

Fall in love with your next book

Let the booksellers of
Gower Street help you
find your perfect match.



Low-rank matrix completion: optimization on manifolds at work

Nicolas Boumal

Joint work with Pierre-Antoine Absil

Université catholique de Louvain

April 2011

Recommender systems tell you which items you might like
based on a huge database of ratings

$$X = \begin{matrix} & \leftarrow n \text{ cols} \rightarrow \\ \begin{pmatrix} ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \end{pmatrix} & \begin{matrix} \uparrow \\ m \text{ rows} \\ \downarrow \end{matrix} \end{matrix}$$

Ratings of items by the users are recorded

One row per item, one column per user



user j

$$X = \begin{pmatrix} ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \end{pmatrix} \begin{matrix} \text{item } i \end{matrix}$$



Ratings of items by the users are recorded

One row per item, one column per user



user j

$$X = \begin{pmatrix} ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & 4 & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \end{pmatrix} \text{ item } i$$



Most ratings are unknown. Our job is to complete X .

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

We could exploit similarities between users and items
to complete the matrix

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

In a global, automated, scalable way?

Scalability will guide the algorithm design

for both time and memory complexity

Netflix 1M\$ prize:

- 17,700 movies;
- 480,000 users;
- 100,000,000 ratings (1%).

Scalability will guide the algorithm design

for both time and memory complexity

Netflix 1M\$ prize:

- 17,700 movies;
- 480,000 users;
- 100,000,000 ratings (1%).

⇒ The whole matrix won't fit into memory,

⇒ but the known ratings will.

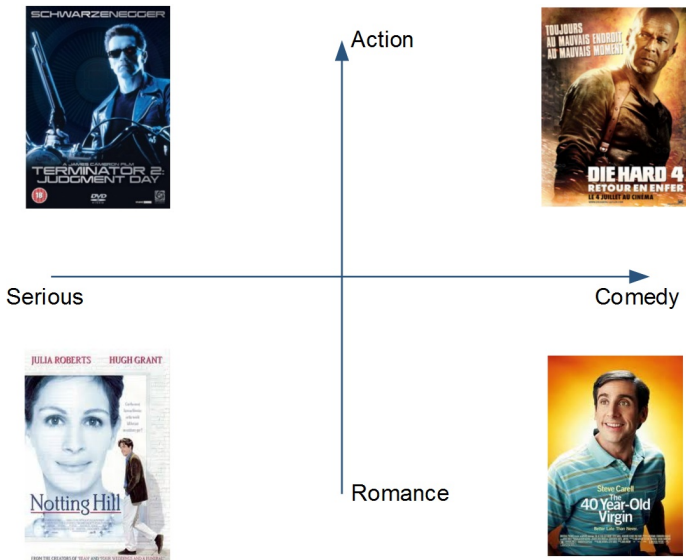
We assume that X has low-rank r

Hence, that ratings are inner products in a small space \mathbb{R}^r

$$X = \begin{pmatrix} 1 & ? & 2 & ? & ? & 5 & ? & ? & ? & ? & 5 & ? & ? & ? & 2 & ? \\ 2 & ? & 2 & ? & ? & 4 & ? & ? & ? & 3 & 4 & ? & 5 & ? & ? & ? \\ 1 & ? & 5 & 2 & ? & 4 & ? & 4 & ? & ? & ? & 2 & ? & ? & ? & ? \\ ? & 1 & ? & 3 & ? & ? & ? & 3 & ? & ? & 3 & ? & 2 & ? & 5 & 5 \\ 4 & 4 & ? & ? & ? & ? & 5 & ? & ? & ? & 1 & ? & ? & 1 & ? & 4 \end{pmatrix}$$
$$\approx \begin{pmatrix} ? & ? \\ ? & ? \\ ? & ? \\ ? & ? \\ ? & ? \end{pmatrix} \cdot \begin{pmatrix} ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \end{pmatrix}$$

Rationale: only a few factors influence our preferences

We map items and users to this small dimensional space
without any human intervention



