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

Lecture 4

Low-rank tensors

Outline

Outline

Rank of order-two tensors

The rank of an order-two tensor $u \in V \otimes W$, denoted $\text{rank}(u)$, is the minimal integer r such that

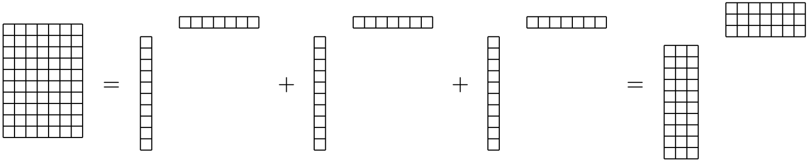
$$u = \sum_{k=1}^{r} v_k \otimes w_k$$

for some $v_k \in V$ and $w_k \in W$.

A tensor $u \in \mathbb{R}^n \otimes \mathbb{R}^m$ is identified with a matrix in $u \in \mathbb{R}^{n \times m}$. The rank of u coincides with the matrix rank, which is the minimal integer r such that

$$u = \sum_{k=1}^{r} v_k w_k^T = VW^T,$$

where $V = (v_1, \ldots, v_r) \in \mathbb{R}^{n \times r}$ and $W = (w_1, \ldots, w_r) \in \mathbb{R}^{m \times r}$.



Low-rank format for order-two tensors

The set of tensors in $V \otimes W$ with rank bounded by r, denoted

$$\mathcal{R}_r = \{v : \text{rank}(v) \leq r\},$$

is not a linear space nor a convex set. However, it has many favorable properties for a numerical use.

In particular, since the application $v \mapsto \text{rank}(v)$ is lower semi-continuous, the set $\mathcal{R}_r$ is closed, which makes best approximation problems in $\mathcal{R}_r$ well posed.

Outline

Canonical rank of higher-order tensors

For tensors $u \in V_1 \otimes \ldots \otimes V_d$ with $d \geq 3$, there are different notions of rank.

The canonical rank, which is the natural extension of the notion of rank for order-two tensors, is the minimal integer r such that

$$u = \sum_{k=1}^{r} v_k^{(1)} \otimes \ldots \otimes v_k^{(d)},$$

for some vectors $v_k^{(\nu)} \in V_\nu$.

A multivariate function $u(x_1, \ldots, x_d)$ with canonical rank bounded by r is such that

$$u(x) = \sum_{k=1}^{r} v_k^{(1)}(x_1) \ldots v_k^{(d)}(x_d),$$

where the $v_k^{(\nu)}(x_\nu)$ are in the function space V_ν.

Canonical format

The subset of tensors in $V = V_1 \otimes \ldots \otimes V_d$ with canonical rank bounded by r is denoted

$$\mathcal{R}_r = \{v \in V : \text{rank}(v) \leq r\}.$$

A tensor in $\mathcal{R}_r$ has a representation

$$v(x_1, \ldots, x_d) = \sum_{k=1}^{r} v_k^{(1)}(x_1) \ldots v_k^{(d)}(x_d) := \sum_{k=1}^{r} v^{(1)}(x_1, k) \ldots v^{(d)}(x_d, k).$$

The storage complexity of tensors in $\mathcal{R}_r$ is

$$\text{storage}(\mathcal{R}_r) = r \sum_{\nu=1}^{d} \dim(V_\nu) = O(rdn)$$

for $\dim(V_\nu) = O(n)$.

The following equivalent representation

$$v(x_1, \ldots, x_d) = \sum_{k=1}^{d} C_k v_k^{(1)}(x_1) \ldots v_k^{(d)}(x_d)$$

allows a normalization of the $v_k^{(\nu)}$.

Canonical format

For $d \geq 3$, the set $\mathcal{R}_r$ looses many of the favorable properties of the case $d = 2$.

- Determining the rank of a given tensor is a NP-hard problem.
- The set $\mathcal{R}_r$ is not an algebraic variety.
- The application $v \mapsto \text{rank}(v)$ is not lower semi-continuous and therefore, $\mathcal{R}_r$ is not closed. The consequence is that for most problems involving approximation in canonical format $\mathcal{R}_r$, there is no robust method when $d > 2$.

Example 1

Consider the 3-order tensor

$$v = a \otimes a \otimes b + a \otimes b \otimes a + b \otimes a \otimes a$$

where a and b are linearly independent vectors in $\mathbb{R}^m$. The rank of v is 3. The sequence of rank-2 tensors

$$v_n = n(a + \frac{1}{n} b) \otimes (a + \frac{1}{n} b) \otimes (a + \frac{1}{n} b) - na \otimes a \otimes a$$

converges to v as $n \to \infty$.

