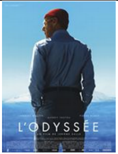
Outils statistiques et algorithmiques pour la complétion de matrices

Pierre Alquier

Machine Learning Meetup - Pau - 24 octobre 2016

Filtrage collaboratif / systèmes de recommandation

				
Claire	4	?	3	...
Nial	?	4	?	...
Brendon	2	?	4	...
Andrew	?	4	?	...
Adrian	1	?	?	...
Pierre	?	5	?	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Prix Netflix

Netflix Prize

COMPLETED

Home | Rules | Leaderboard | Update | Download

Leaderboard

Showing Test Score. Click here to show quiz score

Display top 20 **leaders.**

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace	0.8612	9.59	2009-07-24 17:18:43

Variante binaire

								
Stan			👍		👎		👍	
Pierre	👍	👍	👍	👍	👍	👍	👍	👍
Zoe	👍	👎					👍	
Bob			👍	👎				👎
Oscar				👍		👍		👍
Léa		👍	👎		👍			
Tony			👍	👎				👎

Une formalisation mathématique possible

- m_1 le nombre d'utilisateurs

Une formalisation mathématique possible

- m_1 le nombre d'utilisateurs
- m_2 le nombre de produits

Une formalisation mathématique possible

- m_1 le nombre d'utilisateurs
- m_2 le nombre de produits
- M matrice $m_1 \times m_2$ avec

$$M_{i,j} = \text{ note de l'individu } i \text{ au produit } j$$

Une formalisation mathématique possible

- m_1 le nombre d'utilisateurs
- m_2 le nombre de produits
- M matrice $m_1 \times m_2$ avec

$$M_{i,j} = \text{ note de l'individu } i \text{ au produit } j$$

- données : $M_{i,j}$ pour $(i,j) \in I$

$$n = \text{ nombre d'observations } = \mathrm{card}(I) \ll m_1 m_2$$

Une hypothèse sur la structure de M

$$M = \begin{pmatrix} 3 & 3 & & 1 & \\ & 1 & 1 & & \\ 5 & & & 4 & \\ & 2 & & & 4 \end{pmatrix}$$

Une hypothèse sur la structure de M

$$M = \begin{pmatrix} 3 & 3 & & 1 & \\ & 1 & 1 & & \\ 5 & & & 4 & \\ & 2 & & & 4 \end{pmatrix}$$

- sans information supplémentaire sur M, il est impossible de dire quoi que ce soit

Une hypothèse sur la structure de M

$$M = \begin{pmatrix} 3 & 3 & & 1 & \\ & 1 & 1 & & \\ 5 & & & 4 & \\ & 2 & & & 4 \end{pmatrix}$$

- sans information supplémentaire sur M, il est impossible de dire quoi que ce soit
- nécessité de faire une hypothèse sur la structure de M, qui soit satisfaite dans les applications réelles

Une hypothèse sur la structure de M

$$M = \begin{pmatrix} 3 & 3 & & 1 & \\ & 1 & 1 & & \\ 5 & & & 4 & \\ & 2 & & & 4 \end{pmatrix}$$

- sans information supplémentaire sur M, il est impossible de dire quoi que ce soit
- nécessité de faire une hypothèse sur la structure de M, qui soit satisfaite dans les applications réelles
- plusieurs possibilités ; ici : les goûts des utilisateurs sont des mélanges d'un petit nombre de comportement "de base", i.e M est de faible rang