Toward a reasonable formulation

The optimal choice UW is an m -by- n matrix of rank r in best agreement with the k known entries of X

$$\min_{U \in \mathbb{R}^{m \times r}, W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} \overset{\text{confidence}}{C_{ij}^2} \overset{\text{known ratings}}{((UW)_{ij} - X_{ij})^2}$$

known entries

This is reasonable

$$\min_{U \in \mathbb{R}^{m \times r}, W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2$$

- Very natural;
- Search space of dimension $r(m + n)$;
- Objective computable in $\mathcal{O}(rk)$ time;
- Ideal for Gaussian noise on ratings X_{ij} .

This is reasonable, *but*

$$\min_{U \in \mathbb{R}^{m \times r}, W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2$$

Minimizers are not isolated.

This is reasonable, *but*

$$\min_{U \in \mathbb{R}^{m \times r}, W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2$$

Minimizers are not isolated.

If (U, W) is a minimizer, $(UM, M^{-1}W)$ is too,
for any r -by- r invertible matrix M .

The objective is invariant under invertible transformations

We don't want that. Why?

- The search space $\mathbb{R}^{m \times r} \times \mathbb{R}^{r \times n}$ is bigger than it ought to be;
- Most theoretical guarantees of convergence for iterative optimization methods assume isolated critical points;
- And it may prevent superlinear convergence rates altogether.

Partial solution: force U to be orthonormal

It's a kind of normalization

$$\min_{U \in \mathbb{O}(m,r), W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2$$

|

$$\mathbb{O}(m,r) = \{U \in \mathbb{R}^{m \times r} : U^\top U = I_r\}$$

Partial solution: force U to be orthonormal

It's a kind of normalization

$$\min_{U \in \mathbb{O}(m,r), W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2$$

|

$$\mathbb{O}(m,r) = \{U \in \mathbb{R}^{m \times r} : U^\top U = I_r\}$$

Minimizers are still not isolated.

Partial solution: force U to be orthonormal

It's a kind of normalization

$$\min_{U \in \mathbb{O}(m,r), W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2$$

|

$$\mathbb{O}(m,r) = \{U \in \mathbb{R}^{m \times r} : U^\top U = I_r\}$$

Minimizers are still not isolated.

If (U, W) is a minimizer, $(UQ, Q^\top W)$ is too,
for any r -by- r orthogonal matrix Q .

The set of r -dimensional subspaces of \mathbb{R}^m is the Grassmann manifold $\text{Gr}(m, r)$

- The objective is well defined over the equivalence classes $[U] = \{UQ : Q \in \mathbb{R}^{r \times r}, Q^\top Q = I_r\}$;
- All matrices $UQ \in [U]$ share a common column space: $\text{col}(UQ) = \mathcal{U} \in \text{Gr}(m, r)$;
- We represent a point \mathcal{U} on Grassmann with any orthonormal matrix U such that $\text{col}(U) = \mathcal{U}$.

$$\min_{\mathcal{U} \in \text{Gr}(m,r), W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2$$

U is any m -by- r orthonormal matrix such that $\text{col}(U) = \mathcal{U}$.

$$\min_{\mathcal{U} \in \text{Gr}(m,r)} \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2$$

U is any m -by- r orthonormal matrix such that $\text{col}(U) = \mathcal{U}$.

W is the solution of a simple least squares problem.

The objective is a function of the column space \mathcal{U}

$$\min_{\mathcal{U} \in \text{Gr}(m,r)} \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2$$

\downarrow
 $f(\mathcal{U})$

U is any m -by- r orthonormal matrix such that $\text{col}(U) = \mathcal{U}$.

W is the solution of a simple least squares problem.

f is not continuous :(

$$f(\mathcal{U}) = \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2$$

Indeed, the least squares problem may not always have a unique solution.

This stems from unattended entries, see [Dai, Milenkovic et al., 2010](#).

