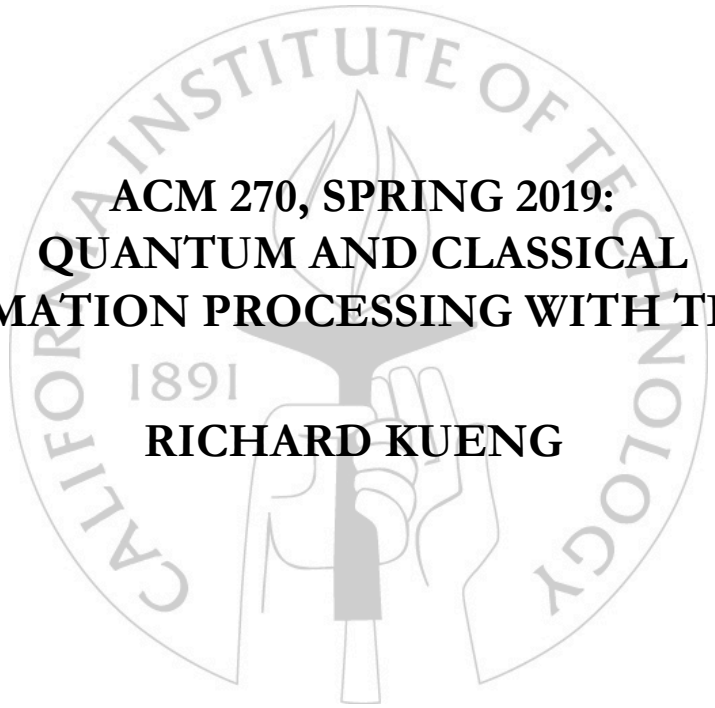


ACM 270, SPRING 2019: QUANTUM AND CLASSICAL INFORMATION PROCESSING WITH TENSORS

RICHARD KUENG

Caltech CMS Lecture Notes No. 2019-03
doi:10.7907/my9r-p178

APPLIED & COMPUTATIONAL MATHEMATICS
CALIFORNIA INSTITUTE OF TECHNOLOGY
mail code 9-94 · pasadena, ca 91125

Quantum and classical information processing with tensors

Caltech: ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp

Contents

1 Classical probability theory and quantum mechanics **1**

2 Tensor products **10**

3 Wiring calculus and entanglement **18**

4 Symmetric and antisymmetric tensors **28**

5 Haar integration **36**

6 Entanglement is ubiquitous **44**

7 Classical reversible circuits **53**

8 Quantum circuits and quantum computing **61**

9 Matrix rank **71**

10 Tensor rank **77**

11 Strassen's algorithm for matrix multiplication **86**

12 Tensorial aspects of matrix multiplication **93**

13 The CP decomposition for tensors **101**

14 The Tucker decomposition for tensors **112**

15 Tensor train decompositions I **120**

16 Tensor train decomposition II **128**

17 Tensor train algorithms (DMRG lite) **137**

Lecture 01: Classical probability theory and quantum mechanics

Scribe: Florian Schäfer

ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp
April 1, 2019

1 Agenda

1. Linear and semidefinite programming
2. Classical (discrete) probability theory
3. Postulates of quantum mechanics
4. Distinguishing classical probability distributions and the maximum likelihood rule
5. Distinguishing quantum distributions and the Holevo-Helstrom Theorem

2 Linear and semidefinite programming

In the first lecture we will show that the difference between classical probability theory and quantum mechanics is a direct analogue of the difference between linear– and semidefinite programming. We begin by introducing the latter two concepts.

2.1 Linear programming

We endow the space $\mathbb{R}^d$ with the standard inner product

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^{d} x_i y_i$$

and define the non-negative orthant

$$\mathbb{R}^d_+ = \{\boldsymbol{x} \in \mathbb{R}^d : x_i \geq 0,\ 1 \leq i \leq d\}.$$

This induces a partial order on $\mathbb{R}^d$ given by

$$\boldsymbol{x} \geq \boldsymbol{y} \ \Leftrightarrow\ \boldsymbol{x} - \boldsymbol{y} \in \mathbb{R}^d_+ \ \Leftrightarrow\ x_i \geq y_i,\ 1 \leq i \leq d.$$

A *linear program* (LP) is an optimization problem of the following form:

$$\begin{aligned} \underset{\boldsymbol{x} \in \mathbb{R}^d}{\text{maximize}} \quad & \langle \boldsymbol{a}, \boldsymbol{x} \rangle \\ \text{subject to} \quad & \langle \boldsymbol{b}_i, \boldsymbol{x} \rangle \rangle = c_i \quad 1 \leq i \leq m, \\ & \boldsymbol{x} \geq \boldsymbol{0}. \end{aligned}$$

The vectors $\boldsymbol{a}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_m \in \mathbb{R}^m$ and numbers $c_1, \ldots, c_m \in \mathbb{R}$ completely specify the problem. Problems of this form can be solved efficiently. Linear programming is a powerful technique from both an analytical and computational point of view.

2.2 Semidefinite programming

We denote the space of $d \times d$ hermitian matrices as $\mathbb{H}^d = \{\boldsymbol{X} \in \mathbb{C}^{d\times d} : \boldsymbol{X}^* = \boldsymbol{X}\}$ and endow it with the Frobenius (or Hilbert-Schmidt) inner product

$$(\boldsymbol{X}, \boldsymbol{Y}) = \mathrm{tr}(\boldsymbol{X}\boldsymbol{Y}).$$

Remark 2.1. We note that while members of $\mathbb{H}^d$ can have complex entries, $\mathbb{H}^d$ is not closed under multiplication with complex numbers and thus forms a d^2-dimensional vectorspace over the real numbers.

A matrix $\boldsymbol{X} \in \mathbb{H}^d$ is positive semidefinite (p.s.d.), if $\langle \boldsymbol{x}, \boldsymbol{X}\boldsymbol{x}\rangle \geq 0$ for all $\boldsymbol{x} \in \mathbb{C}^d$. The set of psd matrices $\mathbb{H}^d_+ \subset \mathbb{H}^d$ forms a convex cone ($\mathbb{H}^d$ is closed under convex mixtures and mutliplication with non-negative scalars). This cone induces the following partial ordering on $\mathbb{H}^d$:

$$\boldsymbol{X} \succeq \boldsymbol{Y} \Leftrightarrow \boldsymbol{X} - \boldsymbol{Y} \in \mathbb{H}^d_+.$$

We succinctly write $\boldsymbol{X} \succeq \boldsymbol{0}$ to indicate that $\boldsymbol{X} \in \mathbb{H}^d$ is psd.

A *semidefinite program* (SDP) is an optimization program of the following form

$$\begin{aligned} \underset{\boldsymbol{X}\in\mathbb{H}^d}{\text{maximize}} \quad & (\boldsymbol{A}, \boldsymbol{X}) \\ \text{subject to} \quad & (\boldsymbol{B}_i, \boldsymbol{X}) = c_i \quad 1 \leq i \leq m, \\ & \boldsymbol{X} \succeq \boldsymbol{0}. \end{aligned}$$

This optimization is completely specified by the matrices $\boldsymbol{A}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_m \in \mathbb{H}^d$ and m numbers $c_1, \ldots, c_m \in \mathbb{R}$.

Like LPs, SDPs are very useful both in theory and practice. We note that LPs and SDPs arose in totally analogous ways from the triples $(\mathbb{R}^n, \langle\cdot,\cdot\rangle, \geq)$ and $(\mathbb{H}^d, (\cdot,\cdot), \succeq)$. We will now show that the difference between classical probability theory and quantum mechanics can equally be understood as replacing the former, with the latter triple.

3 Classical, discrete probability theory

Probability theory is modeled by *probability triples* consisting of a sample space (which contains all potential outcomes), a set of events (to which we might want to assign probabilities), and a probability rule (assigning a probability to each and every event). In the setting of *discrete* probability theory, the set of all possible outcomes is finite ($|\Omega| = d$). In this case, we can simply choose the power set of Ω as the set of events and correspondingly, the probability triple is fully characterized by a *probability density vector* that assigns a probability to each outcome in Ω. Let $\mathbf{1} = (1, \ldots, 1)^T$ denote the all-ones vector in $\mathbb{R}^d$

Definition 3.1 (probability density)**.** A *probability density vector* is a vector

$$\boldsymbol{p} = \begin{pmatrix} p_1 \\ \vdots \\ p_d \end{pmatrix} \in \mathbb{R}^d : \quad \boldsymbol{p} \geq \boldsymbol{0}, \quad \langle \mathbf{1}, \boldsymbol{p}\rangle = \sum_{i=1}^{d} p_i = 1.$$

Probability theory is concerned with characterizing the likelihood of events or, equivalently, the distribution of *measurement* outcomes.

Definition 3.2 (measurement)**.** *Measurements* are resolutions of the identity (vector):

$$\{\boldsymbol{h}_a : \ a \in A\} \subset \mathbb{R}^d : \quad \boldsymbol{h}_a \geq \mathbf{0}, \ a \in A \quad \text{and} \quad \sum_{a \in A} \boldsymbol{h}_a = \mathbf{1}.$$

Here, A is a (finite) set of potential measurement outcomes.

We still need a final ingredient to describe how probability densities (as vectors in $\mathbb{R}^d$) and measurements $\{\boldsymbol{h}_a : \ a \in A\}$ relate to the probability of different measurement outcomes.

Definition 3.3 (probability rule)**.** For a probability density $\boldsymbol{p} \in \mathbb{R}^d$ and a measurement $\{\boldsymbol{h}_a : a \in A\} \subset \mathbb{R}^d$ define the *probability rule*

$$\Pr[a|\boldsymbol{p}] = \langle \boldsymbol{h}_a, \boldsymbol{p} \rangle, \quad \text{for all} \quad a \in A.$$

This assigns a probability to each possible outcome a of the measurement.

Example 3.4 (Fair dice roll)**.** The probability density of a fair dice roll is a flat distribution over 6 potential events: $\boldsymbol{p} = \frac{1}{6}\mathbf{1} \in \mathbb{R}^6$. Suppose that we wish to test whether a single dice roll results in either, $\{1, 2\}$, $\{3, 4\}$, or $\{5, 6\}$. This measurement may be associated with the following resolution of identity:

$$h_{\{1,2\}} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad h_{\{3,4\}} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad h_{\{5,6\}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix},$$

The probability rule then readily implies:

$$\Pr[\{1, 2\}|\boldsymbol{p}] = \Pr[\{3, 4\}|\boldsymbol{p}] = \Pr[\{5, 6\}|\boldsymbol{p}] = \frac{1}{3}.$$

We introduce the *probability simplex* in $\mathbb{R}^d$,

$$\Delta_{d-1} \coloneqq \Big\{ x \in \mathbb{R}^d : \boldsymbol{x} \geq \mathbf{0}, \langle \mathbf{1}, \boldsymbol{x} \rangle = 1 \Big\},$$

and observe that it equal to the convex hull of the standard basis vectors $\boldsymbol{e}_1 = (1, 0, \ldots, 0)^T, \ldots, \boldsymbol{e}_d = (0, \ldots, 0, 1)^T$:

$$\Delta_{d-1} = \operatorname{conv}\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\}.$$

Definition 3.5. A probability distribution $\boldsymbol{p} \in \Delta_{d-1}$ is called *pure*, if it is an extreme point of Δ_{d-1}. This is the case if and only if the probability distribution is deterministic.

The essential concepts of classical probability theory are summarized in Table 1

Concept	Explanation	Mathematical formulation
probability density	normalized, non-negative vectors $\boldsymbol{p} \in \mathbb{R}^d$	$\boldsymbol{p} \geq \mathbf{0}$, $\langle \mathbf{1}, \boldsymbol{p} \rangle = 1$
measurement	resolution of the identity $\{\boldsymbol{h}_a : a \in A\}$	$\boldsymbol{h}_a \geq \mathbf{0}$, $\sum_{a \in A} \boldsymbol{h}_a = \mathbf{1}$
probability rule	standard inner product	$\Pr[a|\boldsymbol{p}] = \langle \boldsymbol{h}_a, \boldsymbol{p} \rangle$

Table 1 *Axioms for classical probability theory:* The structure of discrete probability theory is captured by the following geometric configuration: $\mathbb{R}^d$ endowed with the partial order $\geq$ and the identity element $\mathbf{1} = (1, \ldots, 1)^T$. This closely resembles *linear programming*.

4 Quantum Mechanics

The postulates of quantum mechanics naturally arise from an extension of classical probability theory. Replace the triple $\left(\mathbb{R}^d, \geq, \mathbf{1}\right)$, by the triple $\left(\mathbb{H}^d, \succeq, \mathbf{I}\right)$.

The analogous object to a probability density vector is a *probability density matrix*.

Definition 4.1 (density matrix)**.** The state of a d-dimensional quantum mechanical system is fully described by a *density matrix*

$$\boldsymbol{\rho} \in \mathbb{H}^d : \quad \boldsymbol{\rho} \succeq \mathbf{0}, \quad (\mathbf{I}, \boldsymbol{\rho}) = \operatorname{tr}(\boldsymbol{\rho}) = 1.$$

In analogy to measurements in classical probability theory, we define a quantum measurement as follows.

Definition 4.2 (measurement)**.** A *measurement* is a resolution of the identity (matrix):

$$\{\boldsymbol{H}_a : a \in A\} : \quad \boldsymbol{H}_a \succeq \mathbf{0},\ a \in A, \quad \sum_{a \in A} \boldsymbol{H}_a = \mathbf{I}.$$

If a measurement $\{\boldsymbol{H}_a, a \in A\}$ is performed on a quantum mechanical system with density matrix ρ, the following two things happen.

1. (*Born's rule*) We obtain a random measurement outcome that is distributed according to
$$\mathbb{P}[a|\rho] = (\boldsymbol{H}_a, \boldsymbol{\rho}).$$
2. The quantum system ceases to exist.

The fundamental axioms of quantum mechanics are a straightforward generalization of classical probability theory, see Table 2. The transition from classical to quantum probability theory resembles a transition from linear to semidefinite programming.

Example 4.3 (Stern-Gerlach experiment)**.** Fix $d = 2$ (single "spin") and consider the density matrix

$$\boldsymbol{\rho} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

Concept	Explanation	Mathematical formulation
Probability density	normalized, psd matrix $\boldsymbol{\rho} \in \mathbb{H}^d$	$\boldsymbol{\rho} \succeq \mathbf{0}$, $(\mathbf{I}, \boldsymbol{\rho}) = 1$
measurement	resolution of the identity $\{\boldsymbol{H}_a : a \in A\}$	$\boldsymbol{H}_a \succeq \mathbf{0}$, $\sum_{a \in A} \boldsymbol{H}_a = \mathbf{I}$
probability rule	standard inner product	$\Pr[a\|\boldsymbol{\rho}] = (\boldsymbol{H}_a, \boldsymbol{\rho})$

Table 2 *Axioms for quantum mechanics:* The structure of quantum mechanics is captured by the following geometric configuration: $\mathbb{H}^d$ endowed with the psd order $\succeq$ and the identity matrix $\mathbf{I}$. This closely resembles *semidefinite programming*.

and two distinct potential measurements:

$$\Big\{\boldsymbol{H}_{\pm}^{(z)}\Big\} = \left\{\frac{1}{2}\mathbf{I} \pm \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\right\} = \left\{\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}\right\},$$
$$\Big\{\boldsymbol{H}_{\pm}^{(x)}\Big\} = \left\{\frac{1}{2}\mathbf{I} \pm \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\right\} = \left\{\frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}\right\}.$$

The resulting probabilities are then given by

$$\begin{aligned}
\mathbb{P}[+,(z)|\boldsymbol{\rho}] &= \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\right) = 1, \\
\mathbb{P}[-,(z)|\boldsymbol{\rho}] &= 0, \\
\mathbb{P}[+,(x)|\boldsymbol{\rho}] &= \left(\frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\right) = \frac{1}{2}, \\
\mathbb{P}[-,(x)|\boldsymbol{\rho}] &= \frac{1}{2}.
\end{aligned}$$

This may seem surprising. The state $\boldsymbol{\rho}$ provides completely deterministic measurement outcomes for $\Big\{\boldsymbol{H}_{\pm}^{(z)}\Big\}$. Yet, the outcomes for $\Big\{\boldsymbol{H}_{\pm}^{(x)}\Big\}$ are completely random. This interesting feature of quantum mechanics is the basis of the famous Stern-Gerlach experiment (1923).

The union of all possible quantum states form a convex set in $\mathbb{H}^d$:

$$\mathsf{S}\Big(\mathbb{H}^d\Big) = \Big\{\boldsymbol{X} \in \mathbb{H}^d : \ \boldsymbol{X} \succeq \mathbf{0},\ (\mathbb{I}, \boldsymbol{X}) = \mathrm{tr}(\boldsymbol{X}) = 1\Big\}.$$

This is the quantum analogue of the standard simplex.

Definition 4.4. A density matrix $\boldsymbol{\rho} \in \mathsf{S}(\mathbb{H}^d)$ is called *pure* if it has rank-one, i.e. $\boldsymbol{\rho} = \boldsymbol{x}\boldsymbol{x}^*$ with $\boldsymbol{x} \in \mathbb{C}^d$ normalized to unit Euclidean length.

Pure quantum states correspond to extreme points of the convex set $\mathsf{S}(\mathbb{H}^d)$ and one can show

$$\mathsf{S}(\mathbb{H}^d) = \mathrm{conv}\Big\{\boldsymbol{x}\boldsymbol{x}^* : \ \boldsymbol{x} \in \mathbb{C}^d,\ \langle \boldsymbol{x}, \boldsymbol{x}\rangle = 1\Big\}.$$

This is the quantum version of the decomposition of the standard simplex into the convex hull of its extreme points: $\Delta_{d-1} = \mathrm{conv}\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\}$. Classical density vectors are extreme if and only if they are one-sparse, i.e. only one component is different from zero. Quantum density matrices are extreme if and only if they have rank-one. This is the natural matrix generalization of sparsity: a rank-one matrix is one-sparse in its eigenbasis.

In contrast to pure density vectors (classical), pure density matrices (quantum) are not necessarily deterministic. We have encountered this feature in Example 4.3.

5 Applications: Maximum likelihood rule and Holevo-Helström-theorem

In the last two sections we have illustrated the common structure of classical probability theory and quantum mechanics. Extending these parallels, we will now show the optimality of the maximum likelihood rule, and the Holevo-Helström theorem.

Both address the task of distinguishing two probability densities in the single-shot limit.

5.1 Distinguishing classical probability distributions and the maximum likelihood rule

Suppose that we perfectly know descriptions of two probability distributions $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{R}^d$ and choose to play the following game: a referee chooses one of these distributions uniformly at random and hands it to us. We are allowed to perform a single measurement and – based on its outcome – we must guess which probability distribution was handed to us. We win the game if the guess was correct, otherwise we lose.

Let us now try to come up with an optimal guessing strategy. Since we are faced with a binary question, our decision should take the form of a binary measurement:

$$\{\boldsymbol{h}_p, \boldsymbol{h}_q\} : \ \boldsymbol{h}_q = \mathbf{1} - \boldsymbol{h}_p \quad \text{and} \quad \mathbf{1} \succeq \boldsymbol{h}_p \succeq \mathbf{0}.$$

A brief computation yields the following probability of guessing the distribution correctly, based on this binary measurement: A brief computation yields

$$\begin{aligned} p_{\mathrm{succ}} =& \frac{1}{2}\Pr[p|\boldsymbol{p}] + \frac{1}{2}\Pr[q|\boldsymbol{q}] = \frac{1}{2}(\langle \boldsymbol{h}_p, \boldsymbol{p}\rangle + \langle \boldsymbol{h}_q, \boldsymbol{q}\rangle) \\ =& \frac{1}{2}(\langle \boldsymbol{h}_p, \boldsymbol{p}\rangle + \langle \mathbf{1}, \boldsymbol{q}\rangle - \langle \boldsymbol{h}_p, \boldsymbol{q}\rangle) \\ =& \frac{1}{2} + \frac{1}{2}\langle \boldsymbol{h}_{\boldsymbol{p}}, \boldsymbol{p} - \boldsymbol{q}\rangle \end{aligned}$$

We may rewrite the inner-product in the last line as $\sum_{i=1}^{d} [\boldsymbol{h}_p]_i ([\boldsymbol{p}]_i - [\boldsymbol{q}]_I)$. The factor $1/2$ in front of the expression should not be surprising: we can always achieve a success probability of $1/2$ by mere guessing. Optimizing over measurements $\{\boldsymbol{h}_p, \boldsymbol{h}_q\}$ allows us to further improve upon this basic strategy. This optimization problem assumes the form of a linear program:

$$\begin{aligned} \underset{\boldsymbol{h}_p \in \mathbb{R}^d}{\text{maximize}} \quad & \frac{1}{2} + \frac{1}{2}\langle \boldsymbol{p} - \boldsymbol{q}, \boldsymbol{h}_p\rangle \\ \text{subject to} \quad & \mathbf{1} \geq \boldsymbol{h}_p \geq \mathbf{0}. \end{aligned}$$

This linear program is simple enough to solve it analytically. The optimal measurement is

$$\left[\boldsymbol{h}_p^\sharp\right]_i = \begin{cases} 1, & \text{if } p_i > q_i \\ 0, & \text{else.} \end{cases} \qquad \text{for} \quad 1 \leq i \leq d.$$

The associated guessing strategy is called the *maximum likelihood rule*: upon observing measurement outcome i, we guess $\boldsymbol{p}$ if $[\boldsymbol{p}]_i \geq [\boldsymbol{q}]_i$ and otherwise $\boldsymbol{q}$. In words: we choose the distribution that is most likely to provide the outcome that we observed.

The associated optimal success probability is

$$p_{\text{succ}}^\sharp = \frac{1}{2} + \frac{1}{2}\langle \boldsymbol{h}_{\boldsymbol{p}}, \boldsymbol{p} - \boldsymbol{q}\rangle = \frac{1}{2} + \frac{1}{4}\sum_{i=1}^{d} |p_i - q_i| = \frac{1}{2} + \frac{1}{4}\|\boldsymbol{p} - \boldsymbol{q}\|_{\ell_1}$$

and the bias – the amount by which we improve over the naive guessing strategy – is proportional to the *total variational distance* $\frac{1}{2}\|\boldsymbol{p} - \boldsymbol{q}\|_{\ell_1}$ of the distributions.

5.2 Distinguishing quantum states and the Holevo-Helstrom Theorem

Let us now consider the analogous problem in the quantum setting. A referee hands us a black box that contains one of two quantum states: $\boldsymbol{\rho}$ or $\boldsymbol{\sigma}$. Assume that we know the density matrices associated with both states and the referee chooses each of them with equal probability.

Similarly to before, we are allowed to perform a single quantum measurement to guess which state we obtained. Note that this single-shot limit is very appropriate here. A quantum measurement necessarily destroys the quantum state.

Again, we can base our guessing rule on a two-outcome measurement (the question is binary):

$$\boldsymbol{H}_\rho, \boldsymbol{H}_\sigma = \mathbf{I} - \boldsymbol{H}_\rho.$$

If we observe ρ, we guess $\boldsymbol{\rho}$, otherwise we guess $\boldsymbol{\sigma}$. In analogy to the last section, we compute the success probability associated with such a guessing strategy:

$$\begin{aligned} p_{\text{succ}} =& \frac{1}{2}\Pr[\boldsymbol{H}_\rho|\boldsymbol{\rho}] + \frac{1}{2}\Pr[\boldsymbol{H}_\sigma|\boldsymbol{\sigma}] = \frac{1}{2}(\boldsymbol{H}_\rho, \boldsymbol{\rho}) + \frac{1}{2}(\boldsymbol{H}_\sigma, \boldsymbol{\sigma}) \\ =& \frac{1}{2}((\boldsymbol{H}_\rho, \boldsymbol{\rho}) + (\mathbf{I}, \boldsymbol{\rho}) - (\boldsymbol{H}_\rho, \boldsymbol{\sigma})) \\ =& \frac{1}{2} + \frac{1}{2}(\boldsymbol{H}_{\boldsymbol{\rho}}, \rho - \sigma) \end{aligned}$$

Next, we optimize this expression over all possible choices of measurements:

$$\begin{aligned} \underset{\boldsymbol{H}_\rho \in \mathbb{H}^d}{\text{maximize}} \quad & \frac{1}{2} + \frac{1}{2}(\boldsymbol{H}_\rho, \boldsymbol{\rho} - \boldsymbol{\sigma}) \\ \text{subject to} \quad & \mathbf{I} \succeq \boldsymbol{H}_\rho \succeq \mathbf{0}. \end{aligned}$$

This is a semidefinite program that is simple enough to solve analytically. Apply an eigenvalue decomposition to $\boldsymbol{X} = \boldsymbol{\rho} - \boldsymbol{\sigma} = \sum_{i=1}^d \xi_i \boldsymbol{x}_i \boldsymbol{x}_i^*$. Set $\boldsymbol{P}_+ = \sum_{i=1}^d \mathbb{I}\{\xi_i > 0\}\boldsymbol{x}_i\boldsymbol{x}_i^*$ and $\boldsymbol{P}_- = \sum_{i=1}^d \mathbb{I}\{\xi < 0\}\boldsymbol{x}_i\boldsymbol{x}_i^*$. These are orthogonal projectors onto the positive- and

negative ranges of $\boldsymbol{X} = \boldsymbol{\rho} - \boldsymbol{\sigma}$. They are the natural generalizations of the maximum likelihood rule to the quantum setting. In particular, the choice $\boldsymbol{H}_p^\sharp = \boldsymbol{P}_+$ is optimal and results in the following optimal success probability:

$$p_{\text{succ}}^\sharp = \frac{1}{2} + \frac{1}{4}\|\boldsymbol{\rho} - \boldsymbol{\sigma}\|_*$$

Here, $\|\cdot\|_*$ denotes the nuclear (or trace) norm. It is the natural quantum generalization of the total variational distance.

Theorem 5.1 (Holevo-Helstrom)**.** *The optimal success probability for distinguishing two quantum states $\boldsymbol{\rho}, \boldsymbol{\sigma} \in \mathbb{H}^d$ with a single measurement is*

$$p_{\text{succ}}^\sharp = \frac{1}{2} + \frac{1}{4}\|\boldsymbol{\rho} - \boldsymbol{\sigma}\|_1.$$

The optimal measurement is the projector onto the positive range of $\boldsymbol{\rho} - \boldsymbol{\sigma}$ and depends on the states in question.

This observation dates back to Holevo[1] (1973) and Helstrom (1976) and plays a prominent role in modern quantum information theory. For instance, when estimating density matrices from experimental observations, error bars are typically reported in the nuclear norm.

[1] Alexander Holevo received the *Claude E. Shannon Award* in 2016 for his outstanding contributions to quantum information theory.

Lecture 02: Tensor products

Scribe: Chung-Yi Lin

ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp
April 03, 2017

1 Agenda

1. Natural axioms for vector multiplication
2. Bipartite tensor product spaces $H^{\otimes 2}$
3. Operators on $H^{\otimes 2}$
4. Multi-partite tensor product spaces $H^{\otimes k}$
5. Operators on $H^{\otimes k}$

2 Axiomatic approach to vector products

2.1 Natural axioms for vector multiplication

Tensor products are motivated by the following basic and natural question: *What does it mean to multiply vectors?*

In order to answer this question, we turn to scalar multiplication for guidance. Let $\mathbb{F}$ be a field, e.g. $\mathbb{R}$ or $\mathbb{C}$. Then, the scalar product is additive, homogeneous, has a zero-element and is faithful, as well as symmetric. We refer to Definition 2.1 for a precise definition of these properties.

Now, let H be a d-dimensional inner product space over $\mathbb{F} = \mathbb{R}$, or $\mathbb{F} = \mathbb{C}$, equipped with an inner product[1]

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^{n} \bar{x}_i y_i \quad \text{for} \quad \boldsymbol{x}, \boldsymbol{y} \in H.$$

Based on our intuition about scalar multiplication, we postulate the following "natural" set of properties.

Definition 2.1 (Axioms for vector multiplication)**.** A well-defined product $\boldsymbol{x} \otimes \boldsymbol{y}$ of vectors $\boldsymbol{x}, \boldsymbol{y} \in H$ should obey the following properties:

1. *Additivity:* for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in H$:

$$(\boldsymbol{x} + \boldsymbol{y}) \otimes \boldsymbol{z} = \boldsymbol{x} \otimes \boldsymbol{z} + \boldsymbol{y} \otimes \boldsymbol{z} \quad \text{and} \quad \boldsymbol{x} \otimes (\boldsymbol{y} + \boldsymbol{z}) = \boldsymbol{x} \otimes \boldsymbol{y} + \boldsymbol{x} \otimes \boldsymbol{z}.$$

2. *Homogeneity:* for all $\boldsymbol{x}, \boldsymbol{y} \in H$ and $\alpha \in \mathbb{F}$:

$$\alpha(\boldsymbol{x} \otimes \boldsymbol{y}) = (\alpha \boldsymbol{x}) \otimes \boldsymbol{y} = \boldsymbol{x} \otimes (\alpha \boldsymbol{y}).$$

[1] In contrast to widespread mathematical convention, we define the inner product to be linear in the second argument. This convention will considerably simplify analysis throughout the course of these lectures.

3. *Zero property of multiplication:* Let $\mathbf{0} \in H$ denote the zero element. Then for all $\boldsymbol{x}, \boldsymbol{y} \in H$,
$$\boldsymbol{x} \otimes \mathbf{0} = \mathbf{0} \otimes \boldsymbol{y} = \mathbf{0}.$$
4. *Faithfulness:* If $\boldsymbol{x} \otimes \boldsymbol{y} = \mathbf{0}$, then either $\boldsymbol{x} = \mathbf{0}$, or $\boldsymbol{y} = \mathbf{0}$.

These axioms naturally generalize to multiplication of more than two vectors.

Remark 2.2. Note that we have excluded *symmetry* from this list. A symmetric vector product also obeys $\boldsymbol{x} \otimes \boldsymbol{y} = \boldsymbol{y} \otimes \boldsymbol{x}$. We will discuss such a symmetric vector product in Lecture 4.

After cornering these natural properties, let us analyze familiar notions of vector products.

1. the *dot product:* $H \times H \to \mathbb{F}$, where $(\boldsymbol{x}, \boldsymbol{y}) \mapsto \langle \boldsymbol{x}, \boldsymbol{y} \rangle$. The dot product obeys properties 1. 2. and 3. but is not faithful: two non-zero orthogonal vectors have a vanishing dot-product.
2. the *Schur/Hadamard product:* $H \times H \to H$, where $(\boldsymbol{x}, \boldsymbol{y}) \mapsto [x_i y_i]_{i=1}^d$. This vector product obeys properties 1. 2. and 3. but is not faithful. The Schur product of two non-zero vectors with disjoint supports vanishes.
3. the *Outer product:* $H \times H \to \mathcal{L}(H)$, where $(\boldsymbol{x}, \boldsymbol{y}) \mapsto \boldsymbol{x}\boldsymbol{y}^T$. This product fulfills all the axioms from Definition 2.1. However, it is not obvious how to generalize the outer product to more than two vectors.

2.2 Axiomatic approach to bipartite tensor products

Let "$\otimes$" denote a product operation that obeys the first three properties of Definition 2.1. For $\boldsymbol{x}, \boldsymbol{y} \in H$, we define the *elementary tensor product* $\boldsymbol{x} \otimes \boldsymbol{y}$. For now, we regard this as a formal product of $\boldsymbol{x}$ and $\boldsymbol{y}$. The *tensor product space* $H^{\otimes 2}$ contains all formal[2] linear combinations

$$\boldsymbol{T} = \sum_{i=1}^{r} \alpha_i \boldsymbol{x}_i \otimes \boldsymbol{y}_i \quad r \in \mathbb{N},\ \boldsymbol{x}_i, \boldsymbol{y}_i \in H,\ \alpha_i \in \mathbb{F}. \tag{1}$$

We emphasize that these formal representations of $\boldsymbol{T}$ are not unique: the zero property implies that $\boldsymbol{x} \otimes \mathbf{0}$ and $\mathbf{0} \otimes \boldsymbol{y}$ both yield zero. Adding terms of this form in the summation (1) thus does not change $\boldsymbol{T}$. Moreover, additivity and homogeneity ensure linearity:

$$\begin{aligned}(\alpha\boldsymbol{x} + \beta\boldsymbol{y}) \otimes \boldsymbol{z} &= \alpha\boldsymbol{x} \otimes \boldsymbol{z} + \beta\boldsymbol{y} \otimes \boldsymbol{z},\\ \boldsymbol{x} \otimes (\alpha\boldsymbol{y} + \beta\boldsymbol{z}) &= \alpha\boldsymbol{x} \otimes \boldsymbol{y} + \beta\boldsymbol{y} \otimes \boldsymbol{z}.\end{aligned}$$

This in turn implies that vectors $\boldsymbol{x}_i, \boldsymbol{y}_i \in H$ can be further decomposed into different vectors (e.g. via a basis expansion) and inserting these decompositions into (1) seemingly leads to a different $\boldsymbol{T} \in H^{\otimes 2}$.

We define the set of tensor products as the space of all tensors $\boldsymbol{T}$ modulo these identity transformations:

[2] *Formal* means, that for now we treat these expressions as a collection of symbols, give them a name and perform linear combinations.

Definition 2.3 (Tensor product space)**.** Let H be a finite dimensional vector space over $\mathbb{F}$ and let $\boldsymbol{x} \otimes \boldsymbol{y}$ be a vector product that satisfies Definition 2.1. Then

$$H^{\otimes 2} = \left\{ \sum_{i=1}^{r} \alpha_i \boldsymbol{x}_i \otimes \boldsymbol{y}_i \in H^{\otimes 2} \quad \forall r \in \mathbb{N},\ \forall \alpha_i \in \mathbb{F},\ \forall \boldsymbol{x}_i, \boldsymbol{y}_i \in H \right\} / \text{identity}.$$

We emphasize that the definitions and concepts presented in this sub-section naturally generalize to products of more than two vectors.

3 Bipartite tensor product space $H^{\otimes 2}$

3.1 Bipartite tensor products and bilinear forms

In this section, we define a natural notion of a *tensor product*. It obeys all axioms from Definition 2.1 and is *minimal* in the sense that every true statement about tensor products can be reduced to these defining properties. No additional structure is present.

Definition 3.1 (Bilinear forms)**.** A *bilinear form* is a function of the form $B:\ H \times H \to \mathbb{F}$ with the following properties:

1. For any $\boldsymbol{x} \in H$, $B(\boldsymbol{x}, \cdot)$ is a linear functional on H;
2. For any $\boldsymbol{y} \in H$, $B(\cdot, \boldsymbol{y})$ is a linear functional on H.

Let $\mathrm{Bil}(H, H)$ denote the (linear) space of all bilinear forms.

A concrete model for bilinear forms can be obtained in the following way: Let $\boldsymbol{A} = [a_{i,j}]_{i,j=1}^{d} \in \mathbb{F}^{d \times d}$ be a $d \times d$ matrix. Then, we can associate $\boldsymbol{A}$ with the following bilinear form:

$$B_{\boldsymbol{A}}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i,j=1}^{d} x_i a_{i,j} v_j.$$

Note that this identification is bijective. Conversely, fix a bilinear form $B(\cdot, \cdot)$ and an orthonormal basis $\{\boldsymbol{e}_i\}_{i=1}^{d}$ of H. Then, linearity implies

$$B(\boldsymbol{x}, \boldsymbol{y}) = B\left(\sum_{i=1}^{d} x_i \boldsymbol{e}_i, \sum_{j=1}^{d} y_j \boldsymbol{e}_j \right) = \sum_{i,j=1}^{d} x_i B(\boldsymbol{e}_i, \boldsymbol{e}_j) y_j$$

for any $\boldsymbol{x}, \boldsymbol{y} \in H$ with basis expansion $\boldsymbol{x} = \sum_{i=1}^{n} x_i \boldsymbol{e}_i, \boldsymbol{y} = \sum_{j=1}^{n} y_j \boldsymbol{e}_j$. The $d^2 = (\dim(H))^2$ numbers $B(\boldsymbol{e}_i, \boldsymbol{e}_j)$ are independent degrees of freedom. We can identify these degrees of freedom with entries of a matrix $\boldsymbol{A} = [a_{i,j}]_{i,j=1}^{d}$ that tabulates the action of the bilinear form on different basis vectors: $a_{i,j} = B(\boldsymbol{e}_i, \boldsymbol{e}_j)$.

Definition 3.2 (Tensor product space)**.** Let H be a (finite dimensional) inner product space. The *tensor product space* $H^{\otimes 2}$ is the dual space of $\mathrm{Bil}(H, H)$. In particular, we identify elementary tensor products with the following functional:

$$\boldsymbol{x} \otimes \boldsymbol{y}:\ B \mapsto B(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{F}.$$

Fact 3.3. *This definition of a tensor product obeys all properties listed in Definition 2.1.*

The fact that vector spaces and their duals are both linear and have equal dimension allows us to infer the dimension of $H^{\otimes 2}$ via the correspondence between bilinear forms and matrices:

$$\dim\Big(H^{\otimes 2}\Big) = \dim(\mathrm{Bil}(H,H)^*) = \dim(\mathrm{Bil}(H,H)) = \dim\Big(\mathbb{F}^{\dim(H)\times\dim(H)}\Big) = \dim(H)^2.$$

Moreover, linear extension allows us to define an inner product on $H \otimes H$ that is induced by the inner product $\langle\cdot,\cdot\rangle$ on H. For elementary tensors $\boldsymbol{x}_1 \otimes \boldsymbol{y}_1$ and $\boldsymbol{x}_2 \otimes \boldsymbol{y}_2$, we define

$$\langle \boldsymbol{x}_1 \otimes \boldsymbol{y}_1, \boldsymbol{x}_2 \otimes \boldsymbol{y}_2\rangle = \langle \boldsymbol{x}_1, \boldsymbol{x}_2\rangle\langle \boldsymbol{y}_1, \boldsymbol{y}_2\rangle \quad \forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in H,\ \forall \boldsymbol{y}_1, \boldsymbol{y}_2 \in H. \tag{2}$$

We extend this definition linearly to the space of all linear combinations of elementary tensors (1), i.e. $H^{\otimes 2}$.

Fact 3.4 (Dimension of tensor products)**.** *The tensor product space $H^{\otimes 2}$ equipped with induced the inner product (2) forms an inner product space of dimension* $\dim(H^{\otimes 2}) = \dim(H)^2$.

3.2 Concrete realization of $H^{\otimes 2}$ as the space of all outer products

So far, we have presented a construction of $H^{\otimes 2}$ as the dual space of bilinear forms $\mathrm{Bil}(H,H)$. This is rather abstract, but we can represent $H^{\otimes 2}$ as an *outer product space*. Set $d = \dim(H)$ and define the elementary tensors to be outer products of vectors:

$$\boldsymbol{x} \otimes \boldsymbol{y} := \boldsymbol{x}\boldsymbol{y}^T \in \mathcal{L}(H) \simeq \mathbb{F}^{d\times d}.$$

The linear hull of these outer products corresponds to $\mathbb{F}^{d\times d}$, or equivalently, $\mathcal{L}(H)$. Note that $\dim(\mathcal{L}(H)) = d^2$, in accordance with Fact 3.4. On outer products $\boldsymbol{x}\boldsymbol{y}^T$, the induced inner product $(\cdot,\cdot)$ is

$$\Big(\boldsymbol{x}_1\boldsymbol{y}_1^T, \boldsymbol{x}_2\boldsymbol{y}_2^T\Big) := \langle \boldsymbol{x}_1, \boldsymbol{x}_2\rangle\langle \boldsymbol{y}_1, \boldsymbol{y}_2\rangle = \mathrm{tr}\Big(\Big(\boldsymbol{x}_1\boldsymbol{y}_1^T\Big)^* \boldsymbol{x}_2\boldsymbol{y}_2^T\Big),$$

where "tr" denotes the trace and $\boldsymbol{A}^*$ is the adjoint of $\boldsymbol{A} \in \mathcal{L}(H)$. By linear extension, this inner product becomes the *Frobenius* (or *Hilbert-Schmidt*) *inner product* $(\boldsymbol{A},\boldsymbol{B}) = \mathrm{tr}(\boldsymbol{A}^*\boldsymbol{B})$ on $\mathcal{L}(H)$.