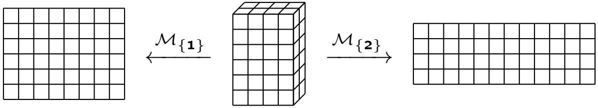
Outline

α-rank

For a non-empty subset α of $D = \{1, \dots, d\}$, a tensor $u \in V = V_1 \otimes \dots \otimes V_d$ can be identified with an order-two tensor

$$\mathcal{M}_\alpha(u) \in V_\alpha \otimes V_{\alpha^c},$$

where $V_\alpha = \bigotimes_{\nu \in \alpha} V_\nu$, and $\alpha^c = D \setminus \alpha$. The operator $\mathcal{M}_\alpha = V \to V_\alpha \otimes V_{\alpha^c}$ is called the matricisation operator.



The α-rank of u, denoted $\text{rank}_\alpha(u)$, is the rank of the order two tensor $\mathcal{M}_\alpha(u)$,

$$\text{rank}_\alpha(u) = \text{rank}(\mathcal{M}_\alpha(u)),$$

which is the minimal integer r_α such that

$$\mathcal{M}_\alpha(u) = \sum_{k=1}^{r_\alpha} v_k^\alpha \otimes w_k^{\alpha^c}$$

for some $v_k^\alpha \in V_\alpha$ and $w_k^{\alpha^c} \in V_{\alpha^c}$. We note that $\text{rank}_\alpha(u) = \text{rank}_{\alpha^c}(u)$.

α-rank

A multivariate function $u(x_1, \ldots, x_d)$ with $\mathrm{rank}_\alpha(u) \leq r_\alpha$ is such that

$$u(x) = \sum_{k=1}^{r_\alpha} v_k^\alpha(x_\alpha) w_k^{\alpha^c}(x_{\alpha^c})$$

for some functions $v_k^\alpha(x_\alpha)$ and $w_k^{\alpha^c}(x_{\alpha^c})$ of groups of variables

$$x_\alpha = \{x_\nu\}_{\nu \in \alpha} \quad \text{and} \quad x_{\alpha^c} = \{x_\nu\}_{\nu \in \alpha^c}.$$

Example 2

$u(x_1, \ldots, x_d) = u_1(x_1) + \ldots + u_d(x_d)$ where $u_1, \ldots, u_d$ are non constant functions satisfies $\mathrm{rank}_\alpha(u) = 2$ for all α.

Example 3

$u(x_1, x_2, x_3) = f(x_1) + g(x_2, x_3)$ where f and g are non constant functions satisfies $\mathrm{rank}_{\{1\}}(u) = \mathrm{rank}_{\{2,3\}}(u) = 2$, $\mathrm{rank}_{\{2\}}(u) = \mathrm{rank}_{\{1,3\}} = \mathrm{rank}(g) + 1$ and $\mathrm{rank}_{\{3\}}(u) = \mathrm{rank}_{\{2,3\}} = \mathrm{rank}(g) + 1$.

α-rank and minimal subspace

For a subset α of $D = \{1, \dots, d\}$, the minimal subspace

$$U_\alpha^{min}(u)$$

of a tensor $u \in V_1 \otimes \dots \otimes V_d$ is defined as the smallest subspace

$$U_\alpha \subset V_\alpha = \bigotimes_{\nu \in \alpha} V_\nu$$

such that

$$\mathcal{M}_\alpha(u) \in U_\alpha \otimes V_{\alpha^c}.$$

The α-rank of u is the dimension of the minimal subspace $U_\alpha^{min}(u)$,

$$\text{rank}_\alpha(u) = \dim(U_\alpha^{min}(u)).$$

Subset of tensors with bounded α-rank

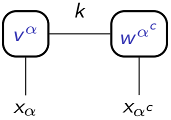
For a given subset $\alpha \subset D$, we define the subset of tensors with α-rank bounded by r_α as

$$\mathcal{T}_{r_\alpha}^{\{\alpha\}} = \{v \in V : \text{rank}_\alpha(v) \leq r_\alpha\}.$$

Elements of $\mathcal{T}_{r_\alpha}^{\{\alpha\}}$ admit the representation

$$v(x) = \sum_{k=1}^{r_\alpha} v_k^\alpha(x_\alpha) w_k^{\alpha^c}(x_{\alpha^c}) := \sum_{k=1}^{r_\alpha} v^\alpha(x_\alpha, k) w^{\alpha^c}(x_{\alpha^c}, k)$$

with order-two tensors v^α and w^{α^c}.



The corresponding storage complexity is

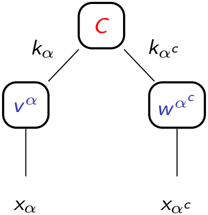
$$\text{storage}(\mathcal{T}_{r_\alpha}^{\{\alpha\}}) = r_\alpha(\prod_{\nu\in\alpha} \dim(V_\nu) + \prod_{\nu\in\alpha^c} \dim(V_\nu)) = O(r_\alpha(n^{\#\alpha} + n^{\#\alpha^c})).$$

Subset of tensors with bounded α-rank

Elements of $\mathcal{T}_{r_\alpha}^{\{\alpha\}}$ also admit the following representation

$$v(x_\alpha, x_{\alpha^c}) = \sum_{k_\alpha=1}^{r_\alpha} \sum_{k_{\alpha^c}=1}^{r_\alpha} C(k_\alpha, k_{\alpha^c}) v^\alpha(x_\alpha, k_\alpha) w^{\alpha^c}(x_{\alpha^c}, k_{\alpha^c})$$

where $C \in \mathbb{R}^{r_\alpha \times r_\alpha}$ and v^α and w^{α^c} are order-two tensors.



