Most Tensor Problems are NP-Hard

C. J. Hillar and L.-H. Lim

Betty Shea

UBC MLRG 2020 Winter Term 1

14-Oct-2020

Today

Overview and Definitions

Where we are

- Second of two papers on the ‘Tensor Basics’ subsection

Where we are

- Second of two papers on the 'Tensor Basics' subsection
- Mark's talk: tensor definitions, operations, concepts (e.g. rank, decomposition, etc.)

Where we are

- Second of two papers on the ‘Tensor Basics’ subsection
- Mark’s talk: tensor definitions, operations, concepts (e.g. rank, decomposition, etc.)
- Today: multilinear algebra complexity

Where we are

- Second of two papers on the 'Tensor Basics' subsection
- Mark's talk: tensor definitions, operations, concepts (e.g. rank, decomposition, etc.)
- Today: multilinear algebra complexity
- Bahare's talk: Tensor factorization in graph representational learning

Theme of paper

1. 'The central message of our paper is that many problems in linear algebra that are efficiently solvable on a Turing machine become NP-hard in multilinear algebra.'

Theme of paper

1. 'The central message of our paper is that many problems in linear algebra that are efficiently solvable on a Turing machine become NP-hard in multilinear algebra.'
2. As much about the 'tractability of a numerical computing problem using the rich collection of NP-complete combinatorial problems....'

Everything is hard

Table I. Tractability of Tensor Problems

Problem	Complexity
Bivariate Matrix Functions over $\mathbb{R}$, $\mathbb{C}$	Undecidable (Proposition 12.2)
Bilinear System over $\mathbb{R}$, $\mathbb{C}$	NP-hard (Theorems 2.6, 3.7, 3.8)
Eigenvalue over $\mathbb{R}$	NP-hard (Theorem 1.3)
Approximating Eigenvector over $\mathbb{R}$	NP-hard (Theorem 1.5)
Symmetric Eigenvalue over $\mathbb{R}$	NP-hard (Theorem 9.3)
Approximating Symmetric Eigenvalue over $\mathbb{R}$	NP-hard (Theorem 9.6)
Singular Value over $\mathbb{R}$, $\mathbb{C}$	NP-hard (Theorem 1.7)
Symmetric Singular Value over $\mathbb{R}$	NP-hard (Theorem 10.2)
Approximating Singular Vector over $\mathbb{R}$, $\mathbb{C}$	NP-hard (Theorem 6.3)
Spectral Norm over $\mathbb{R}$	NP-hard (Theorem 1.10)
Symmetric Spectral Norm over $\mathbb{R}$	NP-hard (Theorem 10.2)
Approximating Spectral Norm over $\mathbb{R}$	NP-hard (Theorem 1.11)
Nonnegative Definiteness	NP-hard (Theorem 11.2)
Best Rank-1 Approximation	NP-hard (Theorem 1.13)
Best Symmetric Rank-1 Approximation	NP-hard (Theorem 10.2)
Rank over $\mathbb{R}$ or $\mathbb{C}$	NP-hard (Theorem 8.2)
Enumerating Eigenvectors over $\mathbb{R}$	#P-hard (Corollary 1.16)
Combinatorial Hyperdeterminant	NP-, #P-, VNP-hard (Theorems 4.1, 4.2, Corollary 4.3)
Geometric Hyperdeterminant	Conjectures 1.9, 13.1
Symmetric Rank	Conjecture 13.2
Bilinear Programming	Conjecture 13.4
Bilinear Least Squares	Conjecture 13.5

Note: Except for positive definiteness and the combinatorial hyperdeterminant, which apply to 4-tensors, all problems refer to the 3-tensor case.

Matrices

Matrix A over field $\mathbb{F}$ is a $m \times n$ array of elements of $\mathbb{F}$

$$A = [a_{ij}]_{i,j=1}^{m,n} \in \mathbb{F}^{m \times n}$$

Matrices

Matrix A over field $\mathbb{F}$ is a $m \times n$ array of elements of $\mathbb{F}$

$$A = [a_{ij}]_{i,j=1}^{m,n} \in \mathbb{F}^{m \times n}$$

Given standard basis $e_1, \ldots, e_d$ in $\mathbb{F}^d$, matrices are also bilinear maps.

$$f : \mathbb{F}^m \times \mathbb{F}^n \to \mathbb{F} \text{ where } a_{ij} = f(e_i, e_j) \in \mathbb{F}$$

By linearity, $f(u, v) = u^\intercal A v$.

