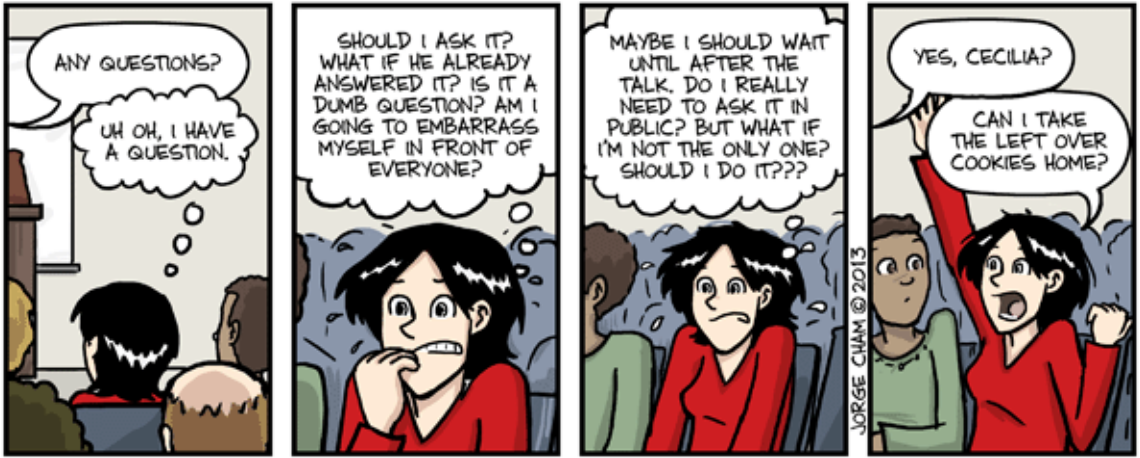
Tensor decompositions and their applications

Lecture 5: Approximation by a tensor rank decomposition

Nick Vannieuwenhoven (KU Leuven)

1. Introduction (5')
2. A complication (15')
3. Alternating least squares methods (30')
4. Riemannian quasi-Newton optimization method (50')
   - Riemannian optimization
   - Riemannian nonlinear least squares problems
   - Trust region globalization
   - Stepping in the search direction
   - A practical implementation
   - Numerical experiments
5. Conclusions


ANY QUESTIONS?
UH OH, I HAVE A QUESTION.
SHOULD I ASK IT? WHAT IF HE ALREADY ANSWERED IT? IS IT A DUMB QUESTION? AM I GOING TO EMBARRASS MYSELF IN FRONT OF EVERYONE?
MAYBE I SHOULD WAIT UNTIL AFTER THE TALK. DO I REALLY NEED TO ASK IT IN PUBLIC? BUT WHAT IF I'M NOT THE ONLY ONE? SHOULD I DO IT???
YES, CECILIA?
CAN I TAKE THE LEFT OVER COOKIES HOME?
JORGE CHAM © 2013
WWW.PHDCOMICS.COM

Overview

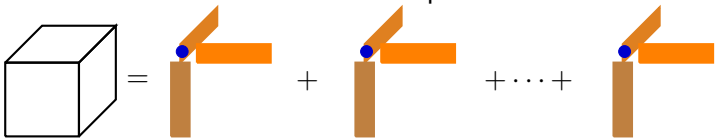
Linear algebra

is extended to

Multilinear algebra

Singular value decomposition

Tensor rank decomposition



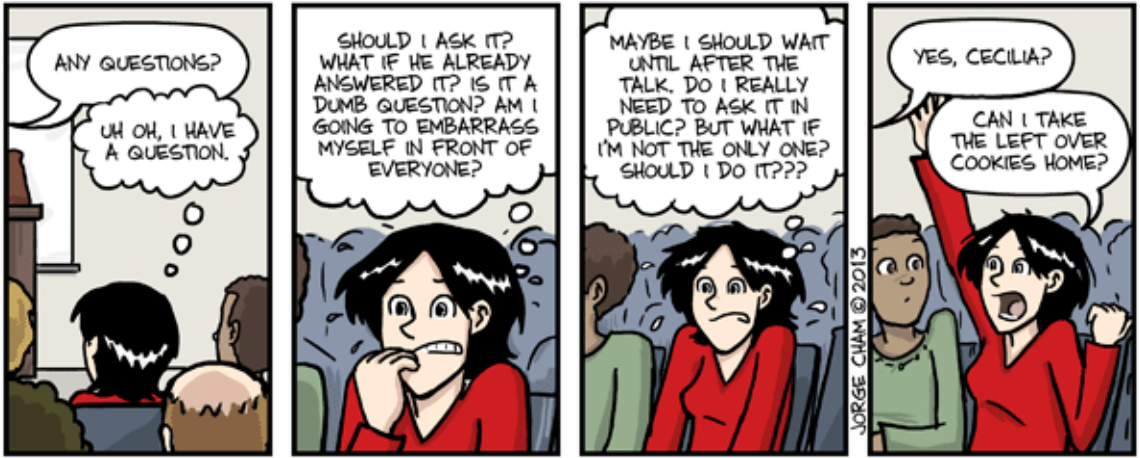
Tensors originating in applications are rarely exactly of low rank due to various sources of errors.

We are thus looking to approximate a tensor $\mathcal{A} \in V_1 \otimes \cdots \otimes V_d$ by a rank-r tensor.
Formulated as an optimization problem, we seek to solve the **distance minimization problem**

$$\min_{\operatorname{rank}(\mathcal{B}) \leq r} \frac{1}{2} \|\mathcal{B} - \mathcal{A}\|_F^2,$$

where $\|\cdot\|_F$ is the **Euclidean norm**.

This formulation can be **extended to incomplete tensors** by only measuring the distance in the known coordinates.


ANY QUESTIONS?
UH OH, I HAVE A QUESTION.
SHOULD I ASK IT? WHAT IF HE ALREADY ANSWERED IT? IS IT A DUMB QUESTION? AM I GOING TO EMBARRASS MYSELF IN FRONT OF EVERYONE?
MAYBE I SHOULD WAIT UNTIL AFTER THE TALK. DO I REALLY NEED TO ASK IT IN PUBLIC? BUT WHAT IF I'M NOT THE ONLY ONE? SHOULD I DO IT???
YES, CECILIA?
CAN I TAKE THE LEFT OVER COOKIES HOME?
JORGE CHAM © 2013
WWW.PHDCOMICS.COM

Overview

A complication

In contrast to the matrix, multilinear, and tensor trains ranks, **the set of tensors of bounded tensor rank**,

$$\sigma_r^0 = \{\mathcal{A} \in V_1 \otimes \cdots \otimes V_d \mid \operatorname{rank}(\mathcal{A}) \leq r\},$$

is not closed.

For example, the tensors on the curve

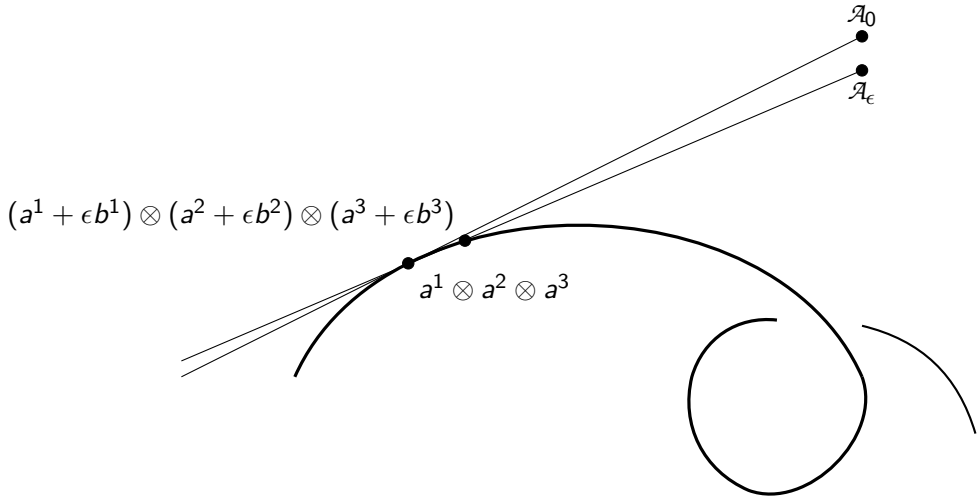
$$\mathcal{A}_\epsilon = \frac{1}{\epsilon}(a^1 + \epsilon b^1) \otimes (a^2 + \epsilon b^2) \otimes (a^3 + \epsilon b^3) - \frac{1}{\epsilon} a^1 \otimes a^2 \otimes a^3$$

converge to the rank-3 tensor

$$\mathcal{A}_0 = b^1 \otimes a^2 \otimes a^3 + a^1 \otimes b^2 \otimes a^3 + a^1 \otimes a^2 \otimes b^3$$

as $\epsilon \to 0$.

Geometrically what happens is that $\mathcal{A}_0$ is a point on a **tangent line** that is not a **secant line**:



The following result, by combining de Silva and Lim (2008) and Breiding and Vannieuwenhoven (2018a) partially explains what happens:

Proposition

Let $\mathcal{A}$ be a tensor of rank $s > r$ and assume there exists a sequence of identifiable rank-r tensors

$$\mathcal{A}^{(k)} = \mathcal{A}_1^{(k)} + \cdots + \mathcal{A}_r^{(k)}$$

such that $\mathcal{A}^{(k)} \to \mathcal{A}$ as $k \to \infty$. Then,

1. *there exist $i \neq j$ such that $\|\mathcal{A}_i^{(k)}\| \to \infty$ and $\|\mathcal{A}_j^{(k)}\| \to \infty$, and*
2. $\kappa[\tau_r](\mathcal{A}^{(k)}) \to \infty$.