Un cas extrêmement simplifié

$$M = \begin{array}{c} \begin{array}{cc} \text{sci-fi} & \text{comedy} \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 3 & 3 & & 1 & \\ & 1 & 1 & & \\ 5 & & & 4 & \\ ? & 2 & ? & ? & 4 \end{pmatrix} \end{array}$$

$$\text{user } = \lambda_1 \text{sci-fi} + \lambda_2 \text{comedy}$$

Un cas extrêmement simplifié

$$M = \begin{array}{c} \begin{array}{ccccc} \multicolumn{2}{c}{\text{sci-fi}} & \multicolumn{3}{c}{\text{comedy}} \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 3 & 3 & & 1 & \\ & 1 & 1 & & \\ 5 & & & 4 & \\ 2 & 2 & 4 & 4 & 4 \end{pmatrix} \end{array}$$

$$\text{user} \; = \lambda_1 \text{sci-fi} + \lambda_2 \text{comedy}$$

$$\lambda_1 = 2 \text{ et } \lambda_2 = 4$$

Un cas extrêmement simplifié

$$M = \begin{array}{c} \begin{array}{cc} \text{sci-fi} & \quad \text{comedy} \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 3 & 3 & & 1 & \\ & 1 & 1 & & \\ 5 & & & 4 & \\ 2 & 2 & 4 & 4 & 4 \end{pmatrix} \end{array}$$

$$\text{user} \;\; = \lambda_1 \text{sci-fi} + \lambda_2 \text{comedy}$$

$$\lambda_1 = 2 \text{ et } \lambda_2 = 4$$

Problème : en pratique, on ne veut pas supposer que l'on connaît les "comportements de base"

Méthode générale

$$\tilde{M} \leftarrow \begin{cases} \min_A \operatorname{rank}(A) \\ \text{s. c. } \forall (i,j) \in I : \quad A_{i,j} = M_{i,j} \end{cases}$$

Méthode générale

$$\tilde{M} \leftarrow \begin{cases} \min_A \operatorname{rank}(A) \\ \text{s. c. } \forall (i,j) \in I : \quad A_{i,j} = M_{i,j} \end{cases}$$

En pratique, on ne sait pas calculer $\tilde{M}$ en un temps rapide.

Méthode générale

$$\tilde{M} \leftarrow \begin{cases} \min_A \operatorname{rank}(A) \\ \text{s. c. } \forall (i,j) \in I : \quad A_{i,j} = M_{i,j} \end{cases}$$

En pratique, on ne sait pas calculer $\tilde{M}$ en un temps rapide.

Soient $\sigma_1(A) \geq \cdots \geq \sigma_K(A) \geq 0$ les valeurs singulières de A,

$$\operatorname{rank}(A) = \text{ le nombre de } \sigma_h(A) \neq 0.$$

Méthode générale

$$\tilde{M} \leftarrow \begin{cases} \min_A \operatorname{rank}(A) \\ \text{s. c. } \forall (i,j) \in I : \quad A_{i,j} = M_{i,j} \end{cases}$$

En pratique, on ne sait pas calculer $\tilde{M}$ en un temps rapide.

Soient $\sigma_1(A) \geq \cdots \geq \sigma_K(A) \geq 0$ les valeurs singulières de A,

$$\operatorname{rank}(A) = \text{ le nombre de } \sigma_h(A) \neq 0.$$

$$\hat{M} \leftarrow \begin{cases} \min_A \sum_{h=1}^{K} \sigma_h(A) \\ \text{s. c. } \forall (i,j) \in I : \quad A_{i,j} = M_{i,j} \end{cases}$$

Un article fondateur

Found Comput Math (2009) 9: 717–772
DOI 10.1007/s10208-009-9045-5

FOUNDATIONS OF COMPUTATIONAL MATHEMATICS
The Journal of the Society for the Foundations of Computational Mathematics

Exact Matrix Completion via Convex Optimization

Emmanuel J. Candès · Benjamin Recht

Received: 30 May 2008 / Revised: 6 February 2009 / Accepted: 14 February 2009 /
Published online: 3 April 2009

Abstract We consider a problem of considerable practical interest: the recovery of a data matrix from a sampling of its entries. Suppose that we observe m entries selected uniformly at random from a matrix $\boldsymbol{M}$. Can we complete the matrix and recover the entries that we have not seen?

We show that one can perfectly recover most low-rank matrices from what appears to be an incomplete set of entries. We prove that if the number m of sampled entries obeys

$$m \geq C\, n^{1.2} r \log n$$

for some positive numerical constant C, then with very high probability, most $n \times n$ matrices of rank r can be perfectly recovered by solving a simple convex optimization program. This program finds the matrix with minimum nuclear norm that fits the data.