Matrices

Matrix A over field $\mathbb{F}$ is a $m \times n$ array of elements of $\mathbb{F}$

$$A = [a_{ij}]_{i,j=1}^{m,n} \in \mathbb{F}^{m \times n}$$

Given standard basis $e_1, \ldots, e_d$ in $\mathbb{F}^d$, matrices are also bilinear maps.

$$f : \mathbb{F}^m \times \mathbb{F}^n \to \mathbb{F} \text{ where } a_{ij} = f(e_i, e_j) \in \mathbb{F}$$

By linearity, $f(u, v) = u^\intercal A v$.

If $m = n$, A is symmetric means that f is invariant under coordinate exchange:

$$f(u, v) = u^\intercal A v = (u^\intercal A v)^\intercal = v^\intercal A^\intercal u = v^\intercal A u = f(v, u)$$

where second to last equality made use of $A = A^\intercal$.

3-Tensor

3-tensor A over field $\mathbb{F}$ is an $l \times m \times n$ array of elements of $\mathbb{F}$

$$A = [a_{ijk}]_{i,j,k=1}^{l,m,n} \in \mathbb{F}^{l \times m \times n} \tag{1}$$

3-Tensor

3-tensor A over field $\mathbb{F}$ is an $l \times m \times n$ array of elements of $\mathbb{F}$

$$A = [a_{ijk}]_{i,j,k=1}^{l,m,n} \in \mathbb{F}^{l \times m \times n} \tag{1}$$

Also a trilinear map

$$f : \mathbb{F}^l \times \mathbb{F}^m \times \mathbb{F}^n \to \mathbb{F} \text{ where } a_{ijk} = f(e_i, e_j, e_k) \in \mathbb{F}$$

3-Tensor

3-tensor A over field $\mathbb{F}$ is an $l \times m \times n$ array of elements of $\mathbb{F}$

$$A = [a_{ijk}]_{i,j,k=1}^{l,m,n} \in \mathbb{F}^{l \times m \times n} \quad (1)$$

Also a trilinear map

$$f : \mathbb{F}^l \times \mathbb{F}^m \times \mathbb{F}^n \to \mathbb{F} \text{ where } a_{ijk} = f(e_i, e_j, e_k) \in \mathbb{F}$$

If $l = m = n$, A is (super-)symmetric means

$$a_{ijk} = a_{jik} = \cdots = a_{kji}$$

OR

$$f(u, v, w) = f(u, w, v) = \cdots = f(w, v, u)$$

Cubic form

Given $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$, define

$$A(x, x) = x^\intercal A x = \sum_{i,j=1}^{n} a_{ij} x_i x_j$$

Cubic form

Given $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$, define

$$A(x, x) = x^\intercal A x = \sum_{i,j=1}^{n} a_{ij} x_i x_j$$

Extended to 3-tensor $A \in \mathbb{R}^{l \times m \times n}$ when $l = m = n$, we get cubic form

$$A(x, x, x) := \sum_{i,j,k=1}^{n} a_{ijk} x_i x_j x_k \tag{2}$$

Cubic form

Given $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$, define

$$A(x,x) = x^\intercal A x = \sum_{i,j=1}^{n} a_{ij} x_i x_j$$

Extended to 3-tensor $A \in \mathbb{R}^{l \times m \times n}$ when $l = m = n$, we get cubic form

$$A(x,x,x) := \sum_{i,j,k=1}^{n} a_{ijk} x_i x_j x_k \tag{2}$$

In general, the trilinear form is

$$A(x,y,z) := \sum_{i,j,k=1}^{l,m,n} a_{ijk} x_i y_j z_k \tag{3}$$

where $x \in \mathbb{R}^l$,$y \in \mathbb{R}^m$,$z \in \mathbb{R}^k$

Inner product, outer product

Let $A = [a_{ijk}]_{i,j,k=1}^{l,m,n}$, $B = [b_{ijk}]_{i,j,k=1}^{l,m,n} \in \mathbb{R}^{l \times m \times n}$

