Low-rank and sparse methods for high-dimensional approximation and model order reduction

—

Lecture 6

Low-rank methods for tensor-structured equations

In this lecture, we present algorithms for computing low-rank approximations of the solution of variational problems

$$\min_{v \in V} \mathcal{J}(v),$$

where V is a tensor space.

Algorithms will be detailed for the case of tensor-structured equations

$$A(u) = b$$

where $\mathcal{J}(v)$ is a certain norm of the residual

$$\mathcal{J}(v) = \|A(v) - b\|^2,$$

or

$$\mathcal{J}(v) = \frac{1}{2}\langle Av, v\rangle - \langle b, v\rangle$$

in the case of symmetric operators.

Outline

1. Alternating minimization algorithms
2. Greedy algorithms
3. Iterative solvers with tensor truncation
4. Iterative solvers and optimization in subsets of low-rank tensors

Optimization in subsets of low-rank tensors

Let $\mathcal{M}_r$ be a subset of tensors in a certain low-rank format with a multilinear parametrization of the form

$$v(i_1, \ldots, i_d) = \sum_{k_1=1}^{r_1} \ldots \sum_{k_L=1}^{r_L} \prod_{\nu=1}^{d} p^{(\nu)}(i_\nu, (k_i)_{i\in S_\nu}) \prod_{\nu=d+1}^{M} p^{(\nu)}((k_i)_{i\in S_\nu})$$

and let

$$\mathcal{M}_r = \{v = \Psi(p^{(1)}, \ldots, p^{(M)}) : p^{(\nu)} \in P^{(\nu)}, 1 \leq \nu \leq M\},$$

where Ψ is a multilinear map.

The problem

$$\min_{v\in\mathcal{M}_r} \mathcal{J}(v)$$

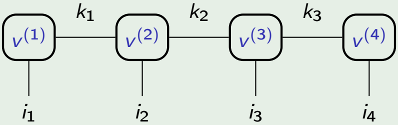
can be written as an optimization problem over the parameters

$$\min_{p^{(1)}} \ldots \min_{p^{(M)}} \mathcal{J}(\Psi(p^{(1)}, \ldots, p^{(M)})).$$

Optimization in subsets of low-rank tensors

Example 1 (Tensor-train format $\mathcal{M}_r = \mathcal{TT}_r$ in $\mathbb{R}^I$)

$$v(i_1, \dots, i_4) = \sum_{k_1=1}^{r_1} \sum_{k_2=1}^{r_2} \sum_{k_3=1}^{r_3} v^{(1)}(i_1, k_1) v^{(2)}(k_1, i_2, k_2) v^{(3)}(k_2, i_3, k_3) v^{(4)}(k_3, i_4)$$



Example 2 (Set of rank-one tensors $\mathcal{R}_1 = \mathcal{TT}_{(1,\dots,1)}$ in $\mathbb{R}^I$)

$$v(i_1, \dots, i_4) = v^{(1)}(i_1) v^{(2)}(i_2) v^{(3)}(i_3) v^{(4)}(i_4)$$

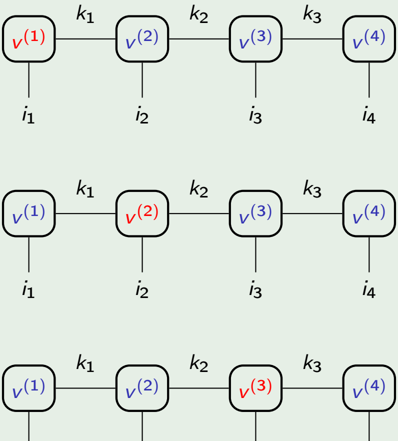
Alternating minimization algorithm

The alternating minimization algorithm consists in solving successively minimization problems

$$\min_{p^{(\nu)}} \mathcal{J}(\Psi(p^{(1)}, \dots, p^{(\nu)}, \dots, p^{(d)})) := \min_{p^{(\nu)}} \mathcal{J}_\nu(p^{(\nu)})$$

over the parameter $p^{(\nu)}$, letting the other parameters $p^{(\eta)}$, $\eta \neq \nu$, fixed.