Un article fondateur

Found Comput Math (2009) 9: 717–772
DOI 10.1007/s10208-009-9045-5

FOUNDATIONS OF COMPUTATIONAL MATHEMATICS
The Journal of the Society for the Foundations of Computational Mathematics

Exact Matrix Completion via Convex Optimization

Emmanuel J. Candès · Benjamin Recht

Received: 30 May 2008 / Revised: 6 February 2009 / Accepted: 14 February 2009 /
Published online: 3 April 2009

Abstract We consider a problem of considerable practical interest: the recovery of a data matrix from a sampling of its entries. Suppose that we observe m entries selected uniformly at random from a matrix $\boldsymbol{M}$. Can we complete the matrix and recover the entries that we have not seen?

We show that one can perfectly recover most low-rank matrices from what appears to be an incomplete set of entries. We prove that if the number m of sampled entries obeys

$$m \geq C n^{1.2} r \log n$$

for some positive numerical constant C, then with very high probability, most $n \times n$ matrices of rank r can be perfectly recovered by solving a simple convex optimization program. This program finds the matrix with minimum nuclear norm that fits the data.

Contenu :

- il existe une méthode rapide pour calculer $\hat{M}$ (optimisation convexe)

Un article fondateur

Found Comput Math (2009) 9: 717–772
DOI 10.1007/s10208-009-9045-5

FOUNDATIONS OF COMPUTATIONAL MATHEMATICS
The Journal of the Society for the Foundations of Computational Mathematics

Exact Matrix Completion via Convex Optimization

Emmanuel J. Candès · Benjamin Recht

Received: 30 May 2008 / Revised: 6 February 2009 / Accepted: 14 February 2009 /
Published online: 3 April 2009

Abstract We consider a problem of considerable practical interest: the recovery of a data matrix from a sampling of its entries. Suppose that we observe m entries selected uniformly at random from a matrix $\boldsymbol{M}$. Can we complete the matrix and recover the entries that we have not seen?

We show that one can perfectly recover most low-rank matrices from what appears to be an incomplete set of entries. We prove that if the number m of sampled entries obeys

$$m \geq C n^{1.2} r \log n$$

for some positive numerical constant C, then with very high probability, most $n \times n$ matrices of rank r can be perfectly recovered by solving a simple convex optimization program. This program finds the matrix with minimum nuclear norm that fits the data.

Contenu :

- il existe une méthode rapide pour calculer $\hat{M}$ (optimisation convexe)
- analyse théorique

Le résultat de Candès et Recht

$$\hat{M} \leftarrow \begin{cases} \min_A \sum_{h=1}^{K} \sigma_i(A) \\ \text{s. c. } \forall (i,j) \in I : \quad A_{i,j} = M_{i,j} \end{cases}$$

Le résultat de Candès et Recht

$$\hat{M} \leftarrow \begin{cases} \min_A \sum_{h=1}^{K} \sigma_i(A) \\ \text{s. c. } \forall (i,j) \in I : \quad A_{i,j} = M_{i,j} \end{cases}$$

- les entres (i, j) tirées uniformément, sans remise

Le résultat de Candès et Recht

$$\hat{M} \leftarrow \begin{cases} \min_A \sum_{h=1}^{K} \sigma_i(A) \\ \text{s. c. } \forall (i,j) \in I : \quad A_{i,j} = M_{i,j} \end{cases}$$

- les entres (i,j) tirées uniformément, sans remise
- hyp. d'*incohérence* sur M et $\mathrm{rang}(M) = r \ll \min(m, T)$,

Le résultat de Candès et Recht

$$\hat{M} \leftarrow \begin{cases} \min_A \sum_{h=1}^{K} \sigma_i(A) \\ \text{s. c. } \forall (i,j) \in I : \quad A_{i,j} = M_{i,j} \end{cases}$$

- les entres (i,j) tirées uniformément, sans remise
- hyp. d'*incohérence* sur M et $\mathrm{rang}(M) = r \ll \min(m, T)$,

Théorème

Avec grande probabilité, si on a observé

$$n \geq C.r \max(m, T)^{\frac{5}{4}} \log[\max(m, T)]$$

entrées, alors on a la reconstruction exacte $\hat{M} = M$.

