st} T(L) \leq_{st} T(A) \quad and \quad S_A \leq_{st} V(T(L)) \leq_{st} V(T(A))$$

- Finding some component-wise lower bound of $B\left(\sum_{i=0}^{\infty} C^i\right) D$ helps to obtain a more accurate bound.
- **Theorem 6** *Let $L1$ and $L2$ such that $A \leq_{el} L1 \leq_{el} L2 \leq_{el} S_A$ element-wise. Then:*

$$\begin{cases} S_A \leq_{st} T(L2) \leq_{st} T(L1) \leq_{st} T(A) \\ S_A \leq_{st} V(T(L2)) \leq_{st} V(T(L1)) \leq_{st} V(T(A)) \end{cases}$$

- The more information you get, the more accurate the bounds (but all informations are not created equal).

Improving the bound-heuristics

- All the rows do not have the same importance for the computation of the bound.
- Due to the monotonicity constraint, the last row is often completely modified by Vincent's algorithm.
- More eﬃcient to try to improve the first row of A than the last one.

Improving the bound-example

$$A = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.1 \\ 0.1 & 0.4 & 0.2 & 0. \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.2 & 0 & 0.4 & 0 \end{bmatrix}$$

$$T(A) = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.4 \\ 0.1 & 0.4 & 0.2 & 0.3 \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.2 & 0 & 0.4 & 0.4 \end{bmatrix} \quad V(T(A)) = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.4 \\ 0.1 & 0.3 & 0.2 & 0.4 \\ 0.1 & 0.2 & 0.3 & 0.4 \\ 0.1 & 0.2 & 0.3 & 0.4 \end{bmatrix}$$

Improving the bound-example

Suppose that one have compute the probability $[0.1, 0.1, 0., 0.1]$ of some paths leaving E from state 4 and entering again set E after a visit in E^c.

$$L1 = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.1 \\ 0.1 & 0.4 & 0.2 & 0. \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.3 & 0.1 & 0.4 & 0.1 \end{bmatrix}$$

$$T(L1) = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.4 \\ 0.1 & 0.4 & 0.2 & 0.3 \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.3 & 0.1 & 0.4 & 0.2 \end{bmatrix} \quad V(T(L1)) = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.4 \\ 0.1 & 0.3 & 0.2 & 0.4 \\ 0.1 & 0.2 & 0.3 & 0.4 \\ 0.1 & 0.2 & 0.3 & 0.4 \end{bmatrix}$$

The bound does not change...

Improving the bound-example

Assume now one have improved the first row and we have got the same vector of probability for the paths: $[0.1, 0.1, 0., 0.1]$.

$$L2 = \begin{bmatrix} 0.2 & 0.4 & 0.2 & 0.2 \\ 0.1 & 0.4 & 0.2 & 0. \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.2 & 0 & 0.4 & 0 \end{bmatrix}$$

$$T(L2) = \begin{bmatrix} 0.2 & 0.4 & 0.2 & 0.2 \\ 0.1 & 0.4 & 0.2 & 0.3 \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.2 & 0. & 0.4 & 0.4 \end{bmatrix} \quad V(T(L2)) = \begin{bmatrix} 0.2 & 0.4 & 0.2 & 0.2 \\ 0.1 & 0.4 & 0.2 & 0.3 \\ 0.1 & 0.2 & 0.4 & 0.3 \\ 0.1 & 0.1 & 0.4 & 0.4 \end{bmatrix}$$

The bound is now much better than the original one.

Graph techniques to find L

- $\mathbf{x} = B\left(\sum_{i=0}^{\infty} D^i\right) C$: sum of probability of paths leaving E (i.e. matrix B) and returning into E (matrix C) after an arbitrary number of visits inside E^c (matrix D).
- We select some paths instead of generating all of them.
- Well-known graph algorithms (Shortest Path, Breadth First search) to select some paths and compute their probability.

Some Details about Paths and Probability

- BFS: only takes into account the number of states in a path.
- Give a depth for the analysis tree.
- The probability of a path is the product of the probability of the arcs.
- SP: the weight of a an arc is equal to $-log(P(i,j))$.
- Thus the SP according to this weight is the path with the highest probability.

Taking Self-Loops into account

- Let $\mathcal{Z}$ be a path selected by the algorithm, p its probability and x a node of $\mathcal{Z}$.
- If there is a self loop in x (i.e. $P(x,x) = q > 0$), consider $\mathcal{L}_i = \mathcal{Z} + i$ loops in state x (for an arbitrary $i > 0$).
- $\mathcal{L}_i$ has probability pq^i.
- $\mathcal{L}_i$ is also a path which can be aggregated to $\mathcal{Z}$ in the analysis and the global probability is $p/(1-q)$.
- The algorithm computes the probability of the path and the list of self loops (with their probability) along the path.