Example 3 (Tensor-train format $\mathcal{M}_r = \mathcal{TT}_r$ in $\mathbb{R}^I$)



Alternating minimization algorithm

For a differentiable functional $\mathcal{J}$, the stationarity condition is given by

$$\langle \nabla \mathcal{J}(\Psi(p^{(1)}, \dots, p^{(\nu)}, \dots, p^{(M)})), \Psi(p^{(1)}, \dots, q^{(\nu)}, \dots, p^{(M)}) \rangle = 0 \quad \forall q^{(\nu)} \in P^{(\nu)}$$

or equivalently

$$\nabla \mathcal{J}_\nu(p^{(\nu)}) = 0.$$

Alternating minimization algorithm

Example 4 (Linear symmetric problem and approximation by a rank-one tensor)

Let consider $\mathcal{J}(v) = \frac{1}{2}\langle Av, v\rangle - \langle b, v\rangle$, where A is a symmetric operator in canonical format $A = \sum_{k=1}^{L} A_k^{(1)} \otimes \ldots \otimes A_k^{(d)}$ and $b = \sum_{k=1}^{R} b_k^{(1)} \otimes \ldots \otimes b_k^{(d)}$.

For the approximation by a rank one tensor

$$v = v^{(1)} \otimes \ldots \otimes v^{(d)}, \quad v^{(\nu)} \in \mathbb{R}^{n_\nu},$$

$$\nabla \mathcal{J}_\nu(v^{(\nu)}) = B^{(\nu)} v^{(\nu)} - c^{(\nu)} \in \mathbb{R}^{n_\nu},$$

with $B^{(\nu)} \in \mathbb{R}^{n_\nu \times n_\nu}$ and $c^{(\nu)} \in \mathbb{R}^{n_\nu}$ such that

$$B^{(\nu)} = \sum_{k=1}^{L} \alpha_k A_k^{(\nu)}, \quad \alpha_k = \prod_{\eta \neq \nu} \langle A_k^{(\eta)} v^{(\eta)}, v^{(\eta)}\rangle,$$

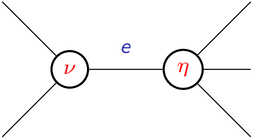
$$c^{(\nu)} = \sum_{k=1}^{R} \beta_k b_k^{(\nu)}, \quad \beta_k = \prod_{\eta \neq \nu} \langle b_k^{(\eta)}, v^{(\eta)}\rangle.$$

Modified alternating minimization algorithm

Modified alternating minimization algorithm[1] is a modification of the alternating minimization algorithm which allows for an automatic rank adaptation.

It can be used for optimization in tree-based tensor formats or more general tensor networks.

At each step of the algorithm, we consider two nodes ν and η connected by an edge e and we update simultaneously the associated parameters $p^{(\nu)}$ and $p^{(\eta)}$.



[1] known as DMRG algorithm (for Density Matrix Renormalization Group) for tensor networks.

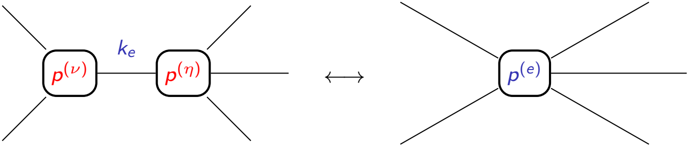
Modified alternating minimization algorithm

In the expression of a tensor $v = \Psi(p^{(1)}, \dots, p^{(M)})$, the two tensors p^{ν} and p^{η} connected by the edge e appear as

$$\sum_{k_e=1}^{r_e} p^{(\nu)}(k_e, ...)p^{(\eta)}(k_e, ...) := p^{(e)}(...)$$

where $p^{(e)}$ is a tensor of order

$$\text{order}(p^{(e)}) = \text{order}(p^{(\nu)}) + \text{order}(p^{(\eta)}) - 2.$$



This corresponds to a new tensor networks where the nodes ν and η and edge e are replaced by a single node e, and a new parametrization

$$v = \Psi^e(\dots, p^{(e)}, \dots).$$

Modified alternating minimization algorithm

We first solve an optimization problem

$$\min_{p^{(e)}} \mathcal{J}(\Psi^e(\ldots, p^{(e)}, \ldots))$$

for obtaining an new value of the tensor $p^{(e)}$.