Mais en réalité ?

En réalité, M ne peut pas être exactement de faible rang. Supposons

$$M = M_0 + E$$

où M_0 est une matrice de faible rang.

Mais en réalité ?

En réalité, M ne peut pas être exactement de faible rang. Supposons

$$M = M_0 + E$$

où M_0 est une matrice de faible rang.

$$\hat{M} \leftarrow \min_A \left\{ \sum_{(i,j)\in I} (A_{i,j} - M_{i,j})^2 + \lambda \sum_{h=1}^{K} \sigma_i(A) \right\}$$

Le cas bruité...

Candès, E. J. and Plan, Y.,
Matrix completion with noise.
Proceedings of the IEEE, 98(6) :925–936, 2010.

Koltchinskii, V., Lounici, K., and Tsybakov, A. B.,
Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion.
The Annals of Statistics, 39(5) :2302–2329, 2011.

Klopp, O.,
Noisy low-rank matrix completion with general sampling distribution.
Bernoulli, 20(1) :282–303, 2014.

T. T. Mai and P. Alquier.
A Bayesian approach for noisy matrix completion : Optimal rate under general sampling distribution.
Electron. J. Statist., 9(1) :823–841, 2015.

Le cas bruité...

Candès, E. J. and Plan, Y.,
Matrix completion with noise.
Proceedings of the IEEE, 98(6) :925–936, 2010.

Koltchinskii, V., Lounici, K., and Tsybakov, A. B.,
Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion.
The Annals of Statistics, 39(5) :2302–2329, 2011.

Klopp, O.,
Noisy low-rank matrix completion with general sampling distribution.
Bernoulli, 20(1) :282–303, 2014.

T. T. Mai and P. Alquier.
A Bayesian approach for noisy matrix completion : Optimal rate under general sampling distribution.
Electron. J. Statist., 9(1) :823–841, 2015.

$$\frac{1}{mT}\sum_{i=1}^{m}\sum_{j=1}^{T}(\hat{M}_{i,j}-(M_0)_{i,j})^2 \lesssim \frac{\sigma^2 \max(m,T)\mathrm{rang}(M_0)\log(m+T)}{N}$$

$$\sigma^2 = \mathrm{Var}(E_{i,j}).$$

Implémentation

$$\hat{M} \leftarrow \min_{A} \left\{ \sum_{(i,j)\in I} (A_{i,j} - M_{i,j})^2 + \lambda \sum_{h=1}^{K} \sigma_i(A) \right\}$$

Implémentation

$$\hat{M} \leftarrow \min_{A} \left\{ \sum_{(i,j)\in I} (A_{i,j} - M_{i,j})^2 + \lambda \sum_{h=1}^{K} \sigma_i(A) \right\}$$

Il existe plusieurs packages, librairies, etc... qui implémentent cet estimateur (et des variantes). A titre d'exemple, je présente le package *softImpute* de Trevor Hastie & Rahul Mazumder.

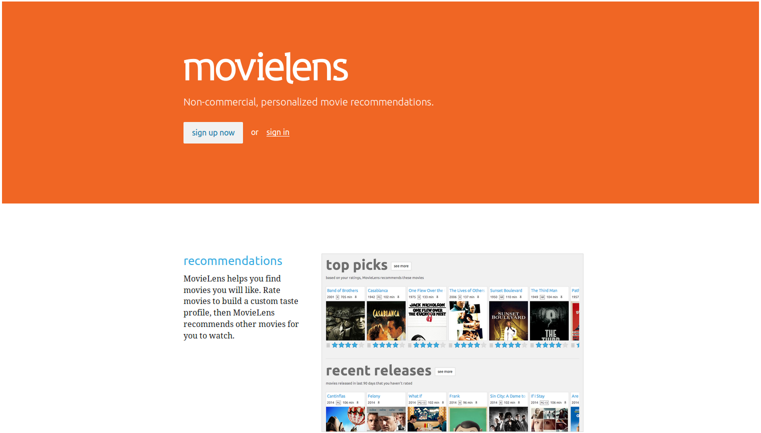
Implémentation

$$\hat{M} \leftarrow \min_{A} \left\{ \sum_{(i,j)\in I} (A_{i,j} - M_{i,j})^2 + \lambda \sum_{h=1}^{K} \sigma_i(A) \right\}$$