4 Operators on $H^{\otimes 2}$

4.1 Definition and useful properties

For $\boldsymbol{A},\boldsymbol{B} \in \mathcal{L}(H)$, we define $\boldsymbol{A}\otimes\boldsymbol{B} \in \mathcal{L}(H^{\otimes 2})$ via the following action on elementary tensors:

$$(\boldsymbol{A}\otimes\boldsymbol{B})(\boldsymbol{x}\otimes\boldsymbol{y}) = (\boldsymbol{A}\boldsymbol{x})\otimes(\boldsymbol{B}\boldsymbol{y}).$$

This action can be extended linearly to all elements of $H^{\otimes 2}$. Linear extensions of the form $\sum_{i=1}^r \alpha_i \boldsymbol{A}_i \otimes \boldsymbol{B}_i$ form the full set of linear operators on $H^{\otimes 2}$. The key property of this construction is:

$$(\boldsymbol{A}\otimes\boldsymbol{B})(\boldsymbol{C}\otimes\boldsymbol{D}) = (\boldsymbol{A}\boldsymbol{C})\otimes(\boldsymbol{B}\boldsymbol{D}). \tag{3}$$

This property has powerful consequences.

Fact 4.1.

1. *Let* $\mathbf{I} \in \mathcal{L}(H)$ *be the identity. Then,* $\mathbf{I} \otimes \mathbf{I}$ *is the identity in* $\mathcal{L}(H^{\otimes 2})$.
2. *Let* $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{L}(H)$ *be invertible. Then,* $(\boldsymbol{A} \otimes \boldsymbol{B})^{-1} = (\boldsymbol{A}^{-1}) \otimes (\boldsymbol{B}^{-1})$.
3. $(\boldsymbol{A} \otimes \boldsymbol{B})^{-1} = (\boldsymbol{A}^{-1}) \otimes (\boldsymbol{B}^{-1})$ *if and only if* $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{L}(H)$ *are invertible,*
4. *Let* $\boldsymbol{A}^*$ *denote the adjoint of* $\boldsymbol{A} \in \mathcal{L}(H)$. *Then,* $(\boldsymbol{A} \otimes \boldsymbol{B})^* = (\boldsymbol{A}^*) \otimes (\boldsymbol{B}^*)$.

All these properties readily follow from the definition and the composition rule (3). For instance, the first claim is a consequence of

$$(\mathbf{I} \otimes \mathbf{I})(\boldsymbol{A} \otimes \boldsymbol{B}) = (\mathbf{I}\boldsymbol{A}) \otimes (\mathbf{I}\boldsymbol{B}) = \boldsymbol{A} \otimes \boldsymbol{B} = (\boldsymbol{A}\mathbf{I}) \otimes (\boldsymbol{B}\mathbf{I}) = (\boldsymbol{A} \otimes \boldsymbol{B})(\mathbf{I} \otimes \mathbf{I}).$$

for all $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{L}(H)$. These facts together with the composition rule (3) imply the following persistence property.

Fact 4.2 (Persistence)**.** *Fix* $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{L}(H)$.

1. *If* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *are positive semidefinite, then so is* $\boldsymbol{A} \otimes \boldsymbol{B}$.
2. *If* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *are self-adjoint, then so is* $\boldsymbol{A} \otimes \boldsymbol{B}$.
3. *If* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *are normal, then so is* $\boldsymbol{A} \otimes \boldsymbol{B}$.
4. *If* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *are unitary, then so is* $\boldsymbol{A} \otimes \boldsymbol{B}$.

Remark 4.3. Converse persistence relations are usually false. The concept of quantum entanglement is closely related to a converse relation of property 1. failing to hold. We refer to Lecture 3 for details.

The *Kronecker product* is a concrete model for tensor products of operators.

4.2 Spectral Theory

To ease notational burden, we will restrict attention to tensor product operators of the form $\boldsymbol{A} \otimes \boldsymbol{A}$. A generalization to asymmetric tensor products $\boldsymbol{A} \otimes \boldsymbol{B}$ is straightforward.

4.2.1 Spectral resolutions

Recall that an operator $\boldsymbol{A} \in \mathcal{L}(H)$ is normal if $\boldsymbol{A}^*\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^* = \mathbf{I}$. The spectral theorem implies that every normal matrix has a spectral resolution:

$$\boldsymbol{A} = \sum_i \lambda_i \boldsymbol{P}_i. \tag{4}$$

Here, λ_i are (potentially complex-valued) eigenvalues and the $\boldsymbol{P}_i$'s are (mutually) orthogonal projectors that form a particular resolution of the identity: $\sum_i \boldsymbol{P}_i = \mathbf{I}$ and $\boldsymbol{P}_i\boldsymbol{P}_j = \delta_{i,j}\boldsymbol{P}_i$.

Fact 4.4 (Spectral resolutions of tensor product operators)**.** *Let* $\boldsymbol{A} \in \mathcal{L}(H)$ *be a normal matrix with spectral resolution* (4). *Then,*

$$\boldsymbol{A} \otimes \boldsymbol{A} = \sum_{i,j} \lambda_i \lambda_j \boldsymbol{P}_i \otimes \boldsymbol{P}_j$$

is a spectral resolution of the tensor product.

4.2.2 Singular value decompositions of tensor product operators

Fact 4.5 (Singular value decompostions of tensor product operators)**.** *Let $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$ be a SVD of $\boldsymbol{A} \in \mathcal{L}(H)$. Then,*

$$\boldsymbol{A} \otimes \boldsymbol{A} = (\boldsymbol{U} \otimes \boldsymbol{U})(\boldsymbol{\Sigma} \otimes \boldsymbol{\Sigma})(\boldsymbol{V} \otimes \boldsymbol{V})^*$$

is a SVD of $\boldsymbol{A} \otimes \boldsymbol{A}$. In particular, the singular values are $\sigma_i \sigma_j$ with $1 \leq i, j \leq d$.

4.2.3 Eigenvalue decompositions of tensor product operators

Recall that every operator admits a *Schur decomposition*

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{T}\boldsymbol{Q}^*.$$

Here, $\boldsymbol{Q}$ is unitary, and $\boldsymbol{T}$ is upper triangular. The diagonal of $\boldsymbol{T}$ contains all eigenvalues of $\boldsymbol{A}$. Persistence implies that these properties readily extend to tensor products ($\boldsymbol{T} \otimes \boldsymbol{T}$ is again upper-triangular with respect to the designated basis used).

Fact 4.6 (Eigenvalues of tensor product operators)**.** *Suppose that $\boldsymbol{A} \in \mathcal{L}(H)$ has eigenvalues $\lambda_1, \ldots, \lambda_d \in \mathbb{C}$. Then, $\boldsymbol{A} \otimes \boldsymbol{A}$ has eigenvalues $\lambda_i \lambda_j$ with $1 \leq i, j \leq d$.*

5 Multi-partite tensor product spaces $H^{\otimes k}$

5.1 Axiomatic approach to tensor spaces of order $k \geq 3$

Formally, we introduce *elementary k-fold tensor*

$$\boldsymbol{x}_1 \otimes \cdots \otimes \boldsymbol{x}_k, \quad \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \in H$$

and define the space of all k-fold tensors by linear extension:

$$H^{\otimes k} = \left\{ \sum_{i=1}^{r} \alpha_i \boldsymbol{x}_{i_1} \otimes \cdots \otimes \boldsymbol{x}_{i_k} : \; r \in \mathbb{N}, \; \alpha_i \in \mathbb{F}, \; \boldsymbol{x}_{i_j} \in H \right\} / \text{identity}$$

Note that this product formalism obeys all desirable axioms for vector multiplication. In particular, the role of zero and faithfulness hold: $\boldsymbol{x}_1 \otimes \cdots \otimes \boldsymbol{x}_k = \boldsymbol{0}$ if and only if $\boldsymbol{x}_i = \boldsymbol{0}$ for at least one coordinate $1 \leq i \leq k$.

5.2 Multi-partite tensor product spaces and multi-linear forms

More concretely, $H^{\otimes k}$ can be identified with the dual space of the space of all multi-linear forms.

Definition 5.1 (Multi-linear forms)**.** A multi-linear form (of order k) is a function

$$M : \; H^{\times k} = \underbrace{H \times \cdots H}_{k \text{ times}} \to \mathbb{F}$$

that is linear in each argument. More precisely:

1. For any $\boldsymbol{x}_2, \ldots, \boldsymbol{x}_k \in H$, $M(\cdot, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k)$ is a linear functional on H;

⋮

k. For any $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1} \in H$, $M(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1}, \cdot)$ is a linear functional on H.

Let $\mathrm{Multi}\Big(H^{\times k}\Big)$ denote the (linear) space of all multi-linear forms.

In complete analogy to the bipartite case, we can identify $H^{\otimes k}$ with the dual space of all multi-linear forms (of order k).

Definition 5.2 (Tensor product space)**.** Let H be a (finite dimensional) inner product space. The *tensor product space* $H^{\otimes k}$ is the dual space of $\mathrm{Multi}\Big(H^{\times k}\Big)$. In particular, we identify elementary tensors with the following functional:

$$\boldsymbol{x}_1 \otimes \cdots \otimes \boldsymbol{x}_k : \; M \mapsto M(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) \in \mathbb{F}.$$

The inner product $\langle \cdot, \cdot \rangle$ on H induces an inner product on $H^{\otimes k}$. Define

$$\langle \boldsymbol{x}_1 \otimes \cdots \otimes \boldsymbol{x}_k, \boldsymbol{y}_1 \otimes \cdots \otimes \boldsymbol{y}_k \rangle = \prod_{j=1}^{k} \langle \boldsymbol{x}_j, \boldsymbol{y}_j \rangle \tag{5}$$

and extend this definition linearly.

Fact 5.3. *Let $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$ be an orthonormal basis of H (with respect to $\langle \cdot, \cdot \rangle$). Then,*

$$\{\boldsymbol{e}_{i_1} \otimes \cdots \otimes \boldsymbol{e}_{i_d} : \; 1 \leq i_1, \ldots, i_d \leq d\}$$

is an orthonormal basis of $H^{\otimes k}$ (with respect to the extended inner product (5)).

Corollary 5.4. *Set $d = \dim(H)$. Then, $\dim\Big(H^{\otimes k}\Big) = d^k$.*

The dimension of tensor product spaces grows exponentially with the order. This is a veritable *curse of dimensionality*.

6 Operators on $H^{\otimes k}$

For $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_k \in \mathcal{L}(H)$ we define tensor product operators $\boldsymbol{A}_1 \otimes \cdots \otimes \boldsymbol{A}_k \in \mathcal{L}\Big(H^{\otimes k}\Big)$ via their action on elementary tensors

$$(\boldsymbol{A}_1 \otimes \cdots \otimes \boldsymbol{A}_k)(\boldsymbol{x}_1 \otimes \cdots \otimes \boldsymbol{x}_k) = (\boldsymbol{A}\boldsymbol{x}_1) \otimes \cdots \otimes (\boldsymbol{A}\boldsymbol{x}_k)$$

and extend this definition by linearity. The results form Section 4 generalize naturally to this k-fold setting. This, in particular, includes the composition rule. For $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_k, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_k \in \mathcal{L}(H)$,

$$(\boldsymbol{A}_1 \otimes \cdots \otimes \boldsymbol{A}_k)(\boldsymbol{B}_1 \otimes \cdots \otimes \boldsymbol{B}_k) = (\boldsymbol{A}_1\boldsymbol{B}_1) \otimes \cdots \otimes (\boldsymbol{A}_k\boldsymbol{B}_k).$$

This composition rule implies persistence. For instance, let $\mathbf{I} \in \mathcal{L}(H)$ be the identity. Then, $\mathbf{I} \otimes \cdots \otimes \mathbf{I}$ is the identity on $\mathcal{L}(H^{\otimes k})$.

Fact 6.1 (Persistence)**.** *Fix* $\boldsymbol{A}_1, \dots, \boldsymbol{A}_k \in \mathcal{L}(H)$.

1. *If* $\boldsymbol{A}_1, \dots, \boldsymbol{A}_k$ *are positive semidefinite, then so is* $\boldsymbol{A}_1 \otimes \cdots \otimes \boldsymbol{A}_k$.
2. *If* $\boldsymbol{A}_1, \dots, \boldsymbol{A}_k$ *are self-adjoint, then so is* $\boldsymbol{A}_1 \otimes \cdots \otimes \boldsymbol{A}_k$.
3. *If* $\boldsymbol{A}_1, \dots, \boldsymbol{A}_k$ *are normal, then so is* $\boldsymbol{A}_1 \otimes \cdots \otimes \boldsymbol{A}_k$.
4. *If* $\boldsymbol{A}_1, \dots, \boldsymbol{A}_k$ *are unitary, then so is* $\boldsymbol{A}_1 \otimes \cdots \otimes \boldsymbol{A}_k$.

The insights about spectral resolutions and decompositions also generalize in a straightforward way.

Fact 6.2. *Fix* $\boldsymbol{A} \in \mathcal{L}(H)$ *and write* $\boldsymbol{A}^{\otimes k} = \boldsymbol{A} \otimes \cdots \otimes \boldsymbol{A}$.

1. *Suppose that* $\boldsymbol{A}$ *is normal with spectral resolution* $\boldsymbol{A} = \sum_i \lambda_i \boldsymbol{P}_i$. *Then,*

$$\boldsymbol{A}^{\otimes k} = \sum_{i_1, \dots, i_k} \lambda_{i_1} \cdots \lambda_{i_k} \boldsymbol{P}_{i_1} \otimes \cdots \otimes \boldsymbol{P}_{i_k}$$

   is again a spectral resolution.

2. *Let* $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$ *be a singular value decomposition with singular values* $\sigma_1, \dots \sigma_d$. *Then,* $(\boldsymbol{U} \otimes \cdots \otimes \boldsymbol{U})(\boldsymbol{\Sigma} \otimes \cdots \otimes \boldsymbol{\Sigma})(\boldsymbol{V} \otimes \cdots \otimes \boldsymbol{V})^*$ *is a singular value decomposition of* $\boldsymbol{A}^{\otimes k}$. *In particular, the singular values are* $\sigma_{i_1} \cdots \sigma_{i_k}$ *for* $1 \leq i_1, \dots, i_k \leq d$.
3. *Suppose that* $\boldsymbol{A}$ *has eigenvalues* $\lambda_1, \dots, \lambda_d \in \mathbb{C}$. *Then, the eigenvalues of* $\boldsymbol{A}^{\otimes k}$ *are* $\lambda_{i_1} \cdots \lambda_{i_k}$ *for* $1 \leq i_1, \dots, i_k \leq d$.

Lecture 03: Wiring calculus and entanglement

Scribe: Erika Ye

ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp
April 8, 2019

1 Agenda

1. Wiring calculus
2. Joint classical probability distributions
3. Joint quantum distributions
4. Entanglement and the Positive-Partial-Transpose (PPT) test

2 Wiring Calculus

Wiring calculus is a graphical formalism that is designed to deal with index contractions among tensors. It has been used in various fields, such as physics (Feynman and Penrose), representation theory (Cvitanovic), knot theory (Bar-Natan and Kontsevich), quantum groups (Reshetikhin), and category theory (Deligne and Vogel). More recently, it has become popular in the field of tensor networks. (We refer to the excellent lecture notes by Bridgeman and Chubb for further information and reading.) Here, we will focus on wiring calculus developed for tensor representation and manipulation.

2.1 Wiring diagrams for vectors and adjoints

Let H be a d-dimensional vector space with designated inner product $\langle \cdot, \cdot \rangle$. Let $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$ denote a designated orthonormal basis of H. The basic building blocks of wiring calculus are boxes for standard basis vectors and their (basis-dependent) transposes:

$$\boldsymbol{e_i} = \boxed{\boldsymbol{e}_i}\!\!- \quad \text{and} \quad \boldsymbol{e_i}^T = -\!\!\boxed{\boldsymbol{e}_i} \quad \text{for} \quad 1 \leq i \leq d.$$

We extend both definitions in a linear and anti-linear fashion to all of H:

$$-\!\!\boxed{\boldsymbol{x}} = \sum_{i=1}^{d} x_i \left(-\!\!\boxed{\boldsymbol{e}_i} \right) \quad \text{and} \quad \boxed{\boldsymbol{x}}\!\!- = \sum_{i=1}^{d} \bar{x}_i \left(\boxed{\boldsymbol{e}_i}\!\!- \right) \tag{1}$$

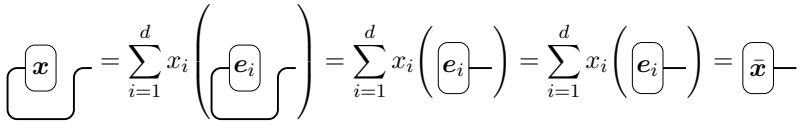
This convention is crucial. Boxes with an emanating line towards the left are standard vectors, while boxes with an emanating line towards the right are adjoint vectors. This convention is designed to appropriately capture contractions, like the inner product:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i,j=1}^{d} \bar{x}_i y_j \langle \boldsymbol{e}_i, \boldsymbol{e}_j \rangle = \sum_{i=1}^{d} \sum_{j=1}^{d} \bar{x}_i y_j \, \boxed{\boldsymbol{e_i}}\!\!-\!\!\boxed{\boldsymbol{e_j}} = \boxed{\boldsymbol{x}}\!\!-\!\!\boxed{\boldsymbol{y}} \tag{2}$$

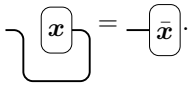
Transposition corresponds to bending an emanating line into the opposite direction. For standard basis vectors, we define

$$\text{⟲}\boxed{\boldsymbol{e}_i}\text{⟳} = \left(-\!\!\boxed{\boldsymbol{e}_i} \right)^T = \boxed{\boldsymbol{e}_i}\!\!- \quad \text{and} \quad \text{⟳}\boxed{\boldsymbol{e}_i}\text{⟲} = \left(\boxed{\boldsymbol{e}_i}\!\!- \right)^T = -\!\!\boxed{\boldsymbol{e}_i}.$$

and extend this action linearly to H:



Similarly:



It is easy to see that doing the transpose twice returns the original vector.

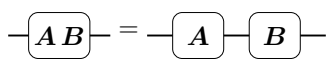
Remark 2.1. Transposition is basis-dependent and turns column vectors (boxes with lines emanating to the left) into row vectors (boxes with lines emanating to the left). Importantly, it does not conjugate the vector entries.

The motivation behind this graphical formalism is as follows: Vectors can be thought of as 1-dimensional arrays $\boldsymbol{x} = [x_i]_{i=1}^d$. They correspond to d numbers labeled by an index $1 \leq i \leq d$. The outgoing lines in Equation (1) represent a free index. The direction of the line tells us whether we should think of the array as a column, or row-vector. The inner product (2) is an index contraction. It perfectly aligns the indices associated with a row-vector and a column vector: $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^d \bar{x}_i y_i$. A closed line represents such a contraction pictorially. This graphically mimics the Einstein summation convention: $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \bar{x}_i y^i$, where the location of the index tells us whether the object is a contra- or co-variant vector and it is implicitly assumed that one sums over indices that appear twice.

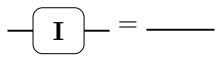
2.2 Wiring diagrams for operators

The wiring diagram formalism readily and consistently extends to operators (matrices). An operator $\boldsymbol{A} \in \mathcal{L}(H)$ "eats" a vector $\boldsymbol{x}$ and spits out another vector in H. In wiring calculus, we write

where $\boldsymbol{A}^* \in \mathcal{L}(H)$ is the adjoint of $\boldsymbol{A}$. Operators are represented by boxes with two emanating indices. This is consistent with the array interpretation. Operators may be characterized by matrices $\boldsymbol{A} = [a_{ij}]_{i,j=1}^d$ which are 2-dimensional arrays. The two indices correspond to two lines that emanate in different directions. Matrix multiplication combines two operators and returns a third one: $\boldsymbol{AB} \in \mathcal{L}(H)$:

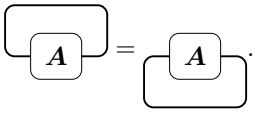


A particularly important operator/matrix is the identity $\boldsymbol{I} \in \mathcal{L}(H)$. It is characterized by the unique property of "doing nothing": $\mathbf{I}\boldsymbol{x} = \boldsymbol{x}$ for all $\boldsymbol{x} \in H$. We pictorially underline this by writing

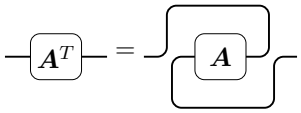


There is an alternative explanation for this notation. We can expand $\mathbf{I} = \sum_{i=1}^{d} \boldsymbol{e}_i \boldsymbol{e}_i^T$. This action perfectly aligns both emanating indices. This resembles the contraction that features in the inner product (2).

The *trace* is the natural index contraction for matrices. It perfectly aligns left- and right-indices: $\mathrm{tr}(\boldsymbol{A}) = \sum_{i=1}^{d} A_{ii}$:

$$\mathrm{tr}(\boldsymbol{A}) =$$  $$=$$.

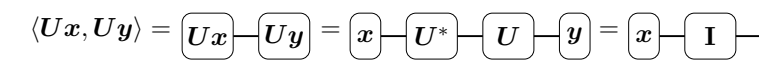
Finally, the (basis dependent) transpose operation swaps the indices associated with a matrix: $\left[\boldsymbol{A}^T\right]_{ij} = \boldsymbol{A}_{ji}$. Pictorially:



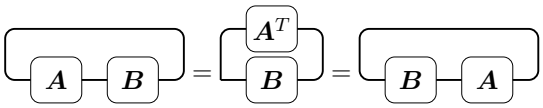
and it is easy to verify that transposing twice returns the original operator diagram.

2.2.1 Graphical proofs for important results in linear algebra

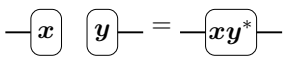
1. *Inner products are basis independent*: Fix a unitary matrix $\boldsymbol{U} \in H$ (basis change). Then, for any $\boldsymbol{x}, \boldsymbol{y} \in H$

   $\langle \boldsymbol{U}\boldsymbol{x}, \boldsymbol{U}\boldsymbol{y} \rangle =$  $= \langle \boldsymbol{x}, \boldsymbol{y} \rangle$

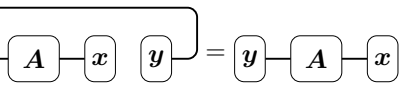
2. *The trace is cyclic:*

3. *Outer products are matrices:*

   In particular,

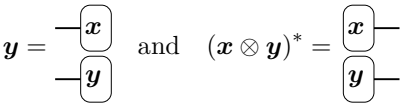
   $\mathrm{tr}\left(\boldsymbol{A}\boldsymbol{x}\boldsymbol{y}^T\right) =$  $=$

2.3 Wiring diagrams for tensors

The wiring formalism readily extends to tensor products. Note that so far, all index lines have been arranged horizontally. Wiring diagrams for operations in H may be thought

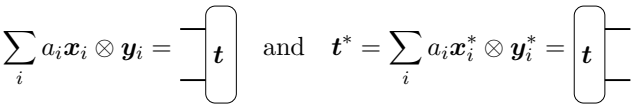
of as wires that connect operations in a serial fashion. Tensor product operations are arranged in a parallel fashion instead:

$$\boldsymbol{x} \otimes \boldsymbol{y} = \quad \text{and} \quad (\boldsymbol{x} \otimes \boldsymbol{y})^* =$$



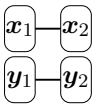
This definition extends linearly to general tensors on $H^{\otimes 2}$:

$$\boldsymbol{t} = \sum_i a_i \boldsymbol{x}_i \otimes \boldsymbol{y}_i = \quad \text{and} \quad \boldsymbol{t}^* = \sum_i a_i \boldsymbol{x}_i^* \otimes \boldsymbol{y}_i^* =$$



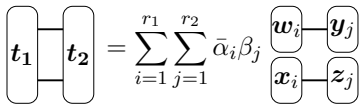
The parallel alignment of tensor products ensures that extended scalar product factorizes appropriately for elementary tensors:

$$\langle \boldsymbol{x}_1 \otimes \boldsymbol{y}_1, \boldsymbol{x}_2 \otimes \boldsymbol{y}_2 \rangle =$$



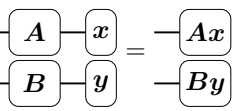
This concept extends linearly to more general tensors. Fix $\boldsymbol{t}_1 = \sum_{i=1}^{r_1} \alpha_i \boldsymbol{w}_i \otimes \boldsymbol{x}_i$ and $\boldsymbol{t}_2 = \sum_{j=1}^{r_2} \beta_j \boldsymbol{y}_j \otimes \boldsymbol{z}_j$. Then,

$$\langle \boldsymbol{t}_1, \boldsymbol{t}_2 \rangle = \quad = \sum_{i=1}^{r_1} \sum_{j=1}^{r_2} \bar{\alpha}_i \beta_j \quad = \sum_{i=1}^{r_1} \sum_{j=1}^{r_2} \bar{\alpha}_i \beta_j \langle \boldsymbol{w}_i, \boldsymbol{y}_j \rangle \langle \boldsymbol{x}_i, \boldsymbol{z}_j \rangle.$$



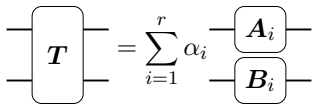
The action of elementary tensor product operators $\boldsymbol{A} \otimes \boldsymbol{B}$ also factorizes appropriately:

$$(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{x} \otimes \boldsymbol{y}) = \quad = \quad .$$



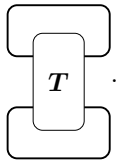
Similar to general tensor products of vectors, we denote general tensor product operators by big boxes with (in total) 4 index lines:

$$= \sum_{i=1}^{r} \alpha_i$$

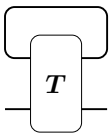
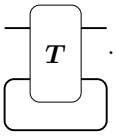


The *trace* on $\mathcal{L}(H \otimes H)$ again corresponds to a full index contraction. It aligns in- and out-going indices on both spaces and sums over both. In wiring notation:

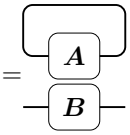
$$\mathrm{tr}(\boldsymbol{T}) = \quad .$$

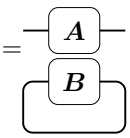


For $\mathcal{L}(H)$, the trace is the only index contraction. For $\mathcal{L}(H \otimes H)$ partial contractions are also possible (align only one pair of indices). The two options correspond to *partial traces* over the first and second tensor factor, respectively:

$\mathrm{tr}_1(\boldsymbol{T}) =$  and $\mathrm{tr}_2(\boldsymbol{T}) =$  .

More formally, these partial contractions are defined for elementary tensor product operators $\boldsymbol{A} \otimes \boldsymbol{B}$ and linearly extended to all of $\mathcal{L}(H \otimes H)$:

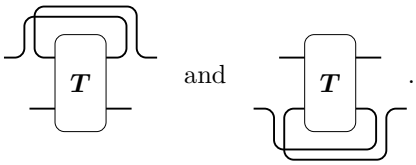
$\mathrm{tr}_1(\boldsymbol{A} \otimes \boldsymbol{B}) =$  $= \mathrm{tr}(\boldsymbol{A})\boldsymbol{B} \in \mathcal{L}(H),$

$\mathrm{tr}_2(\boldsymbol{A} \otimes \boldsymbol{B}) =$  $= \mathrm{tr}(\boldsymbol{B})\boldsymbol{A} \in \mathcal{L}(H).$

Similarly, we define the *partial transposes* on elementary tensor products

$$\mathrm{PT}_1(\boldsymbol{A} \otimes \boldsymbol{B}) = \boldsymbol{A}^T \otimes \boldsymbol{B} \quad \text{and} \quad \mathrm{PT}_2(\boldsymbol{A} \otimes \boldsymbol{B}) = \boldsymbol{A} \otimes \boldsymbol{B}^T$$

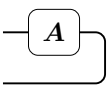
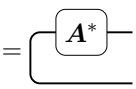
and extend them linearly to $\mathcal{L}(H^{\otimes 2})$. In contrast to the previous definitions, these operations are basis dependent. The wiring formula for ordinary transposes readily generalizes to partial transposition:

 and . (3)

Finally, we introduce a useful correspondence between $\mathcal{L}(H)$ and $H^{\otimes 2}$. Note that both spaces are linear and have the same dimension. *Vectorization* is a bijective map from $\mathcal{L}(H)$ to $H^{\otimes 2}$ that makes this correspondence precise. For $1 \leq i, j \leq d$ define

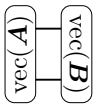
$$\mathrm{vec}\Big(\boldsymbol{e}_i \boldsymbol{e}_j^T\Big) = \boldsymbol{e}_i \otimes \boldsymbol{e}_j \in H^{\otimes 2}.$$

The operators $\boldsymbol{E}_{ij} = \boldsymbol{e}_i \boldsymbol{e}_j^T$ form a basis of $\mathcal{L}(H)$ and allow for generalizing this definition linearly to all $\mathcal{L}$. In wiring notation,

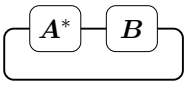
$\mathrm{vec}(\boldsymbol{A}) =$  and $\mathrm{vec}(\boldsymbol{A})^* =$ 

This correspondence is basis dependent, but does preserve the natural inner products associated with both spaces:

$$\langle \mathrm{vec}(\boldsymbol{A}), \mathrm{vec}(\boldsymbol{B}) \rangle =$$



$$=$$



$$= \mathrm{tr}(\boldsymbol{A}^* \boldsymbol{B}) = (\boldsymbol{A}, \boldsymbol{B}).$$

This isometry connects the (extended) Euclidean inner product on $H^{\otimes 2}$ with the Frobenius inner product on $\mathcal{L}(H)$.

Remark 2.2. In Lecture 2, we discussed a concrete realization of $H^{\otimes 2}$ as the linear hull of all outer products $\mathcal{L}(H)$. This realization is equivalent to inverting the vectorization map.

3 Joint Probability Distributions

Recall that a classical (discrete) probability space is fully characterized by a probability vector

$$\boldsymbol{p} \in \Delta_{d-1} = \left\{ \boldsymbol{x} \in \mathbb{R}^d : \ \boldsymbol{x} \geq \boldsymbol{0}, \ \langle \boldsymbol{1}, \boldsymbol{x} \rangle = 1 \right\} = \mathrm{conv}\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\} \subset \mathbb{R}^d.$$

Measurements correspond to resolutions of the identity (vector) $\{\boldsymbol{h}_a : \ a \in A\} \subset \mathbb{R}^d$: $\boldsymbol{h}_a \geq \boldsymbol{0}$ for each $a \in A$ and $\sum_{a \in A} \boldsymbol{h}_a = \boldsymbol{1}$. The probability rule is given by the standard inner product on $\mathbb{R}^d$:

$$\Pr[a|\boldsymbol{p}] = \langle \boldsymbol{h}_a, \boldsymbol{p} \rangle \quad \text{for all} \quad a \in A.$$

This formalism fully describes a *single* d-variate random variable. A natural extension is to consider *joint* random variables. Here we restrict ourselves to joint distributions on a pair of d-variate random variables. Extensions to different dimensions and more distributions are straightforward.

3.1 Independent distributions

Let Δ_{d-1} be the standard probability simplex in $\mathbb{R}^d$. Define

$$\Delta_{d-1} \hat{\otimes} \Delta_{d-1} = \{ \boldsymbol{p} \otimes \boldsymbol{q} : \ \boldsymbol{p}, \boldsymbol{q} \in \Delta_{d-1} \} \subset \mathbb{R}^d \otimes \mathbb{R}^d. \tag{4}$$

The notation $\hat{\otimes}$ underlines that this is not the usual tensor product. We do not allow for convex (or linear) mixtures, only elementary tensor products feature. This in turn implies that $\Delta_{d-1} \hat{\otimes} \Delta_{d-1}$ is not a convex set. The fact that a joint probability distribution corresponds to an elementary tensor product has important consequences. The probability rule associated with such joint distributions factorizes. More precisely, let $\{\boldsymbol{h}_a : a \in A\}, \{\boldsymbol{h}_b : b \in A\} \subset \mathbb{R}^d$ be two measurements that address the first and second random variables respectively. Then, the combined measurement on both random variables becomes

$$\{ \boldsymbol{h}_a \otimes \boldsymbol{h}_b : \ a \in A, B \in B \} \subset \mathbb{R}^d \hat{\otimes} \mathbb{R}^d$$

and the (extended) probability rule factorizes:

$$\Pr[a, b|\boldsymbol{p} \otimes \boldsymbol{q}] = \langle \boldsymbol{h}_a \otimes \boldsymbol{h}_b, \boldsymbol{p} \otimes \boldsymbol{q} \rangle = \langle \boldsymbol{h}_a, \boldsymbol{p} \rangle \langle \boldsymbol{h}_b, \boldsymbol{q} \rangle = \Pr[a|\boldsymbol{p}] \Pr[b|\boldsymbol{q}].$$

This is the defining property of two *independent* random variables.

Fact 3.1. *The set of joint independent probability distributions corresponds to* $\Delta_{d-1} \otimes \Delta_{d-1} = \{\boldsymbol{p} \otimes \boldsymbol{q} : \boldsymbol{p}, \boldsymbol{q} \in \Delta_{d-1}\}$.

Example 3.2 (two fair coins)**.** The probability vector associated with a fair coin toss is $\boldsymbol{p} = 2^{-1}\mathbf{1} \in \mathbb{R}^2$. The joint probability distribution of two independent coin tosses becomes

$$\left(\frac{1}{2}\mathbf{1}\right) \otimes \left(\frac{1}{2}\mathbf{1}\right) \simeq \frac{1}{4}\mathbf{1} \in \mathbb{R}^4.$$

Example 3.3 (Two deterministic distributions)**.** Let $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$ denote the standard basis of $\mathbb{R}^d$. Then, each distribution of the form $\boldsymbol{p} = \boldsymbol{e}_i$ is deterministic. A joint distribution of deterministic probability vectors is independent:

$$\boldsymbol{p}_{\text{joint}} = \boldsymbol{e}_i \otimes \boldsymbol{e}_j \in \Delta_{d-1} \hat{\otimes} \Delta_{d-1}.$$

4 Correlated random variables

Independent joint probability distributions are not everything. Correlated distributions are also possible. Note that every joint probability distribution $\boldsymbol{p}_{\text{joint}} \in \mathbb{R}^d \otimes \mathbb{R}^d$ is necessarily entry-wise positive and normalized. Therefore, it must be contained in the extended probability simplex:

$$\Delta_{d^2-1} = \left\{\boldsymbol{t} \in \mathbb{R}^d \otimes \mathbb{R}^d : \boldsymbol{t} \geq \mathbf{0}, \langle \mathbf{1} \otimes \mathbf{1}, \boldsymbol{t} \rangle = 1\right\}. \tag{5}$$

Importantly, $\Delta_{d-1} \hat{\otimes} \Delta_{d-1} \subset \Delta_{d^2-1}$ and this inclusion is strict.

Example 4.1. Consider a joint distribution of the form

$$\boldsymbol{p}_{\text{joint}} = \frac{1}{2}(\boldsymbol{e}_1 \otimes \boldsymbol{e}_2 + \boldsymbol{e}_2 \otimes \boldsymbol{e}_1) \in \Delta_{d^2-1}.$$

It is easy to check that $\boldsymbol{p}_{\text{joint}} \notin \Delta_{d-1} \otimes \Delta_{d-1}$.

Definition 4.2. We call a joint probability distribution $\boldsymbol{p}_{\text{joint}} \in \mathbb{R}^d \otimes \mathbb{R}^d$ *correlated* if it is not independent, i.e. $\boldsymbol{p}_{\text{joint}} \notin \Delta_{d-1} \hat{\otimes} \Delta_{d-1}$.

The following simple argument provides a geometric connection between independent random variables and every possible joint probability distribution.

Proposition 4.3. *The two sets* (4) *and* (5) *have the following relation:* $\operatorname{conv}\{\Delta_{d-1} \hat{\otimes} \Delta_{d-1}\} = \Delta_{d^2-1}$.

Proof. Persistence ensures $\boldsymbol{p} \otimes \boldsymbol{q} \geq \mathbf{0}$ and $\langle \mathbf{1} \otimes \mathbf{1}, \boldsymbol{p}, \boldsymbol{q} \rangle = \langle \mathbf{1}, \boldsymbol{p} \rangle \langle \mathbf{1}, \boldsymbol{q} \rangle = 1$ for any $\boldsymbol{p}, \boldsymbol{q} \in \Delta_{d-1}$. This readily implies $\Delta_{d-1} \hat{\otimes} \Delta_{d-1} \subseteq \Delta_{d^2-1}$. Conversely, recall that Δ_{d-1} is the convex hull of its extreme points: $\Delta_{d-1} = \operatorname{conv}\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\}$. Tensor products of such deterministic distributions are independent and we conclude

$$\operatorname{conv}\{\Delta_{d-1} \hat{\otimes} \Delta_{d-1}\} \supseteq \operatorname{conv}\{\boldsymbol{e}_i \otimes \boldsymbol{e}_j : 1 \leq i, j \leq d\} = \Delta_{d^2-1},$$

where the last equation follows from the fact that independent deterministic distributions constitute all extreme points of the larger simplex. □

Corollary 4.4 (Full characterization of joint probability distributions)**.** *Joint probability distributions are either independent, or correlated (there is no joint option). Moreover, every correlated distribution corresponds to a convex mixture of independent distributions:*

$$\underbrace{\Delta_{d-1}\hat{\otimes}\Delta_{d-1}}_{\textit{independent}} \subset \underbrace{\mathrm{conv}\{\Delta_{d-1}\hat{\otimes}\Delta_{d-1}\}}_{\textit{correlated}} = \underbrace{\Delta_{d^2-1}}_{\textit{everything}}.$$

5 Joint states of bipartite quantum systems

Recall that a single quantum mechanical system is described by a (probability) *density matrix*:

$$\boldsymbol{\rho} \in \Big\{\boldsymbol{X} \in \mathbb{H}^d :\ \boldsymbol{X} \succeq \mathbf{0},\ (\boldsymbol{I}, \boldsymbol{X}) = \mathrm{tr}(\boldsymbol{X}) = 1\Big\} = \mathsf{S}\Big(\mathbb{H}^d\Big).$$

Next consider a joint quantum system that is comprised of two quantum mechanical particles. The associated joint density matrix lives in the tensor product $\mathbb{H}^d \otimes \mathbb{H}^d \subset \mathcal{L}(\mathbb{C}^d \otimes \mathbb{C}^d)$. In full analogy to the analysis of classical probability distributions, we introduce the following three sets:

(i) independent joint quantum states: $\mathsf{S}(\mathbb{H}^d)\hat{\otimes}\mathsf{S}(\mathbb{H}^d) = \Big\{\boldsymbol{\rho}\otimes\boldsymbol{\sigma} :\ \boldsymbol{\rho},\boldsymbol{\sigma} \in \mathcal{S}(\mathbb{H}^d)\Big\}$,

(ii) Separable joint quantum states: $\mathrm{SEP}\Big(\mathbb{H}^d \otimes \mathbb{H}^d\Big) := \mathrm{conv}\Big\{\mathsf{S}(\mathbb{H}^d)\hat{\otimes}\mathsf{S}(\mathbb{H}^d)\Big\}$. This set encompasses all quantum state that arise as convex mixtures of independent joint quantum states.

(iii) all possible states: $\mathsf{S}\Big(\mathbb{H}^d \otimes \mathbb{H}^d\Big) = \Big\{\boldsymbol{X} \in \mathbb{H}^d \otimes \mathbb{H}^d :\ \boldsymbol{X} \succeq \mathbf{0},\ (\mathbf{I}\otimes\mathbf{I}, \boldsymbol{X}) = 1\Big\}$.

These sets are related by the following inclusions:

$$\underbrace{\mathsf{S}(\mathbb{H}^d)\hat{\otimes}\mathsf{S}(\mathbb{H}^d)}_{\text{independent}} \subseteq \underbrace{\mathrm{SEP}\Big(\mathbb{H}^d \otimes \mathbb{H}^d\Big)}_{\text{correlated}} \subseteq \underbrace{\mathsf{S}(\mathbb{H}^d \otimes \mathbb{H}^d)}_{\text{everything}}. \tag{6}$$

5.1 The Positive Partial Transpose Test

It should not come as a surprise that the first inclusion in Rel. (6) is strict. Correlated joint quantum distributions do exist. A more interesting question is whether correlations span the entire space of joint density matrices.

A seminal test designed by Horodecki[3] and Peres (1996) addresses this question and provides a necessary condition for joint quantum states to be separable.

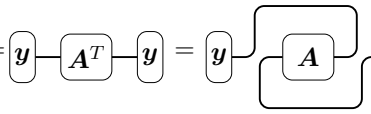
Theorem 5.1 (The Positive-Partial-Transpose (PPT) test)**.** *Every separable joint quantum state* $\boldsymbol{\rho}_{\text{joint}} \in \mathrm{SEP}\Big(\mathbb{H}^d \otimes \mathbb{H}^d\Big)$ *admits positive semidefinite partial transposes:*

$$\mathrm{PT}_1(\boldsymbol{\rho}_{\text{joint}}) \succeq \mathbf{0} \quad \textit{and} \quad \mathrm{PT}_2(\boldsymbol{\rho}_{\text{joint}}) \succeq \mathbf{0}.$$

Here, $\mathrm{PT}_1, \mathrm{PT}_2 :\ \mathbb{H}^d \otimes \mathbb{H}^d \to \mathbb{H}^d \otimes \mathbb{H}^d$ *denote the partial transposes* (3).