Then, we compute a low-rank approximation of the tensor $p^{(e)}$

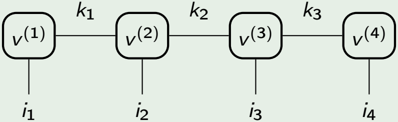
$$p^{(e)}(\ldots) \approx \sum_{k_e=1}^{r_e} p^{(\nu)}(k_e, \ldots) p^{(\eta)}(k_e, \ldots)$$

where the rank r_e in general differs from the initial rank.

In practice, the approximation is obtained using truncated singular value decomposition.

Modified alternating minimization algorithm

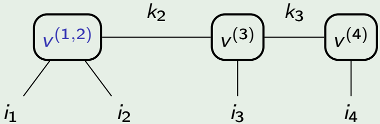
Example 5 (Modified alternating minimization for TT format)



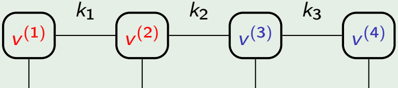
STEP 1

- Solve the optimization problem

$$v^{(1,2)} = \arg\min_{v^{(1,2)}} \mathcal{J}(\Psi^{(1,2)}(v^{(1,2)}, v^{(3)}, v^{(4)}))$$



- Truncate $v^{(1,2)}$ to obtain $v^{(1)}$ and $v^{(2)}$, with a new value of the rank r_1.



Greedy algorithms for canonical format

Standard greedy algorithms can be used to construct a sequence of approximations u^n with increasing canonical rank

$$u^n = \sum_{k=1}^{n} c_k^n w_k, \quad c_k^n \in \mathbb{R},$$

where

$$w_n = w_n^{(1)} \otimes \ldots \otimes w_n^{(d)} \in \mathcal{R}_1$$

is such that

$$w_n \in \arg \min_{w \in \mathcal{R}_1} \mathcal{J}(u^{n-1} + w), \tag{1}$$

and where the coefficients c_k^n can be either taken as $c_k^n = 1$ (for a pure greedy algorithm), or as the solution of

$$\min_{c_1, \ldots, c_n} \mathcal{J}(\sum_{k=1}^{n} c_k w_k). \tag{2}$$

Each step requires to solve an optimization problem in $\mathcal{R}_1$, for which we can rely on an alternating minimization algorithm or other optimization algorithms.

Greedy algorithms for canonical format

Example 6 (Linear equation - Greedy algorithm for canonical format)

Let consider $\mathcal{J}(v) = \|Av - b\|^2$, where A is a linear operator in canonical format $A = \sum_{k=1}^{L} A_k^{(1)} \otimes \ldots \otimes A_k^{(d)}$ and $b = \sum_{k=1}^{R} b_k^{(1)} \otimes \ldots \otimes b_k^{(d)}$.

$u^n = \sum_{l=1}^{n} c_l^n w_l \in \mathcal{R}_n$ is a tensor with canonical rank n, where $w_n = w_n^{(1)} \otimes \ldots \otimes w_n^{(d)}$ is solution of

$$\min_{w \in \mathcal{R}_1} \|A(u^{n-1} + w) - b\|^2 = \min_{w \in \mathcal{R}_1} \|Aw - b^n\|^2,$$

with $b^n = b - Au_{n-1}$ such that

$$b^n = \sum_{k=1}^{R} b_k^{(1)} \otimes \ldots \otimes b_k^{(d)} - \sum_{l=1}^{n-1} \sum_{k=1}^{L} c_l^{n-1} (A_k^{(1)} w_l^{(1)}) \otimes \ldots \otimes (A_k^{(d)} w_l^{(d)}),$$

and where $(c_1^n, \ldots, c_n^n)$ is solution of

$$\min_{c_1, \ldots, c_n} \mathcal{J}(\sum_{k=1}^{n} c_k w_k)$$

Greedy algorithms with dictionary of low-rank tensors

These algorithms are essentially used for the approximation in canonical format but $\mathcal{R}_1$ could be replaced by another subset of low-rank tensors $\mathcal{M}$ containing $\mathcal{R}_1$.