Inner product is defined as

$$\langle A, B \rangle := \sum_{i,j,k=1}^{l,m,n} a_{ijk} b_{ijk}$$

Inner product, outer product

Let $A = [a_{ijk}]_{i,j,k=1}^{l,m,n}$, $B = [b_{ijk}]_{i,j,k=1}^{l,m,n} \in \mathbb{R}^{l \times m \times n}$

Inner product is defined as

$$\langle A, B \rangle := \sum_{i,j,k=1}^{l,m,n} a_{ijk} b_{ijk}$$

Outer product of vectors $x \in \mathbb{F}^l, y \in \mathbb{F}^m, z \in \mathbb{F}^n$, denoted $x \otimes y \otimes z$, gives 3-tensor A where

$$A = [a_{ijk}]_{i,j,k=1}^{l,m,n} \text{ where } a_{ijk} = x_i y_j z_k$$

Note: $\langle A, x \otimes y \otimes z \rangle = A(x, y, z)$ which is in the trilinear form (3)

Norms

Frobenius norm squared of 3-tensor A is defined as

$$\|A\|_F^2 := \sum_{i,j,k=1}^{l,m,n} |a_{ijk}|^2$$

Note: $\|A\|_F^2 = \langle A, A \rangle$

Norms

Frobenius norm squared of 3-tensor A is defined as

$$\|A\|_F^2 := \sum_{i,j,k=1}^{l,m,n} |a_{ijk}|^2$$

Note: $\|A\|_F^2 = \langle A, A \rangle$

Spectral norm of 3-tensor A is defined as

$$\|A\|_{2,2,2} := \sup_{x,y,z \neq 0} \frac{A(x,y,z)}{\|x\|_2 \|y\|_2 \|z\|_2} \tag{4}$$

Computation model

- Mostly follows Sipser 2012.

Computation model

- Mostly follows Sipser 2012.
- Computations on a Turing Machine. Inputs are rational numbers. Outputs are rational vectors or Yes/No.

Computation model

- Mostly follows Sipser 2012.
- Computations on a Turing Machine. Inputs are rational numbers. Outputs are rational vectors or Yes/No.
- Decision problem: the solution is in the form of Yes or No.

Computation model

- Mostly follows Sipser 2012.
- Computations on a Turing Machine. Inputs are rational numbers. Outputs are rational vectors or Yes/No.
- Decision problem: the solution is in the form of Yes or No.
- A decision problem is decidable if there is a Turing machine that will output a Yes/No for all allowable inputs in finitely many steps. Undecidable otherwise.

Complexity

Time complexity measured in units of bit operations, i.e. the number of tape-level instructions on bits. (Input size is also specified in terms of number of bits.)

Complexity

Time complexity measured in units of bit operations, i.e. the number of tape-level instructions on bits. (Input size is also specified in terms of number of bits.)

Measuring whether problems are equivalently difficult.

- Reducibility in the Cook-Karp-Levin sense.
- Very informally, problem P_1 polynomially reduces to P_2 if there is a way to solve P_1 by first solving P_2 and then translating the P_2-solution into a P_1-solution deterministically and in polynomial-time. P_2 is at least as hard as P_1.

Complexity

Time complexity measured in units of bit operations, i.e. the number of tape-level instructions on bits. (Input size is also specified in terms of number of bits.)

Measuring whether problems are equivalently difficult.

- Reducibility in the Cook-Karp-Levin sense.
- Very informally, problem P_1 polynomially reduces to P_2 if there is a way to solve P_1 by first solving P_2 and then translating the P_2-solution into a P_1-solution deterministically and in polynomial-time. P_2 is at least as hard as P_1.
- NP: Problems where solutions could be certified in polynomial time.

Complexity

Time complexity measured in units of bit operations, i.e. the number of tape-level instructions on bits. (Input size is also specified in terms of number of bits.)

Measuring whether problems are equivalently difficult.