Regularization makes f smooth

$$f(\mathcal{U}) = \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2$$

Regularization makes f smooth

$$f(\mathcal{U}) = \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2 + \lambda \sum_{(i,j) \notin \Omega} (UW)_{ij}^2$$

Regularization makes f smooth

$$f(\mathcal{U}) = \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2 + \lambda \sum_{(i,j) \notin \Omega} (UW)_{ij}^2$$

Computation of the inner objective looks like it costs $\mathcal{O}(mnr)$ time :(

A Matrix 101 trick to reduce computational costs

$$f(\mathcal{U}) = \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2 + \lambda \sum_{(i,j) \notin \Omega} (UW)_{ij}^2$$

A Matrix 101 trick to reduce computational costs

$$f(\mathcal{U}) = \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2 + \lambda \sum_{(i,j) \notin \Omega} (UW)_{ij}^2$$

$$\sum_{(i,j) \notin \Omega} (UW)_{ij}^2$$

A Matrix 101 trick to reduce computational costs

$$f(\mathcal{U}) = \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2 + \lambda \sum_{(i,j) \notin \Omega} (UW)_{ij}^2$$

$$\sum_{(i,j) \notin \Omega} (UW)_{ij}^2 + \sum_{(i,j) \in \Omega} (UW)_{ij}^2$$

A Matrix 101 trick to reduce computational costs

$$f(\mathcal{U}) = \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2 + \lambda \sum_{(i,j) \notin \Omega} (UW)_{ij}^2$$

$$\sum_{(i,j) \notin \Omega} (UW)_{ij}^2 + \sum_{(i,j) \in \Omega} (UW)_{ij}^2 = \sum_{(i,j)} (UW)_{ij}^2$$

A Matrix 101 trick to reduce computational costs

$$f(\mathcal{U}) = \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2 + \lambda \sum_{(i,j) \notin \Omega} (UW)_{ij}^2$$

$$\begin{aligned} \sum_{(i,j) \notin \Omega} (UW)_{ij}^2 + \sum_{(i,j) \in \Omega} (UW)_{ij}^2 &= \sum_{(i,j)} (UW)_{ij}^2 \\ &= \|UW\|_F^2 \end{aligned}$$

A Matrix 101 trick to reduce computational costs

$$f(\mathcal{U}) = \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2 + \lambda \sum_{(i,j) \notin \Omega} (UW)_{ij}^2$$

$$\begin{aligned} \sum_{(i,j) \notin \Omega} (UW)_{ij}^2 + \sum_{(i,j) \in \Omega} (UW)_{ij}^2 &= \sum_{(i,j)} (UW)_{ij}^2 \\ &= \|UW\|_F^2 \\ &= \|W\|_F^2 \end{aligned}$$

Complexity: $\mathcal{O}(r(k+n))$.

$$\|UW\|_F^2 = \text{trace}((UW)^T UW) = \text{trace}(W^T U^T UW) = \text{trace}(W^T W) = \|W\|_F^2$$

This (final) objective has many good properties
for the low-rank matrix completion problem

$$f(\mathcal{U}) = \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2 + \lambda \sum_{(i,j) \notin \Omega} (UW)_{ij}^2$$

This (final) objective has many good properties
for the low-rank matrix completion problem

$$f(\mathcal{U}) = \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2 + \lambda \sum_{(i,j) \notin \Omega} (UW)_{ij}^2$$

- It is natural and defined over the “right” space;

This (final) objective has many good properties
for the low-rank matrix completion problem

$$f(\mathcal{U}) = \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2 + \lambda \sum_{(i,j) \notin \Omega} (UW)_{ij}^2$$

- It is natural and defined over the “right” space;
- It has isolated minimizers and it is smooth;

This (final) objective has many good properties

for the low-rank matrix completion problem

$$f(\mathcal{U}) = \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2 + \lambda \sum_{(i,j) \notin \Omega} (UW)_{ij}^2$$

- It is natural and defined over the “right” space;
- It has isolated minimizers and it is smooth;
- It is efficiently computable, and so are $\text{grad } f$ and $\text{Hess } f$;