Convergence is guaranteed under quite general assumptions on $\mathcal{J}$ (strongly convex, differentiable with Lipschitz differential) and the set $\mathcal{M}$ ($\mathcal{M}$ closed, $\operatorname{span} \mathcal{M} = V$).

Greedy algorithms with a dictionary $\mathcal{R}_1$ of rank-one tensors often present a slow convergence compared to the ideal performance of n-term approximations

$$\inf_{v \in \mathcal{R}_n} \mathcal{J}(v).$$

Also, these algorithms do not really exploit the structure of tensors.

Approximation in Tucker format: a subspace point of view

The set $\mathcal{T}_r$ of tensors with Tucker rank bounded by $r = (r_1, \dots, r_d)$ is defined by

$$\mathcal{T}_r = \left\{ v = \sum_{1 \le k_1 \le r_1} \dots \sum_{1 \le k_d \le r_d} C_{k_1, \dots, k_d} v^{(1)}_{k_1} \otimes \dots \otimes v^{(d)}_{k_d} : C \in \mathbb{R}^{r_1 \times \dots \times r_d}, v^{(\nu)}_{k_\nu} \in V_\nu \right\}.$$

It can be equivalenlty parametrized by subspaces

$$\mathcal{T}_r = \{ v : v \in U_1 \otimes \dots \otimes U_d \text{ with } U_\nu \subset V_\nu, \dim(U_\nu) = r_\nu \} .$$

Then, an optimization problem on $\mathcal{T}_r$ can be interpreted as a problem of finding optimal low-dimensional spaces:

$$\min_{v \in \mathcal{T}_r} \mathcal{J}(v) = \min_{\dim(U_1) = r_1} \dots \min_{\dim(U_d) = r_d} \min_{v \in U_1 \otimes \dots \otimes U_d} \mathcal{J}(v).$$

This is a multilinear version of projection-based model order-reduction methods, where an approximation is searched in a tensor product $U_1^{r_1} \otimes \dots \otimes U_d^{r_d}$ of optimal subspaces $U_\nu^{r_\nu}$ of dimension r_ν.

Greedy algorithms for approximation in Tucker format

Greedy algorithms with a subspace point of view, which are similar to greedy algorithms for reduced basis methods, can be introduced for the construction of approximations u^n in an increasing sequence of tensor subspaces

$$U_1^n \otimes \ldots \otimes U_d^n, \quad n \geq 1,$$

with

$$U_\nu^1 \subset \ldots \subset U_\nu^n \subset \ldots, \quad 1 \leq \nu \leq d.$$

Greedy algorithms for approximation in Tucker format

At step n of these algorithms, we have an approximation u^{n-1} and associated subspaces U_ν^{n-1} of dimension r_ν^{n-1}, $1 \leq \nu \leq d$.

Assume that we have selected a set of dimensions $D_n \subset \{1, \dots, d\}$ to be enriched ($D_n = \{1, \dots, d\}$ for an isotropic enrichment).

For $\nu \notin D_n$, we let $U_\nu^n = U_\nu^{n-1}$, and for $\nu \in D_n$ we construct new spaces U_ν^n with dimension $r_\nu^n = r_\nu^{n-1} + 1$ and such that $U_\nu^n \supset U_\nu^{n-1}$.

An optimal greedy algorithm would consist in solving

$$\mathcal{J}(u^n) = \min_{\substack{\dim(U_\nu^n) = r_\nu^n \\ U_\nu^n \supset U_\nu^{n-1} \\ \nu \in D_n}} \min_{v \in U_1^n \otimes \ldots \otimes U_d^n} \mathcal{J}(v)$$