- Reducibility in the Cook-Karp-Levin sense.
- Very informally, problem P_1 polynomially reduces to P_2 if there is a way to solve P_1 by first solving P_2 and then translating the P_2-solution into a P_1-solution deterministically and in polynomial-time. P_2 is at least as hard as P_1.
- NP: Problems where solutions could be certified in polynomial time.
- NP-complete: If one can polynomially reduce any particular NP-complete problem P_1 to a problem P_2, then all NP-complete problems are so reducible to P_2. (Cook–Levin Theorem)

Tensor Approximation

Tensor rank

Recall, the rank of a tensor $A = [a_{ijk}]_{i,j,k=1}^{l,m,n} \in \mathbb{F}^{l \times m \times n}$ is the minimum r for which A is a sum of r rank-1 tensors.

$$\text{rank}(A) := \min \left\{ r : A = \sum_{i=1}^{r} \lambda_i x_i \otimes y_i \otimes z_i \right\}$$

where $\lambda_i \in \mathbb{F}$, $x_i \in \mathbb{F}^l$, $y_i \in \mathbb{F}^m$, and $z_i \in \mathbb{F}^n$.

Tensor rank

Recall, the rank of a tensor $A = [a_{ijk}]_{i,j,k=1}^{l,m,n} \in \mathbb{F}^{l \times m \times n}$ is the minimum r for which A is a sum of r rank-1 tensors.

$$\operatorname{rank}(A) := \min \left\{ r : A = \sum_{i=1}^{r} \lambda_i x_i \otimes y_i \otimes z_i \right\}$$

where $\lambda_i \in \mathbb{F}$, $x_i \in \mathbb{F}^l$, $y_i \in \mathbb{F}^m$, and $z_i \in \mathbb{F}^n$.

1. Rank-1 tensors are tensors that could be expressed as an outer product of vectors.

Tensor rank

Recall, the rank of a tensor $A = [a_{ijk}]_{i,j,k=1}^{l,m,n} \in \mathbb{F}^{l \times m \times n}$ is the minimum r for which A is a sum of r rank-1 tensors.

$$\operatorname{rank}(A) := \min \left\{ r : A = \sum_{i=1}^{r} \lambda_i x_i \otimes y_i \otimes z_i \right\}$$

where $\lambda_i \in \mathbb{F}$, $x_i \in \mathbb{F}^l$, $y_i \in \mathbb{F}^m$, and $z_i \in \mathbb{F}^n$.

1. Rank-1 tensors are tensors that could be expressed as an outer product of vectors.
2. More than one definition of tensor 'rank': e.g. symmetric rank, border rank.

Tensor rank

Recall, the rank of a tensor $A = [a_{ijk}]_{i,j,k=1}^{l,m,n} \in \mathbb{F}^{l \times m \times n}$ is the minimum r for which A is a sum of r rank-1 tensors.

$$\text{rank}(A) := \min \left\{ r : A = \sum_{i=1}^{r} \lambda_i x_i \otimes y_i \otimes z_i \right\}$$

where $\lambda_i \in \mathbb{F}$, $x_i \in \mathbb{F}^l$, $y_i \in \mathbb{F}^m$, and $z_i \in \mathbb{F}^n$.

1. Rank-1 tensors are tensors that could be expressed as an outer product of vectors.
2. More than one definition of tensor 'rank': e.g. symmetric rank, border rank.
3. Unlike matrices, rank of a tensor changes over changing fields.

Matrix approximation

- Last week: Recommender systems with large matrix A of rank r. Netflix competition: best rank $k < r$ matrix that approximates A.

Matrix approximation

- Last week: Recommender systems with large matrix A of rank r. Netflix competition: best rank $k < r$ matrix that approximates A.
- Singular value decomposition gives a sum of rank-1 matrices

$$A = \sum_{i=1}^{r} \sigma_i u_i \otimes v_i \text{ where singular values } \sigma_1 \geq \cdots \geq \sigma_r$$

Matrix approximation

- Last week: Recommender systems with large matrix A of rank r. Netflix competition: best rank $k < r$ matrix that approximates A.
- Singular value decomposition gives a sum of rank-1 matrices

$$A = \sum_{i=1}^{r} \sigma_i u_i \otimes v_i \text{ where singular values } \sigma_1 \geq \cdots \geq \sigma_r$$

- Define $A_k = \sum_{i=1}^{k} \sigma_i u_i \otimes v_i$, $k < r$.