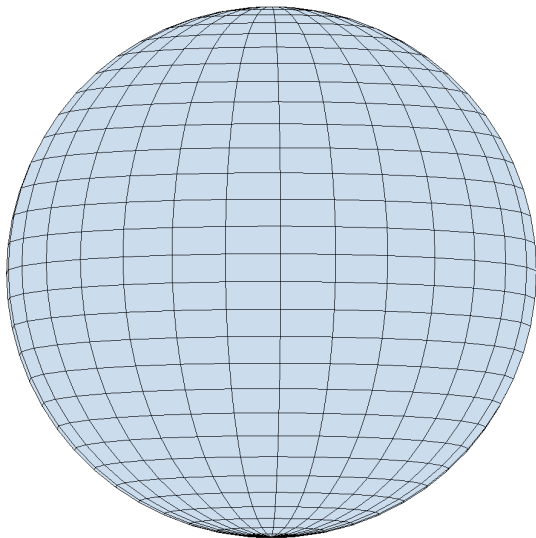
This (final) objective has many good properties

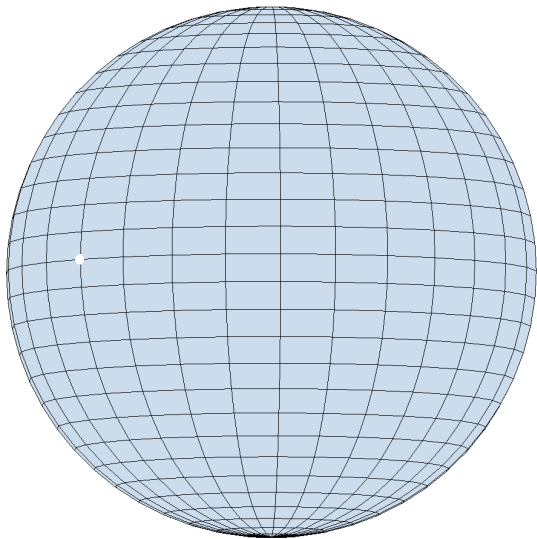
for the low-rank matrix completion problem

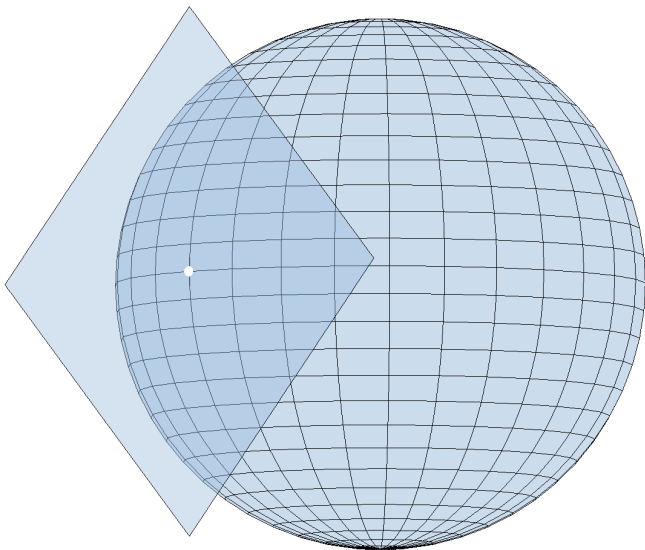
$$f(\mathcal{U}) = \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2 + \lambda \sum_{(i,j) \notin \Omega} (UW)_{ij}^2$$