Here's what happens to the condition number in one of Paatero's (2000) models:

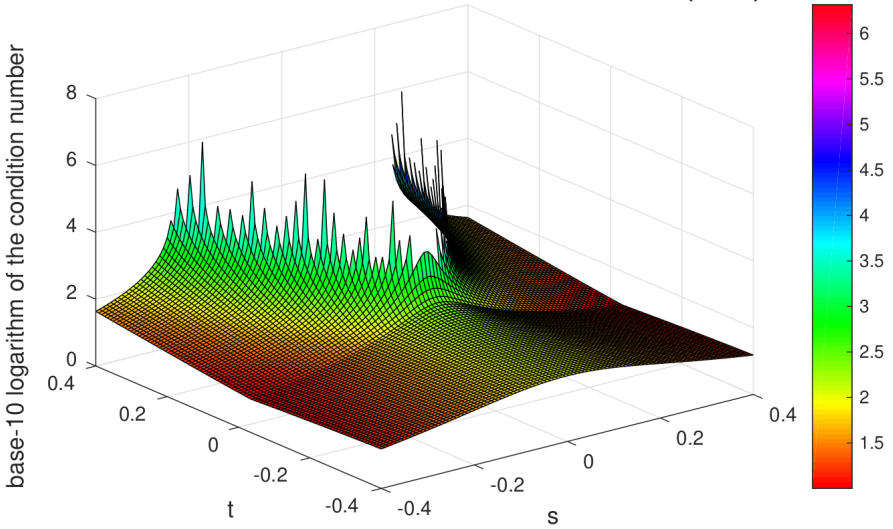
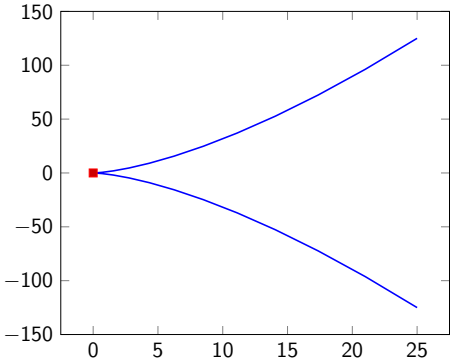


FIG. 3. *The base-10 logarithm* $\log_{10}$ *of the condition number for Paatero's model* (6.4). *The grid consists of* 100×100 *points equally spaced in* $[-0.4, 0.4] \times [-0.4, 0.4]$.

Fortunately, Landsberg (2012) states that the troublesome set of tensors with a rank strictly greater than r has **Lebesgue measure zero** within

$$\sigma_r = \overline{\sigma_r^0} = \left\{ \lim_{k \to \infty} \mathcal{A}^{(k)} \mid \operatorname{rank}\left(\mathcal{A}^{(k)}\right) \leq r \right\}.$$

Unfortunately, for the approximation problem this can still be disastrous! Imagine what happens when σ_r would be like the **nodal cubic** with $\sigma_r \setminus \sigma_r^0$ at the node:



Rank-3 tensors in $\mathbb{R}^{2\times 2\times 2}$ never have a best rank-2 approximation (de Silva and Lim, 2008).

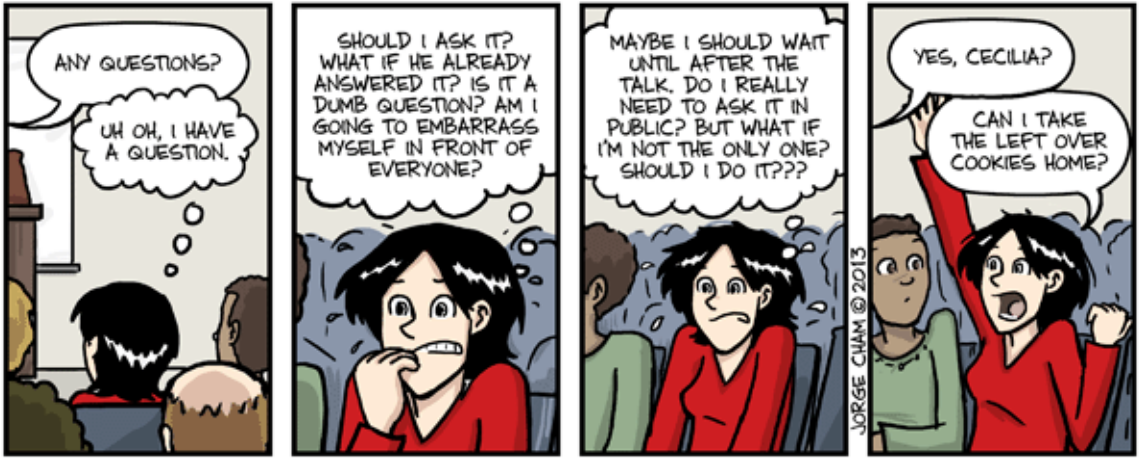
It can nevertheless be shown, for real vector spaces and **real rank**, that there exists an **open tubular neighborhood** $\mathcal{T} \subset V_1 \otimes \cdots \otimes V_d$ of (a dense open subset of) σ_r^0 in the sense of Hirsch (1976) such that for all tensors $\mathcal{A} \in \mathcal{T}$, the approximation problem

$$\min_{\operatorname{rank}(\mathcal{B}) \leq r} \|\mathcal{B} - \mathcal{A}\|_F$$

is **well posed** for all $\mathcal{A} \in \mathcal{T}$ (if $\sigma_r \neq V_1 \otimes \cdots \otimes V_d$).

Qi, Michałek and Lim (2020) proved that for tensor products of complex vector spaces and the corresponding **complex tensor rank**, the above approximation problem is well posed for almost all $\mathcal{A} \in V_1 \otimes_{\mathbb{C}} \cdots \otimes_{\mathbb{C}} V_d$.

I will assume in the remainder an $\mathcal{A}$ is given for which the problem is well posed.


ANY QUESTIONS?
UH OH, I HAVE A QUESTION.
SHOULD I ASK IT? WHAT IF HE ALREADY ANSWERED IT? IS IT A DUMB QUESTION? AM I GOING TO EMBARRASS MYSELF IN FRONT OF EVERYONE?
MAYBE I SHOULD WAIT UNTIL AFTER THE TALK. DO I REALLY NEED TO ASK IT IN PUBLIC? BUT WHAT IF I'M NOT THE ONLY ONE? SHOULD I DO IT???
YES, CECILIA?
CAN I TAKE THE LEFT OVER COOKIES HOME?
JORGE CHAM © 2013
WWW.PHDCOMICS.COM

Overview

1. Introduction (5')
2. A complication (15')
3. Alternating least squares methods (30')
4. Riemannian quasi-Newton optimization method (50')
   - Riemannian optimization
   - Riemannian nonlinear least squares problems
   - Trust region globalization
   - Stepping in the search direction
   - A practical implementation
   - Numerical experiments
5. Conclusions

Alternating least squares methods

Usually we want to find both the **closest rank-r approximation** to $\mathcal{A}$, given as an $n_1 \times \cdots \times n_d$ array in coordinates, and its **decomposition** into rank-1 tensors.

Traditionally, the set of tensors of bounded rank was parameterized by **factor matrices**:

$$p : \mathbb{k}^{n_1 \times r} \times \cdots \times \mathbb{k}^{n_d \times r} \to \mathbb{k}^{n_1 \times \cdots \times n_d}, \quad (A^1, \ldots, A^d) \mapsto \sum_{i=1}^{r} a_i^1 \otimes \cdots \otimes a_i^d.$$

With this parameterization, the optimization problem for finding both the best rank-r approximation and its decomposition was formulated as

$$\min_{(A_1,\ldots,A_d) \in \mathbb{k}^{n_1 \times r} \times \cdots \times \mathbb{k}^{n_d \times r}} \frac{1}{2} \|p(A_1, \ldots, A_d) - \mathcal{A}\|_F^2.$$

The **alternating least squares** (ALS) method by Carroll and Chang (1970) is based on the observation that the **objective function** is equivalent to

$$\min_{\substack{A_j \in \mathbb{k}^{n_k \times r}, \\ j=1,\ldots,d}} \frac{1}{2} \|\mathcal{A}_{(k)} - A_k (A_1 \odot \cdots \odot A_{k-1} \odot A_{k+1} \odot \cdots \odot A_d)^T\|_F^2.$$

for every $k = 1, 2, \ldots, d$, and where

$$A \odot B := [a_i \otimes b_i]_i$$

is the **Khatri–Rao product**.

If A_j, $j \neq k$, are fixed, then finding the optimal A_k becomes a **linear problem**! Indeed, consider the (compact) QR factorization

$$(A_1 \odot \cdots \odot A_{k-1} \odot A_{k+1} \odot \cdots \odot A_d) = QR.$$

Then,

$$\frac{1}{2}\|\mathcal{A}_{(k)}^T - QRA_k^T\|_F^2 = \frac{1}{2}\|QQ^*\mathcal{A}_{(k)}^T - QRA_k^T\|_F^2 + \frac{1}{2}\|(I - QQ^*)\mathcal{A}_{(k)}^T\|_F^2.$$

Since the second summand is constant, it suffices to minimize

$$\frac{1}{2}\|QQ^*\mathcal{A}_{(k)}^T - QRA_k^T\|_F^2 = \frac{1}{2}\|Q^*\mathcal{A}_{(k)}^T - RA_k^T\|_F^2$$

over $A_k \in \mathbb{k}^{n_k \times r}$. Hence, we can take

$$A_k^T = R^{-1}Q^*\mathcal{A}_{(k)}^T$$

as optimal solution.

The standard ALS method then solves the optimization problem by **cyclically fixing** all but one factor matrix A_k and alternatingly updating $A_1, \ldots, A_d$.

Algorithm 1: ALS method

---

input : A tensor $\mathcal{A} \in \mathbb{k}^{n_1 \times \cdots \times n_d}$.
input : A target rank r.
output: Factor matrices $(A_1, \ldots, A_d)$ of a CPD approximating $\mathcal{A}$.

Initialize factor matrices $A_k \in \mathbb{k}^{n_k \times r}$ (e.g., entries sampled i.i.d. from $N(0,1)$, or truncated HOSVD);
while *Not converged* **do**
  for $k = 1, 2, \ldots, d$ **do**
    $Z \leftarrow A_1 \odot \cdots \odot A_{k-1} \odot A_{k+1} \odot \cdots \odot A_d$;
    Compute compact QR decomposition $Z = QR$;
    $A_k \leftarrow (R^{-1} Q^* \mathcal{A}_{(k)}^T)^T$;
  end
end

The ALS scheme produces a sequence of incrementally better approximations. However, the **convergence properties** of the ALS method are poorly understood.