Matrix approximation

- Last week: Recommender systems with large matrix A of rank r. Netflix competition: best rank $k < r$ matrix that approximates A.
- Singular value decomposition gives a sum of rank-1 matrices

$$A = \sum_{i=1}^{r} \sigma_i u_i \otimes v_i \text{ where singular values } \sigma_1 \geq \cdots \geq \sigma_r$$

- Define $A_k = \sum_{i=1}^{k} \sigma_i u_i \otimes v_i$, $k < r$.
- Eckart-Young: If matrix B has rank k, then $\|A - B\|_F \geq \|A - A_k\|_F$. (Works also with $\|\cdot\|_2 = \sigma_1$ and $\|\cdot\|_* = \sigma_1 + \cdots + \sigma_r$)

Rank-r tensor approximation

Fix $\mathbb{F} = \mathbb{R}$

- Rank-r tensor approximation of tensor $A = [a_{ijk}]_{i,j,k=1}^{l,m,n}$ solves the problem

$$\min_{x_i, y_i, z_i} \|A - \lambda_1 x_1 \otimes y_1 \otimes z_1 - \cdots - \lambda_r x_r \otimes y_r \otimes z_r\|_F$$

Rank-r tensor approximation

Fix $\mathbb{F} = \mathbb{R}$

- Rank-r tensor approximation of tensor $A = [a_{ijk}]_{i,j,k=1}^{l,m,n}$ solves the problem

$$\min_{x_i, y_i, z_i} \|A - \lambda_1 x_1 \otimes y_1 \otimes z_1 - \cdots - \lambda_r x_r \otimes y_r \otimes z_r\|_F$$

- Generally unsolvable when $r > 1$ because the set $\{A : \text{rank}_{\mathbb{R}}(A) \leq r\}$ is not closed when $r > 1$. (Details for this in Silva and Lim 2008.)

Rank-r tensor approximation

Fix $\mathbb{F} = \mathbb{R}$

- Rank-r tensor approximation of tensor $A = [a_{ijk}]_{i,j,k=1}^{l,m,n}$ solves the problem

$$\min_{x_i, y_i, z_i} \|A - \lambda_1 x_1 \otimes y_1 \otimes z_1 - \cdots - \lambda_r x_r \otimes y_r \otimes z_r\|_F$$

- Generally unsolvable when $r > 1$ because the set $\{A : \text{rank}_{\mathbb{R}}(A) \leq r\}$ is not closed when $r > 1$. (Details for this in Silva and Lim 2008.)
- Where $r = 1$, however, this set is closed. So consider the smaller problem of rank-1 tensor approximation.

Rank-r tensor approximation

Fix $\mathbb{F} = \mathbb{R}$

- Rank-r tensor approximation of tensor $A = [a_{ijk}]_{i,j,k=1}^{l,m,n}$ solves the problem

$$\min_{x_i, y_i, z_i} \|A - \lambda_1 x_1 \otimes y_1 \otimes z_1 - \cdots - \lambda_r x_r \otimes y_r \otimes z_r\|_F$$

- Generally unsolvable when $r > 1$ because the set $\{A : \text{rank}_{\mathbb{R}}(A) \leq r\}$ is not closed when $r > 1$. (Details for this in Silva and Lim 2008.)

- Where $r = 1$, however, this set is closed. So consider the smaller problem of rank-1 tensor approximation.

- Problem simplifies to

$$\min_{x,y,z} \|A - x \otimes y \otimes z\|_F \tag{5}$$

Rank-1 tensor approximation

- Introduce σ where $x \otimes y \otimes z = \sigma u \otimes v \otimes w$ and $\|u\|_2 = \|v\|_2 = \|w\|_2 = 1$.