The corresponding storage complexity is

$$\text{storage}(\mathcal{T}_{r_\alpha}^{\{\alpha\}}) = r_\alpha^2 + r_\alpha\Big(\prod_{\nu\in\alpha} \dim(V_\nu) + \prod_{\nu\in\alpha^c} \dim(V_\nu)\Big) = O(r_\alpha^2 + r_\alpha(n^{\#\alpha} + n^{\#\alpha^c})).$$

Subset of tensors with bounded α-rank

The motivation behind the definition of tensor formats based on α-ranks is to benefit from the nice properties of the rank of order-two tensors.

The application $v \mapsto \text{rank}_\alpha(v)$ is lower semi-continuous and therefore, $\mathcal{T}_{r_\alpha}^{\{\alpha\}}$ is closed.

For a given $\alpha \subset D$, the determination of the α-rank of a tensor, which is equivalent to the determination of the rank an order-two tensor, is feasible.

Also, $\mathcal{T}_{r_\alpha}^{\{\alpha\}}$ is a smooth manifold.

α-ranks and related low-rank formats

For T a collection of subsets of D, we define the T-rank of a tensor v, denoted $\text{rank}_T(u)$, as the tuple

$$\text{rank}_T(v) = \{\text{rank}_\alpha(v)\}_{\alpha \in T}.$$

The subset of tensors in V with T-rank bounded by $r = (r_\alpha)_{\alpha \in T}$ is

$$\mathcal{T}_r^T = \{v \in V : \text{rank}_T(v) \leq r\} = \bigcap_{\alpha \in T} \mathcal{T}_{r_\alpha}^{\{\alpha\}}.$$

As a finite intersection of subsets $\mathcal{T}_{r_\alpha}^{\{\alpha\}}$, $\mathcal{T}_r^T$ inherits from geometrical and topological properties of the subsets $\mathcal{T}_{r_\alpha}^{\{\alpha\}}$ which are favorable for numerical simulation. In particular, $\mathcal{T}_r^T$ is closed.

α-ranks and related low-rank formats

Different choices for T yield different tensor formats, the standard formats being

- the Tucker format,
- the Tensor Train format,
- and more general tree-based (or hierarchical) Tucker formats.

Tucker format

For

$$T = \{\{1\}, \dots, \{d\}\},$$

the tuple

$$\text{rank}_T(v) = \{\text{rank}_{\{1\}}(v), \dots, \text{rank}_{\{d\}}(v)\}$$

is called the Tucker (or multilinear) rank of the tensor v.

The set of tensors with Tucker rank bounded by $r = (r_1, \dots, r_d)$, denoted

$$\mathcal{T}_r = \{v : \text{rank}_{\{\nu\}}(v) \leq r_\nu, 1 \leq \nu \leq d\},$$

is such that

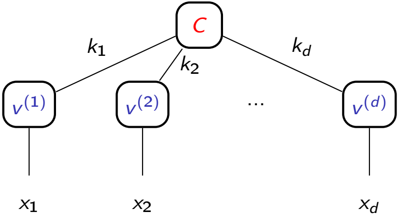
$$\mathcal{T}_r = \{v \in U_1 \otimes \dots \otimes U_d : \dim(U_\nu) = r_\nu, 1 \leq \nu \leq d\}.$$

Tucker format

A tensor in $v \in \mathcal{T}_r$ admits a representation

$$v(x_1, \ldots, x_d) = \sum_{k_1=1}^{r_1} \cdots \sum_{k_d=1}^{r_d} C(k_1, \ldots, k_d) v^{(1)}(x_1, k_1) \ldots v^{(d)}(x_d, k_d).$$

where $C \in \mathbb{R}^{r_1 \times \cdots \times r_d}$ is an order-d tensor and the $v^{(\nu)}$ are order-two tensors.



The storage complexity is

$$\text{storage}(\mathcal{T}_r) = \prod_{\nu=1}^{d} r_\nu + \sum_{\nu=1}^{d} r_\nu \dim(V_\nu) = O(R^d + Rnd)$$

with $r_\nu = O(R)$ and $\dim(V_\nu) = O(n)$. This format still suffers from the curse of dimensionality.

Note that a tensor in canonical format admits a representation in Tucker format with a super-diagonal tensor C such that $C(k_1, \ldots, k_d) \neq 0$ only for $k_1 = k_2 = \ldots = k_d$.