Greedy algorithms for approximation in Tucker format

A practical greedy algorithm consists in computing an optimal rank-one correction of u^{n-1}

$$\mathcal{J}(u^{n-1} + w_n^{(1)} \otimes \ldots \otimes w_n^{(d)}) = \min_{w \in \mathcal{R}_1} \mathcal{J}(u^{n-1} + w),$$

in enriching the spaces according to

$$U_\nu^n = U_\nu^{n-1} + \operatorname{span}(w_n^{(\nu)}), \quad \nu \in D_n,$$

and finally in computing the best approximation u^n in the tensor space $U_1^n \otimes \ldots \otimes U_d^n$ by solving

$$\mathcal{J}(u^n) = \min_{v \in U_1^n \otimes \ldots \otimes U_d^n} \mathcal{J}(v)$$

or

$$\min_{C \in \mathbb{R}^{r_1^n \times \ldots \times r_d^n}} \mathcal{J}\Big(\sum_{1 \le k_1 \le r_1^n} \ldots \sum_{1 \le k_d \le r_d^n} C_k v_{k_1}^{(1)} \otimes \ldots \otimes v_{k_d}^{(d)} \Big) \tag{3}$$

where $\{v_i^{(\nu)}\}_{i=1}^{r_\nu^n}$ is a basis of U_ν^n.

For high-dimensional problems, the practical solution of (3) requires a structured approximation of the tensor C, e.g. using sparse or low-rank formats. Note that if we add the constraint of having a super-diagonal tensor C, we recover a standard greedy algorithm for approximation in canonical format.

1. Alternating minimization algorithms
2. Greedy algorithms
   - Greedy algorithms for canonical format
   - Greedy algorithms for Tucker format
   - Partially greedy algorithms for Tucker format
3. Iterative solvers with tensor truncation
4. Iterative solvers and optimization in subsets of low-rank tensors

Partially greedy algorithms for Tucker format

For order-two tensors in $V_1 \otimes V_2$, greedy algorithms for Tucker format construct a sequence of spaces

$$U^n = U_1^n \otimes U_2^n,$$

with a greedy enrichment of both left and right spaces, and a corresponding sequence of rank-n approximations u^n with

$$\mathcal{J}(u^n) = \min_{v \in U_1^n \otimes U_2^n} \mathcal{J}(v) = \min_{C \in \mathbb{R}^{n\times n}} \mathcal{J}(\sum_{i,j=1}^{n} v_i^{(1)} \otimes v_j^{(2)} C_{i,j})$$

A partially greedy strategy consists in constructing a sequence of spaces

$$U^n = U_1^n \otimes V_2,$$

where only the left spaces are constructed in a greedy fashion.

Partially greedy algorithms for Tucker format

At step n, a suboptimal algorithm consists in computing a rank-one correction of u^{n-1}

$$\mathcal{J}(u^{n-1} + w_n^{(1)} \otimes w_n^{(2)}) = \min_{w^{(1)}, w^{(2)}} \mathcal{J}(u^{n-1} + w^{(1)} \otimes w^{(2)}),$$

in enriching the left subspace according to

$$U_1^n = U_1^{n-1} + \mathrm{span}(w_n^{(1)}),$$

and then in computing an approximation u^n in $U_1^n \otimes V_2$ by solving

$$\mathcal{J}(u^n) = \min_{v \in U_1^n \otimes V_2} \mathcal{J}(v) = \min_{v_1^{(2)}, \dots, v_n^{(2)}} \mathcal{J}(\sum_{i=1}^{n} v_i^{(1)} \otimes v_i^{(2)})$$

where $\{v_i^{(1)}\}_{i=1}^n$ is a basis of U_1^n.

Partially greedy algorithms for parameter-dependent equations

Consider the particular case of parameter-dependent equations

$$A\xi)u(\xi) = b(\xi), \quad \xi \in \Xi,$$

where Ξ is equipped with a measure P_ξ, and

$$u \in L^2_{P_\xi}(\Xi; V) = \overline{V \otimes L^2_{P_\xi}(\Xi)}.$$

Let

$$\mathcal{J}(v) = \int_\Xi \|A(y)u(y) - b(y)\|^2 P_\xi(dy).$$

A partially greedy algorithm constructs a sequence of approximations

$$u^n(\xi) = \sum_{i=1}^n v_i \lambda_i(\xi)$$

where $V_n = \text{span}\{v_1, \ldots, v_n\}$ is an increasing sequence of subspaces in V, and

$$\mathcal{J}(u^n) = \arg \min_{v \in V_n \otimes L^2_{P_\xi}(\Xi)} \mathcal{J}(v) = \min_{\lambda_1, \ldots, \lambda_n} \mathcal{J}(\sum_{i=1}^n v_i \lambda_i).$$