The scheme has accumulation points, but it is not known if they correspond to critical points of the objective function. That is, we do not know if the accumulation points of the ALS scheme correspond to points satisfying the first-order optimality conditions.

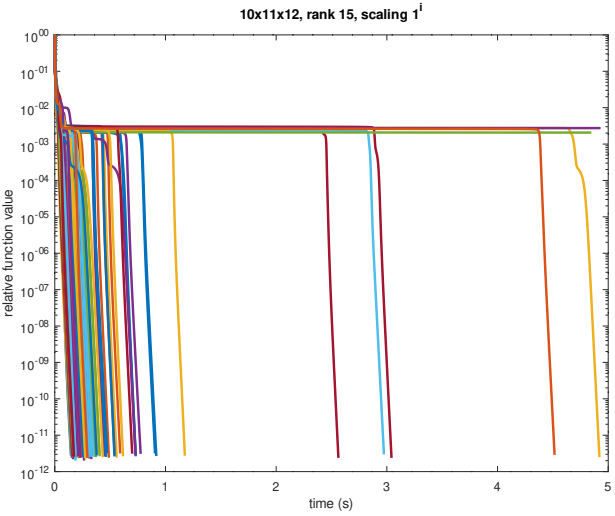
Uschmajew (2012) proved local convergence to critical points where the Hessian is positive semi-definite and of maximal rank.

Numerical experiments

ALS is a very fast, effective and reliable method when generating random $10 \times 11 \times 12$ tensors of rank 15 in the usual way: sample the entries of the factor matrices i.i.d. from $N(0, 1)$.

Observe that 98% of random initializations converge to a solution.

Tensor decomposition, solved!

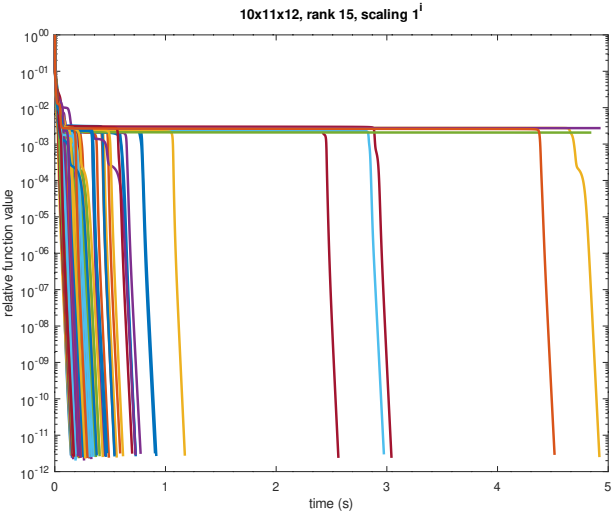


Numerical experiments

ALS is a very fast, effective and reliable method when generating random $10 \times 11 \times 12$ tensors of rank 15 in the usual way: sample the entries of the factor matrices i.i.d. from $N(0, 1)$.

Observe that 98% of random initializations converge to a solution.

Tensor decomposition, solved!
But is it?



The condition number, once more

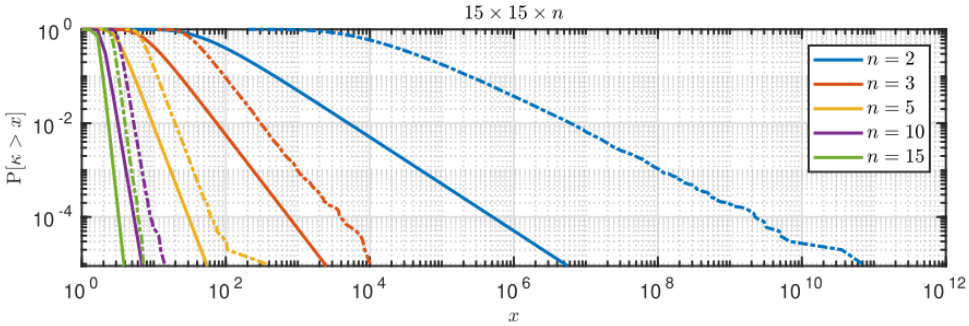
There is widespread evidence, however, that the performance of **iterative solution methods** for a computational problem is often correlated with the **condition number** of the solution.

This holds, among others, for

- solving **linear systems** with the steepest descent and conjugate gradient methods;
- solving systems of **linear inequalities** with the perceptron and ellipsoid methods;
- solving **polyhedral cone feasibility** problems with the interior point method;
- solving **homogeneous polynomial systems** with homotopy continuation; and
- solving (Riemannian) **nonlinear least squares problems** with the (Riemannian) Gauss–Newton method.

The last of these can be found in Breiding and Vannieuwenhoven (2018c) and the others are discussed in Bürgisser and Cucker (2013).

Experiments by Beltrán, Breiding and Vannieuwenhoven (2019) showed that sampling rank-r tensors by sampling random factor matrices from a Gaussian ensemble results in a **very favorable** distribution of condition numbers:



(a) A, B, and C i.i.d. standard normal entries.

FIG. 3.1. *The empirical complementary cumulative distribution function of the condition number for rank-15 tensors of size $15 \times 15 \times n$ is shown in dashed lines. The corresponding solid line shows the lower bound from Theorem 1.4. The tensors $\mathcal{A} = \sum_{i=1}^{15} \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i$ were generated by randomly sampling factor matrices $A \in \mathbb{R}^{15\times 15}$, $B \in \mathbb{R}^{15\times 15}$, and $C \in \mathbb{R}^{n\times 15}$, as indicated.*

We do not believe that sampling factor matrices produces a **realistic probability density function** on σ_r, however.

For a more realistic **Gaussian-like distribution** on σ_r (it equals the Gaussian distribution when σ_r perfectly fills the ambient space), the following was proved.

Theorem (Beltrán, Breiding and Vannieuwenhoven (2021))

Consider $n_1 \times \cdots \times n_d$ tensors with $n_i \geq 2$. Then,

$$\mathop{\mathbb{E}}_{\mathcal{A} \sim N(\sigma_2)} [\kappa[\tau_2](\mathcal{A})] = \infty.$$

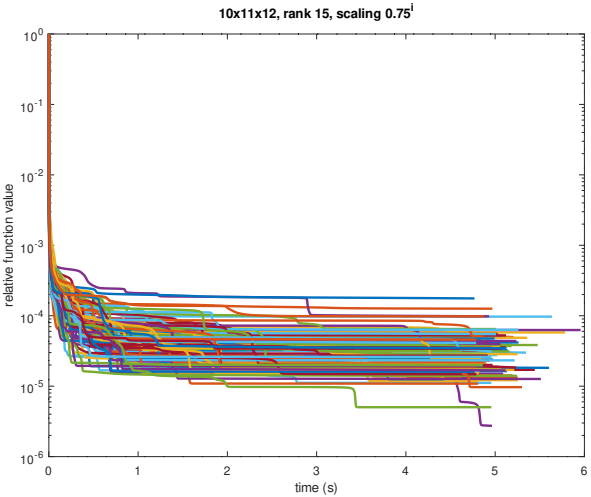
If $n_i \geq 3$ and $r \geq 3$ and additionally r-identifiability holds and $(r-2)$-identifiability holds for $(n_1 - 2) \times \cdots \times (n_d - 2)$ tensors, then

$$\mathop{\mathbb{E}}_{\mathcal{A} \sim N(\sigma_r)} [\kappa[\tau_r](\mathcal{A})] = \infty.$$

Sampling the factor matrices randomly from a Gaussian ensemble does not result in an expected value ∞ of the condition number, empirically. Therefore:

> *Sampling factor matrices randomly oversamples the well-conditioned areas of σ_r, while undersampling the high-condition areas.*

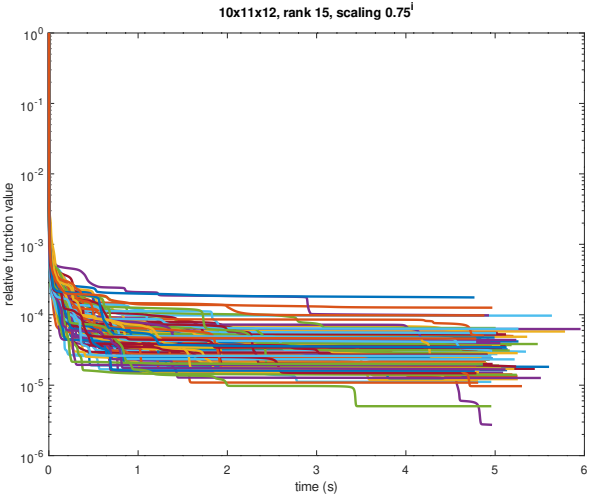
Simply **altering the scaling** of the rank-1 tensors by multiplying the ith tensor with 0.75^i, so the last tensor is approximately 1.5% the size of the first one **catastrophically destroys convergence**.

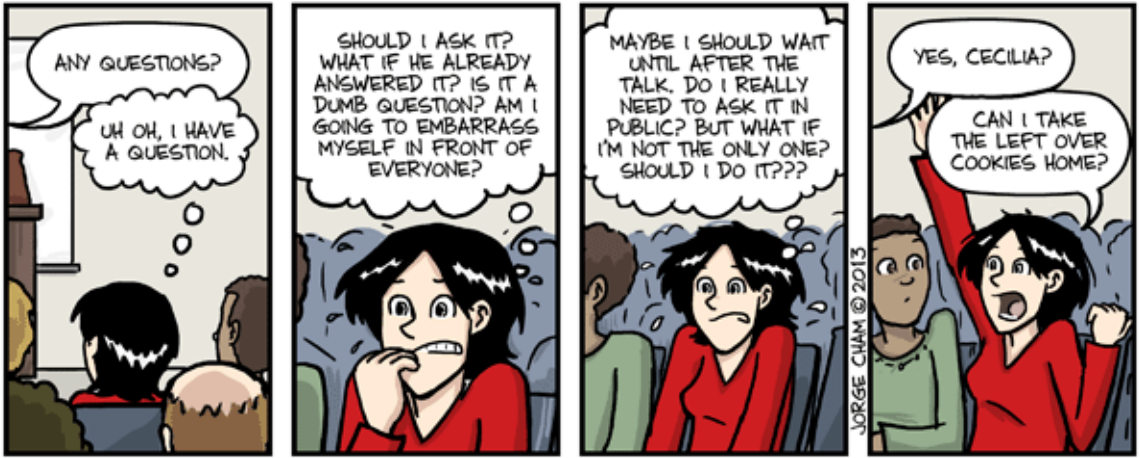


It is interesting to note that the condition number of tensor rank decomposition τ_r, taking a rank-r tensor to its rank-1 tensors, is **invariant under such scaling**! What's happening?