Rank-1 tensor approximation

- Introduce σ where $x \otimes y \otimes z = \sigma u \otimes v \otimes w$ and $\|u\|_2 = \|v\|_2 = \|w\|_2 = 1$.
- Problem becomes

$$\min_{u,v,w} \|A - \sigma u \otimes v \otimes w\|_F$$

$$= \|A\|_F^2 - 2\sigma\langle A, u \otimes v \otimes w\rangle + \sigma^2\|u \otimes v \otimes w\|_F^2 = \|A\|_F^2 - 2\sigma\langle A, u \otimes v \otimes w\rangle$$

Rank-1 tensor approximation

- Introduce σ where $x \otimes y \otimes z = \sigma u \otimes v \otimes w$ and $\|u\|_2 = \|v\|_2 = \|w\|_2 = 1$.
- Problem becomes

$$\min_{u,v,w} \|A - \sigma u \otimes v \otimes w\|_F$$

$$= \|A\|_F^2 - 2\sigma\langle A, u \otimes v \otimes w\rangle + \sigma^2\|u \otimes v \otimes w\|_F^2 = \|A\|_F^2 - 2\sigma\langle A, u \otimes v \otimes w\rangle$$

- Above minimized when σ

$$\sigma = \max_{\|u\|_2=\|v\|_2=\|w\|_2=1} \langle A, u \otimes v \otimes w\rangle$$

Rank-1 tensor approximation

- Introduce σ where $x \otimes y \otimes z = \sigma u \otimes v \otimes w$ and $\|u\|_2 = \|v\|_2 = \|w\|_2 = 1$.
- Problem becomes

$$\min_{u,v,w} \|A - \sigma u \otimes v \otimes w\|_F$$

$$= \|A\|_F^2 - 2\sigma\langle A, u \otimes v \otimes w\rangle + \sigma^2\|u \otimes v \otimes w\|_F^2 = \|A\|_F^2 - 2\sigma\langle A, u \otimes v \otimes w\rangle$$

- Above minimized when σ

$$\sigma = \max_{\|u\|_2=\|v\|_2=\|w\|_2=1} \langle A, u \otimes v \otimes w\rangle$$

- Because $\langle A, u \otimes v \otimes w\rangle = A(u, v, w)$, can rewrite this as finding the spectral norm (4)

$$\sigma = \|A\|_{2,2,2}$$

Rank-1 tensor approximation

- Deciding whether σ is the spectral norm of a tensor is shown in another section of the paper to be NP-hard.

Rank-1 tensor approximation

- Deciding whether σ is the spectral norm of a tensor is shown in another section of the paper to be NP-hard.
- Say we can solve the rank-1 approximation problem (5) efficiently and we get solution (x, y, z). Then we can also find the spectral norm σ by setting

$$\sigma = \|\sigma u \otimes v \otimes w\|_F = \|x\|_2 \|y\|_2 \|z\|_2$$

Rank-1 tensor approximation

- Deciding whether σ is the spectral norm of a tensor is shown in another section of the paper to be NP-hard.
- Say we can solve the rank-1 approximation problem (5) efficiently and we get solution (x, y, z). Then we can also find the spectral norm σ by setting

$$\sigma = \|\sigma u \otimes v \otimes w\|_F = \|x\|_2 \|y\|_2 \|z\|_2$$

- Finding the spectral norm is reducible to best rank-1 approximation.

Rank-1 tensor approximation

- Deciding whether σ is the spectral norm of a tensor is shown in another section of the paper to be NP-hard.
- Say we can solve the rank-1 approximation problem (5) efficiently and we get solution (x, y, z). Then we can also find the spectral norm σ by setting

$$\sigma = \|\sigma u \otimes v \otimes w\|_F = \|x\|_2\|y\|_2\|z\|_2$$

- Finding the spectral norm is reducible to best rank-1 approximation.
- So, best rank-1 approximation is also NP-hard.

Tensor Eigenvalues

Eigenpairs for matrices

Also fix $\mathbb{F} = \mathbb{R}$.

- Given symmetric $A \in \mathbb{R}^{n \times n}$, eigenvalues and eigenvectors are stationary values and points of

$$R(x) = \frac{x^\intercal A x}{x^\intercal x}$$

Eigenpairs for matrices

Also fix $\mathbb{F} = \mathbb{R}$.

- Given symmetric $A \in \mathbb{R}^{n \times n}$, eigenvalues and eigenvectors are stationary values and points of

$$R(x) = \frac{x^\intercal A x}{x^\intercal x}$$

- Equivalently, constrained maximization $\max_{\|x\|_2^2 = 1} x^\intercal A x$ with Lagrangian

$$L(x, \lambda) = x^\intercal A x - \lambda \left(\|x\|_2^2 - 1 \right)$$

Eigenpairs for matrices

Also fix $\mathbb{F} = \mathbb{R}$.