Il existe plusieurs packages, librairies, etc... qui implémentent cet estimateur (et des variantes). A titre d'exemple, je présente le package *softImpute* de Trevor Hastie & Rahul Mazumder.

Application sur un jeu de données provenant de *MovieLens*.

http ://movielens.org



Les films de la base de données MovieLens 100K

1682 films

```
1|Toy Story (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)|0|0|0|1|1|1|0
2|GoldenEye (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?GoldenEye%20(1995)|0|1|1|0|0|0|0
3|Four Rooms (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Four%20Rooms%20(1995)|0|0|0|0|0|0
4|Get Shorty (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Get%20Shorty%20(1995)|0|1|0|0|0|1
5|Copycat (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Copycat%20(1995)|0|0|0|0|0|0|1|0|1|0
6|Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)|01-Jan-1995||http://us.imdb.com/Title?Yao+a+
7|Twelve Monkeys (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Twelve%20Monkeys%20(1995)|0|0
8|Babe (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Babe%20(1995)|0|0|0|0|1|1|0|0|1|0|0|0|0
9|Dead Man Walking (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Dead%20Man%20Walking%20(199
10|Richard III (1995)|22-Jan-1996||http://us.imdb.com/M/title-exact?Richard%20III%20(1995)|0|0|0|0|
11|Seven (Se7en) (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Se7en%20(1995)|0|0|0|0|0|0|1|
12|Usual Suspects, The (1995)|14-Aug-1995||http://us.imdb.com/M/title-exact?Usual%20Suspects,%20The
13|Mighty Aphrodite (1995)|30-Oct-1995||http://us.imdb.com/M/title-exact?Mighty%20Aphrodite%20(1995
14|Postino, Il (1994)|01-Jan-1994||http://us.imdb.com/M/title-exact?Postino,%20Il%20(1994)|0|0|0|0|
15|Mr. Holland's Opus (1995)|29-Jan-1996||http://us.imdb.com/M/title-exact?Mr.%20Holland's%20Opus%2
16|French Twist (Gazon maudit) (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Gazon%20maudit%
17|From Dusk Till Dawn (1996)|05-Feb-1996||http://us.imdb.com/M/title-exact?From%20Dusk%20Till%20Da
18|White Balloon, The (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Badkonake%20Sefid%20(199
19|Antonia's Line (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Antonia%20(1995)|0|0|0|0|0|0
20|Angels and Insects (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Angels%20and%20Insects%2
21|Muppet Treasure Island (1996)|16-Feb-1996||http://us.imdb.com/M/title-exact?Muppet%20Treasure%20
22|Braveheart (1995)|16-Feb-1996||http://us.imdb.com/M/title-exact?Braveheart%20(1995)|0|1|0|0|0|0|
```

Utilisateurs de la base de données MovieLens 100K

943 utilisateurs

```
1|24|M|technician|85711
2|53|F|other|94043
3|23|M|writer|32067
4|24|M|technician|43537
5|33|F|other|15213
6|42|M|executive|98101
7|57|M|administrator|91344
8|36|M|administrator|05201
9|29|M|student|01002
10|53|M|lawyer|90703
11|39|F|other|30329
12|28|F|other|06405
13|47|M|educator|29206
```

Lecture des données

```
> X = read.table("u.data",header=FALSE)
> X
    V1  V2 V3        V4
1  196 242  3 881250949
2  186 302  3 891717742
3   22 377  1 878887116
4  244  51  2 880606923
5  166 346  1 886397596
....
> A = Incomplete(i=X$V1,j=X$V2,x=X$V3)
```