ALS is solving a **different problem**: it recovers the factor matrices rather than rank-1 tensors.

Vannieuwenhoven (2017) describes a condition number for that problem, which does blow up when introducing differences in scale. This potentially explains the behavior.




ANY QUESTIONS?
UH OH, I HAVE A QUESTION.
SHOULD I ASK IT? WHAT IF HE ALREADY ANSWERED IT? IS IT A DUMB QUESTION? AM I GOING TO EMBARRASS MYSELF IN FRONT OF EVERYONE?
MAYBE I SHOULD WAIT UNTIL AFTER THE TALK. DO I REALLY NEED TO ASK IT IN PUBLIC? BUT WHAT IF I'M NOT THE ONLY ONE? SHOULD I DO IT???
YES, CECILIA?
CAN I TAKE THE LEFT OVER COOKIES HOME?
JORGE CHAM © 2013
WWW.PHDCOMICS.COM

Overview

1. Introduction (5')
2. A complication (15')
3. Alternating least squares methods (30')
4. Riemannian quasi-Newton optimization method (50')
   - Riemannian optimization
   - Riemannian nonlinear least squares problems
   - Trust region globalization
   - Stepping in the search direction
   - A practical implementation
   - Numerical experiments
5. Conclusions

Riemannian optimization

The parameterization of a collection of rank-1 tensors via factor matrices has an additional problem: There is an entire $(d-1)$-dimensional family of **equivalent representations**, namely

$$\{(\alpha_1 a^1) \otimes \cdots \otimes (\alpha_d a^d) \mid \alpha_1 \cdots \alpha_d = 1\}.$$

This has the nasty implication that **minimizers** of the optimization problem

$$\min_{(A_1,\ldots,A_d)\in\mathbb{k}^{n_1\times r}\times\cdots\times\mathbb{k}^{n_d\times r}} \frac{1}{2}\|p(A_1,\ldots,A_d) - \mathcal{A}\|_F^2$$

always occur in a **positive-dimensional family**. This implies that the Hessian matrix at a local minimizer is **always singular**, and likewise for the Gauss–Newton approximation.

A way to overcome this problem is to formulate the approximation and decomposition problem as an **optimization problem over rank-1 tensors** rather than over factor matrices. If r-identifiability holds, the positive-dimensional family of equivalent representations disappears.

Given

- a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, and
- a target rank $r \in \mathbb{N}$,

find a minimizer of

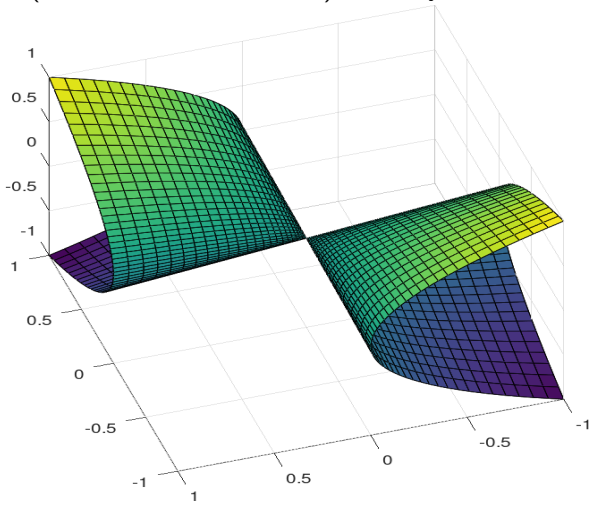
$$\min_{(\mathcal{B}_1,\ldots,\mathcal{B}_r)\in\mathcal{S}\times\cdots\times\mathcal{S}} \frac{1}{2}\left\|\mathcal{B}_1 + \cdots + \mathcal{B}_r - \mathcal{A}\right\|_F^2, \tag{TAP}$$

where $\mathcal{S}$ is the set of all rank-1 tensors:

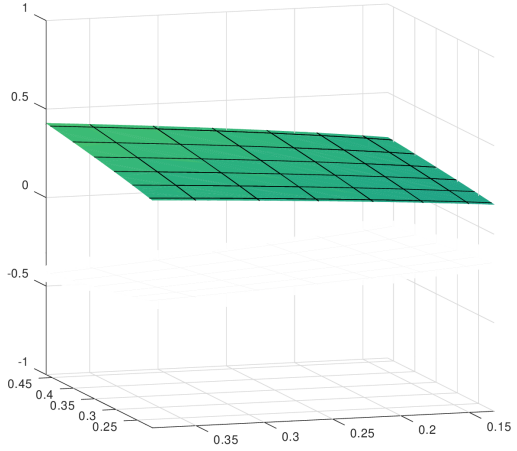
$$\mathcal{S} = \{\mathcal{A} \mid \operatorname{rank}(\mathcal{A}) = 1\}.$$

Breiding and Vannieuwenhoven (2018b; c) introduced this formulation as the **tensor rank approximation problem (TAP)**.

Before we can continue, we need to known more about the set of **rank**-1 **tensors** $\mathcal{S}$. It is the smooth **Segre manifold** (Harris, 1992; Lee, 2013) Globally it looks like a curved object ...

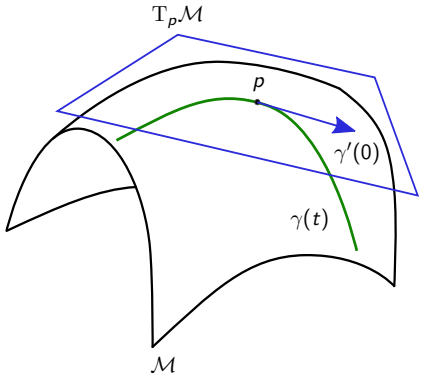


. . . but zooming in ...



... it **locally** looks like a 2-dimensional linear space! For a 2-dimensional manifold, this is true at every point.

A **tangent vector** to an embedded submanifold $\mathcal{M} \subset \mathbb{R}^n$ at p is a vector $t_p \in \mathbb{R}^n$ such that there exists a **smooth curve** $\gamma(t) \subset \mathcal{M}$, $t \in (-1, 1)$, for which $p = \gamma(0)$ and $t_p = \gamma'(0)$.



The **tangent space** $\mathrm{T}_p\mathcal{M} \subset \mathbb{R}^n$ is the set of all tangent vectors. For an m-dimensional manifold it is an m-**dimensional linear subspace**. (It can also be equipped with the inner product from $\mathbb{R}^n$)

Riemannian optimization solves **constrained optimization problems**

$$\min_{x \in \mathcal{M}} f(x)$$

where

1. the **constraint set** $\mathcal{M}$ is a smooth manifold, and
2. the **objective function** $f : \mathcal{M} \to \mathbb{R}$ is a smooth map.

See Absil, Mahoney and Sepulchre (2008) and Boumal (2020).

Continuous optimization methods for minimizing $f : \mathbb{R}^n \to \mathbb{R}$ perform the following steps:

Continuous optimization

1. Choose a starting point x_0;
2. For $k \leftarrow 1, 2, 3, \ldots$
   - 3.a Determine a **search direction** t_k;
   - 3.b Determine a **step length** α_k;
   - 3.c **Go to** $x_{k+1} \leftarrow x_k + \alpha_k t_k$.

Continuous optimization methods mainly differ in the choice of search direction:

method	**search direction** t_k
steepest descent	$-\nabla_{x_k} f$
conjugate gradient	$-\nabla_{x_k} f + \beta_k t_{k-1}$
Newton	$-(\nabla^2_{x_k} f)^{-1}(\nabla_{x_k} f)$

Recall that Newton's method is based on a truncated series expansion of f:

$$f(x_k + t_k) = f(x_k) + t_k^T(\nabla_{x_k} f) + \frac{1}{2} t_k^T(\nabla_{x_k}^2 f) t_k + o(\|\delta\|^2)$$

The minimum is achieved where the gradient

$$\dot{\delta} \mapsto \dot{\delta}^T(\nabla_{x_k} f) + \dot{\delta}^T(\nabla_{x_k}^2 f) t_k$$

vanishes identically. Hence, we need to take

$$t_k = -(\nabla_{x_k}^2 f)^{-1}(\nabla_{x_k} f),$$

insofar as the **Hessian matrix** $\nabla_{x_k}^2 f$ is invertible.

While Newton's method has great **quadratic local convergence**, a plain Newton method is not suitable because:

1. the Hessian matrix can **fail to be invertible**,
2. the Hessian matrix might not be positive definite so the search direction **can fail to be a descent direction**,
3. it has no guaranteed **global convergence**, and
4. the Hessian matrix and its inverse can be **expensive to compute**.

These problems are addressed by

1. **modifying the Hessian matrix** so it is always positive definite,
2. incorporating a **trust region** or **line search** scheme, and
3. using cheap approximations of the true Newton direction $-(\nabla^2_{x_k} f)^{-1}(\nabla_{x_k} f)$ through iterative **Krylov methods**.

Riemannian nonlinear least squares problems

The TAP is a specific type of Riemannian optimization problem, called a **nonlinear least squares problem** because the objective function can be written as

$$f : \mathcal{M} \to \mathbb{R}, \quad x \mapsto \frac{1}{2}\|F(x)\|^2$$

for some smooth map $F : \mathcal{M} \to \mathbb{R}^N$.

Since the TAP is

$$\min_{(\mathcal{B}_1,\ldots,\mathcal{B}_r)\in\mathcal{S}\times\cdots\times\mathcal{S}} \frac{1}{2}\,\|\mathcal{B}_1 + \cdots + \mathcal{B}_r - \mathcal{A}\|_F^2\,,$$

this is indeed a nonlinear least squares problem. A default choice for solving Riemannian nonlinear least squares problems is the **Riemannian Gauss–Newton method**.