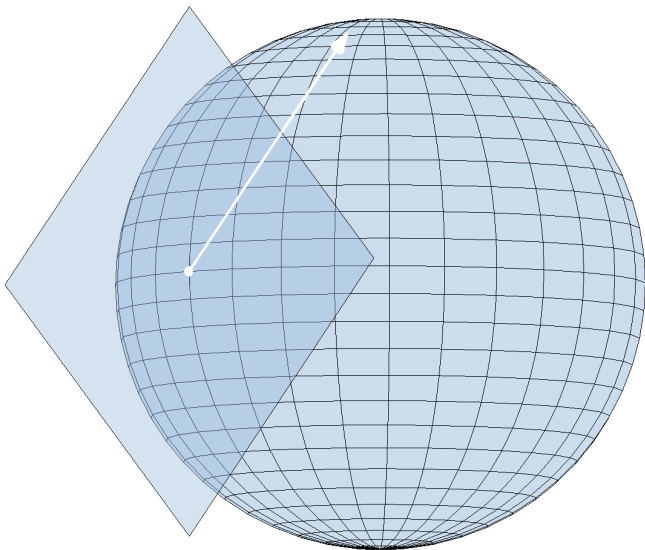
- It is natural and defined over the “right” space;
- It has isolated minimizers and it is smooth;
- It is efficiently computable, and so are $\text{grad } f$ and $\text{Hess } f$;
- It should be able to deal with noise.

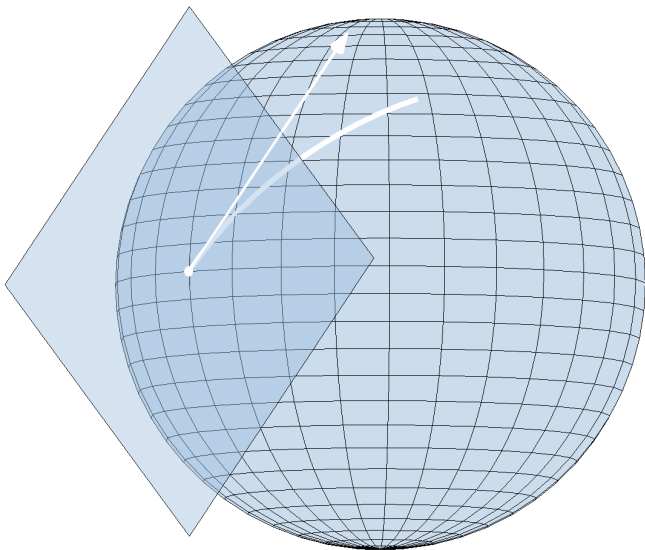
How do you minimize $f(\mathcal{U})$ over Grassmann?











We use a Riemannian trust-region method: GenRTR

In \mathbb{R}^n , we would:

We use a Riemannian trust-region method: GenRTR

In \mathbb{R}^n , we would:

- build a **quadratic model** of f around the current iterate x ,

We use a Riemannian trust-region method: GenRTR

In \mathbb{R}^n , we would:

- build a **quadratic model** of f around the current iterate x ,
- **minimize the model** in a *trust*-region around x ,

We use a Riemannian trust-region method: GenRTR

In \mathbb{R}^n , we would:

- build a **quadratic model** of f around the current iterate x ,
- **minimize the model** in a *trust*-region around x ,
- **make the step** if it decreases f and **rescale the region** accordingly.

We use a Riemannian trust-region method: GenRTR

In \mathbb{R}^n , we would:

- build a **quadratic model** of f around the current iterate x ,
- minimize the model in a *trust*-region around x ,
- make the step if it decreases f and **rescale the region** accordingly.

Absil, Baker and Gallivan (2007) generalized this to manifolds.

Trust-region is much better than a Newton method

The trust-region approach:

- guarantees monotonic objective decrease, and
- can deal with approximate Hessians (even the identity)
- and with **approximate solutions** of the Newton equations.

GenRTR comes with proofs

- Our method is **guaranteed to converge** toward critical points,
- with **second-order speed** once near the limit point.
- Usually, the limit point is a local minimizer.

