Low Rank Decomposition and Stochastic Bounds

J.M. Fourneau

Laboratoire PRiSM, CNRS UMR 8144

Université de Versailles St-Quentin

Joint Work with A. Busic (INRIA DIOGENE and ENS ULM) and M. Ben Mamoun (Univ. Mohammed V, Rabat)

Outline

- Class C and G DTMC
- Low Rank Decomposition of a Markov Chain
- Computation of the steady-state distribution
- Computation of the transient distribution
- Stochastic Bound of an arbitrary DTMC by a DTMC with a low rank decomposition
- All vectors are row vectors.

Class C matrix

Definition 1 *A class* $\mathbf{C}$ *stochastic matrix* $\mathbf{P}$ *is defined by 2 vectors* v *and* c *such that:*

$$\mathbf{P} = [1, 1, 1, \ldots, 1]^T v + [0, 1, 2, \ldots, n-1]^T c,$$

where v *is a positive vector whose norm is equal to* 1, c *is a vector whose entries sum up to* 0 .

Class $\mathbf{C^G}$ matrix

Definition 2 *A class $\mathbf{C^G}$ stochastic matrix $\mathbf{P}$ is defined by 3 vectors (v, r, c) such that:*

$$\mathbf{P} = e^T v + r^T c,$$

where e is a vector whose all entries are equal to 1, v is a positive vector whose norm is equal to 1, c is a vector whose entries sum up to 0 and e^T is the transpose of e.

Exemple

Example 1 *Consider the following class* $\mathbf{C}^{\mathbf{G}}$ *matrix:*

$$\begin{aligned} M &= [1111]^T[0.2, 0.1, 0.4, 0.3] \\ &+ [0, 1, 1, 2]^T[0.1, -0.05, 0.05, -0.1]. \end{aligned}$$

M is equal to:

$$\begin{bmatrix} 0.2 & 0.1 & 0.4 & 0.3 \\ 0.2 & 0.1 & 0.4 & 0.3 \\ 0.2 & 0.1 & 0.4 & 0.3 \\ 0.2 & 0.1 & 0.4 & 0.3 \end{bmatrix} + \begin{bmatrix} 0. & 0. & 0. & 0. \\ 0.1 & -0.05 & 0.05 & -0.1 \\ 0.1 & -0.05 & 0.05 & -0.1 \\ 0.2 & -0.1 & 0.1 & -0.2 \end{bmatrix}$$

Finally,

$$M = \begin{bmatrix} 0.2 & 0.1 & 0.4 & 0.3 \\ 0.3 & 0.05 & 0.45 & 0.2 \\ 0.3 & 0.05 & 0.45 & 0.2 \\ 0.4 & 0.0 & 0.5 & 0.1 \end{bmatrix}$$

Low Rank Decomposition

Definition 3 *We consider the set $\mathcal{A}_k$ of stochastic matrices* $\mathbf{M}$ *defined by:*

$$\mathbf{M} = e^T v + \sum_{i=1}^{k} r_i^T c_i,$$

where c_i are vectors whose entries sum up to 0 *and r_i are vectors such that $r_i[j] \leq 1$ for all i and j. We say that a matrix in $\mathcal{A}_k$ has a decomposition of rank k. When vectors c_i are non zero and independent and similar conditions hold for vectors r_i, we say that matrix has a full rank k decomposition.*

Property 1 *Matrices in $\mathcal{A}_k$ have rank upmost* $k + 1$

Implications

- Computing the steady state distribution: linear time algorithm
- Computing the transient distribution: easier algorithm
- Computing the distribution of the first passage time
- We generalize the first two properties to matrices with a low rank decomposition.

Simplification rules

- For all x, y z row vectors with the same size, $xy^Tz = zxy^T$ as xy^T is a scalar.
- Furthermore by construction $ve^T = 1$ for all i.
- and $c_ie^T = 0$ for all i