Proof. First note that ordinary transposition does not affect positive semidefinitiness. Suppose that $\boldsymbol{A} \in \mathbb{H}^d$ is positive semidefinite and fix $\boldsymbol{y} \in \mathbb{C}^d$. Then,

$$\langle \boldsymbol{y}, \boldsymbol{A}^T \boldsymbol{y} \rangle =$$



$$= \langle \bar{\boldsymbol{y}}, \boldsymbol{A} \bar{\boldsymbol{y}} \rangle$$

which is always non-negative, because $\boldsymbol{A} \succeq \boldsymbol{0}$. Next, choose $\boldsymbol{\rho}_{\text{joint}} \in \text{Sep}\Big(\mathbb{H}^d \otimes \mathbb{H}^d\Big)$ and decompose it as $\boldsymbol{\rho}_{\text{joint}} = \sum_{i=1}^r \tau_i \boldsymbol{\sigma}_i^{(1)} \otimes \boldsymbol{\sigma}_i^{(2)}$. Linearity of the partial transpose then implies

$$\text{PPT}_1(\boldsymbol{\rho}_{\text{joint}}) = \sum_{i=1}^r \tau_i \text{PPT}_1\Big(\boldsymbol{\sigma}_i^{(1)} \otimes \boldsymbol{\sigma}_i^{(2)}\Big) = \sum_{i=1}^r \tau \Big(\boldsymbol{\sigma}_i^{(1)}\Big)^T \otimes \boldsymbol{\sigma}_i^{(2)}.$$

This operator is necessarily positive semidefinite, because of persistence and the fact that psd operators form a convex cone. An analogous argument can be made for the partial transpose over the second tensor product factor. □

5.2 Entanglement

One of the most interesting features of quantum distributions is the following discrepancy: *convex mixtures of independent quantum distributions do not reach all possible joint quantum distributions*. The second inclusion in Rel. (6) is strict!

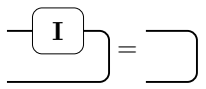
Lemma 5.2. *Set* $\boldsymbol{\Omega} = d^{-1}\text{vec}(\mathbf{I})\text{vec}(\mathbf{I})$*, where* $\mathbf{I} \in \mathbb{H}^d$ *denotes the identity. Then,* $\boldsymbol{\Omega} \in \mathcal{S}\Big(\mathbb{H}^d \otimes \mathbb{H}^d\Big)$*, but* $\text{PPT}_1(\boldsymbol{\Omega})$ *is not positive semidefinite.*

The following statement is an immediate consequence of the PPT-test (Theorem 5.1).

Corollary 5.3. $\boldsymbol{\Omega} \in \mathcal{S}(\mathbb{H}^d \otimes \mathbb{H}^d)$*, but* $\boldsymbol{\Omega} \notin \text{Sep}\Big(\mathbb{H}^d \otimes \mathbb{H}^d\Big)$.

Proof of Lemma 5.2. Vectorization of the identity matrix assumes a particularly simple form in wiring calculus:

$$\text{vec}(\mathbf{I}) =$$

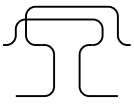


$$\quad \text{and} \quad \boldsymbol{\Omega} = \frac{1}{d}$$

.

This allows us to compute the partial transpose pictorially:

$$\text{PPT}_1(\boldsymbol{\Omega}) = \frac{1}{d}\text{PT}_1\Bigg($$



$$\Bigg) =$$

$$= \frac{1}{d}$$

.

The operator on the right is called the *flip operator*: $\mathbb{F} \in \mathcal{L}(\mathbb{C}^d \otimes \mathbb{C}^d)$. It acts on elementary tensors by permuting them: $\mathbb{F}\boldsymbol{x} \otimes \boldsymbol{y} = \boldsymbol{y} \otimes \boldsymbol{x}$. It is easy to check that the flip operator is self-adjoint and also obeys $\mathbb{F}^2 = \mathbf{I} \otimes \mathbf{I}$. The spectrum of such operators must be contained in $\{\pm 1\}$ and the only hermitian+unitary operator with positive

eigenvalues ($\lambda = 1$ with d^2-fold degeneracy) is the identity. Clearly, $\mathbb{F}$ is not the identity and therefore it must have negative eigenvalues and cannot be positive semidefinite. More concretely, note that

$$\mathbb{F}(\boldsymbol{x} \otimes \boldsymbol{y} - \boldsymbol{y} \otimes \boldsymbol{x}) = -(\boldsymbol{x} \otimes \boldsymbol{y} - \boldsymbol{y} \otimes \boldsymbol{x}).$$

This observation identifies eigenvectors of $\mathbb{F}$ associated to eigenvalue $\lambda = -1$. □

Definition 5.4 (Entanglement)**.** A joint quantum state $\boldsymbol{\rho}_{\text{joint}} \in \mathsf{S}\Big(\mathbb{H}^d \otimes \mathbb{H}^d\Big)$ is called *entangled* if it is not separable, i.e. $\boldsymbol{\rho}_{\text{joint}} \notin \text{Sep}\Big(\mathbb{H}^d \otimes \mathbb{H}^d\Big)$

The name entanglement dates back to Schrödinger who coined this term in a letter to Einstein in 1935 ("Verschränkung" in German, translated to English by Schrödinger himself).

Entangled states arise from correlations that do not have a classical counter-part. This led physicists to use words like "mysterious" and "elusive" to describe entanglement. Today, this strong form of correlation forms the basis of many "cool" quantum technologies, like quantum teleportation (homework), super-dense coding, quantum key distributions and quantum computing.

Lecture 04: Symmetric and antisymmetric tensors

Scribe: Erick Moen

ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp
April 10, 2019

1 Agenda

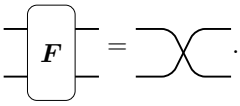
1. Flip operators
2. Symmetric and antisymmetric subspaces of $H^{\otimes k}$
3. The determinant
4. The permanent

2 Symmetric and antisymmetric subspaces of $H^{\otimes k}$

Let H denote a d-dimensional inner product space with designated orthonormal basis $\boldsymbol{e}_1, \dots, \boldsymbol{e}_d$. The *flip operator* $\boldsymbol{F} : H^{\otimes 2} \to H^{\otimes 2}$ permutes elementary tensor products:

$$\boldsymbol{F}\boldsymbol{x} \otimes \boldsymbol{y} = \boldsymbol{y} \otimes \boldsymbol{x} \quad \text{for all} \quad \boldsymbol{x}, \boldsymbol{y} \in H^{\otimes 2}.$$

This action can be extended linearly to all of $H^{\otimes 2}$. In wiring calculus, the flip operator assumes the following form:



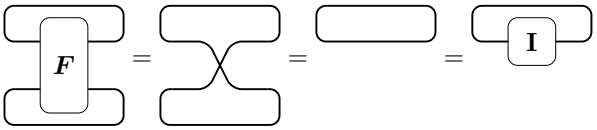
It is easy to check that the flip operator is self-adjoint ($\boldsymbol{F}^* = \boldsymbol{F} \in \mathcal{L}(H^{\otimes 2})$) and unitary:

$$\boldsymbol{F}^2 =$$

$$= \mathbf{I} \otimes \mathbf{I}.$$

The trace of $\boldsymbol{F}$ can also readily be computed using wiring calculus:

$$\operatorname{tr}(\boldsymbol{F}) =$$



$$= \operatorname{tr}(\mathbf{I}) = \dim(H) = d.$$

Define the following operators $\boldsymbol{P}, \boldsymbol{Q} \in \mathcal{L}(H^{\otimes 2})$:

$$\boldsymbol{P} = \frac{1}{2}\left(\mathbf{I}^{\otimes 2} + \boldsymbol{F}\right) =$$

$$\quad \text{and} \quad \boldsymbol{Q} = \frac{1}{2}\left(\mathbf{I}^{\otimes 2} - \boldsymbol{F}\right) =$$

By construction, these operators are self-adjoint and have the following properties:

$$\boldsymbol{P}^2 = \boldsymbol{P}, \ \boldsymbol{Q}^2 = \boldsymbol{Q}, \quad \text{and} \quad \boldsymbol{P}\boldsymbol{Q} = \mathbf{0}.$$

This implies that $\boldsymbol{P}$ and $\boldsymbol{Q}$ are orthogonal projectors onto disjoint subspaces of $H^{\otimes 2}$.

Definition 2.1 (Symmetric and antisymmetric subspaces of $H^{\otimes 2}$)**.** The *symmetric subspace* of $H^{\otimes 2}$ is the range of the orthogonal projector $\boldsymbol{P}_{\vee^2} = \frac{1}{2}(\mathbf{I} + \boldsymbol{F})$:

$$\bigvee^k = \mathrm{range}(\boldsymbol{P}_{\vee^2}) = \mathrm{span}\{\boldsymbol{x} \otimes \boldsymbol{y} + \boldsymbol{y} \otimes \boldsymbol{x} : \ \boldsymbol{x}, \boldsymbol{y} \in H\}.$$

The *antisymmetric subspace* of $H^{\otimes 2}$ is the range of the orthogonal projector $\boldsymbol{P}_{\wedge^2} = \frac{1}{2}(\mathbf{I} - \boldsymbol{F})$:

$$\bigwedge^k = \mathrm{range}(\boldsymbol{P}_{\wedge^2}) = \mathrm{span}\{\boldsymbol{x} \otimes \boldsymbol{y} - \boldsymbol{y} \otimes \boldsymbol{x} : \ \boldsymbol{x}, \boldsymbol{y} \in H\}.$$

This notation is appropriate. The symmetric subspace contains all tensor products that are symmetric under permutation of tensor factors. In contrast, the antisymmetric subspace contains all tensor products that change sign upon permutations. This decomposition of $H^{\otimes 2}$ into symmetric and antisymmetric elements is complete.

Proposition 2.2. *Suppose that H has dimension d. Then,*

$$H^{\otimes 2} = \bigvee^2 \bigoplus \bigwedge^2 \quad \textit{and} \quad \dim\Big(\bigvee^2\Big) = \binom{d+1}{2}, \ \dim\Big(\bigwedge^2\Big) = \binom{d}{2}.$$

Proof. Use $\mathrm{tr}(\mathbf{I} \otimes \mathbf{I}) = \mathrm{tr}(\mathbf{I})^2 = d^2$ and $\mathrm{tr}(\boldsymbol{F}) = d$ to evaluate the dimensions of these subspaces:

$$\begin{aligned}
\dim\Big(\bigvee^2\Big) &= \mathrm{tr}(\boldsymbol{P}_{\vee^2}) = \frac{1}{2}(\mathrm{tr}(\mathbf{I} \otimes \mathbf{I}) + \mathrm{tr}(\boldsymbol{F})) = \frac{1}{2}(d^2 + d) = \binom{d+1}{2}, \\
\dim\Big(\bigwedge^2\Big) &= \mathrm{tr}(\boldsymbol{P}_{\wedge^2}) = \frac{1}{2}(\mathrm{tr}(\mathbf{I} \otimes \mathbf{I}) - \mathrm{tr}(\boldsymbol{F})) = \frac{1}{2}(d^2 - d) = \binom{d}{2}.
\end{aligned}$$

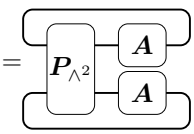
Next, $\boldsymbol{P}_{\vee^2}\boldsymbol{P}_{\wedge^2} = \mathbf{0}$ which ensures that both subspaces are mutually orthogonal. Finally, add the dimensions to check that the direct sum of both subspaces cover all of $H^{\otimes 2}$:

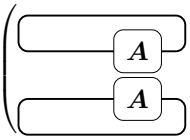
$$\dim\Big(\bigvee^2\Big) + \dim\Big(\bigwedge^2\Big) = \binom{d+1}{2} + \binom{d}{2} = d^2 = \dim\Big(H^{\otimes 2}\Big).$$

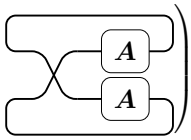
□

We conclude this section with a highly instructive example.

Example 2.3. Fix any operator $\boldsymbol{A} \in \mathcal{L}(H)$. Then,

$\mathrm{tr}(\boldsymbol{P}_{\wedge^2}\boldsymbol{A}) =$ 

$= \frac{1}{2}\Bigg($ 

$-$ 

$\Bigg) = \frac{1}{2}\Big(\mathrm{tr}(\boldsymbol{A})^2 - \mathrm{tr}\Big(\boldsymbol{A}^2\Big)\Big).$

Next, fix $d = 2$ and set $\boldsymbol{A} = [a_{ij}]_{i,j=1}^2$. Then,

$$\mathrm{tr}(\boldsymbol{P}_{\wedge^2}\boldsymbol{A}) = \frac{1}{2}\Big(\mathrm{tr}(\boldsymbol{A})^2 + \mathrm{tr}\Big(\boldsymbol{A}^2\Big)\Big)a_{11}a_{22} - a_{12}a_{21} = \det(\boldsymbol{A}).$$

This is not a coincidence, as we shall see later. For $d = 2$, the antisymmetric subspace has dimension $\binom{2}{2} = 1$. Evaluating the action of $\boldsymbol{A}^{\otimes 2}$ on this one-dimensional subspace produces a famous polynomial: the determinant.

3 Symmetric and antisymmetric subspaces of $H^{\otimes k}$

Let us now generalize the constructions of symmetric and antisymmetric subspaces to tensor spaces of order $k \geq 3$. Let $\mathcal{S}_k$ denote the group of permutations of k elements. For $\pi \in \mathcal{S}_k$ define $\boldsymbol{W}_\pi \in \mathcal{L}(H^{\otimes k})$ via

$$\boldsymbol{W}_\pi \boldsymbol{x}_1 \otimes \cdots \otimes \boldsymbol{x}_k = \boldsymbol{x}_{\pi^{-1}(1)} \otimes \cdots \otimes \boldsymbol{x}_{\pi^{-1}(k)} \quad \text{for} \quad \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \in H$$

and extend this definition linearly. It is easy to check that these operators are unitary and respect the group structure of $\mathcal{S}_k$: $\boldsymbol{W}_\pi \boldsymbol{W}_\tau = \boldsymbol{W}_{\pi\circ\tau}$. Hence, they form a *unitary representation* of the symmetric group $\mathcal{S}_k$ on $H^{\otimes k}$.

Example 3.1 ($k = 2$)**.** For $k = 2$, there are only two permutations: the identity and transposition ("flip"). In wiring notation:

$$\{\boldsymbol{W}_\pi : \pi \in \mathcal{S}_2\} = \Big\{ \quad , \quad \Big\} \subset \mathcal{L}\Big(H^{\otimes 2}\Big)$$

Example 3.2 ($k = 3$)**.** For $k = 3$, there are $3! = 6$ permutation operators:

$$\Big\{ \quad , \quad , \quad , \quad , \quad , \quad \Big\} \subset \mathcal{L}\Big(H^{\otimes 3}\Big)$$

Let $\mathrm{sign}(\pi) \in \{\pm 1\}$ be the signature of a permutation $\pi \in \mathcal{S}_k$. Define

$$\boldsymbol{P}_{\vee^k} = \frac{1}{k!} \sum_{\pi \in \mathcal{S}_k} \boldsymbol{W}_\pi \quad \text{and} \quad \boldsymbol{P}_{\wedge^k} = \frac{1}{k!} \sum_{\pi \in \mathcal{S}_k} \mathrm{sign}(\pi) \boldsymbol{W}_\pi. \tag{1}$$

These are the correct definitions for projectors onto symmetric and antisymmetric subspaces of $H^{\otimes k}$. In order to demonstrate this, we need the following fact about signatures.

Fact 3.3. *The signature is multiplicative:* $\mathrm{sign}(\pi \circ \tau) = \mathrm{sign}(\pi)\mathrm{sign}(\tau)$ *for all* $\pi, \tau \in \mathcal{S}_k$

Proposition 3.4 (Symmetric and antisymmetric subspace of $H^{\otimes k}$)**.** *The operators* $\boldsymbol{P}_{\vee^k}, \boldsymbol{P}_{\wedge^k} \subset \mathcal{L}\Big(H^{\otimes k}\Big)$ *defined in Equation* (1) *are orthogonal projectors onto mutually orthogonal subspaces*

$$\bigvee^k = \mathrm{range}(\boldsymbol{P}_{\vee^k}) \quad \textit{and} \quad \bigwedge^k = \mathrm{range}(\boldsymbol{P}_{\wedge^k}).$$

Proof. First, note that $\boldsymbol{W}_{\pi^{-1}} = \boldsymbol{W}_\pi^*$ for any $\pi \in \mathcal{S}_k$. Moreover, inversion doesn't change the signature. Therefore,

$$\boldsymbol{P}_{\wedge^k}^* = \frac{1}{k!} \sum_{\pi \in \mathcal{S}_k} \mathrm{sign}(\pi) \boldsymbol{W}_\pi^* = \frac{1}{k!} \sum_{\pi \in \mathcal{S}_k} \mathrm{sign}\Big(\pi^{-1}\Big) \boldsymbol{W}_{\pi^{-1}} = \frac{1}{k!} \sum_{\pi' \in \mathcal{S}_k} \mathrm{sign}(\pi') \boldsymbol{W}_{\pi'} = \boldsymbol{P}_{\wedge^k},$$

because permutations form a group and it doesn't matter if we sum over them or their inverses. The operator $\boldsymbol{P}_{\wedge^k}$ is self-adjoint. Next, use multiplicativity of the sign to conclude

$$\begin{aligned}\boldsymbol{P}_{\wedge^k}^2 =& \frac{1}{(k!)^2}\sum_{\pi,\tau\in\mathcal{S}_k}\operatorname{sign}(\pi)\operatorname{sign}(\tau)\boldsymbol{W}_\pi\boldsymbol{W}_\tau = \frac{1}{(k!)^2}\sum_{\pi,\tau\in\mathcal{S}_k}\operatorname{sign}(\pi\circ\tau)\boldsymbol{W}_{\pi\circ\tau}\\ =& \frac{1}{k!}\sum_{\pi\in\mathcal{S}_k}\operatorname{sign}(\pi)\boldsymbol{W}_\pi.\end{aligned}$$

Here, we have once more used the fact that $\mathcal{S}_k$ is a group. Hence, $\boldsymbol{P}_{\wedge^k}$ is indeed an orthogonal projector. A similar argument establishes that $\boldsymbol{P}_{\vee^k}$ is also a projector. Finally, note that

$$\begin{aligned}\boldsymbol{P}_{\wedge^k}\boldsymbol{P}_{\vee^k} =& \frac{1}{(k!)^2}\sum_{\pi,\tau\in\mathcal{S}_k}\operatorname{sign}(\pi)\boldsymbol{W}_\pi\boldsymbol{W}_\tau = \frac{1}{k!}\sum_{\pi\in\mathcal{S}_k}\operatorname{sign}(\pi)\left(\frac{1}{k!}\sum_{\tau\in\mathcal{S}_k}\boldsymbol{W}_{\pi\circ\tau}\right)\\ =& \left(\frac{1}{k!}\sum_{\pi\in\mathcal{S}_k}\operatorname{sign}(\pi)\right)\boldsymbol{P}_{\vee^k} = \boldsymbol{0}.\end{aligned}$$

which establishes orthogonality. The final equation is due to the fact that the sign is antisymmetric. Averaging over all possible signatures must yield zero. □

Definition 3.4 naturally extends symmetry and antisymmetry to higher order tensor spaces.

Lemma 3.5. *Fix $\boldsymbol{x}_1,\dots,\boldsymbol{x}_k\in H$. Then,*

$$\begin{aligned}\boldsymbol{P}_{\vee^k}\boldsymbol{x}_2\otimes\boldsymbol{x}_1\otimes\boldsymbol{x}_3\otimes\cdots\otimes\boldsymbol{x}_k =& \boldsymbol{x}_1\otimes\boldsymbol{x}_2\otimes\boldsymbol{x}_3\otimes\cdots\otimes\boldsymbol{x}_k,\\ \boldsymbol{P}_{\wedge^k}\boldsymbol{x}_2\otimes\boldsymbol{x}_1\otimes\boldsymbol{x}_3\otimes\cdots\otimes\boldsymbol{x}_k =& -\boldsymbol{x}_1\otimes\boldsymbol{x}_2\otimes\boldsymbol{x}_3\otimes\cdots\otimes\boldsymbol{x}_k,\end{aligned}$$

and similarly for any other exchange (flip) of two factors. In particular,

$$\boldsymbol{P}_{\wedge^k}\boldsymbol{x}\otimes\boldsymbol{x}\otimes\boldsymbol{x}_3\otimes\cdots\otimes\boldsymbol{x}_k = \boldsymbol{0}\quad\text{for any}\quad\boldsymbol{x}\in H.$$

Proof. Exchanging two factors corresponds to a certain transposition $\tau\in\mathcal{S}_k$: $\boldsymbol{x}_1\otimes\boldsymbol{x}_1\otimes\cdots\otimes\boldsymbol{x}_k = \boldsymbol{W}_\tau\boldsymbol{x}_1\otimes\cdots\otimes\boldsymbol{x}_k$. Transpositions have sign -1. The group structure of the permutation group then implies

$$\boldsymbol{P}_{\wedge^k}\boldsymbol{W}_\tau = \frac{1}{k!}\sum_{\pi\in\mathcal{S}_k}\operatorname{sign}(\pi)(-\operatorname{sign}(\tau))\boldsymbol{W}_\pi\boldsymbol{W}_\tau = -\frac{1}{k!}\sum_{\pi\in\mathcal{S}_k}\operatorname{sign}(\pi\circ\tau)\boldsymbol{W}_{\pi\circ\tau} = -\boldsymbol{P}_{\wedge^k}.$$

This establishes antisymmetry among permutations. A similar, but simpler, argument establishes symmetry for the projector onto the symmetric subspace: $\boldsymbol{P}_{\vee^k}\boldsymbol{W}_\tau = \boldsymbol{P}_{\vee^k}$ for any transposition.

The final claim is enforced by contradicting requirements. By construction, $\boldsymbol{x}\otimes\boldsymbol{x}\otimes\boldsymbol{x}_3\otimes\cdots\otimes\boldsymbol{x}_k$ is invariant under permuting the first two factors (symmetric). However, the projection onto $\bigwedge^k$ must change sign when the first factors are permuted (antisymmetry). The only tensor product that obeys $-\boldsymbol{t} = +\boldsymbol{t}$ is zero. □

Fact 3.6. *Suppose that H has dimension d. Then,*

$$\dim\Big(\bigvee^k\Big) = \binom{d+k-1}{k} \quad \textit{and} \quad \dim\Big(\bigwedge^k\Big) = \binom{d}{k}.$$

Let $\boldsymbol{e}_1 \ldots, \boldsymbol{e}_d$ be an orthonormal basis of H. Then, $\{\boldsymbol{e}_{i_1} \otimes \cdots \otimes \boldsymbol{e}_{i_k} : 1 \leq i_1, \ldots, i_k \leq d\}$ is an orthonormal basis of $H^{\otimes k}$ with respect to the extended inner product.

Applying $\boldsymbol{P}_{\vee^k}$ and $\boldsymbol{P}_{\wedge^k}$ to these basis vectors produce spanning sets for $\bigvee^k$ and $\bigwedge^k$, respectively. However, many of these symmetrized (anti-symmetrized) standard basis vectors coincide. Removing such redundancies subsequently produces an orthonormal basis of these subspaces. Counting their cardinality yields the following dimension formula.

Proposition 3.7. *Suppose that H has dimension d. Then,*

$$\dim\Big(\bigvee^k\Big) = \binom{d+k-1}{k} \quad \textit{and} \quad \dim\Big(\bigwedge^k\Big) = \binom{d}{k}.$$

Note that these two subspaces in general do not span the entire tensor product space $H^{\otimes k}$ with $k \geq 3$:

$$\dim\Big(\bigvee^k\Big) + \dim\Big(\bigwedge^k\Big) = \binom{d+k-1}{k} + \binom{d}{k} < d^k = \dim\Big(H^{\otimes k}\Big).$$

Also, the dimension of the totally symmetric subspace grows very quickly as k increases. In contrast, the dimension of the totally anti-symmetric subspace achieves its maximum for $k = \lceil d/2 \rceil$. Then, the dimension starts to decrease. A second extreme case is achieved for $k = d$:

$$\dim\Big(\bigwedge^d\Big) = \binom{d}{d} = 1.$$

For larger values of k, the subspace vanishes entirely. The following reformulation explicitly points out that the range of $\boldsymbol{P}_{\vee^d}$ is one-dimensional.

Lemma 3.8. *Let $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$ be an orthonormal basis of H. Then,*

$$\boldsymbol{P}_{\wedge^d} = d! \boldsymbol{P}_{\wedge^d} (\boldsymbol{e}_1 \otimes \cdots \otimes \boldsymbol{e}_d)(\boldsymbol{e}_1 \otimes \cdots \otimes \boldsymbol{e}_d)^* \boldsymbol{P}_{\wedge^d}.$$

Proof. We may expand the identity on H as $\mathbf{I} = \sum_{i=1}^d \boldsymbol{e}_i \boldsymbol{e}_i^*$. Persistence and the fact that $\boldsymbol{P}_{\wedge^d}$ is an orthogonal projector then imply

$$\boldsymbol{P}_{\wedge^d} = \boldsymbol{P}_{\wedge^d} \mathbf{I}^{\otimes d} \boldsymbol{P}_{\wedge^d} = \sum_{i_1, \ldots, i_d = 1}^{d} \boldsymbol{P}_{\wedge^d} (\boldsymbol{e}_{i_1} \otimes \cdots \otimes \boldsymbol{e}_{i_d})(\boldsymbol{e}_{i_1} \otimes \cdots \otimes \boldsymbol{e}_{i_d})^* \boldsymbol{P}_{\wedge^d}.$$

Most of these extended standard basis vectors must vanish when projected onto $\bigvee^d$. Indeed, Lemma 3.5 ensures that the only non-vanishing contributions are permutations of $\boldsymbol{e}_1 \otimes \cdots \otimes \boldsymbol{e}_d$. Up to signs all $k!$ such vectors get projected onto the same vector, namely $\boldsymbol{P}_{\wedge^k} \boldsymbol{e}_1 \otimes \cdots \otimes \boldsymbol{e}_d$. Potential sign flips do not matter, however. Each of these projected tensor products features twice in the expression and potential sign flips cancel out. □

We conclude this section with a powerful property of $\boldsymbol{P}_{\vee^k}$, $\boldsymbol{P}_{\wedge^k}$ and – more generally – permutation operations.

Fact 3.9. *Permutation operators commute with elementary tensor product operators: $\boldsymbol{W}_\pi \in \mathcal{L}(H^{\otimes}k)$ for any $\boldsymbol{A} \in \mathcal{L}(H)$ and any $\pi \in \mathcal{S}_k$. In particular,*

$$\boldsymbol{P}_{\vee^k}\boldsymbol{A}^{\otimes k} = \boldsymbol{A}^{\otimes k}\boldsymbol{P}_{\vee^k} \quad \textit{and} \quad \boldsymbol{P}_{\wedge^k}\boldsymbol{A}^{\otimes k} = \boldsymbol{A}^{\otimes k}\boldsymbol{P}_{\wedge^k}.$$

Intuitively, this makes sense. Tensor product operators $\boldsymbol{A}^{\otimes k}$ act identically on all factors H of $H^{\otimes k}$. Permuting first and then applying this operator is equivalent to first applying the operator and then permuting tensor factors.

4 The determinant

4.1 A tensor product formula for the determinant

Recall that the range of $\boldsymbol{P}_{\wedge^d}$ is one-dimensional. Moreover, Example 2.3 suggests a connection between the projection of $\boldsymbol{A}^{\otimes d}$ onto this 1D-subspace and the determinant (for $d = 2$). The following definition extends this intuition to arbitrary dimensions d.

Definition 4.1 (The determinant)**.** Suppose that H has dimension d. For $\boldsymbol{A} \in \mathcal{L}(H)$ define

$$\det(\boldsymbol{A}) = \operatorname{tr}\Big(\boldsymbol{P}_{\wedge^d}\boldsymbol{A}^{\otimes d}\Big).$$

It is easy to check that this formula is equivalent to the Leibniz formula for the determinant. Let $\boldsymbol{a}_1, \dots, \boldsymbol{a}_d \in H$ denote the columns of $\boldsymbol{A}$. Lemma 3.8 allows for deducing the Leibniz formula from this definition:

$$\det(\boldsymbol{A}) = \operatorname{tr}\Big(\boldsymbol{P}_{\wedge^d}\boldsymbol{A}^{\otimes d}\Big) = \sum_{\pi\in\mathcal{S}_d} \operatorname{sign}(\pi)\langle \boldsymbol{e}_1, \boldsymbol{A}\boldsymbol{e}_{\pi(1)}\rangle \cdots \langle \boldsymbol{e}_d, \boldsymbol{A}\boldsymbol{e}_{\pi(d)}\rangle. \tag{2}$$

4.2 Properties of the determinant

All fundamental properties of the determinant can be established with relative ease in this tensor product formalism.

Lemma 4.2 (Normalization)**.** *Let $\mathbf{I} \in \mathcal{L}(H)$ denote the identity. Then,* $\det(\mathbf{I}) = 1$.

Proof. The antisymmetric subspace $\bigwedge^d \subset H^{\otimes d}$ is one-dimensional. Therefore,

$$\det(\mathbf{I}) = \operatorname{tr}\Big(\boldsymbol{P}_{\wedge^d}\boldsymbol{I}^{\otimes d}\Big) = \operatorname{tr}(\boldsymbol{P}_{\wedge^d}) = \dim\Big(\bigwedge{}^d\Big) = \binom{d}{d} = 1.$$

□

Proposition 4.3 (Invariance under basis changes)**.** *Let $\boldsymbol{Q} \in \mathcal{L}(H)$ be invertible. Then,* $\det(\boldsymbol{Q}\boldsymbol{A}\boldsymbol{Q}^{-1}) = \det(\boldsymbol{A})$ *for any $\boldsymbol{A} \in \mathcal{L}(H)$.*

Proof. The projector $\boldsymbol{P}_{\wedge^d}$ commutes with $\boldsymbol{Q}^{\otimes d}$, see Fact 3.9. Therefore

$$\begin{aligned}\det\Big(\boldsymbol{Q}\boldsymbol{A}\boldsymbol{Q}^{-1}\Big) =& \operatorname{tr}\Big(\boldsymbol{P}_{\wedge^d}\boldsymbol{Q}^{\otimes d}\boldsymbol{A}^{\otimes d}(\boldsymbol{Q}^{-1})^{\otimes d}\Big) = \operatorname{tr}\Big(\boldsymbol{Q}^{\otimes d}\boldsymbol{P}_{\wedge^d}\boldsymbol{A}^{\otimes d}(\boldsymbol{Q}^{-1})^{\otimes d}\Big)\\ =& \operatorname{tr}\Big(\boldsymbol{P}_{\wedge^d}\boldsymbol{A}^{\otimes d}\Big(\boldsymbol{Q}^{-1}\boldsymbol{Q}\Big)^{\otimes d}\Big) = \operatorname{tr}\Big(\boldsymbol{P}_{\wedge^d}\boldsymbol{A}^{\otimes d}\Big) = \det(\boldsymbol{A}),\end{aligned}$$

where we have also used cyclicity of the trace. □

Corollary 4.4 (The determinant is the product of eigenvalues)**.** *Suppose that $\boldsymbol{A} \in \mathcal{L}(H)$ is diagonalizable and has eigenvalues $\lambda_1, \ldots, \lambda_d$. Then,* $\det(\boldsymbol{A}) = \prod_{i=1}^d \lambda_i$.

Proof. Let $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^{-1}$ with $\boldsymbol{D} = \operatorname{diag}(\lambda_1, \ldots, \lambda_d)$ be an eigenvalue decomposition. Note that,

$$\boldsymbol{D}^{\otimes d}\boldsymbol{P}_{\wedge^d}\boldsymbol{e}_1 \otimes \cdots \otimes \boldsymbol{e}_d = \boldsymbol{P}_{\wedge^d}\boldsymbol{D}^{\otimes d}\boldsymbol{e}_1 \otimes \cdots \otimes \boldsymbol{e}_d = \lambda_1 \cdots \lambda_d \boldsymbol{P}_{\bigwedge^d}\boldsymbol{e}_1 \otimes \cdots \otimes \boldsymbol{e}_d.$$

Proposition 4.3 and Lemma 3.8 then imply

$$\begin{aligned}\det(\boldsymbol{A}) =& \det(\boldsymbol{D}) = d!\langle \boldsymbol{e}_1 \otimes \cdots \otimes \boldsymbol{e}_d, \boldsymbol{P}_{\wedge^d}\boldsymbol{D}^{\otimes d}\boldsymbol{P}_{\wedge^d}\boldsymbol{e}_1 \otimes \cdots \otimes \boldsymbol{e}_d\rangle\\ =& \lambda_1 \cdots \lambda_d d!\langle \boldsymbol{e}_1 \otimes \cdots \otimes \boldsymbol{e}_d, \boldsymbol{P}_{\wedge^d}\mathbf{I}^{\otimes d}\boldsymbol{P}_{\wedge^d}\boldsymbol{e}_1 \otimes \cdots \otimes \boldsymbol{e}_d\rangle\\ =& \lambda_1 \cdot \lambda_d \det(\mathbf{I}).\end{aligned}$$

□

Proposition 4.5 (The determinant is multiplicative)**.** *Fix $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{L}(H)$. Then,* $\det(\boldsymbol{A}\boldsymbol{B}) = \det(\boldsymbol{A})\det(\boldsymbol{B})$.

Proof. Recall that $\boldsymbol{P}_{\wedge^d}$ is a rank-one projector that is proportional to the (projected) outer product $d!\boldsymbol{P}_{\wedge^d}\vec{\boldsymbol{e}}\vec{\boldsymbol{e}}^*\boldsymbol{P}_{\wedge^d}$. Here, we have introduced the short-hand notation $\vec{\boldsymbol{e}} = \boldsymbol{e}_1 \otimes \cdots \otimes \boldsymbol{e}_d \in H^{\otimes d}$. Also, $\boldsymbol{P}_{\wedge^d}$ commutes with $\boldsymbol{A}^{\otimes d}$. Therefore,

$$\begin{aligned}\det(\boldsymbol{A}\boldsymbol{B}) =& \operatorname{tr}\Big(\boldsymbol{P}_{\wedge^d}(\boldsymbol{A}\boldsymbol{B})^{\otimes d}\Big) = \operatorname{tr}\Big(\boldsymbol{P}_{\wedge^d}\boldsymbol{A}^{\otimes d}\boldsymbol{P}_{\wedge^d}\boldsymbol{B}^{\otimes d}\Big)\\ =& (d!)^2\operatorname{tr}\Big(\boldsymbol{P}_{\wedge^d}\vec{\boldsymbol{e}}\vec{\boldsymbol{e}}^*\boldsymbol{P}_{\wedge^d}\boldsymbol{A}^{\otimes d}\boldsymbol{P}_{\wedge^d}\vec{\boldsymbol{e}}\vec{\boldsymbol{e}}^*\boldsymbol{P}_{\wedge^d}\boldsymbol{B}^{\otimes d}\Big)\\ =& d!\langle\vec{\boldsymbol{e}}, \boldsymbol{P}_{\wedge^d}\boldsymbol{A}^{\otimes d}\boldsymbol{P}_{\wedge^d}\vec{\boldsymbol{e}}\rangle d!\langle\vec{\boldsymbol{e}}, \boldsymbol{P}_{\wedge^d}\boldsymbol{B}^{\otimes d}\boldsymbol{P}_{\wedge^d}\vec{\boldsymbol{e}}\rangle\\ =& \operatorname{tr}\Big(\boldsymbol{P}_{\wedge^d}\boldsymbol{A}^{\otimes d}\Big)\operatorname{tr}\Big(\boldsymbol{P}_{\wedge^d}\boldsymbol{B}^{\otimes d}\Big) = \det(\boldsymbol{A})\det(\boldsymbol{B}).\end{aligned}$$

□

The following useful fact follows directly from multiplicativity.

Corollary 4.6. *Let $\boldsymbol{Q} \in \mathcal{L}(H)$ be invertible. Then,* $\det(\boldsymbol{Q}^{-1}) = \det(\boldsymbol{Q})$.

Other fundamental properties are evident from the tensor product construction itself. Tensor products of vectors are multi-linear (i.e. linear in each factor) and $\boldsymbol{P}_{\wedge^d}$ anti-symmetrizes. For the determinant, these fundamental properties ensure

1. *Multi-linearity*: $\det(\boldsymbol{A})$ is linear in the columns $\boldsymbol{a}_j = \boldsymbol{A}\boldsymbol{e}_j$ of $\boldsymbol{A}$.
2. *Antisymmetry:* Exchanging two columns of $\boldsymbol{A}$ negates the determinant.

These properties completely characterize the determinant and rule out any other possibility.

Fact 4.7 (Uniqueness)**.** *The determinant is the unique matrix function that is i) multiplicative, ii) multi-linear, iii) antisymmetric and iv) obeys* $\det(\mathbf{I}) = 1$.

5 The permanent

The *permanent* is the symmetric cousin of the determinant. It is typically defined analogously to the Leibniz formula (2):

$$\operatorname{perm}(\boldsymbol{A}) = \sum_{\pi \in \mathcal{S}_d} \langle \boldsymbol{e}_1, \boldsymbol{A}\boldsymbol{e}_{\pi(1)} \rangle \cdots \langle \boldsymbol{e}_d, \boldsymbol{A}\boldsymbol{e}_{\pi(d)} \rangle.$$

Definition 5.1 (The permanent)**.** Suppose that H has dimension d. For $\boldsymbol{A} \in \mathcal{L}(H)$ define

$$\operatorname{perm}(\boldsymbol{A}) = d! \langle \boldsymbol{e}_1 \otimes \cdots \otimes \boldsymbol{e}_d, \boldsymbol{P}_{\vee^d} \boldsymbol{A}^{\otimes d} \boldsymbol{P}_{\vee^d} \boldsymbol{e}_1 \otimes \cdots \otimes \boldsymbol{e}_d \rangle.$$

This definition in terms of tensor products readily implies desirable features. The permanent is *multi-linear* and symmetric under exchanging columns of $\boldsymbol{A}$.

However, there is also a crucial difference. While the range of $\boldsymbol{P}_{\wedge^d}$ is one-dimensional, the range of $\boldsymbol{P}_{\vee^d}$ is huge. This prevents us from rewriting $\operatorname{perm}(\boldsymbol{A})$ as the trace of $\boldsymbol{P}_{\vee^d} \boldsymbol{A}^{\otimes d}$. This clever trick, however, was the basis for establishing multiple nice features of the determinant. For the permanent, such an approach is impossible. In general, rather little is known about the permanent.

The permanent is also notoriously difficult to compute. This computational discrepancy between det (easy to compute) and perm (hard to compute) forms the basic dichotomy of algebraic complexity theory (think P vs NP). Current quantum supremacy experiments ("boson sampling") are also based on the computational hardness associated with computing (generic) permanents.

Lecture 05: Haar integration

Scribe: Oguzhan Teke

ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp
April 15, 2019

1 Agenda

- Polynomials and Tensors
- Closed form expression for Haar Integrals
- Schur-Weyl Duality

2 Motivational Examples

In order to motivate Haar integration, we start this lecture by studying two simple examples. In this regard, we consider the 2-dimensional complex unit sphere

$$\mathbb{S}^1 = \{\boldsymbol{x} \in \mathbb{C}^2 \mid \langle \boldsymbol{x}, \boldsymbol{x} \rangle = 1\} \subseteq \mathbb{C}^2, \tag{1}$$

and two homogeneous polynomials:

$$p_1(\boldsymbol{x}, \bar{\boldsymbol{x}}) = x_1\,\bar{x}_1 + x_2\,\bar{x}_2 \quad \text{and} \quad p_2(\boldsymbol{x}, \bar{\boldsymbol{x}}) = x_1\,\bar{x}_2 + x_2\,\bar{x}_1. \tag{2}$$

The task is to integrate these polynomials over $\mathbb{S}^1$. A moment of thought reveals that the first polynomial is just the squared radius: $p_1(\boldsymbol{x}, \bar{\boldsymbol{x}}) = \langle \boldsymbol{x}, \boldsymbol{x} \rangle = 1$. This polynomial is constant on $\mathbb{S}^1$ and we readily conclude

$$\int_{\mathbb{S}^1} p_1(\boldsymbol{v}, \bar{\boldsymbol{v}})\, \mathrm{d}\mu(\boldsymbol{v}) = 1.$$

The second polynomial flips its sign under negating one corrdinate. Set $\boldsymbol{y} = (x_1, -x_2)$. Then $\boldsymbol{p}(\boldsymbol{y}, \bar{\boldsymbol{y}}) = -\boldsymbol{p}(\boldsymbol{x}, \bar{\boldsymbol{x}})$. This antisymmetry requires that the integral over the entire sphere must vanish:

$$\int_{\mathbb{S}^1} p_2(\boldsymbol{v}, \bar{\boldsymbol{v}})\, \mathrm{d}\mu(\boldsymbol{v}) = 0.$$

Note that we could only evaluate these integration formulas by using clever symmetry tricks. This approach becomes more challenging for higher order polynomials, e.g. $p_2(\boldsymbol{x}, \bar{\boldsymbol{x}})^2$.