Partially greedy algorithms for parameter-dependent equations

With an optimal greedy strategy, v_n and the approximation u^n are obtained by solving

$$\mathcal{J}(u^n) = \min_{v_n} \min_{\lambda_1,\ldots,\lambda_n} \mathcal{J}\Big(\sum_{i=1}^{n-1} v_i \lambda_i + v_n \otimes \lambda_n\Big).$$

With a suboptimal greedy strategy, v_n is first computed by solving

$$\mathcal{J}(u^{n-1} + v_n \otimes \lambda_n) = \min_{v,\lambda} \mathcal{J}(u^{n-1} + v \otimes \lambda).$$

and then the approximation u^n is defined as the best approximation in $V_n \otimes L^2_{P_\xi}(\Xi)$.

Partially greedy algorithms for parameter-dependent equations

If no approximation is introduced for functions in $L^2_\mu(\Xi)$, the obtained approximation $u^n(x)$ is no more than the Minimal Residual Galerkin projection of $u(x)$ in V_n, defined by

$$u^n(x) = \arg\min_{v \in V_n} \|A(x)v - b(x)\|.$$

Note that once a subspace V_n has been constructed, the approximation $u^n(x) \in V_n$ could be defined by other projection methods (Bubnov-Galerkin or Petrov-Galerkin projections) according to

$$\langle w, A(x)u^n(x) - b(x) \rangle = 0 \quad \forall w \in W_n.$$

Outline

Iterative solvers with tensor truncation

Another strategy for solving an operator equation

$$Au = b$$

or a more general optimization problem

$$\min_{v \in V} \mathcal{J}(v)$$

is to rely on classical iterative solvers by interpreting all standard algebraic operations on vector spaces as algebraic operations in tensor spaces.

Iterative solvers with tensor truncation

As a motivating example, consider a simple Richardson algorithm

$$u^n = u^{n-1} - \omega(Au^{n-1} - b).$$

For A and b given in low-rank formats, computing u^n involves standard algebraic operations.

However, the representation rank of the iterates dramatically increases since

$$\text{rank}(u^n) = \text{rank}(A)\,\text{rank}(u^{n-1}) + \text{rank}(u^{n-1}) + \text{rank}(b).$$

This requires additional truncation steps for reducing the ranks of the iterates, such as

$$u^n = T(u^{n-1} - \omega(Au^{n-1} - b)),$$

where $T(v)$ provides a low-rank approximation of v.

We now analyze the behavior of these algorithms depending on the properties of the truncation operator T.

Fixed point iterations algorithm

Let us consider a problem which can be written as a fixed point problem

$$F(u) = u,$$

where $F : V \to V$ is a contractive map, such that for all $u, v \in V$,

$$\|F(u) - F(v)\| \leq \rho \|u - v\|,$$

with $0 \leq \rho < 1$.

Then, consider the fixed point iterations algorithm

$$u^{n+1} = F(u^n)$$

which provides a sequence $(u^n)_{n \geq 1}$ which converges to u, such that

$$\|u - u^n\| \leq \rho^n \|u - u^0\|.$$

Example 7

For a problem $Au = b$, consider $F(u) = u - \omega(Au - b)$, with ω such that $\|I - \omega A\| < 1$. Fixed point iterations $u^{n+1} = u^n - \omega(Au^n - b)$ correspond to Richardson iterations.

Perturbed fixed point iterations algorithm

Now consider the perturbed fixed point iterations

$$v^{n+1} = F(u^n), \quad u^{n+1} = T(v^{n+1})$$

where T is a mapping which for a tensor v provides an approximation (called truncation) $T(v)$ in a certain low-rank format $\mathcal{M}_r$.

Truncations with controlled relative precision

Suppose that the mapping T provides an approximation with relative precision ϵ, i.e.

$$\|T(v) - v\| \leq \epsilon \|v\|.$$

This is made possible by using an adaptation of the ranks.