- Given symmetric $A \in \mathbb{R}^{n \times n}$, eigenvalues and eigenvectors are stationary values and points of
$$R(x) = \frac{x^\intercal A x}{x^\intercal x}$$
- Equivalently, constrained maximization $\max_{\|x\|_2^2=1} x^\intercal A x$ with Lagrangian
$$L(x, \lambda) = x^\intercal A x - \lambda \left(\|x\|_2^2 - 1 \right)$$
- The solution give the eigenvalue equation
$$Ax = \lambda x$$

Eigenpairs for 3-tensor

Conceptually, for tensor $A \in \mathbb{R}^{n \times n \times n}$, find the stationary values and points of cubic form (2)

$$A(x, x, x) := \sum_{i,j,k=1}^{n} a_{ijk} x_i x_j x_k$$

with some generalization of the unit constraint.

Eigenpairs for 3-tensor

Conceptually, for tensor $A \in \mathbb{R}^{n \times n \times n}$, find the stationary values and points of cubic form (2)

$$A(x, x, x) := \sum_{i,j,k=1}^{n} a_{ijk} x_i x_j x_k$$

with some generalization of the unit constraint.

Which generalization?

- $\|x\|_3^3 = 1$?
- $\|x\|_2^2 = 1$?
- $x_1^3 + \cdots + x_n^3 = 1$?

Eigenpairs for 3-tensor

Formally, fix $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$.

$\|x\|_2^2 = 1$: The number $\lambda \in \mathbb{F}$ is called an l^2-eigenvalue of the tensor $A \in \mathbb{F}^{n \times n \times n}$ and $0 \neq x \in \mathbb{F}^n$ its corresponding l^2-eigenvector if

$$\sum_{i,j=1}^{n} a_{ijk} x_i x_j = \lambda x_k \qquad k = 1, \ldots, n \text{ holds}$$

$\|x\|_3^3 = 1$: The number $\lambda \in \mathbb{F}$ is called an l^3-eigenvalue of the tensor $A \in \mathbb{F}^{n \times n \times n}$ and $0 \neq x \in \mathbb{F}^n$ its corresponding l^3-eigenvector if

$$\sum_{i,j=1}^{n} a_{ijk} x_i x_j = \lambda x_k^2 \qquad k = 1, \ldots, n \text{ holds}$$

Tensor eigenvalue over $\mathbb{R}$ is NP-hard

Theorem 1.3 Graph 3-colorability is polynomially reducible to tensor 0-eigenvalue over $\mathbb{R}$. Thus, deciding tensor eigenvalue over $\mathbb{R}$ is NP-hard.

Proof outline

Restrict ourselves to the $\lambda = 0$ case. Both l^2- and l^3-eigenpair equations reduce to

$$\sum_{i,j=1}^{n} a_{ijk} x_i x_j = 0 \qquad k = 1, \ldots, n \text{ holds}$$

The above becomes the square quadratic feasibility problem, which is deciding whether there is a $0 \neq x \in \mathbb{R}^n$ solution to a system of equations $\{x^\intercal A_i x = 0\}_{i=1}^m$.

By a previous result, graph 3-colorability is polynomially reducible to quadratic feasibility.

Conclusion

Hard equals interesting

'Bernd Sturmfels once made the remark to us that "All interesting problems are NP-hard." In light of this, we would like to view our article as evidence that most tensor problems are interesting.'

References

De Silva, V. & Lim, L.-H. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM J. Matrix Anal. Appl.*, **30(3)**: 1084-1127, 2008

Hillar, C.J. & Lim, L.-H.. Most tensor problems are NP-hard. *J. ACM,* **60(6)**:45:1-45:39, 2013.

Robeva, Elina. MATH605D Fall 2020 Tensor decomposition and their applications
`https://sites.google.com/view/ubc-math-605d/class-overview`

Sipser, M. *Introduction to the Theory of Computation, 3rd Ed.*, 2012.

Strang, G. *Linear Algebra and Learning From Data*, 2019.

Thank you