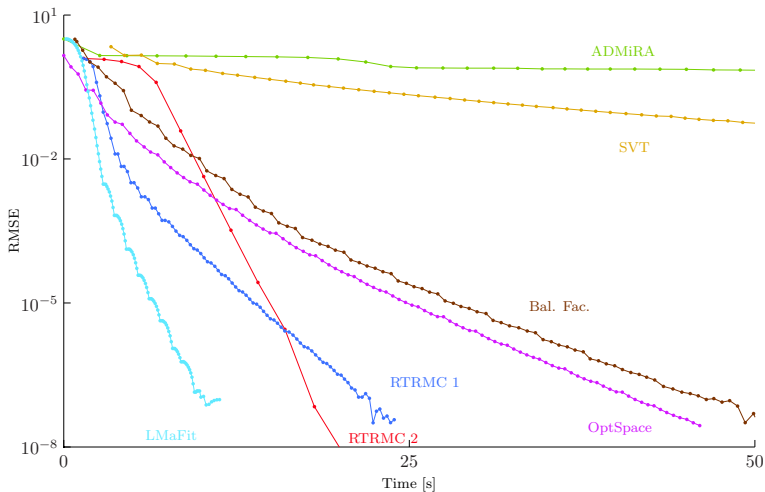
A few numerical tests

We compare seven algorithms

- RTRMC is our method,
with Hessian (2^{nd} order) and without Hessian (1^{st} order);
- OptSpace by Keshavan and Oh;
- Balanced Factorization by Meyer and Sepulchre;
- SVT by Candès and Becker;
- ADMiRA by Lee and Bresler;
- LMaFit by Wen, Yin and Zhang.

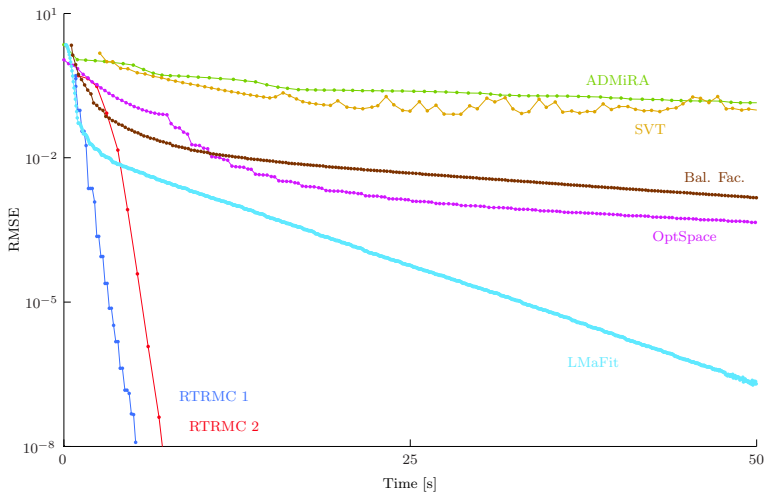
LMaFit is a serious contestant.

$m = n = 10\,000$, $r = 10$, knowledge = 1%



Our method is fastest on rectangular matrices.

$m = 1\,000$, $n = 30\,000$, $r = 5$, knowledge = 2.6%



Final thoughts

Conclusions

Our **contribution** is a sensible objective function fed to a state-of-the-art theory-backed solver with the right complexity;

The algorithm scales well and is easily parallelizable;

In the noiseless case, we get exact completion if enough entries are known;

In the noisy case, we provide a close to optimal reconstruction.

Next step: application to real data sets.

Take home message

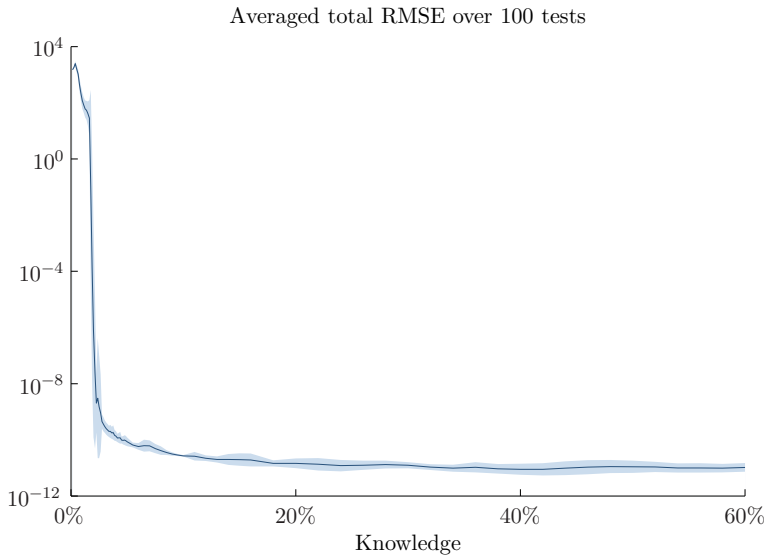
Many practical optimization problems are such that either the objective function presents some invariances, giving rise to a quotient manifold, or are constrained by nonlinear equalities describing a submanifold of \mathbb{R}^n .