La classe d'objets *Incomplete*

```
> A
 [1,] 5 3 4 3 3 5 4 1 5 3 2 5 5 5 5 5 3 4 5 4
 [2,] 4 . . . . . . . . 2 . . 4 4 . . . . 3 .
 [3,] . . . . . . . . . . . . . . . . . . . .
 [4,] . . . . . . . . . . 4 . . . . . . . . .
 [5,] 4 3 . . . . . . . . . . . . . . 4 . . .
 [6,] 4 . . . . . 2 4 4 . . 4 2 5 3 . . . 4 .
 [7,] . . . 5 . . 5 5 5 4 3 5 . . . . . . . .
 [8,] . . . . . . 3 . . . 3 . . . . . . . . .
 [9,] . . . . . 5 4 . . . . . . . . . . . . .
[10,] 4 . . 4 . . 4 . 4 . 4 5 3 . . 4 . . . .
```

La fonction *softImpute*

$$\hat{M} \leftarrow \min_{A} \left\{ \sum_{(i,j)\in I} (A_{i,j} - M_{i,j})^2 + \lambda \sum_{h=1}^{K} \sigma_i(A) \right\}$$

Syntaxe générale :

```
> B = softImpute(A, rank.max = ..., lambda = ...,
                    type = x("als","svd"), ... )
> i = ...
> j = ...
> impute(B,i,j)
```

Exemple

```
> A
[1,] 5 3 4 3 3 5 4 1 5 3 2 5 5 5 5 5 3 4 5 4
[2,] 4 . . . . . . . . 2 . . 4 4 . . . . 3 .
[3,] . . . . . . . . . . . . . . . . . . . .
[4,] . . . . . . . . . . 4 . . . . . . . . .

> B=softImpute(A,rank.max=5,lambda=0,type="svd")
> impute(B,3,1)
[1] 1.361311
> impute(B,4,1)
[1] 2.48563
```

Obtenir plusieurs prédictions

```
> A
[1,] 5 3 4 3 3 5 4 1 5 3 2 5 5 5 5 5 3 4 5 4
[2,] 4 . . . . . . . . 2 . . 4 4 . . . . 3 .
[3,] . . . . . . . . . . . . . . . . . . . .
[4,] . . . . . . . . . . 4 . . . . . . . . .

> B=softImpute(A,rank.max=5,lambda=0,type="svd")
> i = c(3,3,3,4,4,4)
> j = c(1,2,3,1,2,3)
> impute(B,i,j)
[1] 1.3432932 0.3631449 1.0265744 2.7306180
    0.1185635 1.8732696
```

Compléter toute la matrice d'un coup

```
> B=softImpute(A,rank.max=5,lambda=0,type="svd")
> Y = complete(A,B)
```

Attention, ça peut faire exploser votre mémoire !

Compléter toute la matrice d'un coup

```
> round(Y)
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] ...
[1,]    5    3    4    3    3    5    4    1    5
[2,]    4    1    2    2    0    3    4    3    5
[3,]    1    0    1    2    1    1    2    1    3
[4,]    3    0    2    2    1    3    4    2    5
...
> A
[1,] 5 3 4 3 3 5 4 1 5 3 2 5 5 5 5 5 3 4 5 4
[2,] 4 . . . . . . . . 2 . . 4 4 . . . . 3 .
[3,] . . . . . . . . . . . . . . . . . . . .
[4,] . . . . . . . . . . 4 . . . . . . . . .
```

MSE en fonction du rang

```
> data = X[1:80000,]
> A = Incomplete(i=data$V1,j=data$V2,x=data$V3)
> test = X[80001:100000,]
> MSE = c()
> for (k in 1:10)
> {
>  B = softImpute(A,rank.max=k,lambda=0,type="svd")
>  pred = impute(object=B,i=test$V1,j=test$V2)
>  MSE = c(MSE,mean((pred-test$V3)^2))
> }
```

MSE en fonction du rang