Tensor train format

For

$$T = \{\{1\}, \{1,2\}, \dots, \{1, \dots, d-1\}\},$$

the tuple

$$\text{rank}_T(v) = \{\text{rank}_{\{1\}}(v), \text{rank}_{\{1,2\}}(v), \dots, \text{rank}_{\{1,\dots,d-1\}}(v)\}$$

is called the TT-rank of the tensor v.

For a tuple $r = (r_1, \dots, r_{d-1})$, the set $\mathcal{T}_r^T$ of tensors with TT-rank bounded by r is denoted

$$\mathcal{TT}_r = \{v : \text{rank}_{\{1,\dots,\nu\}} = \text{rank}_{\{\nu+1,\dots,d\}}(v) \leq r_\nu, 1 \leq \nu \leq d-1\}.$$

Tensor train format

A tensor v in $\mathcal{TT}_r$, since $\text{rank}_{\{2,\dots,d\}} \leq r_1$, has the representation

$$v(x) = \sum_{k_1=1}^{r_1} v^{(1)}_{k_1}(x_1) w^{(1)}_{k_1}(x_2, \dots, x_d),$$

where $\{v^{(1)}_{k_1}\}_{k_1=1}^{r_1}$ is a basis of $U^{min}_{\{1\}}(v)$ and $\{w^{(1)}_{k_1}\}_{k_1=1}^{r_1}$ is a basis of $U^{min}_{\{2,\dots,d\}}(v)$.

Since $\text{rank}_{\{3,\dots,d\}}(v) \leq r_2$, $w^{(1)}_{k_1}$ has the representation

$$w^{(1)}_{k_1}(x_2, \dots, x_d) = \sum_{k_2=1}^{r_2} v^{(2)}_{k_1,k_2}(x_2) w^{(2)}_{k_2}(x_3, \dots, x_d),$$

where $\{w^{(2)}_{k_2}\}_{k_2=1}^{r_2}$ is a basis of $U^{min}_{\{3,\dots,d\}}(v)$, which yields the following representation of v

$$v(x) = \sum_{k_1=1}^{r_1} \sum_{k_2=1}^{r_2} v^{(1)}_{k_1}(x_1) v^{(2)}_{k_1,k_2}(x_2) w^{(2)}_{k_2}(x_3, \dots, x_d).$$

And we proceed inductively...

Tensor train format

A tensor v in $\mathcal{TT}_r$ has a representation

$$v(x) = \sum_{k_1=1}^{r_1} \dots \sum_{k_{d-1}=1}^{r_{d-1}} v^{(1)}(x_1, k_1) v^{(2)}(k_1, x_2, k_2) \dots v^{(d)}(k_{d-1}, x_d).$$

The storage complexity of an element in $\mathcal{TT}_r$ is

$$\text{storage}(\mathcal{TT}_r) = \sum_{\nu=1}^{d} r_{\nu-1} r_\nu \dim(V_\nu) = O(dnR^2)$$

with $\dim(V_\nu) = O(n)$, $r_\nu = O(R)$. Here we use the convention $r_0 = r_d = 1$.

Exercise 1

Determine a bound of the TT-rank of

$$u(x_1, \ldots, x_d) = u_1(x_1) + \ldots + u_d(x_d)$$

Exercise 2

Determine a bound of the TT-rank of

$$u(x_1, \ldots, x_d) = u_1(x_1) + u_{1,2}(x_1, x_2) + u_2(x_2) + u_{2,3}(x_2, x_3) + \ldots + u_d(x_d)$$

as a function of the ranks $r_{i,j}$ of functions $u_{i,j}$.

Exercise 3

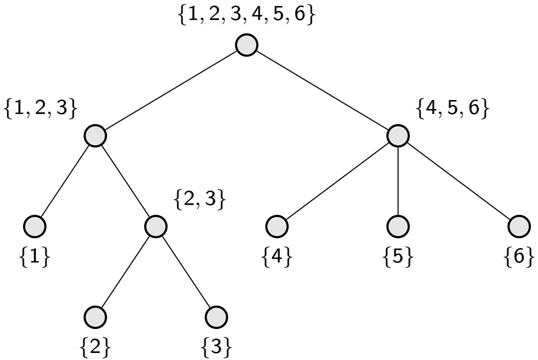
Determine a bound of the TT-rank of

$$u(x_1, \ldots, x_d) = \sum_{i \neq j} u_{i,j}(x_i, x_j)$$

as a function of the ranks $r_{i,j}$ of functions $u_{i,j}$.

Tree-based (hierarchical) Tucker format

Tree-based (or hierarchical) Tucker formats are associated with a partition dimension tree T over $D = \{1, \ldots, d\}$, with root D and leaves $\{\nu\}$, $1 \leq \nu \leq d$.