Steady-State

- Let us define matrix $\mathbf{A}$ by $\mathbf{A}[i,l] = c_l r_i^T$.
- **Property 2** *If matrix* $(\mathbf{Id} - \mathbf{A})$ *is not singular, then the stationary distribution of an ergodic DTMC* $\mathbf{M}$ *in* $\mathcal{A}_k$ *is given as follows:*

$$\pi_{\mathbf{M}} = v + \sum_{i=1}^{k} E[r_i] c_i, \qquad (1)$$

where the values of $E[r_i]$ *which represent the expectations of* r_i *on the steady-state distribution* $\pi_{\mathbf{M}}$ *can be solved from a linear system of size* k*.*

$$E[r_i] = \sum_j r_i[j]v[j] + \sum_{l=1}^{k} E[r_l] \sum_j r_i[j] c_l[j].$$

- Note that we compute the expectations $E[r_i]$ before we compute the steady-state distributions and we do not need these distributions to compute the expectations due to the properties of the matrix.
- **Property 3** *The complexity for computing the steady state distribution is* $O(k^2N)$.
- **Example 2** *Consider matrix* $\mathbf{M}$ *whose decomposition of rank* 2 *is:*

$$
\begin{aligned}
& [1,1,1,1,1,1,1,1]^T \\
& [0.1,0.2,0.1,0.1,0.1,0,0.2,0.2] \\
+ \quad & [0,0.01,0.01,0.1,0.1,0.2,0.5,0.5]^T \\
& [-0.05,-0.05,-0.1,0.2,0,0,0,0] \\
+ \quad & [0,0.5,1,0,1,0,1,0]^T \\
& [0,0,0,0,-0.1,0.1,-0.1,0.1].
\end{aligned}
$$

Solution of the Exemple

- Matrix $\mathbf{A}$ is readily computed: $\begin{bmatrix} 0.0185 & -0.125 \\ -0.01 & -0.2 \end{bmatrix}$,
- and V is $[0.223, 0.5]$.
- Therefore $E = [0.2312031, 0.3925830]$
- the steady state distribution of the matrix is:

$$v + 0.2312031 * c1 + 0.392583 * c2.$$

- One finally get that this distribution is:

$$[0.0884, 0.1884, 0.0769, 0.1462, 0.0608, 0.0393, 0.1607, 0.2393].$$

Transient Distribution

Property 4 *The transient distribution at discrete time n of a DTMC $\mathbf{M}$ in $\mathcal{A}_k$ is given $\pi_{\mathbf{M}}^{(n)} = \pi_{\mathbf{M}}^{(0)}\mathbf{M}^n$ where $\mathbf{M}^n$ is given by:*

$$\mathbf{M}^n = e^T v + \sum_{i=1}^{k} (s_i^n)^T c_i, \tag{2}$$

where the s_i^n are given by the following induction:

$$\left[\begin{array}{lcl} s_i^1 & = & r_i \\ s_i^{n+1} & = & v(s_i^n)^T e + \sum_j c_j (s_j^n)^T r_j \end{array}\right.$$

Lower Complexity.

Approximation of a matrix by a low rank matrix

- Singular Value Decomposition of a matrix $m \times n$ ($\mathbf{A}$).

$$\mathbf{A} = \mathbf{USV}^T,$$

  where $\mathbf{S} = diag(\sigma_1, \sigma_2, \ldots, \sigma_r, 0, \ldots, 0)$, such that the singular values are ordered in decreasing order. r is the rank of $\mathbf{A}$.

- $\mathbf{U}$ and $\mathbf{V}$ are both orthogonal matrices (resp. $(m \times m)$ and $n \times n$) and satisfy $\mathbf{A}v_i^T = \sigma_i u_i^T$ and $u_i A = v_i^T \sigma_i$ for all i.

- u_i are the left singular vectors. v_i are the right singular vectors, σ_i are the singular values.

Approximation of a matrix

- Eckart et Young (36): the best rank k approximation for the Frobenius norm.

  Theorem 1 *A best rank k approximation of $\mathbf{A}$ is given by zeroing the $r-k$ trailing singular values of $\mathbf{A}$, i.e.*

  $$\hat{\mathbf{A}}_k = \mathbf{U}\hat{\mathbf{S}}_k\mathbf{V}^T,$$

  where $\hat{\mathbf{S}}_k = diag(\sigma_1, \sigma_2, \ldots, \sigma_k, 0, \ldots, 0)$, and the Frobenius norm of the difference is given by the Euclidean norm of the singular values which have been zeroed.

  $$||\mathbf{A} - \hat{\mathbf{A}}_k||_F = \sqrt{\sigma_{k+1}^2 + \ldots + \sigma_r^2}$$

- where $||\mathbf{A}||_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{i,j}^2} = \sqrt{\sum_{l=1}^{min(m,n)} \sigma_l^2}$.