Haar integration provides a general means for integrating homogeneous polynomials of any degree over complex unit spheres in any dimensions.

3 Polynomials and Tensors

Throughout this course we restrict our attention to *homogeneous polynomials*.

Definition 3.1. A polynomial $p(\boldsymbol{x})$ of degree k is *homogeneous* if all monomials have degree k. We denote the space of such polynomials by $\mathrm{Hom}^{(k)}(\boldsymbol{x})$.

This restriction is not very severe. Any degree-k polynomial in d variables can be represented as a homogeneous polynomial of the same degree in $d+1$ variables, where we fix the last variable to one:

$$p(x_1, \ldots, x_d) = p_{\text{hom}}(x_1, \ldots, x_d, 1).$$

3.1 Homogeneous polynomials and $H^{\otimes k}$

In fact, there is a close connection between homogeneous polynomials of degree k and tensor product spaces of order k. Let $H = \mathbb{C}^d$ be a d-dimensional complex vector space endowed with the standard inner product. Fix $\boldsymbol{t} \in H^{\otimes k}$ and define

$$p_{\boldsymbol{t}}(\boldsymbol{x}) = \langle \boldsymbol{t},\ \boldsymbol{x} \otimes \cdots \otimes \boldsymbol{x} \rangle \quad \text{for all} \quad \boldsymbol{x} \in H.$$

Here, $\langle \cdot, \cdot \rangle$ denotes the extended inner product on $H^{\otimes k}$. Recall from Lecture 2, that we can expand $\boldsymbol{t}$ as a linear combination of elementary tensor products:

$$\boldsymbol{t} = \sum_{i=1}^{r} \alpha_i\, \boldsymbol{b}_i^{(1)} \otimes \cdots \otimes \boldsymbol{b}_i^{(k)}, \qquad \boldsymbol{b}_i^{(1)}, \ldots, \boldsymbol{b}_i^{(k)} \in H.$$

In turn, $p(\boldsymbol{x})$ becomes

$$\begin{aligned} p_{\boldsymbol{t}}(\boldsymbol{x}) &= \sum_{i=1}^{r} \alpha_i\, \langle \boldsymbol{b}_i^{(1)} \otimes \cdots \otimes \boldsymbol{b}_i^{(k)}, \boldsymbol{x} \otimes \cdots \otimes \boldsymbol{x} \rangle \\ &= \sum_{i=1}^{r} \alpha_i\, \langle \boldsymbol{b}_i^{(1)}, \boldsymbol{x} \rangle \cdots \langle \boldsymbol{b}_i^{(k)}, \boldsymbol{x} \rangle = \sum_{i=1}^{r} \alpha_i \prod_{j=1}^{k} \langle \boldsymbol{b}_i^{(j)}, \boldsymbol{x} \rangle. \end{aligned}$$

It is easy to check that this is a homogeneous polynomial of degree k in d complex variables $\boldsymbol{x} = (x_1, \ldots, x_d)$.

A natural question is to ask whether this correspondence (between the tensor space of order k and the homogeneous polynomials of degree k) is one-to-one? The answer is *no* in general. The tensor product $\boldsymbol{x} \otimes \cdots \otimes \boldsymbol{x}$ is invariant under permuting tensor factors. More generally, let

$$\boldsymbol{P}_{\bigvee^k} = \frac{1}{k!} \sum_{\pi \in \mathcal{S}_k} \boldsymbol{W}_\pi$$

denote the projector onto the totally symmetric subspace $\bigvee^k \subset H^{\otimes k}$. Fix $\boldsymbol{t} \in H^{\otimes k}$. Then,

$$\begin{aligned} p_{\boldsymbol{t}}(\boldsymbol{x}) =& \langle \boldsymbol{t},\ \boldsymbol{x} \otimes \cdots \otimes \boldsymbol{x} \rangle = \langle \boldsymbol{t},\ \boldsymbol{P}_{\bigvee^k}\, \boldsymbol{x} \otimes \cdots \otimes \boldsymbol{x} \rangle = \langle \boldsymbol{P}_{\bigvee^k}\, \boldsymbol{t},\ \boldsymbol{x} \otimes \cdots \otimes \boldsymbol{x} \rangle \\ =& p_{\boldsymbol{P}_{\bigvee^k} \boldsymbol{t}}(\boldsymbol{x}). \end{aligned}$$

Here, we have used the fact that $\mathbf{P}_{\bigvee^k}$ is the projection (hence, unitary) onto the symmetric tensor space, and $\boldsymbol{x} \otimes \cdots \otimes \boldsymbol{x}$ belongs to this subspace. This result shows that both $\boldsymbol{t}$ and $\boldsymbol{P}_{\bigvee^k}\, \boldsymbol{t}$ correspond to the same polynomial. This implies that the correspondence between $H^{\otimes k}$ and $\text{Hom}^{(k)}(\boldsymbol{x})$ is in general not one-to-one. It does, however, become one-to-one if we restrict our attention to the totally symmetric subspace. The following proposition presents this result formally:

Proposition 3.2. *There is a one-to-one correspondence between* $\mathrm{Hom}^{(k)}(\boldsymbol{x})$ *and* $\bigvee^k \in H^{\otimes k}$.

Proof sketch. We note that

$$\dim\Big(\mathrm{Hom}^{(k)}(\boldsymbol{x})\Big) = \binom{d+k-1}{k} = \dim\Big(\bigvee^k\Big), \tag{3}$$

and there is a one-to-one correspondence between the extended standard basis vectors and monomials:

$$p_{\boldsymbol{e}_{i_1}\otimes\cdots\otimes\boldsymbol{e}_{i_k}}(\boldsymbol{x}) = \langle \boldsymbol{e}_{i_1}, \boldsymbol{x}\rangle \cdots \langle \boldsymbol{e}_{i_k}, \boldsymbol{x}\rangle = x_{i_1}\cdots x_{i_k}.$$

Monomials generate $\mathrm{Hom}^{(k)}$, while the extended standard basis spans $\bigvee^{(k)}$. Both dimensions match up which can be used to establish a one-to-one correspondence formally. □

3.2 Doubly homogeneous polynomials and $\mathcal{L}(H^{\otimes k})$

Definition 3.3. Define $\mathrm{Hom}^{(k)}(\boldsymbol{x}, \bar{\boldsymbol{x}})$ to be the space of *doubly homogeneous polynomials* over $\mathbb{C}$, i.e. polynomials that are are k-homogeneous in $\boldsymbol{x}$ and k-homogeneous in $\bar{\boldsymbol{x}}$:

$$\mathrm{Hom}^{(k)}(\boldsymbol{x}, \bar{\boldsymbol{x}}) = \mathrm{Hom}^{(k)}(\boldsymbol{x})\ \mathrm{Hom}^{(k)}(\bar{\boldsymbol{x}}). \tag{4}$$

A typical example for a doubly homogeneous polynomial is the standard Euclidean norm $\langle \boldsymbol{x}, \boldsymbol{x}\rangle$ and its integer powers.

Theorem 3.4. *There is a one-to-one correspondence between* $\mathrm{Hom}^{(k)}(\boldsymbol{x}, \bar{\boldsymbol{x}})$ *and* $\mathcal{L}\Big(\bigvee^k\Big)$ – *the space of linear operators from* $\bigvee^k$ *to itself.*

Proof Sketch. Let $p, q \in \mathrm{Hom}^{(k)}(H)$. Then, $q(\bar{\boldsymbol{x}})\, p(\boldsymbol{x}) \in \mathrm{Hom}^{(k)}(\boldsymbol{x}, \bar{\boldsymbol{x}})$. So,

$$\begin{aligned} q(\bar{\boldsymbol{x}})\, p(\boldsymbol{x}) =& \langle \boldsymbol{x}\otimes\cdots\otimes\boldsymbol{x}\,,\, \boldsymbol{t}_q\rangle\ \langle \boldsymbol{t}_p^*,\ \boldsymbol{x}\otimes\cdots\otimes\boldsymbol{x}\,\rangle & (5)\\ =& \langle \boldsymbol{x}\otimes\cdots\otimes\boldsymbol{x}\ \boldsymbol{t}_q\ \boldsymbol{t}_p^*\ \boldsymbol{x}\otimes\cdots\otimes\boldsymbol{x}\rangle & (6)\end{aligned}$$

for some $\boldsymbol{t}_q, \boldsymbol{t}_p \in \bigvee^k$. Rank-one operators of the form $\boldsymbol{t}_q\, \boldsymbol{t}_p^*$ span the space of all linear operators. The dimension of this space is $\dim(\bigvee^k)^2 = \binom{d+k-1}{k}^2$ which coincides with the space of doubly homogeneous polynomials ($\binom{d+k-1}{k}$ degrees of freedom for homogeneous polynomials in $\bar{\boldsymbol{x}}$ and $\boldsymbol{x}$ each). □

Corollary 3.5. *Fix* $\boldsymbol{A} \in \mathcal{L}(\bigvee^k)$ *(think of it as* $\boldsymbol{P}_{\vee^k}\boldsymbol{A}\boldsymbol{P}_{\vee^k}$ *with* $\boldsymbol{A} \in \mathcal{L}(H^{\otimes k})$*). Then,*

$$p_{\boldsymbol{A}}(\boldsymbol{x}, \bar{\boldsymbol{x}}) = \langle \boldsymbol{x}\otimes\cdots\otimes\boldsymbol{x}\,,\, \boldsymbol{A}\,(\boldsymbol{x}\otimes\cdots\otimes\boldsymbol{x})\rangle = \mathrm{tr}\Big(\boldsymbol{A}\,(\boldsymbol{x}\boldsymbol{x}^*)^{\otimes k}\Big), \tag{7}$$

is doubly-homogeneous of degree k*. Moreover, every polynomial in* $\mathrm{Hom}^{(k)}(\boldsymbol{x}, \bar{\boldsymbol{x}})$ *has this form.*

4 Haar integration

4.1 Motivation

Haar-integration provides closed form expressions for integrating doubly homogeneous polynomials over complex unit spheres. It may be viewed as a complex generalization of Gaussian integration. At the heart is the correspondence between $\mathrm{Hom}^{(k)}(\boldsymbol{x}, \bar{\boldsymbol{x}})$ and $\mathcal{L}(H^{\otimes k})$, see Corollary 3.5. For now, let $\mathrm{d}\mu(\boldsymbol{x})$ be an arbitrary integration mesure. Then,

$$\int p_{\boldsymbol{A}}(\boldsymbol{v}, \bar{\boldsymbol{v}})\, \mathrm{d}\mu(\boldsymbol{v}) = \int \mathrm{tr}\Big(\boldsymbol{A}(\boldsymbol{v}\boldsymbol{v}^*)^{\otimes k}\Big)\, \mathrm{d}\mu(\boldsymbol{v}) = \mathrm{tr}\bigg(\boldsymbol{A} \int (\boldsymbol{v}\boldsymbol{v}^*)^{\otimes k}\, \mathrm{d}\mu(\boldsymbol{v})\bigg).$$

This reformulation has deep implications. A closed-form expression for

$$\boldsymbol{H}^{(k)} = \int (\boldsymbol{v}\boldsymbol{v}^*)^{\otimes k}\, \mathrm{d}\mu(\boldsymbol{v}) \in \mathcal{L}(H^{\otimes k}) \tag{8}$$

would allow us to compute integrals of arbitrary polynomials by contracting tensor product operators:

$$\int p_{\boldsymbol{A}}(\boldsymbol{v}, \bar{\boldsymbol{v}})\, \mathrm{d}\mu(\boldsymbol{v}) = \mathrm{tr}\Big(\boldsymbol{A}\, \boldsymbol{H}^{(k)}\Big).$$

Haar-integration achieves precisely this goal for the *normalized, unitarily invariant measure* $\mathrm{d}\mu(\boldsymbol{v})$ on the complex unit sphere $\mathbb{S}^{d-1} \subset H$. The exceedingly high degree of symmetry allows for deriving an analytic expression for $\boldsymbol{H}^{(k)}$.

Before doing so, a few comments are in order. μ stands for a measure on $\mathbb{S}^{d-1}$ that inherits nice properties from $\mathbb{C}^d$. Normalization means that $\mu\Big(\mathbb{S}^{d-1}\Big) = 1$. Finally, and most importantly for our goals, *unitary invariance* means that the measure is invariant under any unitary transformation $\boldsymbol{U}$:

$$\mu(\boldsymbol{U}\mathcal{A}) = \mu(\mathcal{A}) \quad \text{for every Borel set} \quad \mathcal{A} \subseteq \mathbb{S}^{d-1}.$$

One can show that there is only one measure on $\mathbb{S}^{d-1}$ with these desirable properties. This measure is called the *Haar measure*.

Informally, the Haar measure assigns an infinitesimally small weight to each point $\boldsymbol{x} \in \mathbb{S}^{d-1}$. This assignment is fair in the sense that no vector is weighted less (or more) than any other vector.

4.2 Reformulation of the integration formula

The unitary operators on H form a nice group $\mathsf{U}(d)$. This group is unimodular and carries the structure of a Lie group. Importantly, one can also endow $\mathsf{U}(d)$ with a normalized, unitarily invariant measure $\mathrm{d}\boldsymbol{U}$. In fact, this *Haar measure* on $\mathsf{U}(d)$ induces the unitarily invariant measure on $\mathbb{S}^{d-1}$. Indeed, we can think of the sphere as the set of all possible rotations of a fixed starting vector $\boldsymbol{v}_0 \in \mathbb{S}^{d-1}$, e.g. the "north pole." The precise choice of starting point is irrelevant, because both measures are unitarily

invariant. We can use this reasoning to rewrite $\boldsymbol{H}^{(k)}$ in the following fashion:

$$\begin{aligned}\boldsymbol{H}^{(k)} &= \int_{\mathbb{S}^{d-1}} (\boldsymbol{v}\boldsymbol{v}^*)^{\otimes k} \,\mathrm{d}\mu(\boldsymbol{v}) = \int_{\mathsf{U}(d)} (\boldsymbol{U}\boldsymbol{v}_0\boldsymbol{v}_0^*\boldsymbol{U}^*)^{\otimes k} \,\mathrm{d}\mu(\boldsymbol{U}) \\ &= \int_{\mathsf{U}(d)} \boldsymbol{U}^{\otimes k}(\boldsymbol{v}_0\boldsymbol{v}_0^*)^{\otimes k}(\boldsymbol{U}^*)^{\otimes k} \,\mathrm{d}\mu(\boldsymbol{U}).\end{aligned}$$

This reformulation highlights an interesting property of $\boldsymbol{H}^{(k)}$.

Lemma 4.1. *The operator $\boldsymbol{H}^{\otimes k} \in \mathcal{L}(H^{\otimes k})$ defined in Eq. (8) commutes with any synchronized change of basis in H:*

$$\boldsymbol{H}^{(k)}\boldsymbol{V}^{\otimes k} = \boldsymbol{V}^{\otimes k}\boldsymbol{H}^{(k)} \quad \textit{for all} \quad \boldsymbol{V} \in \mathsf{U}(d).$$

Proof. Fix $\boldsymbol{V} \in \mathsf{U}(d)$. Unitary invariance of the Haar measure implies $\mathrm{d}(\boldsymbol{U}) = \mathrm{d}(\boldsymbol{V}\boldsymbol{U})$. This allows us to perform a simple change of integration variables $\boldsymbol{U}' = \boldsymbol{U} \mapsto \boldsymbol{V}\boldsymbol{U}$ that ensures

$$\begin{aligned}\boldsymbol{V}^{\otimes k}\boldsymbol{H}^{(k)} &= \int_{\mathsf{U}(d)} (\boldsymbol{V}\,\boldsymbol{U})^{\otimes k}(\boldsymbol{v}_0\boldsymbol{v}_0^*)^{\otimes k}(\boldsymbol{U}^*)^{\otimes k} \,\mathrm{d}(\boldsymbol{U}) \\ &= \int_{\mathsf{U}(d)} (\boldsymbol{U}')^{\otimes k}(\boldsymbol{v}_0\boldsymbol{v}_0^*)^{\otimes k}((\boldsymbol{V}^*\,\boldsymbol{U}')^*)^{\otimes k} \,\mathrm{d}(\boldsymbol{U}') \\ &= \int_{U(d)} (\boldsymbol{U}')^{\otimes k}(\boldsymbol{v}_0\boldsymbol{v}_0^*)^{\otimes k}((\boldsymbol{U}')^*)^{\otimes k} \,\mathrm{d}\mu(\boldsymbol{U}')\boldsymbol{V}^{\otimes k} = \boldsymbol{H}^{(k)}\boldsymbol{V}^{\otimes k}.\end{aligned}$$

□

4.3 Haar integration formula for degree $k = 1$

Theorem 4.2. *Set $H = \mathbb{C}^d$ and let $\mathrm{d}\mu(\boldsymbol{v})$ and $\mathrm{d}\mu(\boldsymbol{U})$ denote the Haar measures on $\mathbb{S}^{d-1}$ and $\mathsf{U}(d)$, respectively. Then,*

$$\boldsymbol{H}^{(1)} = \int_{\mathsf{U}(d)} \boldsymbol{U}\boldsymbol{v}_0\boldsymbol{v}_0^*\boldsymbol{U}^* \,\mathrm{d}\mu(\boldsymbol{U}) = \frac{1}{d}\,\mathbf{I}.$$

Proof. Lemma 4.1 implies that $\boldsymbol{H}^{(1)}$ must obey $\boldsymbol{U}\boldsymbol{H}^{(1)}\boldsymbol{U}^* = \boldsymbol{H}^{(1)}$ for any $\boldsymbol{U} \in \mathsf{U}(d)$. In other words: $\boldsymbol{H}^{(1)}$ must have the same matrix representation for *any* choice of basis. There is only one operator with this property – the identity $\mathbf{I} \in \mathcal{L}(H)$. The pre-factor d^{-1} follows from taking the trace:

$$\mathrm{tr}(\boldsymbol{H}^{(1)}) = \int_{\mathsf{U}(d)} \mathrm{tr}(\boldsymbol{U}\boldsymbol{v}_0\boldsymbol{v}_0^*\boldsymbol{U}^*)\mathrm{d}\mu(\boldsymbol{U}) = \langle \boldsymbol{v}_0, \boldsymbol{v}_0\rangle \int_{\mathsf{U}(d)} \mathrm{d}\mu(\boldsymbol{U}) = 1.$$

□

This closed-form expression already allows us to compute integrals of doubly homogeneous polynomials of degree one:

$$\int_{\mathbb{S}^{d-1}} p_{\boldsymbol{A}}(\boldsymbol{v}, \bar{\boldsymbol{v}}) \,\mathrm{d}\mu(\boldsymbol{v}) = \mathrm{tr}\Big(\boldsymbol{A}\boldsymbol{H}^{(1)}\Big) = \frac{\mathrm{tr}(\boldsymbol{A})}{d}.$$

The two example polynomials from the beginning of this lecture fall into this category:

$$p_1(\boldsymbol{x}, \bar{\boldsymbol{x}}) = p_{\boldsymbol{A}_1}(\boldsymbol{x}, \bar{\boldsymbol{x}}) = \operatorname{tr}(\boldsymbol{A}_1\, \boldsymbol{x}\, \boldsymbol{x}^*) \quad \text{with} \quad \boldsymbol{A}_1 = \mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{9}$$

$$p_2(\boldsymbol{x}, \bar{\boldsymbol{x}}) = p_{\boldsymbol{A}_2}(\boldsymbol{x}, \bar{\boldsymbol{x}}) = \operatorname{tr}(\boldsymbol{A}_2\, \boldsymbol{x}\, \boldsymbol{x}^*) \quad \text{with} \quad \boldsymbol{A}_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{10}$$

Theorem 4.2 now allows us to quickly compute the associated integrals. Set $d = 2$ and compute

$$\int_{\mathbb{S}^1} p_1(\boldsymbol{v}, \bar{\boldsymbol{v}})\, \mathrm{d}\mu(\boldsymbol{v}) = \frac{\operatorname{tr}(\boldsymbol{A}_1)}{2} = 1,$$
$$\int_{\mathbb{S}^1} p_2(\boldsymbol{v}, \bar{\boldsymbol{v}})\, \mathrm{d}\mu(\boldsymbol{v}) = \frac{\operatorname{tr}(\boldsymbol{A}_2)}{2} = 0.$$

4.4 Haar integration formula for arbitrary degree

The approach from the previous subsection can be extended to establish closed form expressions for $\boldsymbol{H}^{(k)}$ with $k \geq 2$.

Theorem 4.3 (Haar integration formula)**.** *Set $H = \mathbb{C}^d$ and let $\mathrm{d}\mu(\boldsymbol{v})$ and $\mathrm{d}\mu(\boldsymbol{U})$ denote the Haar measures on $\mathbb{S}^{d-1} \subset H$ and $\mathsf{U}(d)$, respectively. Then, for any $k \geq 2$*

$$\boldsymbol{H}^{(k)} = \int_{\mathbb{S}^{d-1}} (\boldsymbol{v}\boldsymbol{v}^*)^{\otimes k}\, \mathrm{d}\mu(\boldsymbol{v}) = \int_{\mathsf{U}(d)} (\boldsymbol{U}\boldsymbol{v}_0\boldsymbol{v}_0^*\boldsymbol{U}^*)^{\otimes k}\, \mathrm{d}\mu(\boldsymbol{U}) = \binom{d+k-1}{k}^{-1} \boldsymbol{P}_{\vee^k}.$$

Here, $\boldsymbol{P}_{\vee^k} = \frac{1}{k!} \sum_{\pi \in \mathcal{S}_k} \boldsymbol{W}_\pi \in \mathcal{L}(H^{\otimes k})$ denotes the projector onto the totally symmetric subspace and $\binom{d+k-1}{k}$ is the dimension of its range $\bigvee^k \subset H^{\otimes k}$.

The proof of this statement is based on deep results from algebra (representation theory). Recall that we can identify permutations $\pi \in \mathcal{S}_k$ with operators that permute tensor factors:

$$\boldsymbol{W}_\pi : \boldsymbol{x}_1 \otimes \cdots \otimes \boldsymbol{x}_k \mapsto \boldsymbol{x}_{\pi^{-1}(1)} \otimes \cdots \otimes \boldsymbol{x}_{\pi^{-1}(k)}$$

and linearly extended to all of $H^{\otimes k}$. These operators are unitary and respect the group composition rule (see HW I):

$$\boldsymbol{W}^*\pi = \boldsymbol{W}_{\pi^{-1}} = \boldsymbol{W}_\pi^{-1} \quad \text{and} \quad \boldsymbol{W}_\pi \boldsymbol{W}_\tau = \boldsymbol{W}_{\pi \circ \tau} \quad \text{for all} \quad \pi, \tau \in \mathcal{S}_k.$$

This means that the $\boldsymbol{W}_\pi$ form a *unitary representation* of the symmetric group $\mathcal{S}_k$ – a nice, finite group – on the "representation space" $H^{\otimes k}$.

In a similar fashion, the map $\boldsymbol{U} \mapsto \boldsymbol{U}^{\otimes k}$ forms a unitary representation of $\mathsf{U}(d)$ – a nice Lie group – on $H^{\otimes k}$. Crucially, Lemma 4.1 ensures that these two representations always commute:

$$\left[\boldsymbol{U}^{\otimes k}, \boldsymbol{W}_\pi\right] = \boldsymbol{U}^{\otimes k}\boldsymbol{W}_\pi - \boldsymbol{W}_\pi \boldsymbol{U}^{\otimes k} = \mathbf{0} \quad \text{for all} \quad \pi \in \mathcal{S}_k,\ \boldsymbol{U} \in \mathsf{U}(d).$$

This commutation relation has profound consequences. A deep result from algebra (representation theory) states that every matrix that commutes with all $\boldsymbol{U}^{\otimes k}$ must be a linear combination of permutation operators, and vice versa. More precisely: the permutation operators generate the *commutant* of $\left\{\boldsymbol{U}^{\otimes k} : \boldsymbol{U} \in \mathsf{U}(d)\right\}$ and vice versa. We refer to Matthias Christandl's PhD thesis for a short, detailed and insightful analysis of these properties and borrow the following result:

Theorem 4.4. *Let $\boldsymbol{X} \in \mathcal{L}(H^{\otimes k})$ be an operator that commutes with any unitary of the form $\boldsymbol{U}^{\otimes k}$. Then, $\boldsymbol{X}$ must be a linear combination of permutation operators:*

$$\boldsymbol{X} = \sum_{\pi \in \mathcal{S}_k} c_\pi \boldsymbol{W}_\pi.$$

A proof of this claim would go beyond the scope of this lecture. It follows from the double commutant theorem and exploiting both the finite group structure of $\mathcal{S}_k$ and the nice Lie group structure of $\mathsf{U}(d)$. This double-commutant theorem allows us to readily deduce the Haar integration formula.

Proof of Theorem 4.3. Lemma 4.1 implies that $\boldsymbol{H}^{(k)}$ must commute with every tensor product unitary $\boldsymbol{U}^{\otimes k}$. Theorem 4.4 then ensures that this operator must be a linear combination of permutations:

$$\boldsymbol{X} = \sum_{\pi \in \mathcal{S}_k} c_\pi \boldsymbol{W}_\pi. \tag{11}$$

Next, note that there is additional symmetry present: $\boldsymbol{H}^{(k)}$ is also invariant under any permutations. Fix $\pi \in \mathcal{S}_k$ and observe,

$$\begin{aligned}\boldsymbol{W}_\pi \boldsymbol{H}^{(k)} &= \int_{\mathbb{S}^{d-1}} \boldsymbol{W}_\pi (\boldsymbol{v}\boldsymbol{v}^*)^{\otimes k} \,\mathrm{d}\mu(\boldsymbol{v}) = \int_{\mathbb{S}^{d-1}} \left(\boldsymbol{W}_\pi \boldsymbol{v}^{\otimes k}\right)(\boldsymbol{v}^*)^{\otimes k} \,\mathrm{d}\mu(\boldsymbol{v}) \\ &= \int_{\mathbb{S}^{d-1}} (\boldsymbol{v}\boldsymbol{v}^*)^{\otimes k} \,\mathrm{d}\mu(\boldsymbol{v}) = \boldsymbol{H}^{(k)}.\end{aligned}$$

This invariance is only possible if all the expansion coefficients in Eq. (11) are the same:

$$\boldsymbol{H}^{(1)} = \sum_{\pi \in \mathcal{S}_k} c\boldsymbol{W}_\pi = c \sum_{\pi \in \mathcal{S}_k} \boldsymbol{W}_\pi = c\, \boldsymbol{P}_{\vee^k}.$$

Finally, take the trace on both sides to specify this constant:

$$c \binom{d+k-1}{k} = \mathrm{tr}(c\, \boldsymbol{P}_{\vee^k}) = \mathrm{tr}\left(\boldsymbol{H}^{(k)}\right) = 1.$$

□

4.5 Closed form expressions for integrating homogeneous polynomials

Corollary 4.5. *Let $p_{\boldsymbol{A}}(\boldsymbol{x}, \bar{\boldsymbol{x}}) = \mathrm{tr}(\boldsymbol{A}\, \boldsymbol{x}\boldsymbol{x}^*)$ be an arbitrary polynomial in $\mathrm{Hom}^{(k)}(\boldsymbol{x}, \bar{\boldsymbol{x}})$. Let $\mathrm{d}\mu(\boldsymbol{v})$ be the Haar measure on the complex unit sphere $\mathbb{S}^{d-1} \subset \mathbb{C}^d$. Then,*

$$\int_{\mathbb{S}^{d-1}} p_{\boldsymbol{A}}(\boldsymbol{v}, \bar{\boldsymbol{v}}) \,\mathrm{d}\mu(\boldsymbol{v}) = \binom{d+k-1}{k}^{-1} \mathrm{tr}(\boldsymbol{A}\, \boldsymbol{P}_{\vee^k}).$$

Closed form expressions for such integration formulas have a variety of applications. We will discuss several of them in future lectures. For now, we content ourselves with integrating the squares of the homogeneous polynomials defined in Eq. (2):

$$p_i(\boldsymbol{x}, \bar{\boldsymbol{x}})^2 = \operatorname{tr}(\boldsymbol{A}_i\, \boldsymbol{x}\boldsymbol{x}^*)^2 = \operatorname{tr}\Big(\boldsymbol{A}_i^{\otimes 2}\, (\boldsymbol{x}\boldsymbol{x}^*)^{\otimes 2}\Big).$$

Set $d = 2$ and apply Corollary 4.5 to conclude

$$\int_{\mathbb{S}^1} p_i(\boldsymbol{v}, \bar{\boldsymbol{v}})\, \mathrm{d}\mu(\boldsymbol{v}) = \operatorname{tr}\Big(\boldsymbol{A}_i^{\otimes 2}\, \boldsymbol{H}^{(2)}\Big) = \frac{1}{3} \operatorname{tr}\Big(\boldsymbol{A}_i^{\otimes 2}\, \boldsymbol{P}_{\vee^2}\Big),$$

because $\binom{d+1}{2} = \frac{3}{2} = 3$. Next, use $\boldsymbol{P}_{\vee^2} = \frac{1}{2}(\mathbf{I}^{\otimes 2} + \boldsymbol{F})$, where $\boldsymbol{F} : \boldsymbol{x} \otimes \boldsymbol{y} \mapsto \boldsymbol{y} \otimes \boldsymbol{x}$ denotes the flip operator. This ensures,

$$\operatorname{tr}\Big(\boldsymbol{A}_i^{\otimes 2}\, \boldsymbol{P}_{\vee^2}\Big) = \frac{1}{2}\Big(\operatorname{tr}\Big(\boldsymbol{A}_i \mathbf{I}^{\otimes 2}\Big) + \operatorname{tr}\Big(\boldsymbol{A}_i^{\otimes 2} \boldsymbol{F}\Big)\Big) = \frac{1}{2}\Big(\operatorname{tr}(\boldsymbol{A}_i)^2 + \operatorname{tr}\Big(\boldsymbol{A}_i^2\Big)\Big),$$

and we can insert the operator expressions from Equations (9) and (10) to get concrete numbers:

$$\int_{\mathbb{S}^1} p_1(\boldsymbol{v}, \bar{\boldsymbol{v}})^2\, \mathrm{d}\mu(\boldsymbol{v}) = \frac{1}{6}\Big(\operatorname{tr}(\boldsymbol{A}_1)^2 + \operatorname{tr}\Big(\boldsymbol{A}_1^2\Big)\Big) = \frac{1}{6}\Big(2^2 + 2\Big) = 1,$$
$$\int_{\mathbb{S}^1} p_2(\boldsymbol{v}, \bar{\boldsymbol{v}})^2\, \mathrm{d}\mu(\boldsymbol{v}) = \frac{1}{6}\Big(\operatorname{tr}(\boldsymbol{A}_2)^2 + \operatorname{tr}\Big(\boldsymbol{A}_2^2\Big)\Big) = \frac{1}{6}(0 + 2) = \frac{1}{3}.$$

Lecture 06: Entanglement is ubiquitous

Scribe: Jiajie Chen

ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp
April 17, 2019

1 Agenda

1. Pure states and entanglement
2. Almost all pure states are entangled
3. Proof:
   (a) Concentration
   (b) Discretization
   (c) Union bound

2 Pure quantum states and entanglement

2.1 Pure states

Fix $H = \mathbb{C}^d$ and endow it with the standard inner product. Recall that quantum mechanical systems are described by *density matrices*:

$$\boldsymbol{\rho} \in \mathsf{S}(H) = \{\boldsymbol{X} \in \mathcal{L}(H) : \ \boldsymbol{X}^* = \boldsymbol{X}, \ \boldsymbol{X} \succeq \mathbf{0}, \ (\mathbf{I}, \boldsymbol{X}) = 1\}.$$

Density matrices are the SDP-generalization of d-variate probability vectors:

$$\boldsymbol{p} \in \Delta_{d-1} = \left\{\boldsymbol{x} \in \mathbb{R}^d : \ \boldsymbol{x} \geq \mathbf{0}, \ \langle \mathbf{1}, \boldsymbol{x} \rangle = 1\right\}.$$

Pure probability vectors correspond to extreme points of this convex set: $\boldsymbol{p} = \boldsymbol{e}_k$ for $1 \leq k \leq d$. These represent deterministic distributions. Every probability vector corresponds to a convex mixture of these extreme distributions.

The natural quantum analogue is captured by the following definition.

Definition 2.1 (pure state)**.** A density matrix $\boldsymbol{\rho} \in \mathcal{S}(H)$ is *pure* if and only if $\operatorname{rank}(\boldsymbol{\rho}) = 1$, or equivalently: $\boldsymbol{\rho} = \boldsymbol{u}\boldsymbol{u}^*$ for $\boldsymbol{u} \in H$ obeying $\langle \boldsymbol{u}, \boldsymbol{u} \rangle = 1$. We call such density matrices *pure states*.

Let $\mathbb{S}(H)$ denote the complex unit sphere in $H = \mathbb{C}^d$:

$$\mathbb{S}(H) = \{\boldsymbol{x} \in H : \ \langle \boldsymbol{x}, \boldsymbol{x} \rangle = 1\} \subset H.$$

The following properties assert that pure states really mimic pure (deterministic) probability vectors:

- Pure states form the boundary of the convex set $\mathcal{S}(H)$.

- Every density matrix corresponds to a convex mixture of pure states:

$$\mathsf{S}(H) = \mathrm{conv}\{\boldsymbol{u}\boldsymbol{u}^* : \boldsymbol{u} \in \mathbb{S}(H)\}$$

- Every density matrix corresponds to the marginalization of a larger pure state:

$$\boldsymbol{\rho} = \mathrm{tr}_2(\boldsymbol{u}\boldsymbol{u}^*) \quad \text{for some} \quad \boldsymbol{u} \in \mathbb{S}\Big(H^{\otimes 2}\Big).$$

Proofs of these claims are part of Exercise I. In summary: Pure states are the "most extreme" density matrices. Most quantum phenomena+tricks assume their "purest" form for pure states. Their extension to general density matrices is then achieved by convex mixtures. These tend to obfuscate the original properties. An extreme example is the *maximally mixed state:*

$$\boldsymbol{\rho}_0 = \int_{\langle \boldsymbol{u},\boldsymbol{u}\rangle=1} \boldsymbol{u}\boldsymbol{u}^* \mathrm{d}\mu(\boldsymbol{u}) = \frac{1}{d}\mathbf{I}.$$

This is the "most useless" state conceivable. The outcome of any quantum measurement $\{\boldsymbol{H}_a : a \in A\}$ is maximally random:

$$\Pr[a|\boldsymbol{\rho}_0] = (\boldsymbol{H}_a, \boldsymbol{\rho}_0) = \frac{\mathrm{tr}(\boldsymbol{H}_a)}{d} \quad \text{for all} \quad a \in A.$$

Such outcome probabilities can typically be "simulated" by tossing conventional coins.

2.2 Entanglement for pure states

Suppose that a quantum mechanical system is comprised of two smaller system with dimensions d_1 and d_2, respectively. Set $H_1 = \mathbb{C}^{d_1}$ and $H_2 = \mathbb{C}^{d_2}$. Then, the joint quantum state is an element of

$$\boldsymbol{\rho} \in \mathsf{S}(H_1 \otimes H_2) \subset \mathcal{L}(H_1) \otimes \mathcal{L}(H_2) \simeq \mathcal{L}(H_1 \otimes H_2).$$

Pure states assume the following form:

$$\boldsymbol{\rho}_{\mathrm{joint}} = \boldsymbol{u}\boldsymbol{u}^* \quad \text{for} \quad \boldsymbol{u} \in \mathbb{S}(H_1 \otimes H_2) \simeq \mathbb{S}(H),$$

where $H = H_1 \otimes H_2 \simeq \mathbb{C}^D$ with $D = d_1 d_2$.

Recall that there are three possibilities for joint quantum states:

1. *Product states:* $\boldsymbol{\rho}_{\mathrm{joint}} = \boldsymbol{\sigma_1} \otimes \boldsymbol{\sigma}_2$ with $\boldsymbol{\sigma}_1 \in \mathsf{S}(H_1)$ and $\boldsymbol{\sigma_2} \in \mathsf{S}(H_2)$. These behave like independent distributions.
2. *Separable states:* $\boldsymbol{\rho}_{\mathrm{joint}} \in \mathsf{S}(H)1)\hat{\otimes}\mathsf{S}(H_2) = \mathrm{conv}(\boldsymbol{\sigma}_1 \otimes \boldsymbol{\sigma}_2 : \boldsymbol{\sigma}_1 \in \mathsf{S}(H_1),\ \boldsymbol{\sigma}_2 \in \mathsf{S}(H_2)\}$. These correspond to convex mixtures of product states. In classical probability theory, these convex mixtures reach "everything".
3. *Entangled states:* everything that is not separable.

Lemma 2.2. *A joint pure state $\boldsymbol{\rho}_{\mathrm{joint}} = \boldsymbol{u}\boldsymbol{u}^*$ is separable if and only if it is a tensor product of pure states: $\boldsymbol{\rho}_{\mathrm{joint}} = \boldsymbol{a}\boldsymbol{a}^* \otimes \boldsymbol{b}\boldsymbol{b}^*$, $\boldsymbol{a} \in \mathbb{S}(H_1)$, $\boldsymbol{b} \in \mathbb{S}(H_2)$.*

Proof. It is clear that tensor products of pure states are pure product states: $\boldsymbol{a}\boldsymbol{a}^* \otimes \boldsymbol{b}\boldsymbol{b}^* \simeq (\boldsymbol{a} \otimes \boldsymbol{b})(\boldsymbol{a} \otimes \boldsymbol{b})^*$. These are the only pure product states – the rank constraint is very stringent. Indeed, convex mixtures of pure product states necessarily increase the rank – density matrices are psd. So, the intersection of the set of all separable states with the (non-convex) set of all joint pure states returns the (non-convex) set of all pure product states. □

3 Almost all pure states are entangled

A well-known result in quantum states that "almost all states are maximally entangled". The typical argument is as follows: Remember our candidate for a entangled state from Lecture 3 ($H_1 \simeq H_2$):

$$\boldsymbol{\Omega} = \frac{1}{D}\mathrm{vec}(\mathbf{I})\mathrm{vec}(\mathbf{I})^* = \frac{1}{D}\supset\!\subset,$$

This state is entangled and has the following interesting property:

$$\mathrm{tr}_1(\boldsymbol{\Omega}) = \mathrm{tr}_2(\boldsymbol{\Omega}) = \frac{1}{d}\mathbf{I} = \boldsymbol{\rho}_0.$$

Although the joint state is pure, marginalizations produce the maximally mixed state ("garbage"). This is indicative of a very strong correlation in the joint system. Pure states with this property are called *maximally entangled*. Now, suppose that we choose $\boldsymbol{u} \in \mathbb{S}(H_1 \otimes H_2)$ uniformly from the Haar measure. Then,

$$\|\mathrm{tr}_2(\boldsymbol{u}\boldsymbol{u}^*) - \frac{1}{d_1}\mathbf{I}\|_1$$

is small with exceedingly high probability. Proving this will be part of Exercise II. Since the Haar-measure is fair in the sense that it assigns the same infinitesimal weight to any pure state, we can conclude the following quantitative statement: *the marginalization of almost every pure state results is very close to the maximally mixed state.*

Remark 3.1. The trace distance is a natural metric for quantifying deviations among density matrices. Helstrom's theorem (Lecture I) assigns an operational meaning to this quantity: it is proportional to the optimal bias achievable when trying to distinguish the two states in question with a single measurement.

Today, I want to derive a different statement that points in a similar direction. Almost every joint pure state is very far away from *any* product state.