In this setting, the **gradient** of a least-squares objective function f at x is

$$\nabla_x f = (\mathrm{d}_x F)^T(F(x)),$$

where $\mathrm{d}_x F : \mathrm{T}_x\mathcal{M} \to \mathbb{R}^N$ is the derivative (or **Jacobian matrix** in coordinates) of $F : \mathcal{M} \to \mathbb{R}^N$.

The **Riemannian Hessian matrix** of f is

$$\nabla_x^2 f = (\mathrm{d}_x F)^T (\mathrm{d}_x F) + \langle \mathrm{d}_x(\mathrm{d}_x F), F(x) \rangle.$$

Near a solution, we hope to have $F(\mathrm{x}^*) \approx 0$, so that the last term often has a negligible contribution.

This reasoning leads to the **Gauss–Newton approximation** of the Riemannian Hessian matrix

$$\nabla_x^2 f \approx (\mathrm{d}_x F)^T (\mathrm{d}_x F) =: G_x.$$

This matrix is always positive semidefinite!

Replacing the Hessian with the Gauss–Newton approximation yields **local linear convergence**. If $f(\mathrm{x}^*) = 0$ at a solution, then the local convergence is quadratic.

Trust region globalization

The method sketched thus far has no global convergence guarantees. Furthermore, the Gauss–Newton approximation of the Hessian could be very close to singular, resulting in large updates.

The **trust region globalization** scheme can solve both of these problems. The idea is to trust the local Gauss–Newton model at x_k,

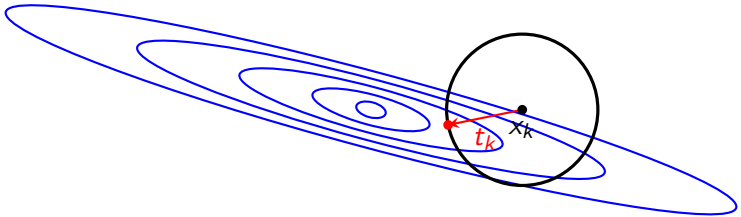
$$m(x_k + t_k) = f(x_k) + t_k^T(\nabla_{x_k} f) + \frac{1}{2} t_k^T G_{x_k} t_k,$$

only in a **small neighborhood** of radius Δ_k around x_k.

Instead of solving for the unconstrained minimizer of $t_k \mapsto m(x_k + t_k)$, a trust region method solves the **trust region subproblem**:

$$\min_{t_k \in \mathrm{T}_{x_k}\mathcal{M}} m(x_k + t_k) \quad \text{subject to } \|t_k\| \leq \Delta_k,$$

where $\Delta_k > 0$ is the **trust region radius**.



The trust region radius is modified in every step according to a fixed scheme. Assume that the update direction is t_k with $\|t_k\| \leq \Delta_k$.

The **trustworthiness** of the Gauss–Newton model is defined as

$$\rho_k = \frac{f(x_k) - f(x_k + t_k)}{m(x_k) - m(x_k + t_k)}.$$

The trust region radius is updated as follows:

- If the trustworthiness ρ_k is high (e.g., $\rho_k \geq 0.75$) and $\|t_k\| \approx \Delta_k$, then the trust region radius is increased (e.g, $\Delta_{k+1} = 2\Delta_k$).
- If the trustworthiness ρ_k is very low (e.g., $\rho_k \leq 0.25$), then the trust region radius is decreased (e.g., $\Delta_{k+1} = \Delta_k/4$).

For (approximately) solving the trust region subproblem one can exploit the following fact. If the unconstrained minimizer

$$t_k^* = -G_{x_k}^{-1}(\nabla_{x_k} f) = (\mathrm{d}_{x_k} F)^\dagger F(x_k)$$

lies in the trusted region, $\|t_k^*\| \leq \Delta_k$, then this is the optimal solution of the trust region subproblem.

Otherwise, there exists a regularizer $\lambda > 0$ such that the optimal solution t_k^* satisfies

$$(G_{x_k} + \lambda I) t_k^* = -(\mathrm{d}_{x_k} F)^T (F(\mathrm{x}_k))$$

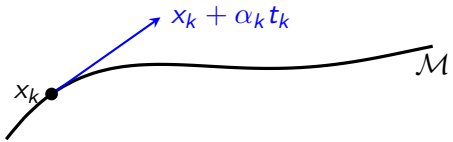
with $\|t_k^*\| = \Delta_k$. Nocedal and Wright (2006) discuss strategies for finding λ.

Stepping in the search direction

The trust region scheme simultaneously determines

- the **search direction** t_k and
- the **step length** $\alpha_k = 1$.

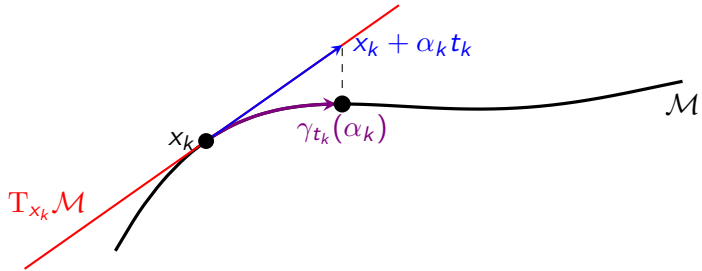
Next, we need to move x_k in the direction of t_k. However, we cannot simply add $x_k + \alpha_k t_k$ as in the Euclidean case ...



The way to generalize this to manifolds is to **construct a smooth curve** $\gamma_{t_k}(t)$ such that

$$\gamma_{t_k}(0) = x_k \text{ and } \gamma'_{t_k}(0) = t_k,$$

and that the selection of the curve it itself smooth in (x_k, t_k). **The exponential map** satisfies these conditions, among others. Other maps that satisfy these properties are called **retraction operators**.



Given a retraction operator γ for the Segre manifold $\mathcal{S}$, a retraction operator Γ for the product manifold $\mathcal{S}^{\times r} = \mathcal{S} \times \cdots \times \mathcal{S}$ at $(\mathcal{A}_1, \ldots, \mathcal{A}_r)$ is

$$\Gamma_{\dot{\mathcal{I}}_1, \ldots, \dot{\mathcal{I}}_r}(t) := \left(\gamma_{\dot{\mathcal{I}}_1}(t), \ldots, \gamma_{\dot{\mathcal{I}}_r}(t)\right),$$

which is called the **product retraction**.

Known **retraction operators for the rank-1 tensors** $\mathcal{S}$ are

- rank-$(1, \ldots, 1)$ T-HOSVD and ST-HOSVD (Kressner, Steinlechner, Vandereycken, 2014);
- the exponential map (Swijsen, Van der Veken, Vannieuwenhoven, 2021).

Skeleton of Riemannian Gauss–Newton method with trust region

Algorithm 2: Riemannian Gauss–Newton method outline

input : A tensor $\mathcal{A} \in \mathbb{k}^{n_1 \times \cdots \times n_d}$.
input : A target rank r.
output: Rank-1 tensors $(\mathcal{A}_1, \ldots, \mathcal{A}_r)$ of a CPD approximating $\mathcal{A}$.

Choose random initial points $\mathcal{A}_i \in \mathcal{S}$;
Let $x_1 \leftarrow (\mathcal{A}_1, \ldots, \mathcal{A}_r)$, and set $k \leftarrow 0$;
Choose a trust region radius $\Delta > 0$;
while *not converged* **do**
    Solve the trust region subproblem, resulting in $t_k \in \mathrm{T}_{\mathfrak{a}}\mathcal{S}^{\times r}$;
    Compute the tentative next iterate $x_{k+1} \leftarrow \Gamma_{t_k}(1)$;
    Accept or reject the next iterate. If the former, increment k;
    Update the trust region radius Δ_k;
end

A practical implementation

In Breiding and Vannieuwenhoven (2018b), we implemented a Riemannian Gauss–Newton method for solving the TAP.

The method was implemented as described above, with the following choices:

1. The trust region subproblem is solved with the **dogleg method**;
2. **hot restarts randomization** was added to escape high-condition areas; and
3. retraction with fast compressed rank-$(1, \ldots, 1)$ ST-HOSVD approximation.

The **dogleg step** approximates the optimal solution t_k^* of the trust region subproblem by

$$t_k = \begin{cases} t_{\mathrm{N}} = -(\mathrm{d}_{x_k} F)^{\dagger}(F(x_k)) & \text{if } \|t_{\mathrm{N}}\| \leq \Delta_k \\ t_{\mathrm{C}} = -\frac{(\nabla_{x_k} f)^T G_{x_k} (\nabla_{x_k} f)}{\|(\nabla_{x_k} f)\|^2} \nabla_{x_k} f & \text{if } \|t_{\mathrm{N}}\| > \Delta_k \text{ and } \|t_{\mathrm{C}}\| \geq \Delta_k \\ t_{\mathrm{I}} := t_{\mathrm{C}} + (\tau - 1)(t_{\mathrm{N}} - t_{\mathrm{C}}) & \text{s.t. } \|t_{\mathrm{I}}\| = \Delta_k, \text{ otherwise} \end{cases}.$$

where $1 \leq \tau \leq 2$ is the solution of $\|t_{\mathrm{C}} + (\tau - 1)(t_{\mathrm{N}} - t_{\mathrm{C}})\|^2 = \Delta_k^2$.

Let

$$\Sigma_r : \mathcal{S} \times \cdots \times \mathcal{S}, \quad (\mathcal{A}_1, \ldots, \mathcal{A}_r) \mapsto \mathcal{A}_1 + \cdots + \mathcal{A}_r.$$

The Gauss–Newton direction

$$t_{\mathrm{N}} = -G_{x_k}^{-1}(\nabla_{x_k} f).$$

is vital to the dogleg step. Unfortunately, $G_{x_k} = (\mathrm{d}_{x_k} \Sigma_r)^T (\mathrm{d}_{x_k} \Sigma_r)$ can be arbitrarily close to a singular matrix (even under r-identifiability).