Yes but we do not do that...

- because of the complexity: i.e. $O(n \times m \times min(n, m))$.
- Instead we compute a monotone upper bound (for the stochastic ordering of the initial matrix)
- Gives a stochastic bound on the steady-state and transient distribution
- Provides a bound on the expectation of any non decreasing reward on the distributions.

Stochastic Monotonicity

Property 5 *A matrix of class* $\mathbf{C^G}$ *is stochastically monotone if* r *is non decreasing and* $c\mathbf{K_{st}}$ *is non negative.*

Property 6 *Generalization rank* k*: If the following conditions hold, the matrix* $\mathbf{P}$ *is st-monotone:*

1. *for all* i*, the vector* r_i *is non decreasing.*
2. *for all* i*, the vectors* $c_i K_{st}$ *are non negative.*

Monotone stochastic bounds with a low rank

- Based on algorithm for class $\mathbf{C^G}$ by Busic and Pekergin
- two extensions: a row partition and a column partition (both in the papers)
- Only the row partition is presented.
- $\mathbf{K_{st}} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix}$

Monotone Upper Bound in Class $\mathbf{C^G}$

Require: $\mathbf{P}$.

Ensure: monotone upper bound in class $\mathbf{C^G}$ described by (v, r, c).

Compute with Vincent's algorithm max, the maximum for the strong stochastic ordering of the rows of $\mathbf{P}$.

$v = \mathbf{P}[1, *]$;

$c = max - v$;

r[N] = 1 ; r[1]=0.

$w = v\mathbf{K_{st}}$; $z = c\mathbf{K_{st}}$;

for $j = 2$ to $N - 1$ **do**

  $s = \mathbf{P}[j, *]\mathbf{K_{st}}$

  $h[j] = Max_{z[k]>0} \frac{s[k]-w[k]}{z[k]}$

  $r[j] = max(h[j], r[j-1])$

end for

Intuition

- the bounding matrix as the same first row and the same last row as the bound provided by Vincent's algorithm.
- The first row is equal to the first row of the initial matrix (Instruction 1) and the last row is max, the maximum for the stochastic ordering.
- The rows between 2 and $N-1$ are linear interpolation larger (with the stochastic ordering) than the corresponding row in the initial matrix.
- Instruction 9) makes the matrix stochastic monotone.

$$\mathbf{P} = \begin{bmatrix} 0.1 & 0.1 & 0.2 & 0.2 & 0.3 & 0.1 \\ 0.1 & 0.2 & 0.3 & 0.2 & 0 & 0.2 \\ 0.2 & 0.1 & 0.1 & 0.2 & 0.3 & 0.1 \\ 0.1 & 0 & 0.5 & 0 & 0.2 & 0.2 \\ 0 & 0.1 & 0.3 & 0.1 & 0.2 & 0.3 \\ 0.1 & 0.1 & 0.1 & 0.4 & 0.2 & 0.1 \end{bmatrix}$$

Vector v is $[0.1, 0.1, 0.2, 0.2, 0.3, 0.1]$. We use Vincent'a algorithm to compute $max = [0, 0.1, 0.3, 0.1, 0.2, 0.3]$. Vector c is $[-0.1, 0, 0.1, -0.1, -0.1, 0.2]$.

$$h[2] = max(0/0.1, -0.1/0.1, -0.2/0.1, 0.1/0.2) = 0.5,$$

$$h[3] = max(-0.1/0.1, -0.1/0.1, 0/0.1, 0/0.1) = 0.0,$$

Finally $r = [0, 0.5, 0.5, 1, 1, 1]$. The decomposition of the upper bound is:

$$\begin{aligned} & e^T[0.1, 0.1, 0.2, 0.2, 0.3, 0.1] \\ + \quad & [0, 0.5, 0.5, 1, 1, 1]^T[-0.1, 0, 0.1, -0.1, -0.1, 0.2], \end{aligned}$$

or with an explicit form:

$$\begin{bmatrix} 0.1 & 0.1 & 0.2 & 0.2 & 0.3 & 0.1 \\ 0.05 & 0.1 & 0.25 & 0.15 & 0.25 & 0.2 \\ 0.05 & 0.1 & 0.25 & 0.15 & 0.25 & 0.2 \\ 0 & 0.1 & 0.3 & 0.1 & 0.2 & 0.3 \\ 0 & 0.1 & 0.3 & 0.1 & 0.2 & 0.3 \\ 0 & 0.1 & 0.3 & 0.1 & 0.2 & 0.3 \end{bmatrix}.$$