Theorem 3.2. *Set $H = H_1 \otimes H_2$ ($\dim(H_1) = d_1$, $\dim(H_2) = d_2$, $\dim(H) = d_1 d_2$). Choose $\boldsymbol{u} \in \mathsf{S}(H)$ uniformly from the complex unit sphere (Haar random). Then,*

$$\Pr\left[\inf_{\boldsymbol{a}\in\mathbb{S}(H_1),\boldsymbol{b}\in\mathbb{S}(H_2)} \|\boldsymbol{u}\boldsymbol{u}^* - \boldsymbol{a}\boldsymbol{a}^* \otimes \boldsymbol{b}\boldsymbol{b}^*\|_1 \leq 1\right] \leq 2\exp\left(4.5(d_1+d_2) - \frac{7d_1d_2}{32}\right)$$

We will use this statement to introduce a very powerful proof technique by example. It is based on three fundamental steps:

- Concentration of individual problem instances (Haar-randomness)
- Discretization: Find covering nets for $\mathcal{S}(H_1)$ and $\mathcal{S}(H_2)$ and combine them to get a covrering net for all pure product sates in $\mathsf{S}(H)$. Finite covering nets allow us to move from controlling an infimum to controlling a minimum.
- Apply a union bound to the (discretized) minimum and use Haar-concentration of each instance to counter-balance the number of different instances.

Although not optimal (in terms of constants) and perhaps cumbersome, it is *very* versatile and can be applied to a variety of problems. Today, we employ it to show that almost every pure state is entangled. Next week, we will employ it to show that almost every pure state is useless for quantum computation. Other examples include:

1. (Quantum): Every quantum channel admits an accurate approximation ("sketch") that has low Kraus rank (Lancien and Winter)
2. (Quantum): Almost every quantum state (unitary channel) has high circuit "complexity" (i.e. takes a long time to generate).
3. (Classical): Randomess in the measurement design allows for recovering sparse vectors and low-rank matrices efficiently from very few measurements.
4. (Classical): control the maximum eigenvalue of a random matrix.

4 Proof of Theorem 3.2

4.1 Preliminaries

Theorem 4.1 (Markov's inequality)**.** *Let $S \in \mathbb{R}$ be a non-negative random variable. Then, for any $\alpha > 0$*

$$\Pr[S \geq \alpha] \leq \frac{\mathbb{E}[S]}{\alpha}.$$

Theorem 4.2 (Union bound, Boole's inequality)**.** *Let $E_1, \ldots, E_N$ be events. Then,*

$$\Pr\left[\bigcup_{k=1}^{N} E_i\right] \leq \sum_{k=1}^{N} \Pr[E_i].$$

In particular, for scalar random variables $S_1, \ldots, S_N \in \mathbb{R}$ we have

$$\Pr\left[\max_{1\leq k\leq N} S_k \geq \alpha\right] \leq N \max_{1\leq k\leq N} \Pr[S_k \geq \alpha].$$

Theorem 4.3 (Haar integration tensor)**.** *Let $\mu(\boldsymbol{u})$ denote the unitarily invariant Haar measure on $\mathbb{S}(H)$, $\dim(H) = D$. Then, for any $k \in \mathbb{N}$:*

$$\int_{\mathbb{S}(H)} (\boldsymbol{u}\boldsymbol{u}^*)^{\otimes k} \mathrm{d}\mu(\boldsymbol{u}) = \binom{D+k-1}{k}^{-1} \boldsymbol{P}_{\vee^k}. \tag{1}$$

Finally, we will need the concept of a covering net for the complex unit spheres $\mathbb{S}(H_1)$ and $\mathbb{S}(H_2)$. A *covering net* of fineness $\theta > 0$ is a finite set of points $\{\boldsymbol{z}_i\}_{i=1}^N \subset \mathbb{S}(H_i)$ that evenly covers the sphere: For every $\boldsymbol{v} \in \mathbb{S}(H_i)$, there is a net element $\boldsymbol{z}_j$ that is (at least) θ-close in Euclidean distance:

$$\|\boldsymbol{v} - \boldsymbol{z}_j\|_{\ell_2} \leq \theta.$$

Theorem 4.4 (Existence of covering nets)**.** *The complex unit sphere $\mathbb{S}(H)$ in $d = \dim(H)$ dimensions admits a θ-covering net of cardinality*

$$N \leq \left(1 + \frac{2}{\theta}\right)^{2d}.$$

The proof follows from embedding the complex unit sphere into a real-valued unit sphere of dimension $2d$ (isometry) and applying a volumetric counting argument: cover the big sphere with many small balls.

4.2 Step I: Concentration

Proposition 4.5. *Suppose that $D = \dim(H)$ and fix $\boldsymbol{v} \in \mathbb{S}(H)$. Choose $\boldsymbol{u} \in \mathbb{S}(H)$ uniformly from the Haar measure. Then, for any $0 < c < 2$*

$$\Pr[\|\boldsymbol{u}\boldsymbol{u}^* - \boldsymbol{v}\boldsymbol{v}^*\|_1 \leq c] \leq 2\mathrm{e}^{-D(1-c^2/4)/2}$$

This strong probabilistic concentration forms the basis of the entire argument. Identify a single instance of the larger problem. Then, apply the Haar integration formula to show that a random vector $\boldsymbol{u}$ avoids this instance with exponentially large probability.

Remark 4.6. Randomness is a misleading term when describing Haar-uniform vectors. They avoid fixed, concrete events in a highly predictable fashion. Sometimes this is constructive (here: we want to show that most states are entangled), sometimes this is destructive (future lecture: Haar-random states are useless for quantum computation).

The proof is based on two steps.

Lemma 4.7 (Reformulation)**.** *Fix $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{S}(H)$. Then,*

$$\|\boldsymbol{u}\boldsymbol{u}^* - \boldsymbol{v}\boldsymbol{v}^*\|_1 = 2\sqrt{1 - |\langle \boldsymbol{u}, \boldsymbol{v}\rangle|^2}.$$

In particular, $\|\boldsymbol{u}\boldsymbol{u}^* - \boldsymbol{v}\boldsymbol{v}^*\|_1 \leq c$ if and only if $|\langle \boldsymbol{v}, \boldsymbol{u}\rangle|^2 \geq 1 - c^2/4$

Proof. Set $\boldsymbol{X} = \boldsymbol{u}\boldsymbol{u}^* - \boldsymbol{v}\boldsymbol{v}^*$. This is a hermitian, traceless matrix with rank at most two. The trace norm collects the absolute values of the eigenvalues: $\|\boldsymbol{X}\|_1 = |\lambda_1| + |\lambda_2|$. A vanishing trace demands $\lambda_1 + \lambda_2 = \mathrm{tr}(\boldsymbol{X}) = 0$, or equivalently $\lambda_{1,2} = \pm\lambda$. Next,

$$2\lambda^2 = \lambda_1^2 + \lambda_2^2 = \mathrm{tr}(\boldsymbol{X}) = 2(1 - \mathrm{tr}(\boldsymbol{u}\boldsymbol{u}^*\, \boldsymbol{v}\boldsymbol{v}^*))$$

and we conclude $\lambda = \sqrt{1 - |\langle \boldsymbol{u}, \boldsymbol{v}\rangle|^2}$. □

Proposition 4.8. *Fix $\boldsymbol{v} \in \mathbb{S}(H)$ and choose $\boldsymbol{u} \in \mathbb{S}(H)$ according to the Haar measure $\mathrm{d}\mu(\boldsymbol{u})$. Then, for any $\tau > 0$*

$$\Pr\Big[|\langle \boldsymbol{v}, \boldsymbol{u}\rangle|^2 \geq \tau\Big] \leq 2\mathrm{e}^{-D\tau/2}.$$

Proof. Define the non-negative, scalar random variable $S_{\boldsymbol{v}}(\boldsymbol{u}) = |\langle \boldsymbol{v}, \boldsymbol{u}\rangle|^2$. The Haar integration formula (1) allows us to compute arbitrary moments: For any $k \in \mathbb{N}$

$$\begin{aligned}
\mathbb{E}\Big[S_{\boldsymbol{v}}(\boldsymbol{u})^k\Big] =& \mathbb{E}\Big[|\langle \boldsymbol{v}, \boldsymbol{u}\rangle|^{2k}\Big] = \mathbb{E}\Big[\mathrm{tr}(\boldsymbol{v}\boldsymbol{v}^*\boldsymbol{u}\boldsymbol{u}^*)^k\Big] = \mathbb{E}\Big[\mathrm{tr}\Big((\boldsymbol{v}\boldsymbol{v}^*)^{\otimes k}\,(\boldsymbol{u}\boldsymbol{u}^*)^{\otimes k}\Big)\Big] \\
=& \mathrm{tr}\left((\boldsymbol{v}\boldsymbol{v}^*)^{\otimes k}\int_{\mathbb{S}(H)}(\boldsymbol{u}\boldsymbol{u}^*)^{\otimes k}\mathrm{d}\mu(\boldsymbol{u})\right) = \binom{D+k-1}{k}^{-1}\mathrm{tr}\Big((\boldsymbol{v}\boldsymbol{v}^*)^{\otimes k}\boldsymbol{P}_{\vee^k}\Big) \\
=& \binom{D+k-1}{k}^{-1} = \frac{k!}{(D+k-1)\cdots(D+1)D} \leq \frac{k!}{D^k}.
\end{aligned}$$

This moment growth indicates sub-exponential tail behavior at a scale proportional to $1/D$. More precisely, choose $\tau > 0$ and observe

$$\Pr[S_{\boldsymbol{v}}(\boldsymbol{u}) \geq \tau] = \Pr[DS_{\boldsymbol{v}}(\boldsymbol{u})/2 \geq D\tau/2] = \Pr[\exp(DS_{\boldsymbol{v}}(\boldsymbol{u})/2) \geq \exp(D\tau/2)].$$

Next, apply Markov's inequality and expand the exponential in a Taylor series:

$$\begin{aligned}
\Pr[\exp(DS_{\boldsymbol{v}}(\boldsymbol{u})/2) \geq \exp(D\tau/2)] \leq& \mathrm{e}^{-D\tau/2}\mathbb{E}\Big[\mathrm{e}^{DS_{\boldsymbol{v}}(\boldsymbol{u})/2}\Big] \\
=& \mathrm{e}^{-D\tau/2}\sum_{k=0}^{\infty}\frac{1}{k!}\frac{D^k}{2^k}\mathbb{E}\Big[S_{\boldsymbol{v}}(\boldsymbol{u})^k\Big] \\
\leq& \mathrm{e}^{-D\tau/2}\sum_{k=0}^{\infty}\frac{1}{2^k} = 2\mathrm{e}^{-D\tau/2}.
\end{aligned}$$

□

Combining both statements readily yields Proposition 4.5.

4.3 Step II: Discretization

Let us now take into account the bi-partite structure: $\dim(H_1) = d_1$, $\dim(H_2) = d_2$ and $H = H_1 \otimes H_2$ has dimension $D = d_1 d_2$.

Let us now choose an arbitrary fixed product state $\boldsymbol{v}\boldsymbol{v}^* = \boldsymbol{a}\boldsymbol{a}^* \otimes \boldsymbol{b}\boldsymbol{b}^*$. Concentration – Proposition 4.5 – ensures that a Haar-random joint state $\boldsymbol{u}\boldsymbol{u}^*$ will be very far away from this reference state:

$$\Pr[\|\boldsymbol{u}\boldsymbol{u}^* - \boldsymbol{a}\boldsymbol{a}^* \otimes \boldsymbol{b}\boldsymbol{b}^*\|_1 \leq c] \leq 2\mathrm{e}^{-D\tilde{c}/2},$$

where $\tilde{c} = 1 - c^2/4 \in (0, 1)$. The probability of being close is exponentially supressed. What is more, the exponent features $D = d_1 d_2$ – the dimension of $H_1 \otimes H_2$.

Intuitively, the set of all possible product space should have a much smaller dimension: it is the tensor product of two unit spheres in d_1 and d_2 dimensions each. The notion

of *covrering nets* allows for quantifying this intuition. Fix $\theta > 0$ and endow $\mathbb{S}(H_1)$ and $\mathbb{S}_2(H_2)$ with two covering nets:

$$\{\boldsymbol{y}_i\}_{i=1}^{N_1} \subset \mathbb{S}(H_1) \quad \text{and} \quad \{\boldsymbol{z}_j\}_{j=1}^{N_2} \subset \mathbb{S}(H_2).$$

It should not come as a surprise that all possible net product states

$$\mathcal{N}_{\text{joint}} = \left\{\boldsymbol{y}_i\boldsymbol{y}_i^* \otimes \boldsymbol{z}_j\boldsymbol{z}_j^* : 1 \le i \le N_1,\ 1 \le j \le N_2\right\} \subset \mathbb{S}(H_1)\tilde{\otimes}\mathbb{S}(H_2) \cap \mathbb{S}(H_1 \otimes H_2).$$

provide an accurate discretization of the set of all product states.

Lemma 4.9. *Fix an arbitrary product state $\boldsymbol{a}\boldsymbol{a}^* \otimes \boldsymbol{b}\boldsymbol{b}^*$ with $\boldsymbol{a} \in \mathbb{S}(H_1)$ and $\boldsymbol{b} \in \mathbb{S}(H_2)$. Then, there is an element $\boldsymbol{y}_i\boldsymbol{y}_i^* \otimes \boldsymbol{z}_i\boldsymbol{z}_j^*$ of the joint net $\mathcal{N}_{\text{joint}}$ that obeys*

$$\|\boldsymbol{a}\boldsymbol{a}^* \otimes \boldsymbol{b}\boldsymbol{b}^* - \boldsymbol{y}\boldsymbol{y}^* \otimes \boldsymbol{z}\boldsymbol{z}^*\|_1 \le 2\theta.$$

Moreover,

$$\inf_{\text{product state } \boldsymbol{v}\boldsymbol{v}^*} \|\boldsymbol{v}\boldsymbol{v}^* - \boldsymbol{u}\boldsymbol{u}^*\|_1 \ge \min_{\boldsymbol{v}\boldsymbol{v}^* \in \mathcal{N}_\theta} \|\boldsymbol{v}\boldsymbol{v}^* - \boldsymbol{u}\boldsymbol{u}^*\|_1 - 2\theta.$$

The second statement achieves our second goal: *discretization*. The infimum over all (infinitely many) product states is well approximated by a finite infimum over states in the net. The cardinality of this net obeys

$$|\mathcal{N}_\theta| = N_1 N_2 \le \left(1 + \frac{2}{\theta}\right)^{d_1+d_2}.$$

This number scales exponentially in $d_1 + d_2$, not $D = d_1 d_2$.

Proof. The covering net assumption ensures that there exist net elements $\boldsymbol{y}$ and $\boldsymbol{z}$ such that

$$\begin{aligned}
\theta^2 \ge& \|\boldsymbol{a} - \boldsymbol{y}\|_2^2 = 2(1 - \operatorname{Re}(\langle \boldsymbol{a}, \boldsymbol{y}\rangle)) \ge 2(1 - |\langle \boldsymbol{a}, \boldsymbol{y}\rangle|),\\
\theta^2 \ge& \|\boldsymbol{b} - \boldsymbol{z}\|_2^2 = 2(1 - \operatorname{Re}(\langle \boldsymbol{b}, \boldsymbol{z}\rangle)) \ge 2(1 - |\langle \boldsymbol{a}, \boldsymbol{y}\rangle|).
\end{aligned}$$

Also, for any $x, y \in [0, 1]$

$$\sqrt{1 - x^2y^2} \le \sqrt{1 - x^2} = \sqrt{(1-x)(1+x)} \le \sqrt{2(1-x)}.$$

Therefore,

$$\begin{aligned}
\|\boldsymbol{a}\boldsymbol{a}^* \otimes \boldsymbol{b}\boldsymbol{b}^* - \boldsymbol{y}\boldsymbol{y}^* \otimes \boldsymbol{z}\boldsymbol{z}^*\|_1 =& 2\sqrt{1 - |\langle \boldsymbol{a} \otimes \boldsymbol{b}, \boldsymbol{y} \otimes \boldsymbol{z}\rangle|^2}\\
=& 2\sqrt{1 - |\langle \boldsymbol{a}, \boldsymbol{y}\rangle|^2 |\langle \boldsymbol{b}, \boldsymbol{z}\rangle|^2}\\
\le& 2\sqrt{1 - |\langle \boldsymbol{a}, \boldsymbol{y}\rangle|^2} \le 2^{3/2}\sqrt{1 - |\langle \boldsymbol{a}, \boldsymbol{y}\rangle|}\\
\le& 2\theta.
\end{aligned}$$

□

4.4 Step III: Union bound

Proposition 4.10. *Let $\mathcal{N}_\theta$ be the net of pure states. Fix $0 < c < 2$. Then,*

$$\Pr\left[\inf_{\textit{product state } \boldsymbol{vv}^*} \|\boldsymbol{vv}^* - \boldsymbol{uu}^*\|_1 \leq c\right] \leq |\mathcal{N}_\theta| \max_{\boldsymbol{vv}^* \in \mathcal{N}_\theta} \Pr[\|\boldsymbol{vv}^* - \boldsymbol{uu}^*\|_1 \leq c + 2\theta].$$

This is the final trick. We replace a infimum over infinitely many points by a minimization over finitely many points. Subsequently, we can apply the union bound to pull out the minimization. The extra cost we incur is the cardinality of the product state net:

$$|\mathcal{N}_\theta| \leq \left(1 + \frac{2}{\theta}\right)^{2(d_1+d_2)}$$

Proof. Fix $\boldsymbol{u} \in \mathbb{S}(H)$. Then

$$\inf_{\text{product state } \boldsymbol{vv}^*} \|\boldsymbol{vv}^* - \boldsymbol{uu}^*\|_1 \leq c \quad \text{implies} \quad \min_{\boldsymbol{vv}^* \in \mathcal{N}_\theta} \|\boldsymbol{vv}^* - \boldsymbol{uu}^*\|_1 \leq c + 2\theta,$$

according to Lemma 4.9. The converse direction must not hold, however. Viewed as events, the right hand side therefore occurs with a larger probability. Importantly, the minimization is over a finite set of net states. Therefore, we may apply the union bound:

$$\begin{aligned}
\Pr\left[\inf_{\text{product state } \boldsymbol{vv}^*} \|\boldsymbol{vv}^* - \boldsymbol{uu}^*\|_1 \leq c\right] \leq & \Pr\left[\min_{\boldsymbol{vv}^* \in \mathcal{N}_\theta} \|\boldsymbol{vv}^* - \boldsymbol{uu}^*\|_1 \leq c + 2\theta\right] \\
= & \Pr\left[\bigcup_{\boldsymbol{vv}^* \in \mathcal{N}_\theta} \{\|\boldsymbol{vv}^* - \boldsymbol{uu}^*\|_1 \leq c + 2\theta\}\right] \\
\leq & \sum_{\boldsymbol{vv}^* \in \mathcal{N}_\theta} \Pr[\|\boldsymbol{vv}^* - \boldsymbol{uu}^*\|_1 \leq c + 2\theta] \\
\leq & |\mathcal{N}_\theta| \max_{\boldsymbol{vv}^* \in \mathcal{N}_\theta} \Pr[\|\boldsymbol{vv}^* - \boldsymbol{uu}^*\|_1 \leq c + 2\theta].
\end{aligned}$$

□

5 Proof of Theorem 3.2

Simply combine Lemma 4.10 with exponentially strong concentration for every net element:

$$\begin{aligned}
\Pr\left[\inf_{\text{product state } \boldsymbol{vv}^*} \|\boldsymbol{vv}^* - \boldsymbol{uu}^*\|_1 \leq c\right] &\leq |\mathcal{N}_\theta| \max_{\boldsymbol{vv}^* \in \mathcal{N}_\theta} \Pr[\|\boldsymbol{vv}^* - \boldsymbol{uu}^*\|_1 \leq c + 2\theta] \\
&\leq |\mathcal{N}_\theta| 2\mathrm{e}^{-D(1-(c+2\theta)^2/4)/2}.
\end{aligned}$$

Finally, recall $D = d_1 d_2$ and use the volumetric bound on the cardinality of $\mathcal{N}_\theta$:

$$\Pr\left[\inf_{\text{product state } \boldsymbol{vv}^*} \|\boldsymbol{vv}^* - \boldsymbol{uu}^*\|_1 \leq c\right] \leq 2\left(1 + \frac{2}{\theta}\right)^{2(d_1+d_2)} \mathrm{e}^{-D(1-(c+2\theta)^2/4)/2}.$$

We could now optimize over the fineness θ of the net.

The naive choice $\theta = 1/4$ suffices for our purpose. Furthermore, specifying $c = 1$ yields

$$\Pr\left[\inf_{\text{product state } \boldsymbol{v}\boldsymbol{v}^*} \|\boldsymbol{v}\boldsymbol{v}^* - \boldsymbol{u}\boldsymbol{u}^*\|_1 \leq 1\right] \leq 29^{2(d_1+d_2)} \mathrm{e}^{-7D/32} \leq 2\exp\left(4.5(d_1+d_2) - \frac{7d_1d_2}{32}\right),$$

as advertised in Theorem 3.2.

Lecture 07: Classical reversible circuits

Scribe: Thom Bohdanowicz

ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp
April 22, 2019

1 Agenda

1. Evolution of classical probability distributions
2. Classical reversible circuits
   (a) Bit strings and tensor products
   (b) logical operations and permutation matrices
   (c) circuit diagrams

2 Evolution of classical probability distributions

2.1 Recapitulation of classical probability distributions

The set of all d-variate probability distributions is the simplex:

$$\Delta_{d-1} = \left\{ \boldsymbol{x} \in \mathbb{R}^d : \ \boldsymbol{x} \geq \boldsymbol{0}, \ \langle \boldsymbol{1}, \boldsymbol{x} \rangle = 1 \right\} \subset \mathbb{R}^d.$$

It has d extreme points, namely the *standard basis vectors*: $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$. The entire simplex can be reached by convex mixtures of these extreme points:

$$\Delta_{d-1} = \operatorname{conv}\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\} = \left\{ \sum_{i=1}^{r} \tau_i \boldsymbol{e}_i : \ \tau_i \geq 0, \ \sum_{i=1}^{r} \tau_i = 1 \right\}$$

Another distinguished point of Δ_{d-1} is its the (bary-) center:

$$\boldsymbol{b} = \frac{1}{d} \sum_{i=1}^{d} \boldsymbol{e}_i = \frac{1}{d} \boldsymbol{1}.$$

This corresponds to the flat (maximally random) distribution of d events.

2.2 Elementary transformations

We now address the question of *evolution* of probability distributions. More precisely, we are looking for *linear* transformations that map probability distributions onto probability distributions.

Example 2.1 (Reset)**.** Fix $\boldsymbol{q} \in \Delta_{d-1}$ and define $\boldsymbol{A} = \boldsymbol{q}\boldsymbol{1}^T \in \mathcal{L}(H)$. Then, for any $\boldsymbol{p} \in \Delta_{d-1}$:

$$\boldsymbol{A}\boldsymbol{p} = \boldsymbol{q} \langle \boldsymbol{1}, \boldsymbol{p} \rangle = \boldsymbol{q}.$$

This evolution resets arbitrary distributions $\boldsymbol{p}$ back to some fixed distribution $\boldsymbol{q}$.

Although valid, resets behave in a rather peculiar fashion. They do not depend on the input at all. In the following we shall restrict our attention to less invasive evolutions.

Definition 2.2. A map $\boldsymbol{A} \in \mathcal{L}(H)$ is *unital* if the flat distribution is a fix-point: $\boldsymbol{A}\mathbf{1} = \mathbf{1}$.

Unital transformations are linear and preserve the barycenter of the simplex. In order to fully preserve the geometric structure of the simplex, a unital transformation must also obey the following three properties:

1. *Non-negativity:* $\boldsymbol{A}_{ij} \geq 0$. Every entry of $\boldsymbol{A}$ (with respect to the standard basis) must be non-negative. Otherwise we could identify a test distribution $\boldsymbol{p} \in \Delta_{d-1}$ that gets mapped onto a vector that has negative entries.
2. *Rows must sum to one:* $\sum_{i=1}^{d} [\boldsymbol{A}]_{k,i} = 1$ for all $1 \leq k \leq d$. This is a consequence of unitality:
$$\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = \mathbf{1} = \boldsymbol{A}\mathbf{1} = \begin{pmatrix} \sum_{i=1}^{d} \boldsymbol{A}_{1i} \\ \vdots \\ \sum_{i=1}^{d} \boldsymbol{A}_{di} \end{pmatrix}$$
3. *Columns must sum to one:* $\sum_{i=1}^{d} \boldsymbol{A}_{ik} = 1$ for all $1 \leq k \leq d$. This is a consequence of the normalization constraint $\langle \boldsymbol{q}, \mathbf{1} \rangle = 1$ for all $\boldsymbol{q} \in \Delta_{d-1}$. Suppose $\boldsymbol{q} = \boldsymbol{A}\boldsymbol{p}$ for some $\boldsymbol{p} \in \Delta_{d-1}$. Then, $1 = \langle \mathbf{1}, \boldsymbol{q} \rangle = \langle \mathbf{1}, \boldsymbol{A}\boldsymbol{p} \rangle = \langle \boldsymbol{A}^T\mathbf{1}, \boldsymbol{p}$. Validity of this normalization for any $\boldsymbol{p} \in \Delta_{d_1}$ enforces $\boldsymbol{A}^T\mathbf{1} = \mathbf{1}$. Point 2 implies that this is equivalent to demanding that the rows of $\boldsymbol{A}^T$ (i.e. the columns of $\boldsymbol{A}$) must sum to one.

Operators $\boldsymbol{A} \in \mathcal{L}(H)$ with these three properties correspond to *doubly stochastic matrices*.

Fact 2.3. *Every unital map $\boldsymbol{A}$ that obeys $\boldsymbol{A}\Delta_{d-1} \subseteq \Delta_{d-1}$ is described by a* doubly stochastic matrix*.*

It is easy to check that the set of all doubly-stochastic matrices is convex. This convex set is called the Birkhoff polytope.

A permutation matrix is an orthogonal matrix $\boldsymbol{\Pi} \in \mathcal{L}(H)$ such that every row and every column contain exactly one entry of one. All other entries are zero.

Example 2.4 (Permutation matrices for $d = 3$)**.**

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Permutation matrices are in one-to-one correspondence with permutations of the standard basis vectors. For instance, the second matrix in the example permutes the

first two standard basis vectors, while leaving the third one invariant:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{e}_1 = \boldsymbol{e}_2, \quad \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{e}_2 = \boldsymbol{e}_1, \quad \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{e}_3 = \boldsymbol{e}_3.$$

It is easy to check that every permutation matrix is doubly stochastic. What is more, permutation matrices seem to correspond to "extreme" versions of doubly stochastic matrices. Most of the entries are zero and therefore exactly saturate the non-negativity constraint $[\boldsymbol{A}_{ij}] \geq 0$. The Birkhoff-von Neumann theorem makes this intuition precise.

Theorem 2.5 (Birkhoff von-Neumann)**.** *The set of doubly stochastic matrices is a convex polytope. Its extreme points correspond to permutation matrices.*

In other words: every doubly stochastic matrix is a convex mixture of permutation matrices. This has profound implications for the study of (unital) evolutions of probability distributions.

Corollary 2.6 (Full characterization of unitary maps that preserve the simplex)**.** *Every unital map $\boldsymbol{A}$ that obeys $\boldsymbol{A}\Delta_{d-1} \subseteq \Delta_{d-1}$ is a convex mixture of permutation matrices.*

Permutation matrices are extreme unital transformations. They simply permute the set of extreme points of Δ_{d_1}:

$$\boldsymbol{\Pi} : \ \boldsymbol{e}_1, \ldots, \boldsymbol{e}_d \mapsto \boldsymbol{e}_{\pi(1)}, \ldots, \boldsymbol{e}_{\pi(d)}.$$

In turn, they leave the simplex invariant:

$$\boldsymbol{\Pi}\Delta_{d-1} = \operatorname{conv}\{\boldsymbol{\Pi}\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\} = \operatorname{conv}\Big\{\boldsymbol{e}_{\pi(1)}, \ldots, \boldsymbol{e}_{\pi(d)}\Big\} = \Delta_{d-1}.$$

More general untital evolutions (convex mixtures) shrink the simplex. This geometric observation may be viewed as a starting point for the beautiful theory of *majorization*.

Remark 2.7. For the sake of simplicity, we have restricted our attention to untial maps from Δ_{d-1} to itself. This restriction is not necessary. Similar arguments allow for characterizing unital evolutions that change the dimension (number of potential outcomes): $\boldsymbol{A} : \ \Delta_{d-1} \to \Delta_{d'-1}$ with $d' \neq d$.

3 Classical reversible circuits

3.1 Bit-strings and the extended standard basis

There is a deep connection between unital evolutions of probability distributions and classical, reversible computation. To make this correspondence as explicit as possible, we introduce the following notation. Fix $d = 2$ (bi-variate distributions) and identify the standard basis with either the zero-bit, or the one-bit:

$$0 \sim \boldsymbol{e}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \ 1 \sim \boldsymbol{e}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

This establishes a connection between $\{0,1\} = \mathbb{Z}_2$ (bit-land) and $\{\boldsymbol{e}_0, \boldsymbol{e}_1\} \subset \Delta_1 \subset \mathbb{R}^2$ (deterministic probability distributions).

We can use tensor products to extend this identification to bit strings of length n:

$$(x_1 \cdots x_n) \in \{0,1\}^n \quad \sim \boldsymbol{e}_{x_1} \otimes \cdots \otimes \boldsymbol{e}_{x_n} \in \left(\mathbb{R}^2\right)^{\otimes n}.$$

We identify length-n bit strings with the labels of the extended standard basis of $\left(\mathbb{R}^2\right)^{\otimes n}$. These extended standard basis vectors form the extreme points of a simplex in 2^n dimensions:

$$\begin{aligned} \Delta_{2^n-1} =& \text{conv}\{\boldsymbol{e}_{x_1} \otimes \cdots \otimes \boldsymbol{e}_{x_n} : \ (x_1 \cdots x_n) \in \{0,1\}^n\} \\ \simeq& \left\{\boldsymbol{x} \in \left(\mathbb{R}^2\right)^{\otimes n} : \ \langle \boldsymbol{x}, \mathbf{1}\rangle = 1, \ \boldsymbol{x} \geq \mathbf{0}\right\}. \end{aligned} \tag{1}$$

Here, the sign "$\simeq$" denotes equivalence up to isomorphisms. The above relation holds true with equality if we identify the standard basis of $\mathbb{R}^{2^n}$ with the extended standard basis of $\left(\mathbb{R}^2\right)^{\otimes n}$.

3.2 Permutation matrices in $\left(\mathbb{R}^2\right)^{\otimes n}$ and logical operations on bit strings

It is highly instructive to consider the symmetry group of the simplex $\Delta_{2^n-1} \subset \left(\mathbb{R}^2\right)^{\otimes n}$ defined in (1).

3.2.1 The permutation matrix associated with negation ($n = 1$)

For $n = 1$, there are only two permutations. The identity $\boldsymbol{\Pi} = \mathbf{I} \in \mathcal{L}(\mathbb{R}^2)$ and transposition. The former leaves standard basis vectors – and their associated bits – invariant, while transposition permutes them:

$$\begin{aligned} T(0) \sim& \boldsymbol{T}\boldsymbol{e}_0 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \boldsymbol{e}_0 = \boldsymbol{e}_1 \sim 1, \\ T(1) \sim& \boldsymbol{T}\boldsymbol{e}_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \boldsymbol{e}_1 = \boldsymbol{e}_0 \sim 1. \end{aligned}$$

On the level of bits, we can associate this transformation with the following truth table:

$x \in \{0,1\}$	$T(x)$
0	1
1	0

.

Hence, the logical operation associated with transposition $\boldsymbol{T} \in \mathcal{L}(\mathbb{R}^2)$ is *negation*:

$$T(x) = \neg x \quad \text{for} \quad x \in \{0,1\}.$$

3.2.2 The permutation matrix associated with XOR ($n = 2$)

For $n = 2$, $\left(\mathbb{R}^2\right)^{\otimes 2} \simeq \mathbb{R}^4$ is accompanied by in total $4! = 24$ permutation matrices. Some of them arise from tensor products of permutation matrices acting on $\mathbb{R}^2$ only.

Concrete examples are the identity (do nothing) and all possible combinations of single bit negations:

$$\begin{aligned} I(x_1x_2) \sim & \mathbf{I} \otimes \mathbf{I} \boldsymbol{e}_{x_1} \otimes \boldsymbol{e}_{x_2} = \boldsymbol{e}_{x_1} \otimes \boldsymbol{e}_{x_2} = x_1x_2, \\ T_1(x_1x_2) \sim & \boldsymbol{T} \otimes \mathbf{I} \boldsymbol{e}_{x_1} \otimes \boldsymbol{e}_{x_2} = \boldsymbol{e}_{\neg x_1} \otimes \boldsymbol{e}_{x_2} \sim \neg x_1x_2, \\ T_2(x_1x_2) \sim & \mathbf{I} \otimes \boldsymbol{T} \boldsymbol{e}_{x_1} \otimes \boldsymbol{e}_{x_2} = \boldsymbol{e}_{x_1} \otimes \boldsymbol{e}_{\neg x_2} \sim x_1\neg x_2, \\ T_{1,2}(x_1x_2) \sim & \boldsymbol{T} \otimes \boldsymbol{T} \boldsymbol{e}_{x_1} \otimes \boldsymbol{e}_{x_2} = \boldsymbol{e}_{\neg x_1} \otimes \boldsymbol{e}_{\neg x_2} \sim \neg x_1\neg x_2 \end{aligned}$$

for all $x_1, x_2 \in \{0, 1\}$. Other permutations are genuine elements of $\mathbb{R}^4$ and cannot be decomposed into tensor products of smaller permutation matrices. A concrete example is the following permutation matrix

$$\boldsymbol{X} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \in \mathcal{L}\Big(\mathbb{R}^4\Big).$$

that we have represented with respect to the extended standard basis $\boldsymbol{e}_{00} = \boldsymbol{e}_0 \otimes \boldsymbol{e}_0, \boldsymbol{e}_{01} = \boldsymbol{e}_0 \otimes \boldsymbol{e}_1, \boldsymbol{e}_{10} = \boldsymbol{e}_1 \otimes \boldsymbol{e}_0, \boldsymbol{e}_{11} = \boldsymbol{e}_1 \otimes \boldsymbol{e}_1$ of $\mathbb{R}^4 \simeq \left(\mathbb{R}^2\right)^{\otimes 2}$. It corresponds to the following logical transformation on length-two bit strings:

$$\begin{aligned} X(00) \sim & \boldsymbol{X}\boldsymbol{e}_0 \otimes \boldsymbol{e}_0 = \boldsymbol{e}_0 \otimes \boldsymbol{e}_0 \sim 00, \\ X(01) \sim & \boldsymbol{X}\boldsymbol{e}_0 \otimes \boldsymbol{e}_1 = \boldsymbol{e}_0 \otimes \boldsymbol{e}_1 \sim 01, \\ X(10) \sim & \boldsymbol{X}\boldsymbol{e}_1 \otimes \boldsymbol{e}_0 = \boldsymbol{e}_1 \otimes \boldsymbol{e}_1 \sim 11, \\ X(11) \sim & \boldsymbol{X}\boldsymbol{e}_1 \otimes \boldsymbol{e}_1 = \boldsymbol{e}_1 \otimes \boldsymbol{e}_0 \sim 10. \end{aligned}$$

On the level of bits, we can associate this transformation with the following truth table:

$x \in \{0,1\}$	$x_2 \in \{0,1\}$	$[X(x_1x_2)]_1$	$[X(x_1x_2)]_2$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

.

Conditioned on the first bit being one ($x_1 = 1$), this operation inverts the second bit. Otherwise it does nothing. This action corresponds to the *reversible XOR* (exclusive ore):

$$\mathrm{XOR}(x_1x_2) = \begin{cases} x_1x_2 & \text{if } x_1 = 0, \\ x_1\neg x_2 & \text{if } x_1 = 1 \end{cases}.$$

3.2.3 Correspondence for general n: Reversible logical functions

The approach outlined generalizes to tensor products of order n, or equivalently: bit strings of length n. Permutation matrices $\boldsymbol{\Pi} \in \left(\mathbb{R}^2\right)^{\otimes n}$ may be interpreted as logical operations on length-n bit strings. We emphasize two key features of permutation matrices:

1. *Orthogonality:* $\mathbf{\Pi}^{-1} = \mathbf{\Pi}^T$ for any permutation matrix
2. *Group structure:* Permutation matrices form a finite group.

Both features have profound implications for the associated logical functions. The first statement highlights that reading a permutation backwards, corresponds to taking the inverse. In particular: $\mathbf{\Pi}^T\mathbf{\Pi} = \mathbf{I}$ implies $\pi^T(\pi(x_1 \cdots x_n)) = x_1 \cdots x_n$. Here, $\pi^T = \pi^{-1}$ denotes the "reverse" of a logical function.

Fact 3.1. *Permutation matrices* $\mathbf{\Pi}$ *acting on* $\left(\mathbb{R}^2\right)^{\otimes n}$ *can represent any reversible logical function.*

The group structure also has profound implications. Finite groups G typically have a small set of generators $\boldsymbol{G}_1, \ldots, \boldsymbol{G}_m \subset G$. These can be combined to generate any element of the group. In particular, a small number of permutation matrices suffices to "build" arbitrary permutation matrices. Fact 3.1 allows for extending this structure to reversible logical functions.

Fact 3.2. *Any reversible logical function* $\pi : \{0,1\}^n \to \{0,1\}^n$ *can be decomposed into a product of smaller (more elementary) logical functions. This procedure is called a* circuit decomposition.

The logical functions associated with generators of permutation matrices are called an *elementary (reversible) gate set*. Perhaps surprisingly, a single logical function on 3 bits suffices to decompose any reversible n-bit function[1]. This magic function is the *Toffoli gate*. It's permutation matrix corresponds to

$$\boldsymbol{TOF} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

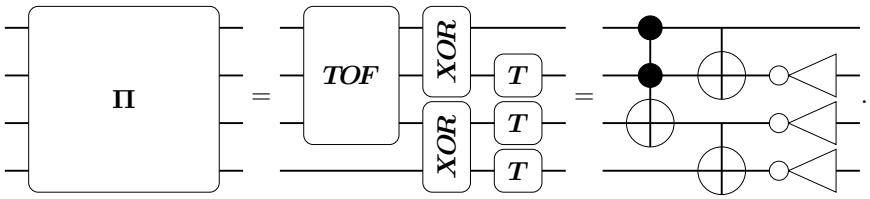
3.3 Circuit diagrams for reversible computation

Suppose that we have found a decomposition of a general permutation matrix $\mathbf{\Pi} \in \mathcal{L}\left((\mathbb{R}^2)^{\otimes n}\right)$ into a product of simpler permutation matrices that act only on a sub-set of the tensor factors. We can then use the wiring formalism to visualize this decomposition. For instance, suppose that $\mathbf{\Pi}$ acts on four tensor factors (bits) and may be decomposed as

$$\mathbf{\Pi} = (\boldsymbol{TOF} \otimes \mathbf{I})(\boldsymbol{XOR} \otimes \boldsymbol{XOR})\mathbf{I} \otimes \boldsymbol{T}^{\otimes 3} : \left(\mathbb{R}^2\right)^{\otimes 4} \to \left(\mathbb{R}^2\right)^{\otimes 4}.$$

[1] In order to achieve this decomposition, we may have to embed the logical function $\pi : \{0,1\}^n \to \{0,1\}^n$ into a larger bit space $\{0,1\}^{n+a}$ and use the additional a bits as "ancillas".

The graphical visualization of this decomposition is



On the right hand side, we have replaced the individual boxes with standard expressions from the field of logical circuits. The diagram on the right is called a *circuit diagram*. The identification of bit strings with extended standard basis vectors in $(\mathbb{R}^2)^{\otimes n}$ naturally produces this important framework from electrical engineering. However, there is a slight twist. We read wiring diagrams from right to left, while circuit diagrams are typically read from left to right.

This example can be generalized to arbitrary logical functions that are associated with big permutation matrices $\mathbf{\Pi}$. The fact that permutation matrices form a group asserts that *any* such logical function can be decomposed into a sequence of more elementary logical functions. Wiring diagrams, or circuit diagrams, visualize such a decomposition in a graphical fashion.

Fact 3.3. *Wiring diagrams may be viewed as a natural extension of classical circuit diagrams.*

Reversible logical functions (big permutation matrix) form the basic building block of reversible computation.

Definition 3.4. A classical reversible computation is a three-step procedure:

1. input: $x_1 \cdots x_n \mapsto \boldsymbol{e}_{x_1} \otimes \cdots \otimes \boldsymbol{e}_{x_b}$
2. computation: run the reversible circuit on $x_1 \cdots x_n$.
3. read-out: Perform the measurement $p(y_1 \cdots y_d) = \langle \boldsymbol{e}_{y_1} \otimes \cdots \otimes \boldsymbol{e}_{y_n} \mathbf{\Pi} \boldsymbol{e}_{\boldsymbol{x}_1} \otimes \cdots \boldsymbol{e}_{x_d} \rangle$. This measurement yields a deterministic read-out.