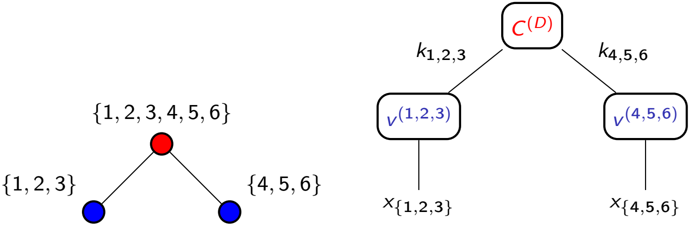
The tree-based rank of a tensor v is the tuple $\text{rank}_T(v) = (\text{rank}_\alpha(v))_{\alpha \in T}$.

Tree-based (hierarchical) Tucker format

Let v be a tensor in $\mathcal{T}_r^T$ with $r = (r_\alpha)_{\alpha \in T}$. At the first level, v admits the representation

$$v(x) = \sum_{k_{\beta_1}=1}^{r_{\beta_1}} \dots \sum_{k_{\beta_s}=1}^{r_{\beta_s}} C^{(D)}(k_{\beta_1}, \dots, k_{\beta_s}) v^{(\beta_1)}(x_{\beta_1}, k_{\beta_1}) \dots v^{(\beta_s)}(x_{\beta_s}, k_{\beta_s}).$$

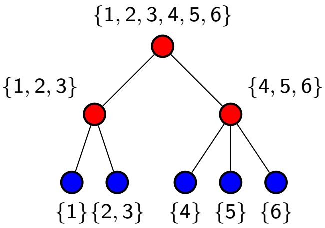
where $\{\beta_1, \dots, \beta_s\} = S(D)$ are the children of the root node D.

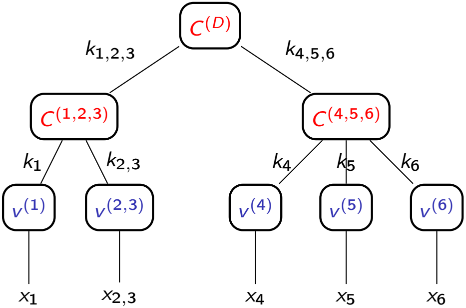


Tree-based (hierarchical) Tucker format

Then, for an interior node α of the tree, with children $S(\alpha) = \{\beta_1, \ldots, \beta_s\}$, the tensor v^α admits the representation

$$v^\alpha(x_\alpha, k_\alpha) = \sum_{k_{\beta_1}=1}^{r_{\beta_1}} \cdots \sum_{k_{\beta_s}=1}^{r_{\beta_s}} C^{(\alpha)}(k_\alpha, k_{\beta_1}, \ldots, k_{\beta_s}) v^{(\beta_1)}(x_{\beta_1}, k_{\beta_1}) \ldots v^{(\beta_s)}(x_{\beta_s}, k_{\beta_s}).$$



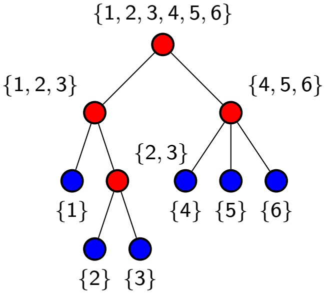


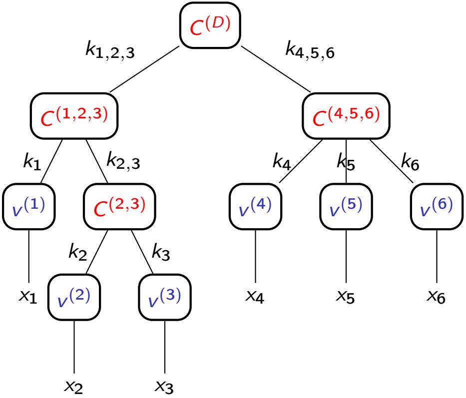
Tree-based (hierarchical) Tucker format

Finally, denoting by $\mathcal{L}(T) = \{\{\nu\} : \nu \in D\}$ the leaves of the tree, the tensor v admits the Tucker-like representation

$$v(x) = \sum_{\substack{1\le k_\nu \le r_\nu \\ \nu\in\{1,\dots,d\}}} \Big(\sum_{\substack{1\le k_\alpha \le r_\alpha \\ \alpha\in T\setminus\mathcal{L}(T)}} \prod_{\mu\in T\setminus\mathcal{L}(T)} C^{(\mu)}(k_\mu, (k_\beta)_{\beta\in S(\alpha)}) \Big) v^{(1)}(x_1, k_1)\dots v^{(d)}(x_d, k_d)$$

where we use the convention $C^{(D)}_{(k_\beta)_{\beta\in S(D)}} = C^{(D)}_{1,(k_\beta)_{\beta\in S(D)}}$ and $r_D = 1$.