For the sake of comparison, we also give the bounds provided by Vincent's algorithm:

$$\begin{bmatrix} 0.1 & 0.1 & 0.2 & 0.2 & 0.3 & 0.1 \\ 0.1 & 0.1 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.1 & 0.1 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.1 & 0 & 0.3 & 0.2 & 0.2 & 0.2 \\ 0 & 0.1 & 0.3 & 0.1 & 0.2 & 0.3 \\ 0 & 0.1 & 0.3 & 0.1 & 0.2 & 0.3 \end{bmatrix}.$$

A generalization based on row selection

- The bound given by the initial Algorithm is the rank 1 matrix, with the rows that are convex combinations of the first and the last row of matrix **Q**.
- The main idea of the new Algorithm is to construct an upper bound of rank k, using additional $k-1$ rows of matrix **Q**.
- 2 rows (first and last) + k-1 rows chosen by you.
- Use the initial algorithm between rows l_i and l_{i+1}.

Require: $\mathbf{P}$; $1 = l_1 < l_2 < \ldots < l_k < l_{k+1} = N$.

Ensure: monotone upper bound $\mathbf{R}$.

Compute $\mathbf{Q}$ with Vincent's Algorithm.

$v_i = \mathbf{Q}[l_i, *]$ for all i

for $i = 1$ to k **do**

  $c_i = v_{i+1} - v_i$; $z = c_i \mathbf{K_{st}}$; $w = v_i \mathbf{K_{st}}$

  $r_i[j] = 0$ for all j from 1 to l_i

  for $j = l_i + 1$ to $l_{i+1} - 1$ **do**

    $s = \mathbf{P}[j, *]\mathbf{K_{st}}$;

    $h_i[j] = Max_{z[m]>0} \frac{s[m]-w[m]}{z[m]}$;

    $r_i[j] = max(h_i[j], r_i[j-1])$

  end for

  for $j = l_{i+1}$ to N **do**

    $r_i[j] = 1$

  end for

end for

Main Result

Property 7 *For a stochastic matrix* $\mathbf{P}$ *let* $\mathbf{Q}$ *be the bound obtained by Vincent's Algorithm and* $\mathbf{R}$ *the bound obtained by the new Algorithm.*

1. $\mathbf{R}$ *is a monotone upper bound of matrix* $\mathbf{P}$ *with a rank* k *decomposition.*
2. $\mathbf{R}$ *is stochastically smaller than the matrix computed by the initial Algorithm.*
3. *Rows* $1 = l_1 < l_2 < \ldots < l_k < l_{k+1} = N$ *satisfy* $\mathbf{R}[l_i, *] = \mathbf{Q}[l_i, *]$. *Rows* $l_i < j < l_{i+1}$ *of the bound* $\mathbf{R}$ *are convex combinations of row* l_i *and* l_{i+1} *of matrix* $\mathbf{Q}$.
4. *The bounding matrix* $\mathbf{R}'$ *obtained using* $l' = \{l'_1, \ldots, l'_{k'+1}\}$ *such that* $l \subset l'$, *is stochastically smaller than* $\mathbf{R}$. *For* $l = \{1, \ldots, N\}$, $\mathbf{R} = \mathbf{Q}$.

Same Matrix

Set $k = 2$ and $l = (1, 3, 6)$. We get

$$c_1 = [0, 0, 0, 0, -0.1, 0.1] \text{ and } c_2 = [-0.1, 0, 0, 0, 0, 0.1].$$

The bound is:

$$\begin{bmatrix} 0.1 & 0.1 & 0.2 & 0.2 & 0.3 & 0.1 \\ 0.1 & 0.1 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.1 & 0.1 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0 & 0.1 & 0.3 & 0.1 & 0.2 & 0.3 \\ 0 & 0.1 & 0.3 & 0.1 & 0.2 & 0.3 \\ 0 & 0.1 & 0.3 & 0.1 & 0.2 & 0.3 \end{bmatrix}$$

Conclusion

- What about the distribution of the first passage time (already known for class $\mathbf{C^G}$) ?
- CTMC ?
- Lower Bounds
- Links with SVD
- Links with Perfect Simulation
- Links with Mixing Time