Efficient tools are readily available to fully exploit the geometry of these problems such as, e.g., GenRTR.

These tools are backed up by a solid theory, described for example in the book by Absil, Mahony and Sepulchre: Optimization Algorithms on Matrix Manifolds, 2008.

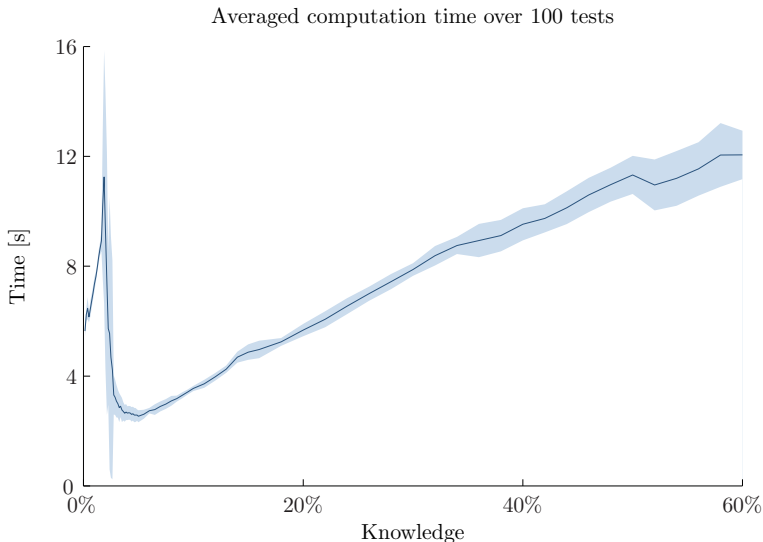
Given enough information, we consistently recover X .

Noiseless, $m = n = 1\,000$, $r = 5$



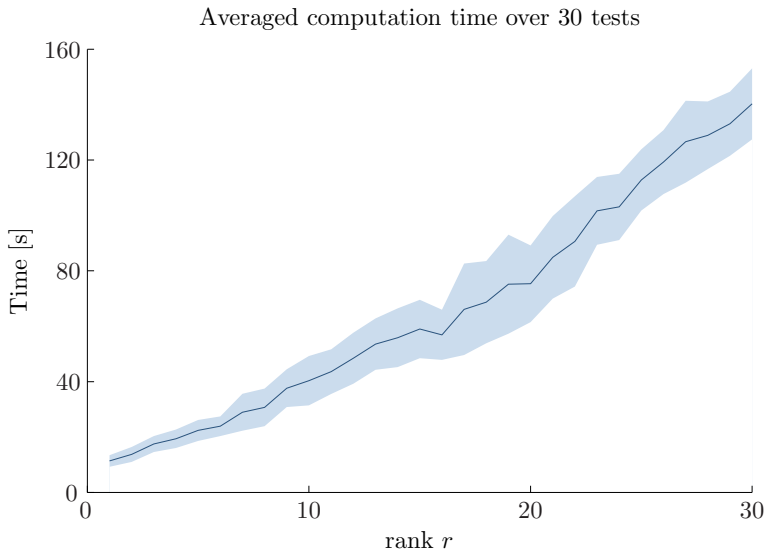
Computation time is proportional to # known entries.

Noiseless, $m = n = 1\,000$, $r = 5$



Computation time scales reasonably with the rank.

Noiseless, $m = n = 1\,000$, knowledge = 25%



In the presence of noise, we are close to optimal.

Noisy, $m = n = 500$, $r = 4$