It is worthwhile to point out the following feature of this formalism. Reading the diagram/circuit backwards necessarily produces $\mathbf{\Pi}^T$. This is the inverse of $\mathbf{\Pi}$. A hardware implementation of such a circuit therefore has the following appealing feature. We can re-set a concrete computation by running the circuit backwards! This automatically re-sets the bits to the original input:

$$x_1 \cdots x_n \overset{\text{circuit: right to left}}{\longrightarrow} y_1 \cdots y_n \overset{\text{circuit: left to right}}{\longrightarrow} x_1 \cdots x_n.$$

Standard, i.e. non-reversible, circuits do not have this feature. They must be re-set by force after a computation is completed. A fundamental thermodynamic restriction, called *Landauer's principle*, states that such re-sets necessarily cost work/energy, because they erase information. Reversible computation is in principle capable of bypassing

this fundamental threshold. For this reason, the study of reversible computations has recently gained some traction again. The ultimate goal is to derive hardware that requires considerable less energy.

We conclude this section with an important remark regarding generalization. The tensor product representation of reversible circuits is remarkably powerful. We can easily extend it to reason about randomized inputs and randomized computations. Simply replace the deterministic input $\boldsymbol{e}_{x_1} \otimes \cdots \otimes \boldsymbol{e}_{x_n}$ by a more general probability distribution $\boldsymbol{p} \in \Delta_{2^n-1}$, and replace permutation matrices $\mathbf{\Pi}$ by doubly stochastic matrices.

Lecture 08: Quantum circuits and quantum computing

Scribe: Alex Dalzell

ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp
April 24, 2019

1 Agenda

1. Evolution of quantum states
2. Quantum circuits and quantum computation
3. Where is the quantum magic?

   Unitaries vs. Permutation?

   Hadamard + superposition?

   Entanglement?

In the last lecture, we focused on the evolution of classical probability distributions. These are described by doubly stochastic matrices and correspond to the convex hull of all permutation matrices. Subsequently, we restricted our attention to permutation matrices (extreme evolutions) and standard basis vectors (extreme probability distributions) on the tensor product space $(\mathbb{R}^2)^{\otimes n}$. In doing so, we developed the theory of reversible classical computation: bit strings are associated with extended standard basis vectors and reversible logical functions permute them. Here, we apply the same reasoning to (probability) density matrices. This will give rise to *quantum computing* – the natural (SDP) extension of reversible computing that has received a tremendous amount of attention over the past decades.

2 Evolution of quantum states

2.1 Recapitulation of quantum states

Set $H = \mathbb{C}^d$. The set of all quantum states is

$$\mathsf{S}(H) = \{\boldsymbol{X} \in \mathcal{L}(H) : \ \boldsymbol{X}^* = \boldsymbol{X}, \ \boldsymbol{X} \succeq \mathbf{0}, \ (\mathbb{I}, \boldsymbol{X})\}.$$

This is a convex set. The extreme points are given by *pure quantum states* $\boldsymbol{v}\boldsymbol{v}^*$, where $\boldsymbol{v} \in \mathbb{C}^d$ has unit norm:

$$\mathsf{S}(H) = \operatorname{conv}\Big\{\boldsymbol{v}\boldsymbol{v}^* : \ \boldsymbol{v} \in \mathbb{C}^d, \ \langle \boldsymbol{v}, \boldsymbol{v} \rangle = 1\Big\}.$$

This formula highlights that any density matrix $\boldsymbol{\rho}$ corresponds to a convex mixture of pure states. Another distinguished point is the bary-center:

$$\boldsymbol{B} = \int_{\mathbb{S}^{d-1}} \mathrm{d}\mu(\boldsymbol{v}) \boldsymbol{v}\boldsymbol{v}^* = \int \mathrm{d}\mu(\boldsymbol{U}) \boldsymbol{U}\boldsymbol{v}_0\boldsymbol{v}_0^*\boldsymbol{U}^* = \frac{1}{d}\mathbf{I}.$$

This is called the *maximally mixed state* and is the quantum analogue of the completely flat distribution.

2.2 Elementary evolution of density matrices

We now address the question of *evolution*. We are looking for *linear* transformations that map the set of quantum states onto itself:

$$\mathcal{A} : \mathcal{L}(H) \to \mathcal{L}(H) \quad \text{s.t.} \quad \mathcal{A}(\mathsf{S}(H)) \subseteq \mathsf{S}(H).$$

These operators act on operators, i.e. $\mathcal{A} \in \mathcal{L}(\mathcal{L}(H))$ and we will call them *channels*.

Example 2.1 (Reset)**.** Fix $\boldsymbol{\sigma} \in \mathsf{S}(H)$ and define $\mathcal{A}(\boldsymbol{X}) = \boldsymbol{\sigma}\mathrm{tr}(\boldsymbol{X})$ for $\boldsymbol{X} \in \mathcal{L}(H)$. This is a valid quantum channel: $\mathcal{A}(\boldsymbol{\rho}) = \boldsymbol{\sigma} \in \mathsf{S}(H)$ for all density matrices $\boldsymbol{\rho} \in \mathsf{S}(H)$.

Although valid channels, resets behave in a peculiar fashion and do only respect the geometry of quantum states in an extreme sense. All of $\mathsf{S}(H)$ is mapped onto a single point. In the following, we shall restrict our attention to less invasive evolutions that preserve the barycenter of state space.

Definition 2.2. A channel $\mathcal{A} : \mathcal{L}(H) \to \mathcal{L}(H)$ is called *unital*, if $\mathcal{A}(\mathbf{I}) = \mathbf{I}$.

Let us now focus on the natural quantum extension of permutation matrices.

Proposition 2.3. *Fix a unitary matrix $\boldsymbol{U} \in U(d)$. Then, the channel $\mathcal{U}(\boldsymbol{X}) = \boldsymbol{U}\boldsymbol{X}\boldsymbol{U}^*$ maps pure states onto pure states. These maps are also unital: $\mathcal{U}(\mathbf{I}) = \boldsymbol{U}\mathbf{I}\boldsymbol{U}^* = \mathbf{I}$.*

Unitary channels appropriately mimic the features of permutation matrices acting on Δ_{d-1}. They fix the barycenter and map quantum state space onto itself:

$$\mathcal{U}(\mathsf{S}(H)) = \{\boldsymbol{U}\boldsymbol{X}\boldsymbol{U}^* : \ \boldsymbol{X}^* = \boldsymbol{X}, \ \boldsymbol{X} \succeq \mathbf{0}, \ (\boldsymbol{I}, \boldsymbol{X}) = 1\}.$$

More general unital channels necessarily shrink $\mathsf{S}(H)$. Typically one defines the set of unital quantum channels via the following three conditions:

1. complete positivity: a channel must map every psd matrix onto a psd matrix (in a strong sense): $\mathcal{A} \otimes \mathcal{I}(\boldsymbol{\rho}) \succeq \mathbf{0}$ for all $\boldsymbol{\rho} \in \mathcal{L}(H) \otimes \mathcal{L}(H)$.
2. unitality: $\mathcal{A}(\mathbf{I}) = \mathbf{I}$.
3. trace preservation: $(\mathbf{I}, \mathcal{A}(\boldsymbol{X})) = (\mathbf{I}, \boldsymbol{X})$.

These three requirements single out a convex set of channels. Unitary channels correspond to extreme points of this set. However, this set of extreme points is not complete!

Fact 2.4 (Quantum von-Neumann theorem is false)**.** *Convex mixtures of unitary channels $\mathcal{A} = \sum_{i=1}^{r} \tau_i \mathcal{U}_i$* do not *generate the set of all unital quantum channels. There are exotic exceptions.*

This apparent lack of structure renders the study of quantum channels more complicated than their classical counter-part (the set of doubly stochastic matrices is a polytope and permutation matrices are its extreme points). Nonetheless, it is useful to consider unitary channels as the most extreme quantum evolutions. They do share

several desirable properties with permutation matrices. In particular, subsequent applications of unitary channels correspond to products of unitaries. Let $\mathcal{V}_1(\boldsymbol{X}) = \boldsymbol{V}_1\boldsymbol{X}\boldsymbol{V}_1^*$ and $\mathcal{V}_2(\boldsymbol{X}) = \boldsymbol{V}_2\boldsymbol{X}\boldsymbol{V}_2^*$ be two unitary channels. Then,

$$\mathcal{V}_2 \circ \mathcal{V}_1(\boldsymbol{\rho}) = \mathcal{V}_2(\mathcal{V}_1(\boldsymbol{\rho})) = \mathcal{V}_2(\boldsymbol{V}_1\boldsymbol{\rho}\boldsymbol{V}_1^*) = \boldsymbol{V}_2\boldsymbol{V}_1\boldsymbol{\rho}(\boldsymbol{V}_1\boldsymbol{V}_2)^* = \mathcal{U}(\boldsymbol{\rho})$$

is again a unitary channel described by $\boldsymbol{U} = \boldsymbol{V}_1\boldsymbol{V}_2$. What is more, unitary matrices form a group $\mathcal{U}(d)$. This allows us to decompose general unitary channels as a product of simpler unitary channels.

3 Quantum circuits and computations

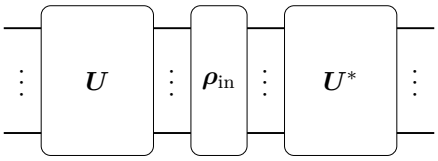
We can now basically repeat our analysis of reversible circuits from last lecture. For concreteness, we will be working explicitly with circuits acting on a colection of qubits ($d = 2$): thus denote

$$H = \mathbb{C}^2, \qquad \mathsf{S}(H) = \operatorname{conv}\{\boldsymbol{v}\boldsymbol{v}^* : \boldsymbol{v} \in H,\ \langle \boldsymbol{v}, \boldsymbol{v}\rangle = 1\}$$

The qubits are the hardware part of a quantum circuit. If we have n qubits, then the quantum circuit acts on an initial quantum state $\boldsymbol{\rho}_{\text{joint}} \in \mathsf{S}(H^{\otimes n}) \subseteq \mathcal{L}(H^{\otimes n})$. The state $\boldsymbol{\rho}_{\text{joint}}$ plays the role that the initial tensor $\boldsymbol{e}_{x_1} \otimes \ldots \otimes \boldsymbol{e}_{x_n} \in (\mathbb{R}^2)^n$ played for classical computation. Then a quantum circuit is simply a big unitary operator $\boldsymbol{U} \in \mathcal{L}(H^{\otimes n})$, which induces the *unitary channel*

$$\mathcal{U}(\boldsymbol{\rho}_{\text{in}}) = \boldsymbol{U}\boldsymbol{\rho}_{\text{in}}\boldsymbol{U}^*. \tag{1}$$

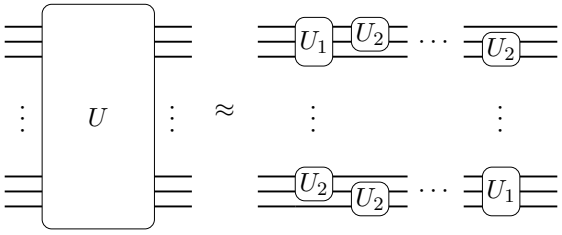
In wiring notation, this unitary channel acts in the following fashion:



Fact 3.1. *The set of unitary matrices on $H^{\otimes n}$ forms a group.*

Moreover, the set of unitaries is approximately generated by a fixed finite set of small unitaries.

Theorem 3.2 (Solovay, Kitaev)**.** *A small number of small (i.e acting on two or three neighboring tensor factors) suffices to approximate any unitary $U \in \mathcal{L}(H^{\otimes n})$*[1]*.*



[1]Specifically, they showed that any U can be approximated to accuracy ε using a sequence composed of smaller unitaries of length only polynomial in $\log(1/\varepsilon)$.

A set of gates that is capable of approximately generating any unitary in $\mathcal{L}(H^{\otimes n})$ is called a *universal gate set*. In the decomposition of U into a sequence of unitaries drawn from such a generating set, the length of the sequence is called the *circuit length*.

Remark 3.3. Parallelization can substantially reduce the length of the circuit. The parallelized circuit length is called the *circuit depth*.

For example, there are six unitaries shown in the decomposition above, so they would contribute six toward the circuit length but only three toward the circuit depth since they can be implemented in three parallel layers.

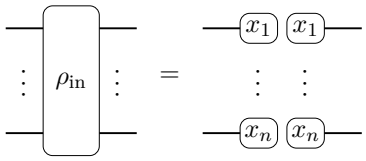
Decomposition of unitary matrices readily implies decomposition of unitary channels (quantum circuits). Following Eq. (1), if $\boldsymbol{U} = \boldsymbol{U}_1 \dots \boldsymbol{U}_L$, then

$$\mathcal{U}(\boldsymbol{\rho}_{\text{in}}) = \boldsymbol{U}_1 \dots \boldsymbol{U}_L \boldsymbol{\rho}_{\text{in}} \boldsymbol{U}_L^* \dots \boldsymbol{U}_1^* = \mathcal{U}_1(\mathcal{U}_2(\dots \mathcal{U}_L(\rho_{\text{in}}) \dots)) = \mathcal{U}_1 \circ \dots \circ \mathcal{U}_L(\boldsymbol{\rho}_{\text{in}}) \tag{2}$$

Now we can formally define a quantum computation.

Definition 3.4 (Quantum computation)**.** A quantum computation consists of the following ingredients.

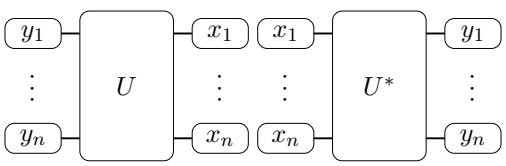
1. Input: Given a classical bit string $x_1, \dots, x_n$, initialize $\boldsymbol{\rho}_{\text{in}}$ to be in a tensor product state: $x_1, \dots, x_n \mapsto \rho_{\text{in}} = \boldsymbol{e}_{x_1}\boldsymbol{e}_{x_1}^* \otimes \dots \otimes \boldsymbol{e}_{x_n}\boldsymbol{e}_{x_n}^*$:



2. Computation: Apply a unitary channel $\boldsymbol{\rho}_{\text{out}} = \mathcal{U}(\boldsymbol{\rho}_{\text{in}} = \boldsymbol{U}\rho_{\text{in}}\boldsymbol{U}^*$
3. Output: Perform a fixed quantum measurement to retrieve a classical string $y_1, \dots, y_n$ from $\boldsymbol{\rho}_{\text{out}}$.

$$\boldsymbol{H}_{y_1,\dots,y_n} = \boldsymbol{e}_{y_1}\boldsymbol{e}_{y_1}^* \otimes \dots \otimes \boldsymbol{e}_{y_n}\boldsymbol{e}_{y_1}^* \succeq \boldsymbol{0} \quad \text{and} \quad \sum_{y_1 \dots y_n = 0}^{1} \boldsymbol{H}_{y_1,\dots,y_n} = \mathbf{I}. \tag{3}$$

A quantum computation can then be represented by the following diagram, which computes the probability of measuring output $y_1, \dots, y_n$ after starting with input $x_1, \dots, x_n$.



(4)

This diagram splits into two separate diagrams that are complex conjugates of one another. Recall that a classical reversible circuit diagram looked very similar to a single copy of one of these constituent diagrams.

The complexity of the quantum computation is the circuit depth of $\boldsymbol{U}$. A quantum computation is said to be polynomial-size if its circuit depth is less than a polynomial in the number of qubits n. The ultimate goal of research in quantum algorithms is to find problems that can be solved by polynomial-sized quantum computations but not polynomial-sized classical computations. For example, Shor's algorithm describes a polynomial-sized quantum computation for factoring integers, a problem for which there is no known classical polynomial-sized computation. Another example of a problem where quantum computations may provide a drastic advantage is in the simulation of quantum systems. But there are other examples where quantum computation may offer a significant, but less drastic speedup, or where it is suspected quantum computation may be useful but no rigorous proof has been provided. Such examples include searching for items within a large unstructured search space, solving linear equations, combinatorial optimization problems, and even certain machine learning tasks.

This begs the question: what is it about quantum computations that leads to their advantage over classical computations? This is the subject of the next section.

4 What is special about quantum computing?

4.1 Unitaries vs. Permutations

A clear difference between quantum computations and classical computations is that quantum computations apply unitary mtarices on an initial tensor product of basis vectors, while classical circuits perform permutation matrices on the basis vectors.

It was discussed in the previous lecture how the "Controlled-controlled-NOT" or "Toffoli" gate is universal for reversible classical computation, since it can generate any permutation matrix. The Toffoli gate is both a permutation matrix and a unitary matrix, so it qualifies as a quantum gate, but it does not alone form a universal set for quantum computation. However, combining the Toffoli gate with the unitary "Hadamard" gate

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

is sufficient to form a universal gate set[2]. Thus we can see that quantum computation is at least as powerful as classical computation, since a designer of quantum computations may always simply forget about the Hadamard gate and perform any classical computation within the framework of a quantum computation using only Toffolis. This illustrates the necessity of the Hadamard gate if one wishes to find any sort of quantum speedup.

Another important distinction between permutation and unitary matrices is that unitary matrices form a continuous group. For example, rotation matrices

$$R_\theta = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

[2]Technically, Toffoli and Hadamard generate the group of real-valued unitaries. This, however, turns out to be sufficient for quantum computation.

are unitary for any rotation angle $\theta \in [0, 2\pi)$. Thus the degrees of freedom in a quantum computation may vary continuously while the circuit is running. The power provided by this fact alone is illustrated through the following example.[3]

Example 4.1. Suppose n people are standing in a line and each is given a number x_i, such that $X = \sum_{i=1}^{n} x_i$ is an integer. They wish to collectively compute whether X is even or odd, but they are only allowed to communicate one bit of information to the person behind them in line.

Classically, if each x_i is an integer, then this is possible: the first person computes $x_1 \mod 2$ and sends the result to the second person, who adds their number and sends $x_1 + x_2 \mod 2$ to the third person, etc. until the final person is able to compute $X \mod 2$. However, if the x_i are not integers but only rational numbers, then this strategy does not work and there is no way to successfully compute the parity of X given only the ability to communicate one classical bit.

What if the people may communicate a single qubit instead of a single classical bit? In this case, the problem can be solved in both the case that x_i are integers and in the case they are rationals. Here the first person begins with the single-qubit state $\boldsymbol{e}_0\boldsymbol{e}_0^*$ and applies the unitary operation $\boldsymbol{R}_{\pi x_1/2}$ yielding $\boldsymbol{R}_{\pi x_1/2}\boldsymbol{e}_0\boldsymbol{e}_0^*\boldsymbol{R}_{\pi x_1/2}^*$. Person i receives the qubit from person $i-1$ and applies $\boldsymbol{R}_{\pi x_i/2}$. The final state of the qubit before measurement is $\boldsymbol{R}_{\pi X/2}\boldsymbol{e}_0\boldsymbol{e}_0^*\boldsymbol{R}_{\pi X/2}^*$, which is $\boldsymbol{e}_0\boldsymbol{e}_0^*$ if X is even and $\boldsymbol{e}_1\boldsymbol{e}_1^*$ if X is odd. Thus the cases can be deterministically distinguished by the measurement with $\boldsymbol{H}_0 = \boldsymbol{e}_0\boldsymbol{e}_0^*$, $\boldsymbol{H}_1 = \boldsymbol{e}_1\boldsymbol{e}_1^*$. This works even in the rational case precisely because the unitary group is continuous and rotations can be performed by arbitrary angles.

Again through this example we see the importance of the Hadamard gate since Toffoli gates alone could not be used to approximate rotation gates by arbitrary angles. These arbitary angle rotations put the quantum data into a *superposition* of multiple basis states, which is not possible for classical computation. Indeed, as we will see in the following section, the Hadamard gate and superposition are intimately related.

4.2 Hadamard and superposition

The Hadamard gate is an extremely useful operation all by itself. It creates superpositions when acting on standard basis elements:

$$\begin{aligned}
\boldsymbol{H}\boldsymbol{e}_0 &= \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} = (\boldsymbol{e}_0 + \boldsymbol{e}_1)/\sqrt{2}\\
\boldsymbol{H}\boldsymbol{e}_1 &= \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix} = (\boldsymbol{e}_0 - \boldsymbol{e}_1)/\sqrt{2}
\end{aligned}$$

and thus

$$\boldsymbol{H}^{\otimes n}\boldsymbol{e}_0^{\otimes n} = \frac{1}{2^{n/2}}(\boldsymbol{e}_0 + \boldsymbol{e}_1)^{\otimes n} = \frac{1}{2^{n/2}}\sum_{x_1,\ldots,x_n=0}^{1} \boldsymbol{e}_{x_1} \otimes \ldots \otimes \boldsymbol{e}_{x_n}.$$

[3] This example was communicated to me by Renato Renner.

This equation shows that performing a Hadamard on each of n qubits in the $\boldsymbol{e}_0$ basis vector state yields the uniform superposition over all 2^n extended basis states on $H^{\otimes n}$. This superposition property can be exploited to yield large speedups over classical computation. A good example of this is the Deutsch-Josza algorithm.

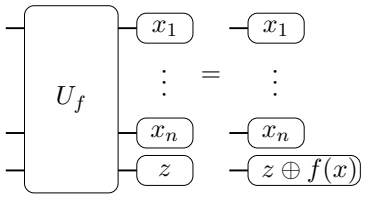
Example 4.2 (Deutsch-Josza algorithm)**.** The task solved by the Deutsch-Josza algorithm is as follows. Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function satisfying the promise that either f is constant (i.e. $f(z) = 0\ \forall z$ or $f(z) = 1\ \forall z$) or f is balanced (i.e. $|\{z : f(z) = 0\}| = |\{z : f(z) = 1\}|$). We only have black-box access to f. This means we may query $f(z)$ for a certain input z, but we can learn nothing else about f. We would like to determine whether f is constant or balanced using as few queries as possible.

Classically, the optimal strategy is simply to begin querying f for different values of z. The minimum number of queries we would need is 2: if the value of f disagrees on the first two queries we know that f cannot be constant and thus must be balanced. However, the maximum number of queries we will need in the worst case is $2^{n-1} + 1$, since if all of the first 2^{n-1} queries agree, it is still possible for the function to be balanced or for it to be constant.

Quantumly, just *one* query is enough to solve the problem. The quantum computation that illustrates this consists only of Hadamards and queries to f. We will assume that f can be queried by applying a unitary $\boldsymbol{U}_f$ that acts on basis states as

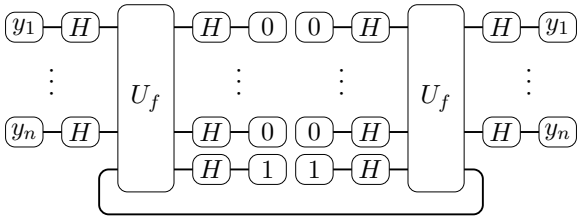
$$\boldsymbol{U}_f \boldsymbol{e}_{x_1} \otimes \ldots \otimes \boldsymbol{e}_{x_n} \otimes \boldsymbol{e}_z = \boldsymbol{e}_{x_1} \otimes \ldots \otimes \boldsymbol{e}_{x_n} \otimes \boldsymbol{e}_{z \oplus f(x_1,\ldots,x_n)} \tag{5}$$

and is linearly extended to the rest of $H^{\otimes n}$, where $\oplus$ represents addition modulo 2. Note that a similar construction would be required to construct a classical reversible computation involving queries to f; indeed, $\boldsymbol{U}_f$ is both a unitary and a permutation matrix. In wiring notation:



for all $x_1, x_2, \ldots, x_n \in \{0,1\}$. (6)

Using this quantum circuit implementation of $\boldsymbol{U}_f$, the Deutsch-Josza problem is solved by the following quantum computation:



(7)

To see that it works, we can track the data at different points in the computation. Define

$$\begin{aligned}
\boldsymbol{u}_1 &:= \boldsymbol{H}^{\otimes(n+1)}\left(\boldsymbol{e}_0^{\otimes n} \otimes \boldsymbol{e}_1\right) = 2^{-(n+1)/2} \sum_{x_1,\ldots,x_n=0}^{1} \boldsymbol{e}_{x_1} \otimes \ldots \otimes \boldsymbol{e}_{x_n} \otimes (\boldsymbol{e}_0 - \boldsymbol{e}_1) \\
\boldsymbol{u}_2 &:= \boldsymbol{U}_f \boldsymbol{u}_1 = 2^{-(n+1)/2} \sum_{x_1,\ldots,x_n=0}^{1} \boldsymbol{e}_{x_1} \otimes \ldots \otimes \boldsymbol{e}_{x_n} \otimes (\boldsymbol{e}_{f(x)} - \boldsymbol{e}_{1\oplus f(x)}) \\
&= 2^{-(n+1)/2} \sum_{x_1,\ldots,x_n=0}^{1} (-1)^{f(x_1,\ldots,x_n)} \boldsymbol{e}_{x_1} \otimes \ldots \otimes \boldsymbol{e}_{x_n} \otimes (\boldsymbol{e}_0 - \boldsymbol{e}_1) \qquad (8)
\end{aligned}$$

where the last line follows from the fact that the last qubit is $\boldsymbol{e}_0 - \boldsymbol{e}_1$ if $f(x) = 0$, and simply the negation $\boldsymbol{e}_1 - \boldsymbol{e}_0$ if $f(x) = 1$. When we take $\mathrm{tr}_{n+1}(\boldsymbol{u}_2\boldsymbol{u}_2^*)$ we still have a rank one matrix and can write it as $\boldsymbol{u}_3\boldsymbol{u}_3^*$ with

$$\boldsymbol{u}_3 := 2^{-n/2} \sum_{x_1,\ldots,x_n=0}^{1} (-1)^{f(x_1,\ldots,x_n)} \boldsymbol{e}_{x_1} \otimes \ldots \otimes \boldsymbol{e}_{x_n} \qquad (9)$$

The final step requires applying n Hadamard gates again which act as (note below the implied sum over repeated index i)

$$\begin{aligned}
\boldsymbol{u}_4 &:= \boldsymbol{H}^{\otimes n}\boldsymbol{u}_3 = 2^{-n} \sum_{x_1,\ldots,x_n=0}^{1} (-1)^{f(x_1,\ldots,x_n)} \sum_{w_1,\ldots,w_n=0}^{1} (-1)^{w_i x_i} \boldsymbol{e}_{w_1} \otimes \ldots \otimes \boldsymbol{e}_{w_n} \\
&= 2^{-n} \sum_{w_1,\ldots,w_n=0}^{1} \left(\sum_{x_1,\ldots,x_n=0}^{1} (-1)^{f(x_1,\ldots,x_n)} (-1)^{w_i x_i} \right) \boldsymbol{e}_{w_1} \otimes \ldots \otimes \boldsymbol{e}_{w_n} \qquad (10)
\end{aligned}$$

Then, we may express the probability of the measurement outcome $y_1, \ldots, y_n = 0$ by

$$\begin{aligned}
\Pr(0\ldots0|\boldsymbol{u}_4\boldsymbol{u}_4^*) &= \langle \boldsymbol{e}_0^{\otimes n}, \boldsymbol{u}_4\boldsymbol{u}_4^*\boldsymbol{e}_0^{\otimes n}\rangle = 2^{-2n} \left(\sum_{x_1,\ldots,x_n=0}^{1} (-1)^{f(x_1,\ldots,x_n)} \right)^2 \\
&= \begin{cases} 0 \text{ if } f \text{ is balanced} \\ 1 \text{ if } f \text{ is constant} \end{cases}
\end{aligned}$$

Thus one may deterministically distinguish between the two cases using only one application of $\boldsymbol{U}_f$.

In the Deutsch-Josza example, the computation consisted only of Hadamards and queries to f. The Hadamards orchestrated a system of superpositions and, later, cancellations of the coefficients for the various basis vectors in such a way that exactly solved the problem, albeit a problem designed specifically to be easy for such a simple quantum computation.

But are superposition and cancellations really the crux of what makes quantum special? A result from Schwarz and Van den Nest challenges this idea.

Theorem 4.3 (Schwarz, Van den Nest)**.** *A wide variety of architectures for quantum computations (including ones that are similar to Shor's algorithm and the Deutsch-Josza algorithm) can be simulated efficiently if the classical outcome probabilities are very "peaky."*

We gain from this result an intuition that quantum computations relying on the superposition and interference between different basis states can only yield an exponential speedup if the final output distribution is not concentrated on too few of the possible outcomes. In other words, there is more to the story than simply interference.

4.3 Entanglement

We refer the reader to the Quora article[4] by current Caltech postdoc Andru Gheorghiu, which gives an excellent perspective on the utility of entanglement as a resource for quantum algorithms.

But is entanglement sufficient for quantum computation? No. In fact, we will now illustrate why most quantum states are actually useless for quantum computation despite having lots of entanglement.

Theorem 4.4 (Gross, Flammia, Eisert)**.** *Most quantum states are useless for quantum computation.*

Proof strategy. For an n qubit computation, we imagine taking a random pure state $\boldsymbol{\rho}_{\text{in}} = \boldsymbol{u}\boldsymbol{u}^*$ on $H^{\otimes n}$ as the input to the quantum computation. Meanwhile, we restrict the quantum computation to be polynomial-size and require the unitary operation $\boldsymbol{V}$ that it implements to have some bounded length L. We suppose that such a quantum computation could allow one to solve an interesting problem, and then we show that you could do the same thing by tossing n (classical) coins, meaning the interesting problem could also be solved by an efficient randomized classical computation.

For a fixed $\boldsymbol{V}$, $\boldsymbol{u}$, and $y_1, \ldots, y_n$, let

$$\begin{aligned} \alpha &= \Pr(y_1 \ldots y_n | \boldsymbol{V}\boldsymbol{u}\boldsymbol{u}^*\boldsymbol{V}^*) = \operatorname{tr}(\boldsymbol{e}_{y_1}^* \otimes \ldots \otimes \boldsymbol{e}_{y_n}^* \boldsymbol{V}\boldsymbol{u}\boldsymbol{u}^*\boldsymbol{V}^* \boldsymbol{e}_{y_1} \otimes \ldots \otimes \boldsymbol{e}_{y_n} \\ &= \operatorname{tr}(\boldsymbol{v}\boldsymbol{v}^*\boldsymbol{u}\boldsymbol{u}^*) \end{aligned}$$

where we have implicitly defined $\boldsymbol{v} = V^*\boldsymbol{e}_{y_1} \otimes \ldots \otimes \boldsymbol{e}_{y_n}$. Now if we consider $\boldsymbol{V}$ fixed but randomize over $\boldsymbol{u}$, we can compute the expectation value of this random number (over $\boldsymbol{u}$):

$$\mathbb{E}[\alpha] = \operatorname{tr}\left(\boldsymbol{v}\boldsymbol{v}^* \int_{\mathbb{S}(H^{\otimes n}} \boldsymbol{u}\boldsymbol{u}^* \mathrm{d}\mu(\boldsymbol{u})\right) = \frac{\operatorname{tr}(\boldsymbol{v}\boldsymbol{v}^*)}{\dim(H^{\otimes n})} = \frac{\langle \boldsymbol{v}, \boldsymbol{v}\rangle}{2^n} = \frac{1}{2^n}.$$

Here, we have used the Haar-integration formula for $k = 1$. Since $\boldsymbol{u} \in \mathbb{S}(H^{\otimes n})$ is a Haar-random vector, we can use higher order Haar integration to establish exponential concentration around this mean value.

Proposition 4.5. *Fix a unit vector $\boldsymbol{v} \in \mathbb{S}(H^{\otimes n})$ and choose $\boldsymbol{u} \in \mathbb{S}(H^{\otimes n})$ uniformly from the Haar measure. Then, there is a constant $c > 0$ such that for any $\tau > 0$*

$$\Pr\left[\left|\operatorname{tr}(\boldsymbol{v}\boldsymbol{v}^*\boldsymbol{u}\boldsymbol{u}^*) - \frac{1}{2^n}\right| \geq \tau\right] \leq \exp\left(-c2^n\tau^2\right).$$

[4] https://www.quora.com/How-is-quantum-entanglement-beneficial-in-quantum-computers

The argument is very similar to the concentration step in Lecture 6 (bound moments using the Haar-integration formula and apply the exponential Markov inequality). Alternatively, one could also use concentration of measure (Levi's Lemma).

Proposition 4.5 implies that the output distribution will be close to flat after randomizing over $\boldsymbol{u}$, and would be simulatable by flipping classical coins.

The only chance to circumvent this argument is to choose $\boldsymbol{u}$ randomly but then choose $\boldsymbol{V}$ wisely in a $\boldsymbol{u}$-dependent way. However, a counting argument shows that we simply don't have enough knobs to turn in $\boldsymbol{V}$ to make a difference. There are 2^n extended standard basis vectors $y_1 \dots y_n$ (measurement), and $n^L|G|^L$ different unitaries $\boldsymbol{V}$ of length L, where $|G|$ is the number of gates in the unviersal gate set. Thus,

$$\Pr_{\boldsymbol{u}}\left[\left(\max_{\boldsymbol{V}}\max_{y_1\dots y_n}\left|\Pr[y_1\dots y_n|\boldsymbol{V}\boldsymbol{u}\boldsymbol{u}^*\boldsymbol{V}^*]-\frac{1}{2^n}\right|\right)\geq\varepsilon\right]\leq n^L|G|^L 2^n\exp(-c2^n\varepsilon^2) \tag{11}$$

where the maximum is taken over all basis states and all unitaries $\boldsymbol{V}$ of length L. For constant ε and $L=\mathrm{poly}(n)$, this probability will be small for sufficiently large n, and thus the output distribution is approximately the uniform distribution and can be simulated by flipping n classical coins. □

Lecture 09: Matrix rank

Scribe: Hsin-Yuan Huang

ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp
April 29, 2019

1 Agenda

This lecture is devoted to recalling and gathering fundamental and useful properties of the *matrix rank*.

1. Definition
2. Computing the matrix rank
3. Uniqueness of decompositions
4. Upper bounds on the maximal rank
5. Typical rank
6. Low-rank approximation and the Eckart-Young Theorem

Lecture 10 is then devoted to generalizing the matrix rank to tensors. We shall see that the tensor rank behaves very differently and is much more challenging to handle.

2 Definition of matrix rank

Let V and W be finite dimensional inner product spaces. Let V^* and W^* denote their dual spaces and fix $\boldsymbol{v}^* \in V^*$ and $\boldsymbol{w} \in W$. Consider the linear map $\boldsymbol{w} \otimes \boldsymbol{v}^* : V \to W$ defined by

$$\boldsymbol{x} \mapsto \boldsymbol{w}\boldsymbol{v}^*\boldsymbol{x} = \boldsymbol{w}\langle \boldsymbol{v}, \boldsymbol{x} \rangle$$

The linear hull of all such elementary maps forms $\mathcal{L}(V, W)$ – the space of linear operators from V to W.

Definition 2.1 (rank-one operator)**.** A linear operator $\boldsymbol{X} \in \mathcal{L}(V, W)$ has *rank one* if it corresponds to an elementary tensor product: $\boldsymbol{X} = \boldsymbol{w}\boldsymbol{v}^*$ for $\boldsymbol{v}^* \in V^*$ ($\boldsymbol{v} \in V$) and $\boldsymbol{w} \in W$.

The matrix rank is a straightforward generalization of this concept.

Definition 2.2 (matrix rank)**.** The rank of an operator $\boldsymbol{X} \in \mathcal{L}(V, W)$ is the smallest number r such that $\boldsymbol{X}$ can be represented as a sum of r rank-one tensor products:

$$\boldsymbol{X} = \sum_{i=1}^{r} \boldsymbol{w}_i \boldsymbol{v}_i^*.$$

We emphasize that this definition may seem non-standard. The following equivalence relation shows that it nonetheless captures the same concept.

Proposition 2.3. *Fix $\boldsymbol{X} \in \mathcal{L}(V, W)$. The following are equivalent:*

1. rank: $\boldsymbol{X}$ *has rank* r
2. column rank: $r = \dim(\mathrm{Im}(\boldsymbol{X}))$
3. row rank: $r = \dim(V) - \dim(\ker(\boldsymbol{X}))$
4. determinantal rank: *all size* $(r+1)$ *minors of any matrix representation have determinant zero.*

Proof sketch. We will show how 1. implies 2. For the remaining equivalences, we refer to standard textbooks and lecture notes on linear algebra. Note that minimiality of the decomposition $\boldsymbol{X} = \sum_{i=1}^{r} \boldsymbol{w}_i \otimes \boldsymbol{v}_i^*$ ensures that $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r\} \subset V$ and $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_r\} \subset W$ are linearly independent. Indeed, suppose that this were not the case: $\boldsymbol{w}_r = \sum_{i=1}^{r-1} \lambda_j \boldsymbol{w}_j$. Then,

$$\boldsymbol{X} = \sum_{i=1}^{r} \boldsymbol{w}_i \otimes \boldsymbol{v}_i^* = \sum_{i=1}^{r-1} \boldsymbol{w}_i \otimes \boldsymbol{v}_i^* + \sum_{i=1}^{r-1} \lambda_i \boldsymbol{w}_i \otimes \boldsymbol{v}_r^* = \sum_{i=1}^{r-1} \boldsymbol{w}_i \otimes (\boldsymbol{v}_i + \lambda_i \boldsymbol{v}_r)^*$$

which would contradict minimality. Next note that

$$\boldsymbol{X}\boldsymbol{x} = \sum_{i=1}^{r} \boldsymbol{w}_i \langle \boldsymbol{v}_i, \boldsymbol{x} \rangle = \sum_{i=1}^{r} \xi_i \boldsymbol{w}_i \in \mathrm{span}\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_r\}.$$

Choosing different inputs $\boldsymbol{x}$ leads to different coefficients ξ_i and we can reach all of $\mathrm{span}\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_r\}$. Linear independence ensures that this is a r-dimensional subspace of W. □

3 Computing the matrix rank

There is an one-to-one correspondence between linear operators $\boldsymbol{X} \in \mathcal{L}(V, W)$ and $\dim(W) \times \dim(V)$ matrices.

Fact 3.1. *Computing the matrix rank is easy!*

Several efficient algorithms exist, each with their own advantages and drawbacks. All of them requires $\Theta(d^3)$ arithmatic operations to compute the matrix rank when $\dim(V) = \dim(W) = d$. For example,

1. Gaussian elimination: This can be done analytically but floating point algorithms can become unreliable.
2. Singular value decomposition: It is stable and can provide the minimum rank decomposition but slightly more expensive.
3. QR decomposition with column pivoting: Less expensive than SVD and more robust than Gaussian elimination.

4 Minimal rank decompositions and uniqueness

Fix a linear operator $\boldsymbol{X} \in \mathcal{L}(V, W)$. The singular value decomposition readily provides a minimal rank decomposition.

Theorem 4.1 (Singular value decomposition)**.** *The singular value decomposition (SVD) decomposes any matrix $\boldsymbol{X} \in \mathcal{L}(V, W)$ into a triple of structured matrices:*

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_r]\mathrm{diag}(\sigma_1, \dots, \sigma_r)[\boldsymbol{v}_1, \dots, \boldsymbol{v}_r]^* = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^*.$$

The matrices $\boldsymbol{U} \in \mathcal{L}(W, W)$ and $\boldsymbol{V} \in \mathcal{L}(V, V)$ are linear isometries (equivalently: $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_r\} \subset W$ and $\{\boldsymbol{v}_1, \dots, \boldsymbol{v}_r\} \subset V$ are orthonormal sets of vectors) and $\sigma_1, \dots, \sigma_r > 0$ are strictly positive numbers (singular values). Computing the SVD requires $\mathcal{O}(d^3)$ arithmetic operations.

The SVD is the work-horse of numerical linear algebra. For our purposes it provides two types of highly relevant information:

1. The matrix rank r is just the number of non-zero singular values.
2. A minimal rank decomposition with rich structure:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^* \simeq \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \otimes \boldsymbol{v}_i^*.$$

Fact 4.2. *Computing minimal rank decompositions of matrices is easy!*

Next we want to address another fundamental question: are these minimal rank decomposition unique? Firstly, note that each matrix factorization carries two kinds of trivial ambiguities:

1. *Permutation of factors pairs:* Choose $\pi \in \mathcal{S}_r$. Then, permuting factor pairs $(\boldsymbol{u}_{\pi(i)}, \boldsymbol{v}_{\pi(i)})$ does not change the decomposition:

$$\sum_{i=1}^{r} \boldsymbol{u}_{\pi(i)} \otimes \boldsymbol{v}_{\pi(i)}^* = \sum_{i=1}^{r} \boldsymbol{u}_i \otimes \boldsymbol{v}_i^* = \boldsymbol{X}.$$

2. *Scaling of factor pairs:* Fix non-zero $\alpha_1, \dots, \alpha_r \in \mathbb{F}$. Then, scaling each factor pair $(\boldsymbol{u}_i, \boldsymbol{v}_i) \mapsto \left(\alpha_i^{-1}\boldsymbol{u}_i, \bar{\alpha}\boldsymbol{v}_i\right)$ does not change the decomposition:

$$\sum_{i=1}^{r} \alpha_i^{-1} \boldsymbol{u}_i \otimes (\bar{\alpha}_i \boldsymbol{v}_i)^* = \sum_{i=1}^{r} \frac{\alpha_i}{\alpha_i} \boldsymbol{u}_i \otimes \boldsymbol{v}_i^*.$$

Such symmetries are intrinsic to any factorization and cannot be avoided.