> *In fact, we showed that G_{x_k} is an ill-conditioned matrix if and only if the CPD represented by x_k is ill-conditioned.*

Whenever G_{x_k} is close to a singular matrix we apply **random perturbations** to the current decomposition $x_k \in \mathcal{S}^{\times r}$ until G_{x_k} is sufficiently well-behaved. We call this **hot restarting**.

Numerical experiments

Breiding and Vannieuwenhoven (2018b) extensively compared their Riemannian Gauss–Newton implementation with some **state-of-the-art Euclidean nonlinear least squares solvers** in Tensorlab v3.0 by Vervliet *et al.* (2016). Specifically, both `nls_lm` and `nls_gndl` were tested with the `LargeScale` option turned both off and on.

These alternative approaches parameterize the rank-1 tensors via factor matrices.

We consider parameterized[1] tensors in $\mathbb{R}^{n_1 \times n_2 \times n_3}$ with varying condition numbers. There are three parameters:

1. $c \in [0, 1]$ regulates the "colinearity" of the factor matrices
2. $s \geq 1$ regulates the scaling, and
3. r is the rank.

Typically,

1. increasing c increases the condition number from Breiding and Vannieuwenhoven (2018a).
2. increasing s increases the factor matrices condition number from Vannieuwenhoven (2017).
3. increasing r decreases the probability of finding a decomposition.

[1] See the afternotes for the precise construction.

The true rank-r tensor is then

$$\mathcal{A} = \sum_{i=1}^{r} \mathsf{a}_i^1 \otimes \mathsf{a}_i^2 \otimes \mathsf{a}_i^3.$$

Finally, we normalize the tensor and add random Gaussian noise $\mathcal{E} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ of magnitude τ:

$$\mathcal{B} = \frac{\mathcal{A}}{\|\mathcal{A}\|_F} + \tau \frac{\mathcal{E}}{\|\mathcal{E}\|_F}.$$

The tensor $\mathcal{B}$ is the one we would like to approximate by a tensor of rank r.

We will choose k random starting points and then apply each of the methods to each of the starting points.

Our **performance criterion** (on a single processor) is the **expected time to success** (TTS).

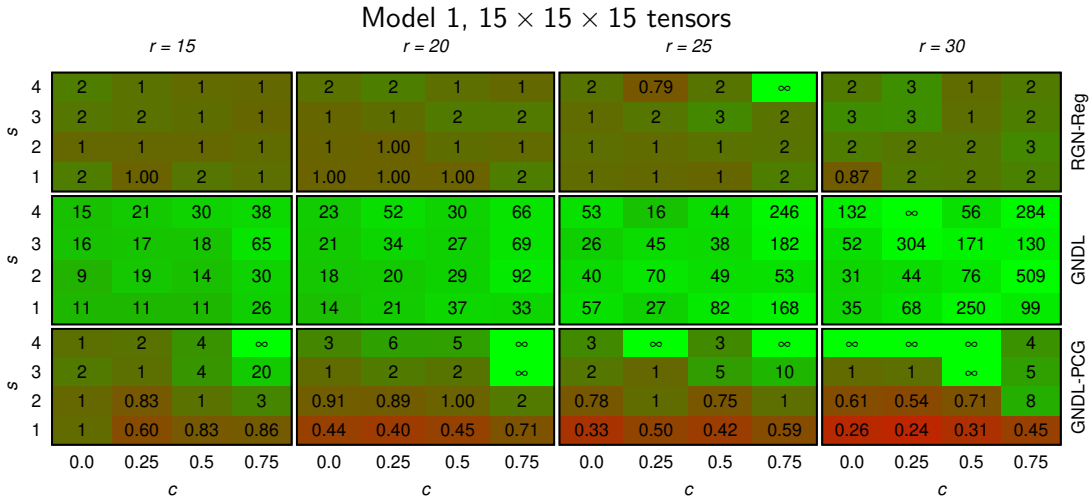
Let

1. the probability of success be p_S,
2. the probability of failure be $p_F = 1 - p_S$,
3. a successful decomposition take m_S seconds, and
4. a failed decomposition take m_F seconds.

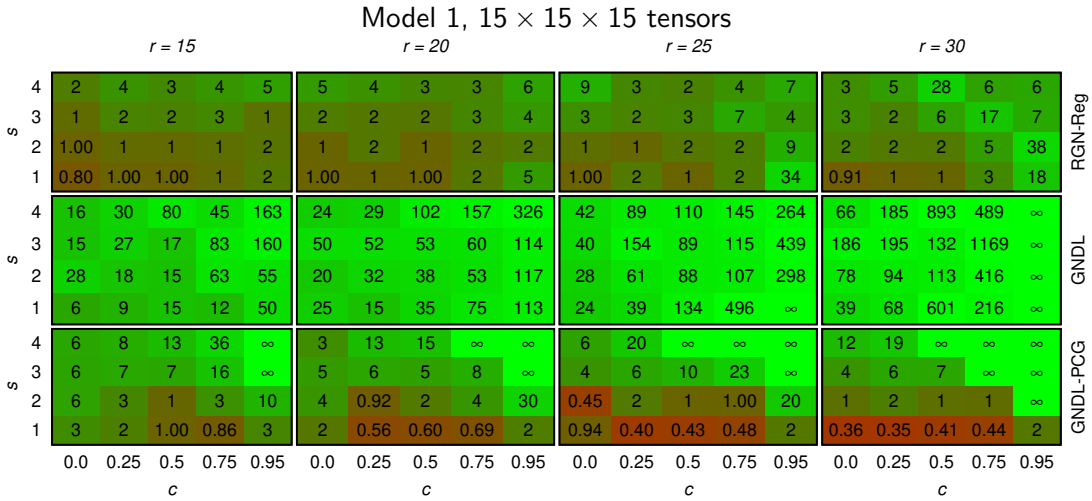
Then, the expected time to a first success is

$$\mathrm{E}[\mathrm{TTS}] = \sum_{k=0}^{\infty} p_F^{k-1} p_S (m_S + (k-1)m_F) = \frac{p_S m_S + p_F m_F}{p_S}.$$

Speedup of RGN-HR

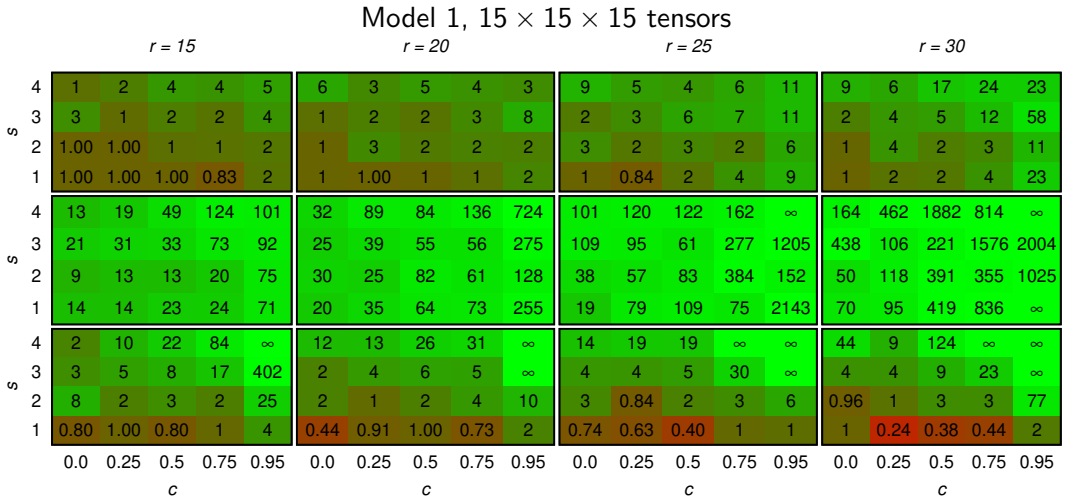


Speedup of RGN-HR



noise level $\tau = 10^{-5}$

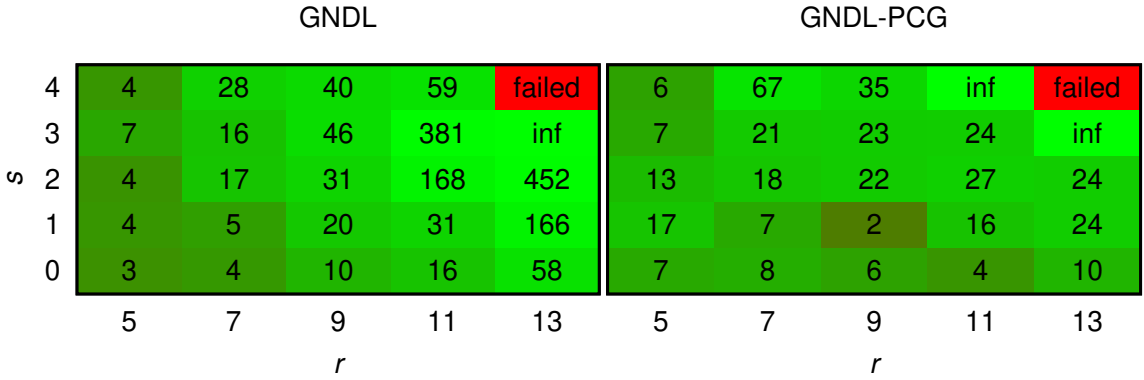
Speedup of RGN-HR



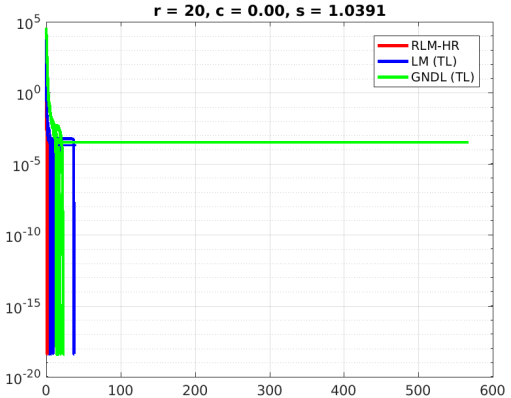
noise level $\tau = 10^{-7}$

Speedup of RGN-HR

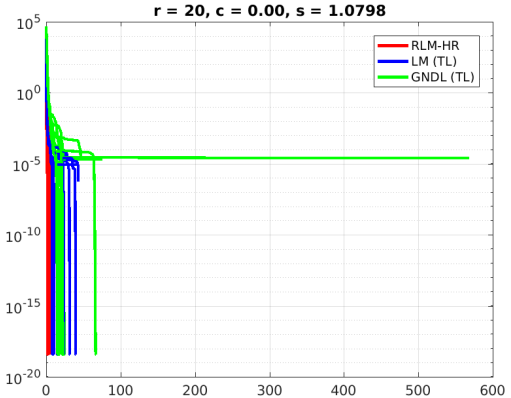
Model 2, $13 \times 11 \times 9$ tensors