Then the sequence $(u^n)_{n\geq 1}$ is such that

$$\|u - u^n\| \leq \gamma^n \|u - u^0\| + \frac{\epsilon}{1-\gamma}\|u\|,$$

with $\gamma = \rho(1+\epsilon)$. Therefore, if $\gamma < 1$

$$\limsup_{n\to\infty} \|u - u^n\| \leq \frac{\epsilon}{1-\gamma}\|u\|$$

which means that the sequence tends to enter a neighborhood of u with radius $\frac{\epsilon}{1-\gamma}\|u\|$.

The drawback of this algorithm is that the ranks of the iterates are not controlled and may become very high during the iterations.

Truncations in fixed subsets

Now consider that the mapping T provides an approximation in a fixed subset of tensors $\mathcal{M}_r$ with rank bounded by r.

Let us assume that for all v, $T(v)$ provides a quasi-optimal approximation of v such that

$$\|T(v) - v\| \leq C \min_{w \in \mathcal{M}_r} \|v - w\|. \tag{4}$$

A practical realization of a mapping T verifying (4) is provided by truncated higher-order singular value decompositions, where

$$C = O(\sqrt{d}).$$

Truncations in fixed subsets

Let u_r be an element of best approximation of u, with

$$\|u - u_r\| = \min_{v \in \mathcal{M}_r} \|u - v\|.$$

The sequence $(u^n)_{n \geq 1}$ is such that

$$\|u - u^n\| \leq \gamma^n \|u - u^0\| + \frac{C}{1-\gamma} \|u - u_r\|,$$

with $\gamma = \rho(1 + C)$. If $\gamma < 1$ (which may be quite restrictive on ρ), we obtain

$$\limsup_{n \to \infty} \|u - u^n\| \leq \frac{C}{1-\gamma} \min_{v \in \mathcal{M}_r} \|u - v\|,$$

which means that the sequence tends to enter a neighborhood of u with radius $\frac{C}{1-\gamma}\sigma_r$, where σ_r is the best approximation error of u by elements of $\mathcal{M}_r$.

An advantage of this approach is that the ranks of the iterates are controlled. A drawback is that the condition $\gamma < 1$ imposes to rely on an iterative solver with small contractivity constant $\rho < (1 + C)^{-1}$, which may be quite restrictive (requires good preconditioners).

Truncations with non-expansive maps

Now we assume that the mapping T providing an approximation in low-rank format is non-expansive, i.e.

$$\|T(v) - T(w)\| \leq \|v - w\| \tag{5}$$

The sequence u^n is defined by

$$u^{n+1} = G(u^n),$$

where $G = T \circ F$ is a contractive mapping with the same contractivity constant ρ as F. Therefore, the sequence u^n converges to the unique fixed point $u^\star$ of G such that

$$G(u^\star) = u^\star,$$

with

$$\|u^\star - u^n\| \leq \rho^n \|u^\star - u^0\|.$$

The obtained approximation $u^\star$ is such that

$$(1+\rho)^{-1}\|u - T(u)\| \leq \|u - u^\star\| \leq (1-\rho)^{-1}\|u - T(u)\|.$$

A practical realization of a mapping T verifying (4) is provided by the soft singular values thresholding operator. The ranks of the iterates are not controlled. However, it is observed in practice that the ranks of iterates are usually lower than with truncations with controlled relative precision.

Outline

Iterative solvers and optimization in subsets of low-rank tensors

Advanced iterative solvers do not provide the iterates explicitly but require the solution of successive linear systems

$$B^n u^n = c^n.$$

As an example, consider an operator splitting $A = B - C$ and the fixed point iterations

$$Bu^n = Cu^{n-1} + b,$$

where B is a linear operator given in low-rank format.

As a second example, consider the Newton (or Quasi-Newton) iterations

$$B^n(u^n - u^{n-1}) = b - A(u^n),$$

where B^n is the tangent operator to $v \mapsto A(v)$ at u^n (or some approximation of the tangent operator).

Iterative solvers and optimization in subsets of low-rank tensors

Successive linear problems can be recasted as optimization problems

$$\min_{v \in V} \|B^n v - c^n\|^2$$

and the strategies based on optimization in subsets of low-rank tensors can be used.