Definition 4.3. We call a minimal rank decomposition of operators $\boldsymbol{X} \in \mathcal{L}(V, W) \simeq W \otimes V^*$ *unique* if it is uniquely determined up to trivial symmetries (permutation and scaling).

Proposition 4.4. *Minimal rank factorizations of operators $\boldsymbol{X} \in \mathcal{L}(V,W)$ are never unique.*

Proof. Fix $\boldsymbol{X} \in \mathcal{L}(V,W)$ and apply an SVD: $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$. Choosing $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r \in W$ to be the columns of $\boldsymbol{U}$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r \in V$ to be the rows of $\boldsymbol{V}$ gives rise to a minimal rank decomposition. Use $\mathbf{I} = \sum_{i=1}^{r} \boldsymbol{e}_i \boldsymbol{e}_i^*$ t conclude

$$\sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^* = \sum_{i=1}^{r} \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{e}_i \boldsymbol{e}_i^* \boldsymbol{V}^* = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* = \boldsymbol{X}.$$

Evidently, this is a minimal rank decomposition. However, we could also have included in additional invertible map $\boldsymbol{R}$:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{R}\boldsymbol{R}^{-1}\boldsymbol{V}^* = \sum_{i=1}^{r} \tilde{\boldsymbol{v}}_i \tilde{\boldsymbol{w}}_i^*.$$

This is also a valid minimal rank decomposition. Unless $\boldsymbol{R}$ is a signed permutation matrix or a diagonal scaling matrix, this alternative decomposition is not related to the orginal one via a trivial symmetry operation. □

5 Upper bounds on the maximal rank

Theorem 5.1. *The rank of any $\boldsymbol{X} \in \mathcal{L}(V,W)$ obeys $r \leq \min\{\dim(V), \dim(W)\}$.*

Proof. Use the equivalent definitions for rank from Proposition 2.3. The column-rank definition readily implies

$$r = \dim(\mathrm{Im}(\boldsymbol{X})) \leq \dim(V),$$

while a row-rank definition ensures

$$r = \dim(\mathrm{Im}(\boldsymbol{X}^*)) \leq \dim(W).$$

Both bounds are necessarily valid and we can without loss choose the minimum of both to make the bound as tight as possible. □

6 Typical rank

Typical rank addresses the following question: what matrix rank do we expect to see for generic or typical matrices? Such typical statements require endowing the space of all operators $\mathcal{L}(V,W)$ with a "fair" measure. One way to achieve this is to consider random matrices with independent entries that follow a continuous distribution. Standard Gaussian matrices meet all these desiderata:

$$\boldsymbol{X} = \begin{pmatrix} g_{11} & \cdots & g_{1n} \\ \vdots & \ddots & \vdots \\ g_{n1} & \cdots & g_{nn} \end{pmatrix},$$

where each g_{ij} is an independent instance of a standard Gaussian random variable: $g_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ for real-valued matrices and $g_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 2^{-1/2}) + i\mathcal{N}(0, 2^{-1/2})$ for complex-valued matrices. Such Gaussian matrices correspond to matrix representations of generic operators. Rotational invariance moreover ensures that the choice of basis is irrelevant.

Fact 6.1. *A typical/generic matrix – e.g. a matrix with standard Gaussian entries – saturates the rank inequality $r \leq \min\{\dim(V), \dim(W)\}$ with probability one.*

This fundamental fact from random matrix theory holds true regardless whether we work with real-valued or complex matrices. It highlights that the matrix rank bound is tight in a strong sense: it is saturated for almost all matrices.

7 Low-rank approximations and the Eckart-Young theorem

Theorem 7.1 (Eckart-Young Mirski theorem)**.** *Let $\boldsymbol{X} \in \mathcal{L}(V, W)$ be a matrix with SVD $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$. Then, the best rank-k approximation is the truncated SVD:*

$$\boldsymbol{X}_k = \boldsymbol{U}\mathrm{diag}(\sigma_1, \ldots, \sigma_r, 0, \ldots, 0)\boldsymbol{V}^*.$$

It achieves

$$\|\boldsymbol{X}_r - \boldsymbol{X}\|_F^2 = \sum_{i=k+1}^{r} \sigma_i.$$

Remark 7.2. The original Eckart-Young theorem proves optimality of the truncated SDP for approximation in operator norm.

Proof. We try to find the solution to the following problem:

$$\begin{aligned} \text{minimize} \quad & \|\boldsymbol{Z} - \boldsymbol{X}\|_F^2 \\ \text{subject to} \quad & \mathrm{rank}(\boldsymbol{Z}) = k. \end{aligned}$$

First, note that

$$\begin{aligned} \|\boldsymbol{Z} - \boldsymbol{X}\|_F^2 =& \mathrm{tr}(\boldsymbol{Z}\boldsymbol{Z}^*) - \mathrm{tr}(\boldsymbol{X}\boldsymbol{Z}^*) - \mathrm{tr}(\boldsymbol{Z}^*\boldsymbol{X}) + \mathrm{tr}(\boldsymbol{X}\boldsymbol{X}^*) \\ =& \|\boldsymbol{Z}\|_F^2 + \|\boldsymbol{X}\|_2^2 - \mathrm{tr}(\boldsymbol{X}\boldsymbol{Z}^*) - \mathrm{tr}(\boldsymbol{X}^*\boldsymbol{Z}). \end{aligned}$$

We need to make these trace-inner products as large as possible. Von-Neumann's trace inequality (which uses Birkhoff-von Neumann) states that

$$|\mathrm{tr}(\boldsymbol{X}\boldsymbol{Z}^*)| \leq \sum_{i=1}^{r} \sigma_i(\boldsymbol{X})\sigma_i(\boldsymbol{Z})$$

with equality if and only if $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$ and $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^*$. This tells us that the SVD provides us with the "right basis rotations: $\boldsymbol{Z} = \sum_{i=1}^{r} \lambda_i \boldsymbol{u}_i \boldsymbol{v}_i^*$. But at most r singular values can be non-zero. Therefore,

$$\|\boldsymbol{Z} - \boldsymbol{X}\|_F^2 = \sum_{i=1}^{r} \sigma_i^2 + \sum_{i=1}^{r} \lambda_i^2 - 2\sum_{i=1}^{r} \lambda_i \sigma_i = \sum_{i=1}^{r} (\sigma_i - \lambda_i)^2.$$

This expression is minimized if we set $\lambda_1 = \sigma_1, \ldots, \lambda_k = \sigma_k, \lambda_{k+1} = \cdots = \lambda_r = 0$. □

This theorem has profound implications for data processing.

- *Greed is good:* The theorem justifies a greedy approach to matrix factorization: find the largest rank-one factor, peel it off and repeat. You can expect progress by going from rank-k to rank-$(k+1)$.
- *Dimension reduction:* The Eckart-Young-Mirski theorem provides the foundation for dimension reduction (such as principal component analysis) in data analysis – an important subroutine in machine learning. The idea of low-rank approximation is also used extensively in modern recommendation system such as those powering YouTube and Netflix.
- *Noise resilience:* If you wiggle the original matrix a little bit, the best rank-k approximation changes slightly but is still a good approximation to the original matrix. Hence it is robust against (small) noise corruption.

Fact 7.3. *The Eckart-Young theorem justifies greedy approaches to matrix factorization: find the largest rank-one factor, peel it off and iterate. Increasing the rank in an approximation gets you closer to the true underlying matrix.*

Lecture 10: Tensor rank

Scribe: Richard Kueng

ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp
May 01, 2019

1 Agenda

Last lecture was devoted to studying several desirable features of matrix rank, as well as their implications. Today, we will extend this rank discussion to *tensors*. We shall see that, as a rule, tensor rank behaves very differently from its matrix-counterpart and is considerably more challenging to handle.

1. Definition of tensor rank
2. Computation of tensor rank
3. Uniqueness of minimal rank decompositions
4. Upper bounds on the maximal rank
5. Tensor rank depends on underlying field ($\mathbb{C}$ vs. $\mathbb{R}$)
6. Typical rank
7. Low-rank approximations and border rank
8. Examples: the standard inner product and the Hadamard product as tensors

We will mostly restrict our attention to tensors of order three. Generalizations to tensors of higher order are straightforward.

2 Recapitulation: Matrix rank

The rank of an operator $\boldsymbol{X} \in \mathcal{L}(V, W)$ is the smallest number of rank-one tensors required to represent $\boldsymbol{X}$:

$$\boldsymbol{X} = \sum_{i=1}^{r} \boldsymbol{w}_i \otimes \boldsymbol{v}_i^*.$$

It has many interesting and desirable properties:

1. *Computation:* several efficient algorithms exist for computing the matrix rank.
2. *Identifying minimal rank decompositions and uniqueness:* The SVD provides an efficient way to compute minimal rank decompositions. These are *never* unique
3. *Upper bound:* $r \leq \min\{\dim(V), \dim(W)\}$
4. *typical rank:* a generic matrix saturates the rank bound with probability one (regardless whether we work over $\mathbb{R}$, or $\mathbb{C}$)
5. *Low rank approximations:* The Eckart-Young-Mirsky states that the best rank-k approximation is a truncated SVD. Increasing k can only increase approximation accuracy.

3 Definition of tensor rank

Let A, B, C be inner product spaces with dimensions a, b and c. We also assume

$$a \geq b \geq c.$$

Recall that the tensor product space $A \otimes B \otimes C$ is the linear hull of all elementary tensor products:

$$\boldsymbol{t} = \boldsymbol{a} \otimes \boldsymbol{b} \otimes \boldsymbol{c} \quad \boldsymbol{a} \in A,\ \boldsymbol{b} \in B,\ \boldsymbol{c} \in C.$$

Definition 3.1. A tensor $\boldsymbol{t} \in A \otimes B \otimes C$ has *rank-one* if it corresponds to an elementary tensor product.

Definition 3.2. The *rank* of a tensor $\boldsymbol{t} \in A \otimes B \otimes C$ is the smallest number r such that $\boldsymbol{t}$ can be represented as a sum of r rank-one tensors:

$$\boldsymbol{t} = \sum_{i=1}^{r} \boldsymbol{a}_i \otimes \boldsymbol{b}_i \otimes \boldsymbol{c}_i.$$

4 Computing the tensor rank

In contrast to the matrix rank – where we can choose between several different efficient numerical algorithms to compute it – computing the tensor rank is in general very challenging.

Fact 4.1 (Hastad, 1990)**.** *Computing the tensor rank of $\boldsymbol{t} \in A \otimes B \otimes C$ is difficult. It is NP-complete for any finite field and NP-hard over the rational numbers.*

The proof follows from a standard reduction of 3SAT which is known to be NP complete.

5 Uniqueness of minimal rank decompositions

Any factorization into sums of rank-one tensors is necessarily accompanied by trivial ambiguities. Let

$$\boldsymbol{t} = \sum_{i=1}^{r} \boldsymbol{a}_i \otimes \boldsymbol{b}_i \otimes \boldsymbol{c}_i$$

be a minimal rank decomposition of $\boldsymbol{t} \in A \otimes B \otimes C$. Then, this decomposition is invariant under *permuting factor triples:* $(\boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{c}_i) \mapsto (\boldsymbol{a}_{\pi(i)}, \boldsymbol{b}_{\pi(i)}, \boldsymbol{c}_{\pi(i)})$. Here, $\pi \in \mathcal{S}_r$ can be any permutation. Indeed,

$$\sum_{i=1}^{r} \boldsymbol{a}_{\pi(i)} \otimes \boldsymbol{b}_{\pi(i)} \otimes \boldsymbol{c}_{\pi(i)} = \sum_{i=1}^{r} \boldsymbol{a}_i \otimes \boldsymbol{b}_i \otimes \boldsymbol{c}_i = \boldsymbol{t}.$$

Similarly, *scaling* of individual factors also leaves the final decomposition invariant. Choose $\alpha_i, \beta_i \neq 0$. Then, $(\boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{c}_i) \mapsto (\alpha_i \boldsymbol{a}_i, \beta_i \boldsymbol{b}_i, (\alpha\beta)^{-1} \boldsymbol{c}_i)$ provides a a rank-r decomposition of $\boldsymbol{t}$. Permutation and scaling of factors are trivial ambiguities that cannot be avoided. A minimal rank decomposition is *unique up to trivial ambiguities*

if it is unique up to scaling and permutation of factors. Recall that minimal rank decompositions of operators (matrices) are never unique – unless one imposes strong additional assumptions and constraints. The situation for tensors of order $k \geq 3$ is very different. A seminal result by Kruskal ensures uniqueness of minimal rank decompositions under much weaker conditions.

To state this result, we are going to introduce the following useful notation by Kolda. Let $\boldsymbol{t} = \sum_{i=1}^{r} \boldsymbol{a}_i \otimes \boldsymbol{b}_i \otimes \boldsymbol{c}_i$ be a decomposition of $\boldsymbol{t} \in A \otimes B \otimes C$ into rank-one tensors. Define the three *factor matrices*

$$\boldsymbol{A} = [\boldsymbol{a}_1 \cdots \boldsymbol{a}_r],\ \boldsymbol{B} = [\boldsymbol{b}_1, \cdots \boldsymbol{b}_r],\ \boldsymbol{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_r].$$

and set

$$\boldsymbol{t} = \sum_{i=1}^{r} \boldsymbol{a}_i \otimes \boldsymbol{b}_i \otimes \boldsymbol{c}_i =: [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!].$$

It is easy to keep track of permutation ambiguities in this decomposition. Let $\boldsymbol{\Pi} \in \mathbb{R}^{r \times r}$ be a permutation matrix. Then,

$$[\![\boldsymbol{A}\boldsymbol{\Pi}, \boldsymbol{B}\boldsymbol{\Pi}, \boldsymbol{C}\boldsymbol{\Pi}]\!] = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!].$$

Scaling ambiguities act in a similar fashion.

Definition 5.1. The *k-rank* of a matrix $\boldsymbol{A}$ is the largest number $k_{\boldsymbol{A}}$ such that any k columns are linearly independent.

This concept is closely related to the spark of a matrix (the smallest number k such that there exists a set of k linearly dependent columns).

Theorem 5.2 (Kruskal, 1977)**.** *Suppose that $\boldsymbol{t} \in A \otimes B \otimes C$ admits a decomposition*

$$\boldsymbol{t} = \sum_{i=1}^{r} \boldsymbol{a}_i \otimes \boldsymbol{b}_i \otimes \boldsymbol{c}_i = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!].$$

Suppose that

$$r \leq \frac{1}{2}(k_{\boldsymbol{A}} + k_{\boldsymbol{B}} + k_{\boldsymbol{C}}) - 1.$$

Then $\boldsymbol{t}$ has rank-r and this decomposition is unique up to trivial symmetries (permutations and scaling).

6 Upper bounds on the maximal tensor rank

Upper bounds on the maximal rank of tensors exist, but they are much weaker than in the matrix case.

Theorem 6.1. *Consider $A \otimes B \otimes C$ with dimensions a, b and c. Then, the rank of any $\boldsymbol{t} \in A \otimes B \otimes C$ obeys*

$$r(\boldsymbol{t}) \leq \min\{ab, ac, bc\}.$$

The proof of this claim is instructive, because it follows a basic line of thoughts. We know very little about tensors, but a lot about matrices. Therefore, it is often beneficial to convert tensor problems into matrix problems. The following technical result follows from such a reduction argument.

Lemma 6.2. *Let $\boldsymbol{t} \in A \otimes B \otimes C$. Then, its rank r equals the number of rank-one matrices that are required to span (a space containing) all possible marginalizations*

$$\boldsymbol{t}(A^*) := \operatorname{span}\left\{ (\boldsymbol{a}^* \otimes \mathbf{I} \otimes \mathbf{I})\boldsymbol{t} = \sum_{i=1}^{r} \alpha(\boldsymbol{a}_i) \otimes \boldsymbol{b}_i \otimes \boldsymbol{c}_i : \ \boldsymbol{a}^* \in A^* \right\} \subset B \otimes C \simeq \mathcal{L}(C, B).$$

Proof. Suppose that $\boldsymbol{t}$ has rank r and express it as $\boldsymbol{t} = \sum_{i=1}^{r} \boldsymbol{a}_i \otimes \boldsymbol{b}_i \otimes \boldsymbol{c}_i$ (Note that in contrast to matrices, the vectors $\boldsymbol{a}_i$, $\boldsymbol{b}_i$ and $\boldsymbol{c}_i$ need not be linearly independent). Therefore,

$$\boldsymbol{t}(A^*) \subset \{\boldsymbol{b}_1 \otimes \boldsymbol{c}_1, \dots, \boldsymbol{b}_r \otimes \boldsymbol{c}_r\}$$

is spanned by at most r rank-one matrices. Conversely, suppose that $\boldsymbol{t}(A^*)$ is spanned by r rank-one matrices $\boldsymbol{b}_1 \otimes \boldsymbol{c}_1, \dots, \boldsymbol{b}_r \otimes \boldsymbol{c}_r$. Choose a (orthonormal) basis $\boldsymbol{a}_1^*, \dots, \boldsymbol{a}_r^*$ of A^*. Then,

$$\boldsymbol{t}(\boldsymbol{a}_k^*) = \sum_{i=1}^{r} x_{i,k} \boldsymbol{b}_i \otimes \boldsymbol{c}_i.$$

Next, let $\boldsymbol{a}_1, \dots, \boldsymbol{a}_r$ be the dual vectors associated with $\boldsymbol{a}_1^*, \dots, \boldsymbol{a}_1^*$ (column vs. row vectors). Then,

$$\boldsymbol{t} = \sum_{k,i} x_{i,k} \boldsymbol{a}_k \otimes \boldsymbol{b}_i \otimes \boldsymbol{c}_i = \sum_{i=1}^{r} \underbrace{\left(\sum_k x_{ik} \boldsymbol{a}_k \right)}_{\tilde{\boldsymbol{a}}_i} \otimes \boldsymbol{b}_i \otimes \boldsymbol{c}_i.$$

This is a valid decomposition into exactly r rank-one factors. Therefore, the tensor rank can be at most r. □

Proof of Theorem 6.1. The space $\mathcal{L}(C, B)$ has dimension $bc = \dim(C)\dim(B)$. This is also the maximum number of matrices that are required to span this operator space. With the previous Lemma, we conclude

$$r \leq \dim(\boldsymbol{t}(A^*)) = \dim(\mathcal{L}(C, B)) = bc$$

Permuting tensor factors also establishes ab and ac as upper bounds. □

7 The tensor rank depends on the underlying field

Choose real-valued vector spaces $V = \mathbb{R}^m$ and $W = \mathbb{R}^n$. Then, we may represent $\boldsymbol{X} \in \mathcal{L}(V, W)$ as a real-valued matrix. Alternatively, we could embed $V \subset \mathbb{C}^m$, $W \subset \mathbb{C}^m$ and extend $\boldsymbol{X}$ linearly to $\mathcal{L}(\tilde{V}, \tilde{W})$. The matrix rank does not care: it is the same in both cases.

This is not the case for tensors, as the following example from Kruskal shows. Fix $\dim(A) = \dim(B) = \dim(C) = 2$.

$$\boldsymbol{t} = \boldsymbol{e}_1 \otimes \boldsymbol{e}_1 \otimes \boldsymbol{e}_1 + \boldsymbol{e}_1 \otimes \boldsymbol{e}_2 \otimes \boldsymbol{e}_2 - \boldsymbol{e}_2 \otimes \boldsymbol{e}_2 \otimes \boldsymbol{e}_1 - \boldsymbol{e}_2 \otimes \boldsymbol{e}_1 \otimes \boldsymbol{e}_2.$$

Over $\mathbb{R}$, this tensor has rank three:

$$\boldsymbol{t} = \left[\!\left[\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 0 \\ -1 & 1 & 1 \end{pmatrix} \right]\!\right]$$

However, over $\mathbb{C}$, we can find a rank-2 decomposition:

$$\boldsymbol{t} = \left[\!\left[\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -i & i \end{pmatrix}, \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix} \right]\!\right].$$

8 Typical rank

The matrix rank bound $r \leq \min\{\dim(V), \dim(W)\}$ is useful. A typical matrix is going to saturate it.

For tensors, the situation is quite different. A simple parameter counting argument suggests the following typical behavior. Consider the tensor product space $A_1 \otimes \cdots \otimes A_k$ with dimensions $a_1, \ldots, a_n$. The number of degrees of freedom for a rank-r tensor is

$$r(\alpha_1 + \cdots + \alpha_k) - r(k-1) = r(a_1 + \cdots a_k - (k-1)),$$

while the total degrees of freedom are

$$\dim(A_1 \otimes \cdots \otimes A_k) = a_1 \times \cdots \times a_k.$$

We expect that the typical rank occurs precisely at the threshold where both numbers become equal:

$$r = \lceil \frac{a_1 \cdots a_k}{a_1 + \cdots + a_k - k + 1} \rceil.$$

This simple counting argument is approximately correct. Here are some rigorous results that provide some insights.

Theorem 8.1. *1. (Strassen) The typical rank of an element of $\mathbb{C}^3 \otimes \mathbb{C}^3 \otimes \mathbb{C}^3$ is five (not the expected four).*

2. (Strassen-Lickteig) For all $d \neq 3$, the typical rank of an element of $\mathbb{C}^d \otimes \mathbb{C}^d \otimes \mathbb{C}^d$ is $\lceil \frac{d^3}{3d-2} \rceil$ (as expected).

3. The typical rank of an element of $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^3$ and $\mathbb{C}^2 \otimes \mathbb{C}^3 \otimes \mathbb{C}^3$ is three (as expected).

For tensor products with equal dimension ($a_1 = \ldots a_k = a$), we have

$$r = \lceil \frac{a^k}{k(a-1)+1} \rceil \sim \frac{a^{k-1}}{k} < a^{k-1}.$$

The typical rank *does not* saturate the maximal rank bound!

Finally, the typical rank also depends on the underlying field. Over $\mathbb{C}$ the typical rank is unique, but over $\mathbb{R}$ this need not be the case. Tensors in $\mathbb{R}^2 \otimes \mathbb{R}^2 \otimes \mathbb{R}^2$ have typical rank two and three over $\mathbb{R}$. Monte Carlo experiments reveal that rank-two tensors fill about 79% of the space, while rank-three tensors fill the remaining 21%. Rank-one tensors are possible, but occur with probability zero.

9 Low-rank approximations and border rank

Recall that the best rank-k approximation of a matrix is given by the truncated SVD (Eckart-Young Theorem). This supports greedy, iterative strategies to approximate a matrix.

Similar approaches seem like a promising avenue to generalize to tensors. Suppose $\boldsymbol{t} \in A \otimes B \otimes C$ admits a decomposition

$$\boldsymbol{t} = \sum_{i=1}^{r} \lambda_i \boldsymbol{a}_i \otimes \boldsymbol{b}_i \otimes \boldsymbol{c}_i,$$

where $\langle \boldsymbol{a}_i, \boldsymbol{a}_i \rangle = \langle \boldsymbol{b}_i, \boldsymbol{b}_i \rangle = \langle \boldsymbol{c}_i, \boldsymbol{c}_i \rangle = 1$ for all $1 \leq i \leq r$. Assume moreover, that the weights are arranged in non-increasing order: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r$. Apparent parallels to the SVD suggest the following iterative approach for approximating $\boldsymbol{t}$:

1. identify the largest contributing rank-one factor $\lambda_1 \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 \otimes \boldsymbol{c}_1$ (somehow).
2. Subtract its contribution and iterate.

We expect that this greedy method provides us with better and better approximations of $\boldsymbol{t}$.

Unfortunately, this intuition is flawed. Tensors are much more complicated than matrices. We illustrate this with the following simple example:

$$\boldsymbol{t} = \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 \otimes \boldsymbol{c}_2 + \boldsymbol{a}_1 \otimes \boldsymbol{b}_2 \otimes \boldsymbol{c}_1 + \boldsymbol{a}_2 \otimes \boldsymbol{b}_1 \otimes \boldsymbol{c}_1,$$

where $\boldsymbol{a}_1, \boldsymbol{a}_2 \in A$, $\boldsymbol{b}_1, \boldsymbol{b}_2 \in B$ and $\boldsymbol{c}_1, \boldsymbol{c}_2 \in C$ are linearly independent each. Evidently, this tensor has rank three. It can, however, be approximated to arbitrary accuracy by a rank-two tensor:

$$\boldsymbol{s}(\varepsilon) = \frac{1}{\varepsilon}((\boldsymbol{a}_1 + \varepsilon \boldsymbol{a}_2) \otimes (\boldsymbol{b}_1 + \varepsilon \boldsymbol{b}_2) \otimes (\boldsymbol{c}_1 + \varepsilon \boldsymbol{c}_2) - \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 \otimes \boldsymbol{c}_1).$$

More precisely, let $\|\cdot\|$ be the Euclidean norm induces by the extended standard inner product on $A \otimes B \otimes C$. Then,

$$\|\boldsymbol{t} - \boldsymbol{s}(\varepsilon)\| = \varepsilon \|\boldsymbol{a}_2 \otimes \boldsymbol{b}_2 \otimes \boldsymbol{c}_1 + \boldsymbol{a}_2 \otimes \boldsymbol{b}_1 \otimes \boldsymbol{c}_2 + \boldsymbol{a}_1 \otimes \boldsymbol{b}_2 \otimes \boldsymbol{c}_2 + \varepsilon \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 \otimes \boldsymbol{c}_2\|$$

which can be made arbitrarily small. Many different simple examples for this behavior are known. These examples motivate the following definition:

Definition 9.1 (border rank)**.** A tensor $\boldsymbol{t}$ has border rank $\underline{r}(\boldsymbol{t}) = r$ if it is a limit of tensors of rank r, but not a limit of tensors with rank $s < r$.

There is an elegant geometric interpretation of this behavior. Intuitively, $\boldsymbol{s}(\varepsilon)$ is a point on the line spanned by rank-one tensors *inside* the space of rank-two tensors. Taking the limit results in a point in the tangent space of $\boldsymbol{a}_1 \otimes \boldsymbol{b}_1 \otimes \boldsymbol{c}_1$. This point on the tangent line is itself not contained in the set of rank-two-tensors, but infinitesimally close.

The property of having border rank at most r is an algebraic property – similar to the matrix rank-characterization via vanishing minors. As such it can in principle be precisely tested by checking whether certain polynomial equations vanish identically. While this is not efficient by any means, it provides at least a strategy that can be executed vor very small tensor products.

Remark 9.2 (Relation between rank and border rank). Very little is known about the relation between rank and border rank. For $\boldsymbol{t} \in A_1 \otimes \cdots A_k$ with border rank 2, the actual rank can be anywhere between 2 and n. More is known for symmetric tensors $\boldsymbol{t} \in \bigvee^k(A)$, because they are closely related to homogeneous polynomials.

10 Examples

10.1 The standard inner product as a order-2 tensor (matrix)

Fix $V = \mathbb{F}^d$ and define the standard inner product:

$$V^* \times V \to \mathbb{F} \quad \langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^{d} \bar{x}_i y_i.$$

This is a bilinear form. The space of bilinear forms is closely related to the tensor product $V \otimes V$. In fact, we defined $V \otimes V$ to be the dual space of the space of all bilinear forms. In particular,

$$\boldsymbol{x} \otimes \boldsymbol{y} : \mathrm{Bil}(V, V) \to \mathbb{F} \quad B \mapsto B(\boldsymbol{x}, \boldsymbol{y}).$$

A moment of thought reveals that the tensor associated with the inner product is

$$\sum_{i=1}^{d} \boldsymbol{e}_i \otimes \boldsymbol{e}_i^* \in V \otimes V^*,$$

which corresponds to the identity operator $\mathbf{I} \in \mathcal{L}(V, V)$. We know that this operator has matrix rank d and minimal decompositions are never unique:

$$\mathbf{I} = \sum_{i=1}^{r} (\boldsymbol{U}\boldsymbol{e}_i) \otimes (\boldsymbol{U}\boldsymbol{e}_i)^* \quad \text{for any unitary} \quad \boldsymbol{U}.$$

This is just a fancy way of saying that the standard inner product is basis-independent.

Next, we turn to the question of computing the inner product:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^* \mathbf{I} \boldsymbol{y} = \sum_{i=1}^{d} \boldsymbol{x}^* \boldsymbol{u}_i \boldsymbol{u}_i^* \boldsymbol{y} = \sum_{i=1}^{d} \langle \boldsymbol{x}, \boldsymbol{u}_i \rangle \langle \boldsymbol{u}_i, \boldsymbol{x} \rangle.$$

Evaluating this expression requires computing $2d$ different inner products in general. A smart choice of basis substantially reduces the cost for individual scalar product evaluations. If we opt for the standard basis, both $\langle \boldsymbol{x}, \boldsymbol{e}_i \rangle$ and $\langle \boldsymbol{e}_i, \boldsymbol{y} \rangle$ are very cheap to evaluate. The total arithmetic cost becomes $\mathcal{O}(n)$.

We could also use a very bad – e.g. a generic ONB. In this case the arithmetic cost could blow up to $\mathcal{O}(n^2)$.

10.2 The Hadamard product as an order 3-tensor

Endow $V = \mathbb{F}^d$ with the standard basis $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$. The Hadamard product is typically defined as

$$V \times V \to V : \boldsymbol{x} \odot \boldsymbol{y} = \sum_{i=1}^{d} \boldsymbol{e}_i \langle \boldsymbol{e}_i, \boldsymbol{x} \rangle \langle \boldsymbol{e}_i, \boldsymbol{y} \rangle.$$

We can view this as a tensor in $V \otimes V^* \otimes V^*$:

$$\boldsymbol{h} = \sum_{i=1}^{d} \boldsymbol{e}_i \otimes \boldsymbol{e}_i^* \otimes \boldsymbol{e}_i^*.$$

Modulo vector space dualities ($V \simeq V^*$), this tensor looks like the standard extension of the identity to order 3:

$$\boldsymbol{h} = \sum_{i=1}^{d} \boldsymbol{e}_i \otimes \boldsymbol{e}_i \otimes \boldsymbol{e}_i = [\![\mathbf{I}, \mathbf{I}, \mathbf{I}]\!].$$

Kruskal's theorem on uniqueness shows that this order-three tensor is essentially unique.

Corollary 10.1. *The Hadamard tensor $\boldsymbol{h} \in V^{\otimes 3}$ has rank $r = d$ and is unique up to trivial ambiguities (permutations and scaling), provided that $d \geq 2$.*

Proof. The decomposition (10.2) has rank $r = d$ and the individual factor matrices obey $k_{\mathbf{I}} = d$ each. Correctness of the matrix rank and uniqueness then follow from checking Kruskal's condition:

$$d = r \leq \frac{1}{2}(k_{\mathbf{I}} + k_{\mathbf{I}} + k_{\mathbf{I}}) - 1 = \frac{3}{2}d - 1.$$

The resulting inequality is true provided that $d \geq 2$. □

The uniqueness requirement $d \geq 2$ is perhaps tautological, but worth noting. For $d = 1$, Hadamard and standard inner product coincide. Closer to home, we conclude the following well-known fact

Fact 10.2. *In contrast to standard-inner, wedge and tensor products, the Hadamard product is basis-dependent.*

The Hadamard tensor also does not saturate the upper bound on tensor rank:

$$d = r(\boldsymbol{h}) \ll d^2.$$

This has meaningful consequences for the computational cost. Similar to the matrix case, the cost of evaluating tensors is strongly connected to the rank. A rank-r tensor will require at least r individual arithmetic operations.

For the Hadamard product this cost $r = d$ is tight. We can use desirable properties of the standard basis to compute $\langle \boldsymbol{e}_i, \boldsymbol{x} \rangle$, $\langle \boldsymbol{e}_i, \boldsymbol{y} \rangle$ and their product at unit cost.

Fact 10.3. *The arithmetic cost of computing the Hadamard product is proportional to* $\operatorname{rank}(\boldsymbol{h}) = d$.

Lecture 11: Strassen's algorithm for matrix multiplication

Scribe: Richard Kueng

ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp
April 29, 2019

1 Agenda

1. Recapitulation: matrix multiplication
2. Strassen's matrix multiplication algorithm
3. The arithmetic complexity model and rigorous improvements for the complexity of matrix multiplication

2 Recapitulation: matrix multiplication

Let $\boldsymbol{Y}, \boldsymbol{Z} \in \mathbb{R}^{n\times n}$. We use Einstein notation to label the individual matrix entries: $X^i_{\ j}$ denotes the entry in the i-th row (upper-case index) and the j-th column (lower-case index). The matrix product $\boldsymbol{Z} = \boldsymbol{Y}\boldsymbol{Z} \in \mathbb{R}^{n\times n}$ of two square matrices is defined component-wise:

$$\begin{pmatrix} X^1_{\ 1} & \cdots & X^1_{\ n} \\ \vdots & \ddots & \vdots \\ X^n_{\ 1} & \cdots & X^n_{\ n} \end{pmatrix} \begin{pmatrix} Y^1_{\ 1} & \cdots & Y^1_{\ n} \\ \vdots & \ddots & \vdots \\ Y^n_{\ 1} & \cdots & Y^n_{\ n} \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^n X^1_{\ k} Y^k_{\ 1} & \cdots & \sum_{k=1}^n X^1_{\ k} Y^k_{\ n} \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^n X^n_{\ k} Y^k_{\ 1} & \cdots & \sum_{k=1}^n X^n_{\ k} Y^k_{\ n} \end{pmatrix}.$$

More succinctly: Let $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{Y} \in \mathbb{R}^{n\times n}$ be the matrix product. Then, its coefficients correspond to

$$Z^i_{\ j} = \sum_{k=1}^n X^i_{\ k} Y^k_{\ j} \quad \text{for all} \quad 1 \le i, j \le n. \tag{1}$$

Remark 2.1 (Restriction to real-valued square matrices). For today, we will restrict our attention to matrix products of real-valued, square $n \times n$ matrices. More general rectangular matrices can be converted into square matrices by adding rows/columns of zero. Moreover, the product of complex-valued matrices can be decomposed into real- and imaginary parts. This decomposition results in four sub-multiplications that are effectively real-valued.

Formula (1) suggests the following general cost for computing the product $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{Y}$ of two $n \times n$ matrices. Computing a coefficient $Z^i_{\ j}$ requires n elementary multiplications ($X^i_{\ k} Y^k_{\ j}$ for $1 \le k \le n$) and n subsequent elementary additions. There are in total n^2 coefficients, so the total arithmetic cost is n^3 elementary multiplications and n^3 elementary additions.

Fact 2.2. *The arithmetic cost of matrix multiplication by means of Formula* (1) *is* $2n^3$.

This total arithmetic cost is of order $\mathcal{O}(n^3)$. A natural question is whether this scaling in the problem size is optimal. Any improvement could lead to faster matrix

multiplication algorithms. This is highly desirable in practice, since matrix multiplications are at the very core of most numerical linear algebra techniques. We use the following notation to indicate such potential improvements.

Definition 2.3 (exponent of matrix multiplication)**.** The exponent of matrix multiplication ω is the smallest number such that a (potentially asymptotic) matrix multiplication algorithm exists whose arithmetic cost obeys $\mathcal{O}(n^\omega)$.

Fact 2.2 asserts $\omega \leq 3$. Also, we do not make any assumption about additional structure in the matrices that we wish to multiply. A general $n \times n$ matrix has n^2 degrees of freedom. This imposes a fundamental lower bound on the matrix multiplication exponent: $n^\omega \geq 2n^2$ arithmetic operations are necessary to input the problem description. Combining both yields

$$2 < \omega \leq 3. \tag{2}$$

Naively, one might assume that $\omega = 3$ (standard matrix multiplication) is optimal. However, this is not the case. The current record is

$$\omega \leq 2.3729$$

and was achieved by Le Gall in 2014. This remarkable improvement is a consequence of tensor analysis. We will devote this lecture and the next one to point out the ideas and methods behind these impressive developments.

3 Strassen's algorithms

3.1 Strassen's algorithm for multiplying 2×2 matrices

All fundamental improvements in the cost of matrix multiplication date back to a key observation that is due to Strassen from 1969. According to Landsberg, Strassen tried to prove that the naive matrix multiplication cost of $\omega = 3$ is optimal. In order to achieve this goal, he focused on 2×2 matrices defined on a finite field, where an exhaustive analysis is possible. His thorough analysis had quite the opposite effect. He found an alternative way of doing matrix multiplication that readily generalizes to any field. Set $n = 2$ and define the following seven numbers:

$$\begin{aligned}
m_1 &= \left(X^1{}_1 + X^2{}_2\right)\left(Y^1_1 + Y^2_2\right),\\
m_2 &= \left(X^2_1 + X^2_2\right)\boldsymbol{B}^1{}_1,\\
m_3 &= X^1_1\left(Y^1_2 - Y^2_2\right),\\
m_4 &= X^2_2\left(Y^2_1 - Y^1_1\right),\\
m_5 &= \left(X^1_1 + X^1_2\right)Y^2_2,\\
m_6 &= \left(X^2_1 - X^1_1\right)\left(Y^1_1 + Y^1_2\right),\\
m_7 &= \left(X^1_2 - X^2_2\right)\left(Y^2_1 + Y^2_2\right).
\end{aligned}$$

One can then check that all entries of the product $\boldsymbol{Z} = \boldsymbol{XY} \in \mathbb{R}^{2\times 2}$ correspond to elementary linear combinations of these seven numbers:

$$\boldsymbol{Z} = \begin{pmatrix} m_1 + m_4 - m_5 + m_7 & m_3 + m_5 \\ m_2 + m_4 & m_1 - m_2 + m_3 + m_6 \end{pmatrix} \in \mathbb{R}^{2\times 2}.$$

It is instructive to group this algorithm into three stages:

1. Linear pre-processing: Compute $r = 7$ linear combinations of each original input matrix:

$$\begin{aligned} a_1 =& X^1_{\ 1} + X^2_{\ 2}, a_2 = X^2_{\ 1} + X^2_{\ 2}, \ldots, a_7 = X^1_{\ 2} - X^2_{\ 2}, \\ b_1 =& Y^1_{\ 1} + Y^2_{\ 2}, b_2 = Y^1_{\ 1}, \ldots, b_7 = Y^2_{\ 1} + Y^2_{\ 2}. \end{aligned}$$

2. Elementary multiplications: Compute $m_i = a_i b_i$ for each $1 \leq i \leq 7 = r$.
3. Linear post-processing: infer the entries of the final matrix product by computing linear combinations of the m_i's.

We emphasize that scalar multiplications are isolated and only occur in stage 2. What is more, Strassen's algorithm requires fewer scalar multiplications than naive matrix multiplication: 7 instead of 8. This reduction in multiplication seems to come at an additional price in elementary additions: 18 (10 for pre-processing plus 8 for post-processing) instead of 8 for standard matrix multiplication.

3.2 Strassen's algorithm for multiplying $2^d \times 2^d$ matrices

Strassen's basic algorithm seems more resource-demanding than the naive procedure. However, it does get by with fewer multiplications. This small saving in multiplications does not (yet) offset the extra cost in linear pre- and post-processing. Perhaps surprisingly, this offset changes when we extend Strassen's basic algorithm to higher-dimensional matrix products.