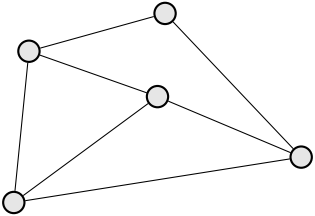




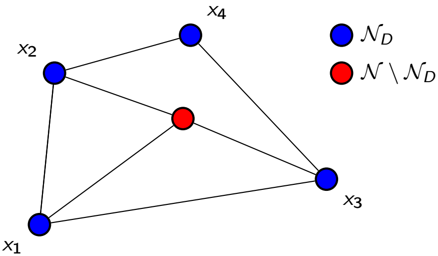
Outline

Tensor networks

Let $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ be a graph with nodes $\mathcal{N}$ and edges $\mathcal{E}$.



Let $\mathcal{N}_D$ be the subset of d nodes attached to the variables x_ν, $1 \leq \nu \leq d$.

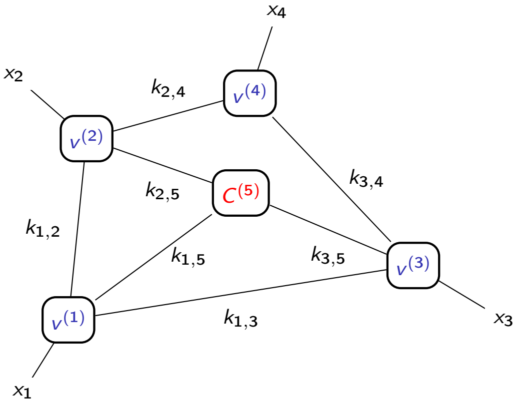


Tensor networks

A tensor v in tensor networks format has a representation of the form

$$v = \sum_{1 \leq k_1 \leq r_1} \dots \sum_{1 \leq k_{\#\mathcal{E}} \leq r_{\#\mathcal{E}}} \prod_{\nu \in \mathcal{N}_d} v^{(\nu)}(x_\nu, (k_e)_{e \in \mathcal{E}_\nu}) \prod_{\nu \in \mathcal{N} \setminus \mathcal{N}_d} C^{(\nu)}((k_e)_{e \in \mathcal{E}_\nu})$$

where $\mathcal{E}_\alpha$ are the edges connected to the node α. The tuple $r = (r_e)_{e \in \mathcal{E}}$ is called the representation rank of the tensor.



Tensor networks

The subset of tensors associated with a graph $\mathcal{G}$ and representation ranks $r = (r_e)_{e \in \mathcal{E}}$ is denoted $\mathcal{T}_r^{\mathcal{G}}$.

Tensor formats related to the notion of T-ranks with T a dimension partition tree, such as the Tucker format, the Tensor Train format or more general tree-based tensor formats, are particular cases of tensor formats $\mathcal{T}_r^{\mathcal{G}}$ where the graph $\mathcal{G}$ is a dimension partition tree or a subset of such a tree. These are called tree tensor networks.

For other types of graphs $\mathcal{G}$, tensor formats $\mathcal{T}_r^{\mathcal{G}}$ are not related to the notion of α-ranks and do not benefit from their favorable properties. In particular, it has been proved that the set of tensors $\mathcal{T}_r^{\mathcal{G}}$ where $\mathcal{G}$ has closed loops is not closed in the Zariski sense.

Canonical format is not a particular case of formats based on tensor networks.

Outline

Parametrization and storage of low-rank tensor formats

Ultimately, a tensor in a certain low-rank tensor format $\mathcal{M}_r$ admits a multilinear parametrization of the form

$$v(x_1, \dots, x_d) = \sum_{k_1=1}^{r_1} \dots \sum_{k_L=1}^{r_L} \prod_{\nu=1}^{d} p^{(\nu)}(x_\nu, (k_i)_{i \in S_\nu}) \prod_{\nu=d+1}^{M} p^{(\nu)}((k_i)_{i \in S_\nu})$$

where the parameter $p^{(\nu)}$ is an element of a tensor space $P^{(\nu)}$ which depends on a subset of summation variables $(k_i)_{i \in S_\nu} := k_{S_\nu}$.

The storage complexity is

$$\text{storage}(\mathcal{M}_r) = \sum_{\nu=1}^{d} \dim(V_\nu) \prod_{i \in S_\nu} r_i + \sum_{\nu=d+1}^{L} \prod_{i \in S_\nu} r_i.$$

If $r_i = O(R)$, $\dim(V_\nu) = O(n)$, $\#S_\nu = O(s)$ for $\nu \leq d$ and $\#S_\nu = O(s')$ for $\nu > d$, then

$$\text{storage}(\mathcal{M}_r) = O(dnR^s + (M-d)R^{s'}).$$

The key to break the curse of dimensionality is to consider low-rank formats with $s = O(1)$ and $s' = O(1)$.