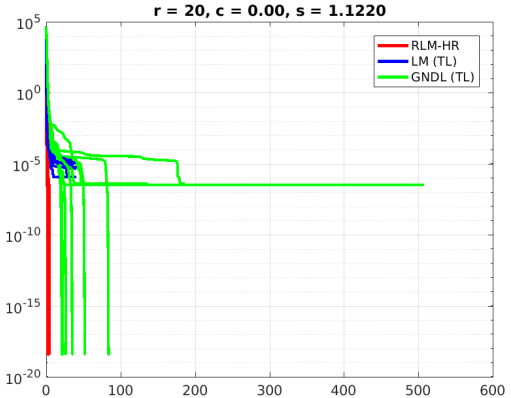
noise level $\tau = 10^{-5}$



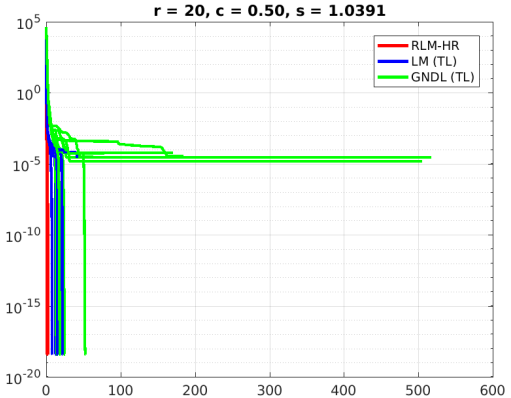
$\kappa = 3.0$	RGN-HR	LM (TL)	GNDL (TL)
Success	100%	100%	90%
E[TTS]	1.4	15.6	81.6
Speedup	-	11.1	58.3



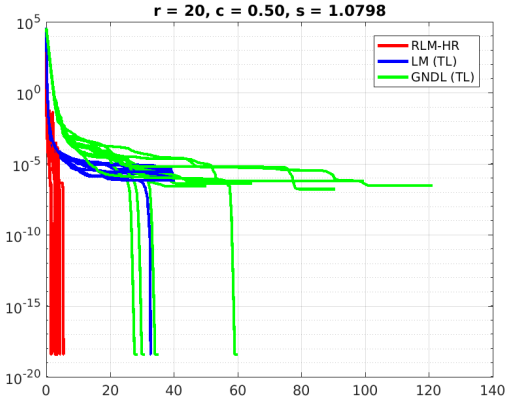
$\kappa = 3.1$	RGN-HR	LM (TL)	GNDL (TL)
Success	100%	70%	70%
E[TTS]	2.5	38.9	127.2
Speedup	-	15.6	50.9



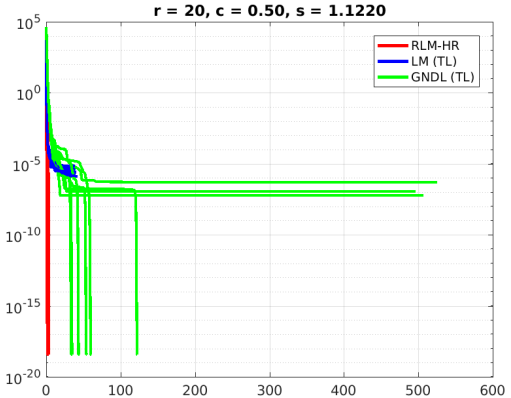
$\kappa = 3.8$	RGN-HR	LM (TL)	GNDL (TL)
Success	100%	0%	60%
E[TTS]	2.7	Inf	188.1
Speedup	-	Inf	69.7



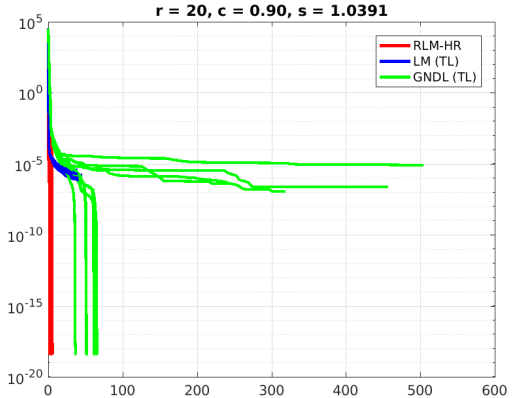
$\kappa = 14.3$	RGN-HR	LM (TL)	GNDL (TL)
Success	100%	70%	40%
E[TTS]	1.5	33.1	407.8
Speedup	-	22.1	271.9



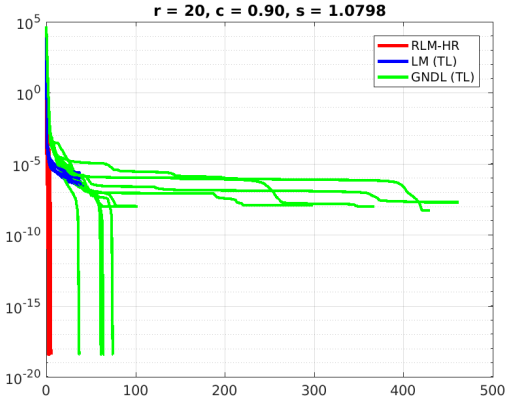
$\kappa = 13.7$	RGN-HR	LM (TL)	GNDL (TL)
Success	100%	10%	40%
E[TTS]	3.2	387.0	167.7
Speedup	-	120.9	52.4



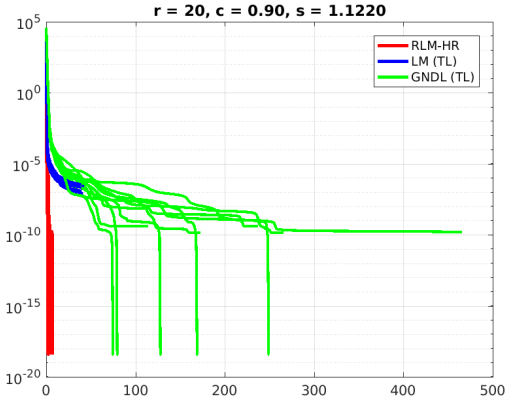
$\kappa = 9.4$	RGN-HR	LM (TL)	GNDL (TL)
Success	100%	0%	60%
E[TTS]	2.3	Inf	330.9
Speedup	-	Inf	143.9



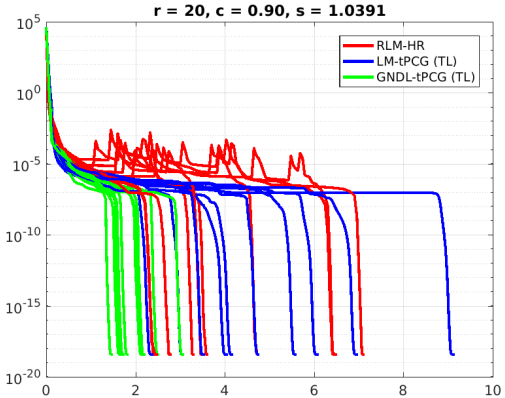
$\kappa = 123.4$	RGN-HR	LM (TL)	GNDL (TL)
Success	100%	0%	60%
E[TTS]	3.7	Inf	310.0
Speedup	-	Inf	83.8



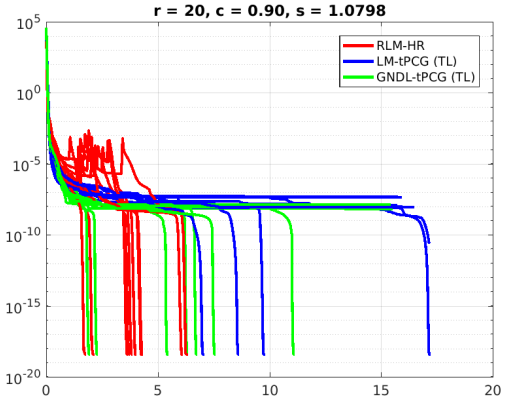
$\kappa = 114.8$	RGN-HR	LM (TL)	GNDL (TL)
Success	100%	0%	40%
E[TTS]	3.2	Inf	500.1
Speedup	-	Inf	156.3



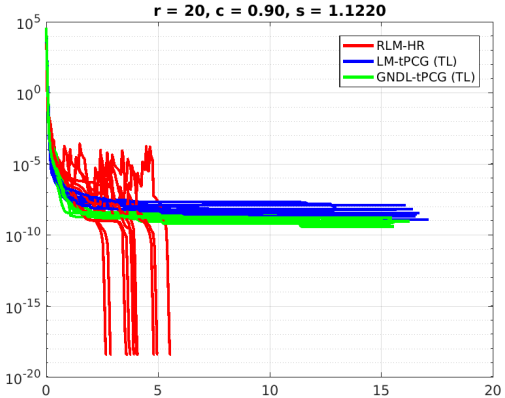
$\kappa = 130.3$	RGN-HR	LM (TL)	GNDL (TL)
Success	100%	0%	50%
E[TTS]	3.0	Inf	391.9
Speedup	-	Inf	130.6



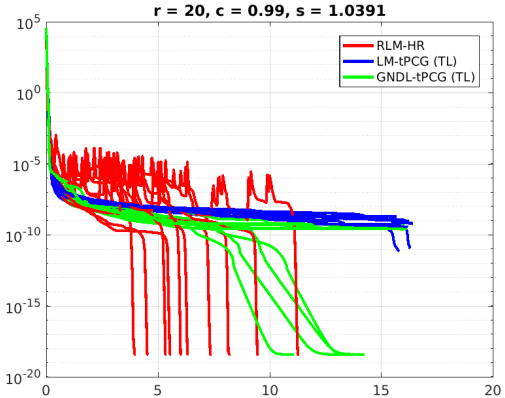
$\kappa = 190.7$	RGN-HR	LM-tPCG (TL)	GNDL-tPCG (TL)
Success	100%	100%	100%
E[TTS]	4.3	5.0	2.1
Speedup	-	1.2	0.49