Parametrization and storage of low-rank tensor formats

Examples

- Canonical format: $L = 1$, $M = d$, $S_\nu = \{1\}$ for all ν.

$$\text{storage}(\mathcal{R}_r) = O(ndR)$$

- Tucker format: $L = d$, $M = d + 1$, $S_\nu = \{\nu\}$ for $1 \leq \nu \leq d$, and $S_{d+1} = \{1, \dots, d\}$.

$$\text{storage}(\mathcal{T}_r) = O(ndR + R^d)$$

- Tensor train format: $L = d - 1$, $M = d$, $S_1 = \{1\}$, $S_d = \{d - 1\}$ and $S_\nu = \{\nu - 1, \nu\}$ for $2 \leq \nu \leq d - 1$.

$$\text{storage}(\mathcal{TT}_r) = O(ndR^2)$$

- Tree-based tensor format (for a dimension partition tree T): $L = \#T - 1$, $M = \#T$, $S_\nu = \{\nu\}$ for $1 \leq \nu \leq d$ and S_ν cointains the sons of the node $\{\nu\}$ for $\nu > d$.

$$\text{storage}(\mathcal{T}_r^T) = O(ndR + dR^{k+1})$$

  where k is the maximal number of sons of the nodes ($k = 2$ for a binary tree).

- Tensor networks: arbitrary L and M and $\#\{\nu : i \in S_\nu\} = 2$ for all $1 \leq i \leq L$.

Parametrization and storage of low-rank tensor formats

For a low-rank tensor format $\mathcal{M}_r$, there exists a multilinear map

$$\Psi : P^{(1)} \times \ldots \times P^{(M)} \to V$$

which associates to a set of parameters $\{p^{(1)}, \ldots, p^{(M)}\}$ the tensor

$$v = \Psi(p^{(1)}, \ldots, p^{(M)}).$$

Approximation in low-rank tensor formats is the first step between linear approximation and nonlinear approximation.

Sparsity in the parameters $p^{(\nu)}$ can be exploited to further reduce the complexity of the representation.

Outline

Approximation in low-rank tensor formats

Here, we present different problems involving a minimization problem

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

using low-rank tensor formats.

Then, we will present an overview of possible algorithms for the solution of such optimization problems.

Different contexts

- For the approximation of a given tensor u with respect to a certain norm,

$$\mathcal{J}(v) = \|u - v\|.$$

  Here, the aim is the compression of u or the extraction of information from u (data analysis).

- For the solution of an equation $Au = b$, the functional $\mathcal{J}(v)$ will measure some distance between u and the approximation v, e.g.

$$\mathcal{J}(v) = \|Av - b\|.$$

  The aim is here to obtain an approximation of the solution u with a low computational complexity.

Different contexts

- In tensor completion,
$$\mathcal{J}(v) = \sum_{i\in\Omega} |u(i) - v(i)|^2,$$
where $\Omega \subset I$ is a set of known entries of the tensor. The aim is here to recover (or complete) a tensor from partial information, by exploiting low-rank structures of the tensor.
- For inverse problems, where we want identify a tensor u from indirect and partial observations, the functional $\mathcal{J}(v)$ measures some distance between observations y and a prediction Av, where A is an observation map:
$$\mathcal{J}(v) = d(y, Av).$$
Exploiting low-rank structures in u allows to reduce the number of parameters to estimate and possibly makes the problem well-posed.

Different contexts

- For the estimation of the density u of a random variable X from samples $\{x_k\}_{k=1}^n$, the functional may be the log-likelihood function

$$\mathcal{J}(v) = \sum_{k=1}^{n} \log(v(x^k)).$$

  Here the aim of using low-rank approximations of the density is to make this highly ill-posed problem a well-posed problem.

- For supervised learning, where we want to learn the relation between a random variable Y and another random variable X, $\mathcal{J}(v)$ will be a risk functional of the form

$$\mathcal{J}(v) = \frac{1}{n} \sum_{k=1}^{n} \ell(y^k, v(x^k))$$

  where $\{(x^k, y^k)\}_{k=1}^n$ are samples of (X, Y) (a training set), and ℓ is a loss function such that $\ell(y, v(x))$ measures a distance between an observation y and a prediction $v(x)$ given be the approximation v.

- For least-squares approximation of a function $u(X)$,

$$\mathcal{J}(v) = \frac{1}{n} \sum_{k=1}^{n} (u(x^k) - v(x^k))^2.$$

Direct optimization in subsets of low-rank tensors

A first approach consists in solving the optimization problem

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

in a given subset of low-rank tensors $\mathcal{M}_r$.

Using the parametrization of subsets of low-rank tensors

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

with Ψ a multilinear map, the problem

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

can be recasted into an optimization problem over the parameters

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

Alternating minimization algorithm

A prominent algorithm for solving (1) is the alternating minimization algorithm (or block coordinate descent algorithm) which consists in solving successively the minimization problems

$$\min_{p^{(\nu)} \in P^{(\nu)}} \mathcal{J}(\Psi(p^{(1)}, \dots, p^{(\nu)}, \dots, p^{(M)})) := \min_{p^{(\nu)} \in P^{(\nu)}} \mathcal{J}_\nu(p^{(\nu)}) \tag{2}$$

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

When $P^{(\nu)}$ is a linear vector space, problem (2) is a linear approximation problem.

If $\mathcal{J}$ is a convex (resp. differentiable) functional, then $\mathcal{J}_\nu$ is a convex (resp. differentiable) functional.

Exploiting sparsity in the parameters

Sparsity (as well as other properties) of the parameters $p^{(\nu)}$ can be exploited at each step of the algorithm by using one of the standard strategies for sparse approximation, such as greedy algorithms, working set algorithms or convex relaxation methods.

Convex relaxation methods consists in replacing the initial optimization problem by

$$\min_{p^{(1)},\dots,p^{(M)}} \mathcal{J}(\Psi(p^{(1)},\dots,p^{(M)})) + \sum_{\nu=1}^{M} \lambda_\nu \Omega_\nu(p^{(\nu)}), \tag{3}$$

where $\Omega_\nu(p^{(\nu)})$ is a sparsity-inducing penalization term.

Applying an alternating minimization algorithm for solving (3) then yields a succession of standard optimization problems with sparsity-inducing penalization

$$\min_{p^{(\nu)}\in P^{(\nu)}} \mathcal{J}_\nu(p^{(\nu)}) + \lambda_\nu \Omega_\nu(p^{(\nu)}). \tag{4}$$

Direct optimization in subsets of low-rank tensors

Other optimization algorithms (e.g. gradient descent, Newton) can be used, possibly exploiting the geometry of low-rank tensor manifolds $\mathcal{M}_r$ (see later...).

Under rather standard assumptions, some results have been obtained for the convergence of algorithms: local convergence to a global optimizer, or global convergence to stationary points.

Up to now, there is no available algorithm for obtaining a global optimizer of a general (even convex) functional in a subset of low-rank tensors.

Greedy algorithms

A tensor $v \in \mathcal{R}_r$ with canonical rank r can be written as a sum of r rank-one tensors

$$v = \sum_{k=1}^{r} \alpha_k w_k, \quad w_k \in \mathcal{R}_1.$$

Therefore, v can be interpreted as a *n*-sparse element with respect to dictionary of rank-one tensors $\mathcal{R}_1$, which is such that

$$\mathrm{span}(\mathcal{R}_1) = V.$$

Standard greedy algorithms for sparse approximation can be used to construct a sequence of approximations $(v_r)_{r\geq 1}$ with increasing canonical rank such that $v_r \in \mathrm{span}\{w_1, \ldots, w_r\}$ with w_r the solution of the optimization problem

$$\min_{w\in\mathcal{R}_1} \mathcal{J}(v_{r-1} + w), \tag{5}$$

which can be solved using algorithms for direct optimization in $\mathcal{R}_1$.

Greedy algorithms

The dictionary $\mathcal{R}_1$ can be replaced by any subset $\mathcal{M}_r$ of low-rank tensors, with fixed rank r. Since $\mathcal{M}_r \supset \mathcal{R}_1$ for any tensor format, $\text{span}(\mathcal{M}_r) = V$ and standard greedy algorithms can be applied for the construction of a sequence of n-sparse approximations

$$v_n \in \Sigma_n = \{v = \sum_{k=1}^{n} \alpha_k w_k : \alpha_k \in \mathbb{R}, w_k \in \mathcal{M}_r\}.$$

Note that greedy algorithms for projection-based model order reduction are also greedy algorithms for low-rank approximation, but are related to the Tucker format (or subspace-based format). Greedy algorithms for the approximation of higher-order tensors in Tucker format will be presented in another lecture.

Convex relaxation methods

For a given subset of low-rank tensors $\mathcal{M}_r = \{v : \text{rank}(v) \leq r\}$, the optimization problem

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

can be replaced by the related formulation

$$\min_{v} \mathcal{J}(v) + \lambda \,\text{rank}(v).$$

Increasing λ yields solutions with lower and lower ranks.

Convex relaxation methods then consist in replacing the above problem by

$$\min_{v} \mathcal{J}(v) + \lambda \Omega(v)$$

where $\Omega(v)$ is a convex functional promoting solutions with low rank.

Convex relaxation methods for low-rank approximation of matrices

The rank of a matrix v is given by the number of non-zero singular values of the matrix. Denoting by $\sigma(v)$ the set of singular values, we have

$$\text{rank}(v) = \|\sigma(v)\|_0.$$

The functional $\Omega(v)$ can here be chosen as the convex relaxation of the function $\text{rank}(v)$,

$$\Omega(v) = \|\sigma(v)\|_1 = \|v\|_*,$$

which is known as the nuclear norm of the matrix v.

Extensions to higher order low-rank formats and functional tensors, as well as algorithms for solving the resulting non-differentiable convex optimization problems, will be presented in another lecture.