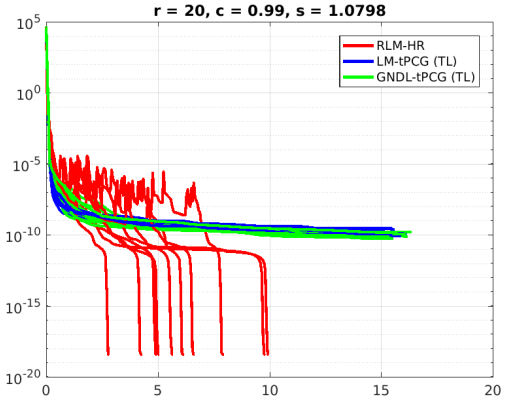
$\kappa = 223.9$	RGN-HR	LM-tPCG (TL)	GNDL-tPCG (TL)
Success	100%	40%	70%
E[TTS]	4.0	34.9	12.6
Speedup	-	8.7	3.2



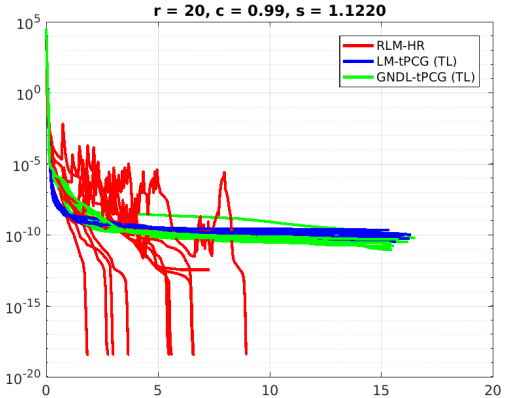
$\kappa = 132.2$	RGN-HR	LM-tPCG (TL)	GNDL-tPCG (TL)
Success	100%	0%	0%
E[TTS]	3.0	Inf	Inf
Speedup	-	Inf	Inf



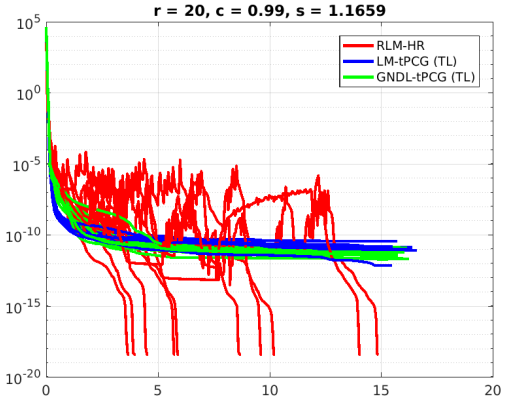
$\kappa = 3308.1$	RGN-HR	LM-tPCG (TL)	GNDL-tPCG (TL)
Success	100%	0%	40%
E[TTS]	6.9	Inf	36.8
Speedup	-	Inf	5.3



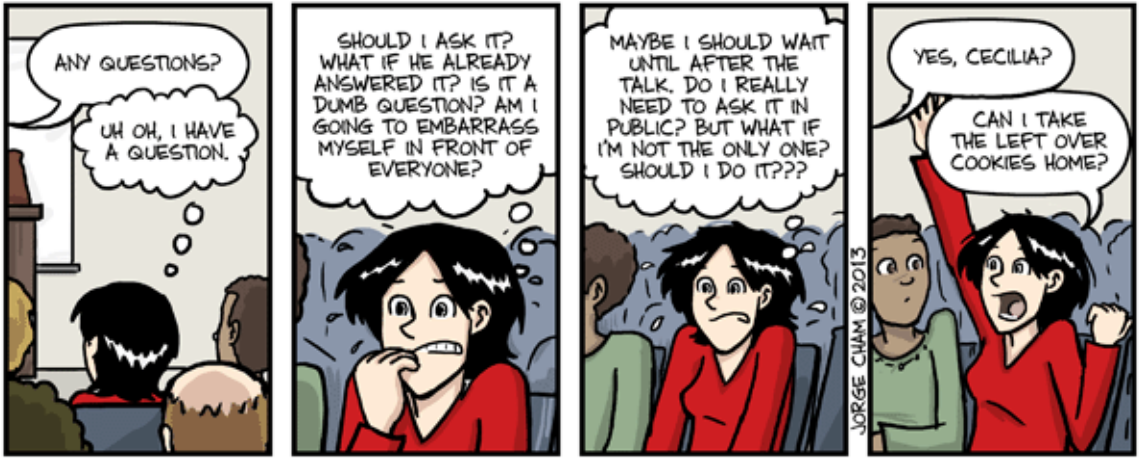
$\kappa = 5931.0$	RGN-HR	LM-tPCG (TL)	GNDL-tPCG (TL)
Success	100%	0%	0%
E[TTS]	6.3	Inf	Inf
Speedup	-	Inf	Inf



$\kappa = 4332.2$	RGN-HR	LM-tPCG (TL)	GNDL-tPCG (TL)
Success	90%	0%	0%
E[TTS]	5.8	Inf	Inf
Speedup	-	Inf	Inf



$\kappa = 6554.3$	RGN-HR	LM-tPCG (TL)	GNDL-tPCG (TL)
Success	100%	0%	0%
E[TTS]	8.2	Inf	Inf
Speedup	-	Inf	Inf


ANY QUESTIONS?
UH OH, I HAVE A QUESTION.
SHOULD I ASK IT? WHAT IF HE ALREADY ANSWERED IT? IS IT A DUMB QUESTION? AM I GOING TO EMBARRASS MYSELF IN FRONT OF EVERYONE?
MAYBE I SHOULD WAIT UNTIL AFTER THE TALK. DO I REALLY NEED TO ASK IT IN PUBLIC? BUT WHAT IF I'M NOT THE ONLY ONE? SHOULD I DO IT???
YES, CECILIA?
CAN I TAKE THE LEFT OVER COOKIES HOME?
JORGE CHAM © 2013
WWW.PHDCOMICS.COM

Overview

Conclusions

Formulating the approximation and decomposition problem as a Riemannian optimization problem on the product Segre manifold results in state-of-the-art methods for approximating a tensor by a low-rank CPD, especially for more difficult models.

Model 2, $13 \times 11 \times 9$ tensors

	GNDL					GNDL-PCG				
s \ r	5	7	9	11	13	5	7	9	11	13
4	4	28	40	59	failed	6	67	35	inf	failed
3	7	16	46	381	inf	7	21	23	24	inf
2	4	17	31	168	452	13	18	22	27	24
1	4	5	20	31	166	17	7	2	16	24
0	3	4	10	16	58	7	8	6	4	10

noise level $\tau = 10^{-5}$

References

- Absil, Mahony, Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, 2008.
- Beltrán, Breiding, Vannieuwenhoven, *Pencil-based algorithms for tensor rank decomposition are not stable*, SIAM Journal on Matrix Analysis and Applications, 2019.
- Beltrán, Breiding, Vannieuwenhoven, *The average condition number of tensor rank decomposition is infinite*, Foundations of Computational Mathematics, 2021. (accepted)
- Boumal, *An Introduction to Optimization on Smooth Manifolds*, available online, 2020.
- Breiding, Vannnieuwenhoven, *The condition number of join decompositions*, SIAM Journal on Matrix Analysis and Applications, 2018a.
- Breiding, Vannnieuwenhoven, *A Riemannian trust region method for the canonical tensor rank approximation problem*, SIAM Journal on Optimization, 2018b.
- Breiding, Vannnieuwenhoven, *Convergence analysis of Riemannian Gauss–Newton methods and its connection with the geometric condition number*, Applied Mathematics Letters, 2018c.
- Bürgisser, Cucker, *Condition: The Geometry of Numerical Algorithms*, Springer, 2013.
- Carroll, Chang, *Carroll, Chang - Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition*, Psychometrika, 1970.

- De Lathauwer, De Moor, Vandewalle, *A multilinear singular value decomposition*, SIAM Journal on Matrix Analysis and Applications, 2000.
- Harris, *Algebraic Geometry: A First Course*, Springer, 1992.
- Hirsch, *Differential Topology*, Springer, 1976.
- de Silva, Lim, *Tensor rank and the ill-posedness of the best low-rank approximation problem*, SIAM Journal on Matrix Analysis and Applications, 2008.
- Hackbusch, *Tensor Spaces and Numerical Tensor Calculus*, Springer, 2012.
- Harris, Algebraic Geometry: A First Course, Springer, 1992;
- Kressner, Steinlechner, Vandereycken, *Low-rank tensor completion by Riemannian optimization*, BIT Numerical Mathematics, 2014.
- Landsberg, *Tensors: Geometry and Applications*, AMS, 2012.
- Lee, Introduction to Smooth Manifolds, Springer, 2012.
- Nocedal, Wright, *Numerical Optimization*, 2nd edition, Springer, 2006.
- Qi, Michałek, Lim, *Complex best r-term approximations almost always exist in finite dimensions*, Applied and Computational Harmonic Analysis, 2020.

- Swijsen, Van der Veken, Vannieuwenhoven, *Tensor completion using geodesics on Segre manifolds*, arXiv:2108.00735, 2021.
- Uschmajew, *Local convergence of the Alternating Least Squares algorithm for canonical tensor approximation*, SIAM Journal on Matrix Analysis and Applications, 2012.
- Vannieuwenhoven, *Condition numbers for the tensor rank decomposition*, Linear Algebra and its Applications, 2017.
- Vannieuwenhoven, Vandebril, Meerbergen, *A new truncation strategy for the higher-order singular value decomposition*, SIAM Journal on Scientific Computing, 2012.
- Vervliet, Debals, Sorber, Van Barel, De Lathauwer, *Tensorlab 3.0*, Available online at `http://www.tensorlab.net`, 2016.