The *divide and conquer rule* allows for readily generalizing Strassen's algorithm to matrix multiplication of $2^d \times 2^d$ matrices. Simply divide $\boldsymbol{X}$ and $\boldsymbol{Y}$ into 2×2 block matrices

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}^1_{\ 1} & \boldsymbol{X}^1_{\ 2} \\ \boldsymbol{X}^2_{\ 1} & \boldsymbol{X}^2_{\ 2} \end{pmatrix}, \ \boldsymbol{Y} = \begin{pmatrix} \boldsymbol{Y}^1_{\ 1} & \boldsymbol{Y}^1_{\ 2} \\ \boldsymbol{Y}^2_{\ 1} & \boldsymbol{Y}^2_{\ 2} \end{pmatrix},$$

where each block is a $2^{d-1} \times 2^{d-1}$ matrix. The intermediate values in Strassen's algorithm readily generalize to matrices:

$$\begin{aligned} \boldsymbol{M}_1 =& \left(\boldsymbol{X}^1_{\ 1} + \boldsymbol{X}^2_{\ 2}\right)\left(\boldsymbol{Y}^1_{\ 1} + \boldsymbol{Y}^2_{\ 2}\right) \in \mathbb{R}^{2^{d-1}\times 2^{d-1}}, \\ & \vdots \\ \boldsymbol{M}_7 =& \left(\boldsymbol{X}^1_{\ 2} - \boldsymbol{X}^2_{\ 2}\right)\left(\boldsymbol{Y}^2_{\ 1} + \boldsymbol{Y}^2_{\ 2}\right) \in \mathbb{R}^{2^{d-1}\times 2^{d-1}}. \end{aligned}$$

So does linear post-processing

$$\boldsymbol{Z} = \begin{pmatrix} \boldsymbol{M}_1 + \boldsymbol{M}_4 - \boldsymbol{M}_5 + \boldsymbol{M}_7 & \boldsymbol{M}_3 + \boldsymbol{M}_5 \\ \boldsymbol{M}_2 + \boldsymbol{M}_4 & \boldsymbol{M}_1 - \boldsymbol{M}_2 + \boldsymbol{M}_3 + \boldsymbol{M}_6 \end{pmatrix} \in \mathbb{R}^{2^d\times 2^d}.$$

This ansatz reduces the task of computing a single $2^d \times 2^d$ matrix multiplication to seven matrix multiplications of size $2^{d-1} \times 2^{d-1}$. Nothing prevents us from repeating this argument inductively: re-use Strassen to decompose each $2^{d-1} \times 2^{d-1}$ matrix multiplication into seven matrix multiplications of size $2^{d-2} \times 2^{d-2}$. Iterate this division recursively d times until the submatrices degenerate into numbers. For $n = 2^d$, this recursive procedure results in

$$7^d = \left(2^d\right)^{\log_2(7)} \simeq n^{2.807}$$

arithmetic multiplications. This is strictly smaller than the n^3 elementary multiplications associated with naive matrix multiplication. This simple counting argument does not (yet) take into account the extra effort in linear pre- and post-processing that is required for sequentially applying Strassen multiplication. We will devote Section 4 to a thorough analysis of the size of this extra cost. This study will highlight that the number of multiplications asymptotically dominate the total arithmetic effort:

Theorem 3.1 (Strassen's improvement for matrix multiplication)**.** *Asymptotically, a recursive application of Strassen's algorithm achieves a matrix multiplication exponent* $\omega \simeq 2.807 < 3$.

We conclude this section with a couple of remarks. Strassen's algorithm only works for matrices whose dimension is a power of two. This reshaping may be achieved by zero-padding: extend the original matrices with zero-rows and columns until they have a $2^d \times 2^d$ shape. Theorem 3.1 asserts that this seemingly counter-intuitive step may speed up the computation. For sufficiently large dimensions, it is beneficial to first increase the problem dimension to subsequently apply a faster algorithm.

Secondly, Strassen's algorithm – like most divide and conquer methods – may be parallelized to a considerable degree. A smooth working of the algorithm, however, requires a considerable amount of data transfer within the cores at each recursion step.

Finally, Strassen's algorithm is used in practice. Basic Linear Algebra Subprograms (BLAS) use it as a subroutine. The reduction in arithmetic cost comes at the prize of additional memory and reduced numerical stability. For these reasons, Strassen's algorithm is mostly used for integer matrix multiplication. Moreover, these practical implementations switch to the naive matrix multiplication procedure, once the submatrices are small enough. They do not carry out the full recursive reduction.

4 Asymptotic dominance of multiplications in Strassen's algorithm

4.1 The algebraic complexity model

In theoretical computer science, the complexity of an algorithm is usually measured in runtime. This measure the number of steps that a Turing machine would execute before terminating and providing the output.

Today, we shall focus on a conceptually similar, but slightly different computation model: the *algebraic complexity model*. There an algorithm is a sequence of algebraic steps. With matrix multiplication in mind, step l is a statement of the following form:

1. constant initialization: $t_l \leftarrow c$ for any $c \in \mathbb{R}$,
2. read in the problem description: $t_l \leftarrow X^i_j$, or $t_l \leftarrow Y^i_j$ for $1 \leq i, j \leq n$,
3. arithmetic computation: $t_l \leftarrow t_p \circ t_q$, where[1] $\circ \in \{+, -, \times\}$ and $p, q < l$,
4. solution output: $Z^i_j \leftarrow t_p$ for some $p < l$ and $1 \leq i, j \leq n$.

We say that such an arithmetic algorithm computes a matrix product $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{Y}$, if it outputs $Z^i_j = \sum_{k=1}^n X^i_k Y^k_j$ for all $1 \leq i, j \leq n$. The *running time*, or *complexity*, of the algorithm is the total number of steps disregarding read-in (step 2) and output (step 4). For matrix multiplication, this simplification is justified. The lower bound (2) on the complexity of computing matrix products is strictly larger than the quadratic cost associated with problem read-in and solution output.

When dealing with matrix multiplication algorithms in the algebraic complexity model, it is sufficient to focus on algorithms of a very special and desirable form.

Definition 4.1 (Normal form for matrix multiplication)**.** We say that a matrix multiplication algorithm is in normal algebraic form if it computes $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{Y}$ by executing the following steps:

1. For $1 \leq i \leq r$: compute α_i, a linear combination of the entries of $\boldsymbol{X}$,
2. For $1 \leq i \leq r$: compute β_i, a linear combination of the entries of $\boldsymbol{Y}$,
3. For $1 \leq i \leq r$: compute $p_i = \alpha_i \beta_i$.
4. For $1 \leq i, j \leq n$: compute $\boldsymbol{Z}^i_j$ as a linear combination of the p_i's.

All these linear combinations are fixed, i.e. they don't depend on the inputs $\boldsymbol{X}$ and $\boldsymbol{Y}$. The size of this normal form algorithm is characterized by r.

This normal form mimics the original presentation of Strassen's algorithm from Subsection 3.1.

Fact 4.2. *Strassen's original* 2×2 *algorithm has normal form with* $r = 7$.

Definition 4.1 may seem somewhat ad-hoc and geared towards Strassen's basic algorithm. This, however, is not the case. Any arithmetic algorithm for matrix multiplication can be converted into this normal form at comparatively little extra cost.

Theorem 4.3. *Suppose there exists an arithmetic algorithm for matrix multiplication that has runtime* T*. Then, there is a normal form algorithm of size* $r = 2T$*.*

The proof of this statement is somewhat pedantic and we refer to, for instance, Yuval Filmus' lecture notes for details. The key idea is that matrix multiplication has a rich bi-linear structure: the final output must be linear in the entries of $\boldsymbol{X}$ and linear in the entries of $\boldsymbol{Y}$. This restriction alone imposes severe constraints on the arithmetic

[1] Typically, the algebraic complexity model also allows for division. However, division does not feature in our algorithms for matrix multiplication and behaves somewhat differently in the algorithmic analysis. This is why we choose to omit it here.

expressions that can occur throughout the course of a general arithmetic algorithm. For instance, no polynomials of order three (or higher) can feature in the final expression. Should the algorithm compute a third order polynomial at any step, this must cancel out again at a later time and we can safely ignore it. Elementary arguments like this allow to considerably trim the arithmetic representation of a general matrix multiplication algorithm. Subsequently, this allows for a conversion into normal form at relatively little extra cost.

4.2 Dominance of multiplications in the arithmetic complexity model

The following statement provides a rigorous connection between the size r of a matrix multiplication algorithm in normal form and the associated runtime (measured in the arithmetic complexity model).

Theorem 4.4. *Suppose there is an $q \in \mathbb{N}$ such that there exists a normal form algorithm that multiplies two $q \times q$ matrices and has size $r = q^\alpha$. Then, the exponent of matrix multiplication obeys $\omega \leq \alpha$.*

Theorem 3.1 is an immediate consequence of this general result. Strassen's basic algorithm meets the requirements of this statement for $q = 2$ and $r = 7 = 2^{\log_2(7)}$. Applying it ensures,

$$\omega \leq \log_2(7) \simeq 2.807.$$

Proof of Theorem 4.4. We follow the divide and conquer approach sketched for Strassen's algorithm We can recursively extend an algorithm for $q \times q$ matrix multiplication to an algorithm for multiplying two $q^k \times q^k$ matrices. Let us denote the associated runtime by $T(k)$. We will establish the claim by induction over k. The assumption $T(1) = q^\alpha$ establishes the base case. For the induction step, we bound $T(k+1)$ in terms of $T(k)$ and additional dimension-dependent factors.

Divide and conquer allows us to reduce matrix multiplication of $q^{k+1} \times q^{k+1}$ matrices to a sequence of in total r smaller matrix multiplications. To do this, we divide $\boldsymbol{X}$ and $\boldsymbol{Y}$ into q^2 blocks of size $q^k \times q^k$ each. We then apply the algorithm to compute the matrix product block-wise. The normal form assumption ensures, that this involves exactly r linear combinations of sub-blocks $\boldsymbol{X}^i{}_j$ and r linear combinations of sub-blocks $\boldsymbol{Y}^i_j$. Moreover, let us assume for simplicity that each linear combination only involves a constant number of blocks[2] (this is true for Strassen's algorithm). Since $\boldsymbol{X}^i{}_j$ and $\boldsymbol{Y}^i_j$ have size $q^k \times q^k$, computing each linear combination takes at most $c(q^k)^2 = cq^{2k}$ arithmetic operations. The normal form ensures that we need to compute exactly r linear combinations for sub-blocks of $\boldsymbol{X}$ and $\boldsymbol{Y}$ each. The total arithmetic cost of pre-processing is therefore bounded by

$$T_{\text{pre}}(k+1) \leq 2crq^{2k}.$$

[2] A more involved argument allows for by-passing this simplifying assumption, but somewhat obscures the main conceptual ideas.

Next, we multiply these linear combinations using the normal form algorithm for $q^k \times q^k$ matrices. The induction hypothesis asserts that this step requires

$$T_{\text{mult}}(k+1) = rT(k)$$

arithmetic operations. Finally, we compute all q^2 blocks of the target matrix $\boldsymbol{Z} = \boldsymbol{XY}$:

$$T_{\text{comb}}(k+1) \leq q^2 \times c'(q^k)^2 = c'q^2q^{2k}.$$

Adding all these individual runtime bounds results in the following bound on the overall runtime:

$$T(k+1) \leq rT(k) + \Big(2cr + c'q^2\Big)q^{2k} = q^\alpha T(d) + \Big(2cq^\alpha + c'q^2\Big)q^{2k}.$$

Finally, recall the lower bound on the minimal cost of matrix multiplication from (2): multiplying two $q^k \times q^k$ matrices requires more than q^{2k} arithmetic operations. Applied to the problem at hand, this ensures $\alpha > 2$ and we may further simplify the inductive runtime bound:

$$T(k+1) \leq q^\alpha\Big(T(k) + Cq^{2k}\Big).$$

Since $\alpha > 2$, $T(k)$ asymptotically dominates Cq^{2k} for any value of the constant C. In turn, the asymptotic solution to this implicit recurrence is

$$T(k) = \mathcal{O}\Big(q^{\alpha k}\Big) = \mathcal{O}\Big(\Big(q^k\Big)^\alpha\Big).$$

In terms of the matrix size $n = q^k$, this is $T(k) = \mathcal{O}(n^\alpha)$ and provides an upper bound on the exponent of matrix multiplication. □

Lecture 12: Tensorial aspects of matrix multiplication

Scribe: Richard Kueng

ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp
May 01, 2019

1 Agenda and Outline

1. Matrix multiplication as a tensor
2. Connections between tensor rank and the exponent of matrix multiplication
3. Schönhage's approach
4. Current records and a rough sketch of the laser method

Last lecture was devoted to a detailed analysis of Strassen's algorithm for fast matrix multiplication. Today, we will re-visit this idea and expand upon it using tensor methods. This alternative point of view has led to spectacular improvements in the fundamental algorithmic cost associated with matrix multiplication. These speed-ups are general, i.e. they do not depend on advantageous matrix structure, like sparsity. Since matrix multiplication is *the* dominant subroutine in numerical linear algebra – and, by extension, data analysis – results of this type are of great relevance to many scientific communities.

Naively, one would expect that the arithmetic cost of multiplying two $n \times n$ matrices is $\mathcal{O}(n^3)$. This would imply that the *exponent of matrix multiplication* is $\omega = 3$. Since Strassen's discovery in 1969, many researches believe that the true cost of matrix multiplication is "almost" linear in the problem size, i.e. the exponent obeys $\omega = 2 + \varepsilon$, where $\varepsilon > 0$ is small. The current record in this direction is

$$\mathcal{O}(n^\omega) \quad \text{where} \quad \omega \leq 2.3728639$$

and was established by Le Gall in 2014. Subsequent work by Ambainis, Fimus and Le Gall highlights that the potential of the underlying approach is almost exhausted: $\omega = 2.3725$ cannot be overcome by incremental improvements of current techniques. Scientifically, this is an exciting state of the art: Further improvements seem to require truly novel ideas.

2 Matrix multiplication as a tensor

We will restrict our attention to real-valued matrix multiplication. Fix $\boldsymbol{X} \in \mathbb{R}^{n \times m}$, as well as $\boldsymbol{Y} \in \mathbb{R}^{m \times p}$ and denote their entries by $X^i_{\ j}$ (entry in the i-th row and j-th column) and $Y^j_{\ k}$ (entry in the j-th row and k-th column), respectively. Then, the product $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{Y}$ is a $n \times p$ matrix whose entries are defined by the matrix multiplication rule:

$$Z^i_{\ k} = \sum_{j=1}^{m} X^i_{\ j} Y^j_{\ k} \quad \text{for} \quad 1 \leq i \leq n,\ 1 \leq k \leq p.$$

Matrix multiplication can be regarded as the following bilinear map:

$$\begin{aligned}\mathbb{R}^{n\times m}\times\mathbb{R}^{m\times p}\to&\mathbb{R}^{n\times p},\\(\boldsymbol{X},\boldsymbol{Y})\mapsto&\boldsymbol{Z}=\boldsymbol{X}\boldsymbol{Y}.\end{aligned}$$

Indeed, it is easy to check that this map is linear in both inputs. Next, note that the output space $\mathbb{R}^{n\times p}$ is a linear vector space with (finite) dimension np. We can endow it with the Frobenius inner product $(\boldsymbol{Z}_1,\boldsymbol{Z}_2)=\mathrm{tr}\Big(\boldsymbol{Z}_1^T\boldsymbol{Z}_2\Big)$. This inner product establishes a one-to-one relation (isomorphism) between $\mathbb{R}^{n\times p}$ and its dual space $(\mathbb{R}^{n\times p})^*\simeq\mathbb{R}^{n\times p}$ – the space of all linear functionals on $\mathbb{R}^{n\times p}$:

$$\varphi(\boldsymbol{Z})=(\boldsymbol{\Phi},\boldsymbol{Z})\quad\text{for some}\quad\boldsymbol{\Phi}\in\mathbb{R}^{n\times p}.$$

Dualizing the image space allows us to convert matrix multiplication into a tri-linear form:

$$\begin{aligned}\mathbb{R}^{n\times m}\times\mathbb{R}^{m\times p}\times\big(\mathbb{R}^{n\times p}\big)^*\to&\mathbb{R},\\(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}^*)\mapsto&(\boldsymbol{Z}^*,\boldsymbol{X}\boldsymbol{Y})=\mathrm{tr}(\boldsymbol{Z}\boldsymbol{X}\boldsymbol{Y})=\mathrm{tr}(\boldsymbol{X}\boldsymbol{Y}\boldsymbol{Z}).\end{aligned}$$

The space of tri-linear forms is the canonical dual space of order three tensor products. This correspondence allows us to associate matrix multiplication with a tensor. This tensor becomes concrete if we choose bases for the individual matrix spaces (viewed as finite dimensional vector spaces). Denote the standard basis of $\mathbb{R}^{n\times m}$ by $\boldsymbol{x}^i{}_j$, the standard basis of $\mathbb{R}^{m\times p}$ by $\boldsymbol{y}^j_k$ and let $\boldsymbol{z}^k_i$ be the standard basis of $\mathbb{R}^{p\times n}$. Then,

$$\langle n,m,p\rangle=\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{p}\boldsymbol{x}^i{}_j\otimes\boldsymbol{y}^j_k\otimes\boldsymbol{z}^k_i\tag{1}$$

is the *matrix multiplication tensor*. It has a very symmetric structure, see Figure 1.

It is wortwhile to underline this correspondence with a more concrete calculation. For $\boldsymbol{A}\in\mathbb{R}^{n\times m}$, $\boldsymbol{B}\in\mathbb{R}^{m\times p}$ and $\boldsymbol{C}\in\mathbb{R}^{p\times n}$, we obtain

$$\begin{aligned}(\langle n,m,p\rangle,\boldsymbol{A}\otimes\boldsymbol{B}\otimes\boldsymbol{C})=&\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{p}\Big(\boldsymbol{x}^i{}_j,\boldsymbol{A}\Big)\Big(\boldsymbol{y}^j_k,\boldsymbol{B}\Big)\Big(\boldsymbol{z}^k_i,\boldsymbol{Z}\Big)\\=&\sum_{i,j,k}A^i{}_jB^j{}_kZ^k{}_i=\mathrm{tr}(\boldsymbol{A}\boldsymbol{B}\boldsymbol{C}).\end{aligned}$$

Specifying $\boldsymbol{C}=(\boldsymbol{e}_i\boldsymbol{e}_k)^T$ allows us to read off the (i,k)-th entry of the matrix product $\boldsymbol{A}\boldsymbol{B}\in\mathbb{R}^{n\times p}$:

$$\Big(\langle n,m,p\rangle,\boldsymbol{A}\otimes\boldsymbol{B}\otimes(\boldsymbol{e}_i\boldsymbol{e}_k)^T\Big)=\mathrm{tr}\Big(\boldsymbol{A}\boldsymbol{B}(\boldsymbol{e}_i\boldsymbol{e}_j)^T\Big)=\boldsymbol{e}_i^T\boldsymbol{A}\boldsymbol{B}\boldsymbol{e}_j$$

for all $1\le i\le n$ and $1\le k\le p$.

The matrix multiplication tensor (1) is a sum of nmp elementary tensor products. Evaluating the contribution of each elementary tensor to the overall matrix product is

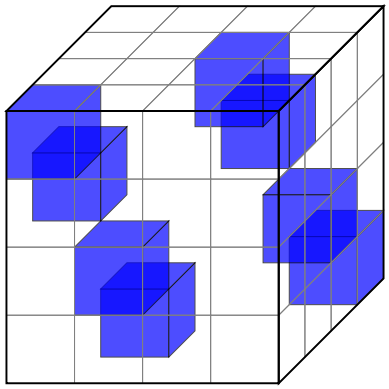


Figure 1 Visualization of $\langle 2,2,2\rangle$ viewed as a 3-dimensional array in $\mathbb{R}^4 \otimes \mathbb{R}^4 \otimes \mathbb{R}^4 \simeq \mathbb{R}^{4\times 4\times 4}$. The blue boxes indicate an entry of one, while white boxes denote zero entries.

cheap: simply read in the corresponding matrix entries and compute a single product of three numbers. This suggests that the computational cost is dominated by the number of rank-one terms that constitute the tensor $\langle n,m,p\rangle$ – in other words: the *tensor rank* matters. The decomposition (1) provides an upper bound on the tensor rank:

$$r(\langle n,m,p\rangle) \leq nmp.$$

Note that this upper bound is proportional to the number of scalar multiplications that is required for standard matrix multiplication.

3 Connections between tensor rank and the exponent of matrix multiplication

3.1 Strassen's Algorithm from a tensor perspective

In 1969 Strassen found an way of multiplying two 2×2 matrices that gets by with fewer multiplications than the naive algorithm. We refer to the previous lecture for details. Here, we emphasize that Strassen's procedure implies an alternative way of decomposing the matrix multiplication tensor $\langle 2,2,2\rangle$ (1) into elementary rank-one tensors:

$$\begin{aligned}\langle 2,2,2\rangle =& \left(\boldsymbol{x}_1^1+\boldsymbol{x}_2^2\right)\otimes\left(\boldsymbol{y}_1^1+\boldsymbol{y}_2^2\right)\otimes\left(\boldsymbol{z}_1^1+\boldsymbol{z}_2^2\right)+\left(\boldsymbol{x}_1^2+\boldsymbol{x}_2^2\right)\otimes\boldsymbol{y}_1^1\otimes\left(\boldsymbol{z}_1^2-\boldsymbol{z}_2^2\right)\\ &+\boldsymbol{x}_1^1\otimes\left(\boldsymbol{y}_2^1-\boldsymbol{y}_2^2\right)\otimes\left(\boldsymbol{z}_2^1+\boldsymbol{z}_2^2\right)+\boldsymbol{x}_2^2\otimes\left(\boldsymbol{y}_1^2-\boldsymbol{y}_1^1\right)\otimes\left(\boldsymbol{z}_1^1+\boldsymbol{z}_1^2\right)\\ &+\left(\boldsymbol{x}_1^1+\boldsymbol{x}_2^1\right)\otimes\boldsymbol{y}_2^2\otimes\left(-\boldsymbol{z}_1^1+\boldsymbol{z}_2^1\right)+\left(\boldsymbol{x}_1^2-\boldsymbol{x}_1^1\right)\otimes\left(\boldsymbol{y}_1^1+\boldsymbol{y}_2^1\right)\boldsymbol{z}_2^2\\ &+\left(\boldsymbol{x}_2^1-\boldsymbol{x}_2^2\right)\otimes\left(\boldsymbol{y}_1^2+\boldsymbol{y}_2^2\right)\otimes\boldsymbol{z}_1^1.\end{aligned}$$

This is a sum of only seven elementary tensor products.

Theorem 3.1 (Strassen, 1969)**.** *The matrix multiplication tensor* $\langle 2,2,2\rangle \in \left(\mathbb{R}^{2\times 2}\right)^{\otimes 3}$ *has rank at most seven (today, we know that it is exactly seven).*

3.2 Connection between tensor rank and the complexity of matrix multiplication

Strassen's results suggest a connection between tensor rank and the number of elementary multiplications required for matrix multiplication. Let ω be the *exponent of matrix multiplication*, i.e. the smallest number such that a (potentially asymptotic) algorithm exists that multiplies two square matrices using $\mathcal{O}(n^\omega)$ arithmetic operations.

We can combine our new tensor observation with the technical results from last lecture to derive the following profound correspondence.

Theorem 3.2. *The tensor rank $r(\langle n,n,n\rangle)$ of any square matrix multiplication tensor provides an upper bound on the exponent of matrix multiplication:*

$$\omega \leq \log_n(r(\langle n,n,n\rangle)) \quad \textit{for any} \quad n \in \mathbb{N}.$$

Combining this insight with Theorem 3.1 readily reproduces the main result from the previous lecture:

$$\omega \leq \log_2(r(\langle 2,2,2\rangle)) \leq \log_2(7) \simeq 2.807. \tag{2}$$

Proof sketch of Theorem 3.2. Fix $n \in \mathbb{N}$ and suppose that $\langle n,n,n\rangle$ admits a decomposition into r elementary tensors (i.e. $\langle n,n,n\rangle$ has rank at most r). Then, we can decompose the matrix multiplication tensor into a sum of r elementary tensor products. We can then use this decomposition as a guideline to construct an arithmetic algorithm in normal form that multiplies two $n \times n$ matrices. The size of this algorithm is governed by r, because the number of arithmetic multiplications exactly corresponds to the number of rank-one tensor contributions. Linear pre- and post-processing steps take into account that the elementary tensor factors need not be matrix standard basis elements. We leave a detailed establishment of this correspondence as an instructive exercise.

Subsequently, we can apply Theorem 4.4 from Lecture 11 to draw a precise connection to the exponent of matrix multiplication. Recall that this result uses divide and conquer to extend this arithmetic algorithm to matrix product of size $n^k \times n^k$ and let k go to infinity. The resulting recurrence establishes the advertised correspondence between the size of the algorithm r and the exponent of matrix multiplication. □

3.3 Extension to asymmetric matrix multiplication tensors

Theorem 3.2 hinges on the assumption that the underlying tensor describes matrix multiplication of square matrices. This structural requirement is essential for the divide and conquer strategy that establishes the connection between tensor rank and the exponent of matrix multiplication.

Viewed from this angle, the tensor rank is somewhat more flexible. There are certain symmetrization operations on tensors that do not increase the tensor rank. Suppose that we have established a non-trivial bound on the rank of $\langle n,m,p\rangle$. We can then "symmetrize" the tensor to effectively convert $\langle n,m,p\rangle$ into $\langle nmp,nmp,nmp\rangle$ – a square matrix multiplication – while simultaneously maintaining key aspects of the original rank bound. This is the general idea behind a proof of the following extension of Theorem 3.2.

Theorem 3.3. *The tensor rank of a general matrix multiplication tensor provides an upper bound on the exponent of matrix multiplication:*

$$\omega \leq 3 \log_{nmp}(r(\langle n, m, p\rangle)).$$

Proof. Let $\boldsymbol{T} = \sum_{i,j,k} T_{ijk} \boldsymbol{a}_i \otimes \boldsymbol{b}_j \otimes \boldsymbol{c}_k$ be a general order-three tensor. Define the *rotations*

$$\boldsymbol{T}^C = \sum_{i,j,k} T_{ijk} \boldsymbol{y}_j \otimes \boldsymbol{z}_k \otimes \boldsymbol{x}_i \quad \text{and} \quad \boldsymbol{T}^{C^2} = \sum_{i,j,k} T_{ijk} \boldsymbol{z}_k \otimes \boldsymbol{x}_i \otimes \boldsymbol{y}_j.$$

This operation extends the notion of transposition to higher order tensors. It is easy to check that rotations do not affect the tensor rank. Next, note that we can also form the tensor product of two order-three tensors. Formally this results in a tensor of order six. The tensor rank is sub-multiplicative under tensoring: $r(\boldsymbol{T} \otimes \boldsymbol{T}') \leq r(\boldsymbol{T}) r(\boldsymbol{T}')$.

By combining these operations, we can use $r = \operatorname{rank}(\langle n, m, p\rangle)$ to bound the tensor rank of $\langle nmp, nmp, nmp\rangle$:

$$\begin{aligned} r(\langle nmp, nmp, nmp\rangle) =& r(\langle n, m, p\rangle \otimes \langle m, p, n\rangle \otimes \langle p, n, m\rangle) \\ \leq& r(\langle n, m, p\rangle) r(\langle m, p, n\rangle) r(\langle p, n, m\rangle) = r(\langle n, m, p\rangle)^3 = r^3. \end{aligned}$$

The last line follows from the fact that $\langle m, p, n\rangle$ and $\langle p, n, m\rangle$ are rotations of $\langle n, m, p\rangle$. Inserting the bound $r(\langle nmp, nmp, nmp\rangle) \leq r^3$ into Theorem 3.2 establishes the claim. □

3.4 Extension to border rank

Tensors behave very differently from matrices. Recall that certain rank-r tensors $\boldsymbol{T}$ can be approximated to arbitrary accuracy by tensors that have much smaller rank. The minimal rank of such approximating tensors is called the *border rank* $\underline{r}(\boldsymbol{T})$.

At first sight, the conversion of such approximations into accurate numerical algorithms for matrix multiplications seems challenging. However, it turns out that this is not the case. By adapting the divide and conquer strategy appropriately, one can show that the approximation accuracy ε becomes almost irrelevant when extending the original algorithm recursively to very large matrix dimensions. We content ourselves with highlighting the result, while referring to the literature for rigorous proofs.

Theorem 3.4. *The border rank of a general matrix multiplication tensor provides an upper bound on the exponent of matrix multiplication:*

$$\omega \leq 3 \log_{nmp}(\underline{r}(\langle n, m, p\rangle)).$$

4 Improved bounds on the exponent of matrix multiplication

The previous results may seem somewhat technical. However, Theorem 3.4 forms the basis of virtually all improvements on the size of ω since Strassen's original discovery.

In a nutshell, all of these improvements arise from variants and refinements of the following basic strategy. Identify (small) numbers $n, m, p \in \mathbb{N}$ and a matrix multiplication tensor $\langle n, m, p\rangle$ whose border rank is as small as possible. Then, use Theorem 3.4 (or refinements thereof) to convert this insight about border rank into an upper bound on ω.

4.1 Schoenhage's Theorem

In 1981, Schoenhage established the following upper bound on the exponent of matrix multiplication:

$$\omega \leq 2.55. \tag{3}$$

This substantial improvement over Strassen's bound (2) is a consequence of *Schoenhage's identity*:

$$\underline{r}(\langle 4,1,4\rangle \oplus \langle 1,9,1\rangle) \leq 17. \tag{4}$$

Here, $\oplus$ denotes the direct sum of two matrix multiplication tensors. The direct sum for tensors is defined in an analogous fashion to the direct sum of two matrices $\boldsymbol{A} \oplus \boldsymbol{B}$. Each tensor factor is decomposed into two orthogonal subspaces and each tensor only acts on one subset of these subspaces. From an operational perspective, Schoenhage's identity bounds the joint border rank of an outer product of two 4-dimensional vectors *and* an inner product of two completely unrelated 9-dimensional vectors. This bound is remarkable, because border ranks of both individual operations are well understood[1]:

$$\underline{r}(\langle 4,1,4\rangle) = 16 \quad \text{and} \quad \underline{r}(\langle 1,9,1\rangle) = 9.$$

If we associate the border rank (qualitatively) with the number of multiplications required to compute outer and inner products, we obtain the following puzzling interpretation of (4): Computing the outer product of two 4-dimensional vectors requires 16 multiplications. At the cost of one additional multiplication, we get an additional inner product of two completely unrelated 9-dimensional vectors for free!

Schoenhage capitalized on this counter-intuitive tensor phenomenon by extending Theorem 3.4 to direct sums of different rectangular matrix multiplication tensors.

Theorem 4.1 (Asymptotic sum inequality)**.** *The following bound is true for any triple sequences* $(n_1, m_1, p_1), \ldots, (n_l, m_l, p_l) \in \mathbb{N} \times \mathbb{N} \times \mathbb{N}$

$$\sum_{i=1}^{l} (n_i m_i p_i)^{\omega/3} \leq \underline{r}\left(\bigotimes_{i=1}^{l} \langle n_i, m_i, p_i\rangle\right)$$

Schoenhage's bound (3) follows from combining the identity (4) with the asymptotic sum inequality and capitalizing on the insight that the border rank is sub-multiplicative under taking tensor products. Choose any $N \in \mathbb{N}$ and note that

$$\underline{r}\Big((\langle 4,1,4\rangle \oplus \langle 1,9,1\rangle)^{\otimes N}\Big) \leq 17^N.$$

We may interpret this tensor product as a direct sum of many independent matrix multiplications. Applying the direct sum inequality and subsequently transforming back to (tensor) products yields

$$17^N \geq \underline{r}\Big((\langle 4,1,4\rangle \oplus \langle 1,9,1\rangle)^{\otimes N}\Big) = \Big((4\times 1\times 4)^{\omega/3} + (1\times 9\times 1)^{\omega/3}\Big)^N = \Big(16^{\omega/3} + 9^{\omega/3}\Big)^N$$

Solving $16^{\omega/3} + 9^{\omega/3} = 17$ for ωestablishes Schoenhage's improvement (3).

[1] Exact numbers for rank and border rank readily follow from the fact that both operations may be described by matrices.

4.2 A rough sketch of the main idea behind recent top scores

Strassen's bound on the exponent of matrix multiplication arises from finding a single square matrix multiplication tensor $\langle 2,2,2\rangle$ whose rank is smaller than naively anticipated (7 vs. 8). The divide and conquer strategy subsequently allows for converting this gain in tensor rank into a genuine speed-up for multiplying large square matrices. This approach can be readily extended to handle non-square matrix multiplication tensors (Theorem 3.3) and border rank (Theorem 3.4).

Schoenhage deviated from this straightforward approach by considering direct sums of different matrix multiplication tensors. This affects the relation to matrix multiplication. The correspondence is more involved and mediated by the direct sum inequality (Theorem 4.1) which is non-trivial to prove. However, this relaxation allowed Schoenhage to analyze the border rank of more "exotic" order-three tensors. The identity (4) achieved a substantially smaller border rank than Strassen's observation (and, more generally: any known border rank bound for matrix multiplication tensors). This resulted in a much better bound: $\omega \leq 2.55$.

More recent developments are based on pushing Schoenhage's idea further: deviate even more from nice matrix multiplication tensors and search this larger set of target tensors for specimen that have a particularly small border rank. This is the main idea behind the so-called *laser method*. At the basis of this method is a border rank identity due to Coppersmith and Winograd from 1990. For any $q \in \mathbb{N}$,

$$\underline{r}\Big(\langle 1,1,q\rangle^{[0,1,1]} + \langle q,1,1\rangle^{[1,0,1]} + \langle 1,q,1\rangle^{[1,1,0]} + \langle 1,1,1\rangle^{[0,0,2]} + \langle 1,1,1\rangle^{[0,2.0]} + \langle 1,1,1\rangle^{[2,0,0]}\Big) \leq q+2.$$

Here $\langle n,m,p\rangle^{[I,J,K]}$ denotes a matrix multiplication tensor equivalent to $\langle n,m,p\rangle$, but whose support is restricted to a certain subset of matrix entries. The super-script indicates which subset of matrix entries is affected by the multiplication tensor. After flattening the matrix standard basis vectors, these partitions are

$$\boldsymbol{x}_0^{[0]}, \boldsymbol{x}_1^{[1]}, \ldots, \boldsymbol{x}_q^{[q]}, \boldsymbol{x}_{q+1}^{[1]},\ \boldsymbol{y}_0^{[0]},\ \boldsymbol{y}_1^{[1]}, \ldots, \boldsymbol{y}_q^{[q]}, \boldsymbol{y}_{q+1}^{[1]},\ \text{and}\ \boldsymbol{z}_0^{[0]}, \boldsymbol{z}_1^{[1]}, \ldots, \boldsymbol{z}_q^{[q]}, \boldsymbol{z}_{q+1}^{[1]}.$$

Importantly, the Coppersmith and Winograd identity is valid for any $q \in \mathbb{N}$. However, it is not directly related to a simple matrix multiplication procedure. Forcing certain matrix entries to zero, however, allows for reducing the tensor to something that looks much more like a standard matrix multiplication $\langle n(q), m(q), p(q)\rangle$. Tighter bounds on ω follow from identifying zero-out patterns that are as sparse as possible and nonetheless enforce a nice matrix multiplication structure. Zeroing out can be done for the original tensor directly (sub-optimal), or for high-order tensor products and direct sums, respectively. In 2014 Le Gall automated this search for efficient zero-out patterns in a large search space (vary q and the size of the tensor product) using numerical algorithms based on convex optimization. In doing so, he scored the current record regarding the asymptotic cost of matrix multiplication:

$$\omega \leq 2.3728639.$$

The laser method yields impressive results, but its potential is almost exhausted. In 2015, Ambainis, Filmus and Le Gall proved that it is impossible to go beyond $\omega = 2.3725$

using the Coppersmith-Winograd identity. This no-go result follows from putting the above ideas into a rigorous framework. Tight bounds on the convergence of ω to certain entropy functions can be established. These bounds cut both ways: Lower bounds establish upper bounds for ω. This is what Le Gall implicitly found in 2014. Upper bounds limit the capabilities of the entire framework and highlight a veritable bottleneck.

Lecture 13: The CP decomposition for tensors

Scribe: Richard Kueng

ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp
May 13, 2019

1 Agenda

1. Practical tools for handling tensors
   (a) Tensors as multi-dimensional arrays
   (b) Tensor slices
   (c) Matriciation
2. Useful (tensor) products and identities
3. The CP decomposition
   (a) Motivation
   (b) Definition
   (c) Computation

2 Practical tools for handling tensors

Today we will focus exclusively on real-valued tensor products of order three. The methods discussed readily extend to tensors of arbitrary order and may also be generalized to complex-valued tensors. However, the latter generalization may require some care: transposition and conjugation are not equivalent for complex vector spaces.

Fix $A = \mathbb{R}^{d_1}$, $B = \mathbb{R}^{d_2}$ and $C = \mathbb{R}^{d_3}$ and endow each space with the standard basis $\boldsymbol{e}_1, \dots, \boldsymbol{e}_{d_l}$, $l = 1, 2, 3$. We consider tensors in the tensor product space $A \otimes B \otimes C$:

$$\boldsymbol{T} = \sum_{i=1}^{r} \lambda_i \boldsymbol{a}_i \otimes \boldsymbol{b}_i \otimes \boldsymbol{c}_i \quad \boldsymbol{a}_i \in A,\ \boldsymbol{b}_i \in B,\ \boldsymbol{c}_i \in C,\ \lambda_i \in \mathbb{R}.$$

2.1 Bases and inner products

The individual standard bases give rise to an extended standard basis on the tensor product $A \otimes B \otimes C$:

$$\boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k \quad 1 \le i \le d_1,\ 1 \le j \le d_2,\ 1 \le k \le d_3$$

The canonical inner products on $\langle \cdot, \cdot \rangle$ extend as well. Expand $\boldsymbol{T}, \boldsymbol{T}' \in A \otimes B \otimes C$ with respect to the extended standard basis

$$\boldsymbol{T} = \sum_{i,j,k} t_{ijk} \boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k \quad \text{and} \quad \boldsymbol{T}' = \sum_{i,j,k} t'_{ijk} \boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k$$

and set

$$\begin{aligned}\langle \boldsymbol{T}, \boldsymbol{T}' \rangle &= \sum_{i,j,k} \sum_{i',j',k'} t_{ijk} t'_{i'j'k'} \langle \boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k, \boldsymbol{e}_{i'} \otimes \boldsymbol{e}_{j'} \otimes \boldsymbol{e}_{k'} \rangle \\ &= \sum_{i,j,k} \sum_{i',j',k'} t_{ijk} t'_{i'j'k'} \langle \boldsymbol{e}_i, \boldsymbol{e}_{i'} \rangle \langle \boldsymbol{e}_j, \boldsymbol{e}_{j'} \rangle \langle \boldsymbol{e}_k, \boldsymbol{e}_{k'} \rangle \\ &= \sum_{i,j,k} t_{ijk} t'_{ijk}.\end{aligned}$$

This in particular endows $A \otimes B \otimes C$ with a *Euclidean norm*:

$$\|\boldsymbol{T}\|_2^2 = \sum_{i,j,k} t_{ijk}^2. \tag{1}$$

The extended standard basis representation highlights a useful interpretation of tensors. They correspond to multi-dimensional arrays:

$$\boldsymbol{T} = [t_{ijk}]_{i,k,k} \in A \otimes B \otimes C = \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \times \mathbb{R}^{d_3}.$$

Vectors are 1-dimensional arrays and matrices are 2-dimensional arrays. Tensors correspond to higher order arrays. We emphasize, however, that this array interpretation is manifestly basis-dependent.

Example 2.1 (Movie frames)**.** We can associate a digital picture with a matrix of pixels. The (i, j)-th entry of this matrix encodes the color of the pixel that sits at position $1 \leq i \leq x_{\max}$ in the x-direction and $1 \leq j \leq y_{\max}$ in the y-direction. Movies correspond to a sequence of at least 24 frames per second. Throughout a single scene (no cut), these individual frames are typically closely related to each other. It therefore makes sense to represent a movie scene as a 3-dimensional data array, where the third axes encodes time.

2.2 Tensor fibres and slices

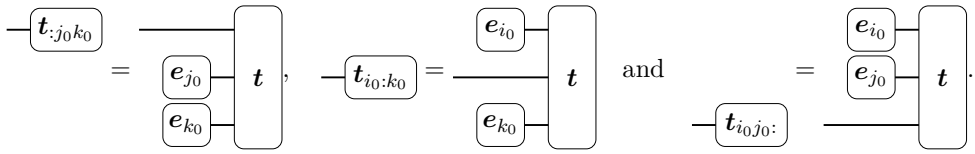
Let $\boldsymbol{T} \in A \otimes B \otimes C$ be a tensor with basis expansion $[t_{ijk}]$ for $1 \leq i \leq d_1$, $1 \leq j \leq d_2$ and $1 \leq k \leq d_3$.

Definition 2.2 (Fibre)**.** A fibre is the higher order analogue of matrix rows and columns. For $\boldsymbol{t} \in \mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \mathbb{R}^{d_3}$, fix $1 \leq i_0 \leq d_1$, $1 \leq j_0 \leq d_2$, $1 \leq k_0 \leq d_3$ and define

$$\boldsymbol{t}_{:j_0k_0} = \sum_{i=1}^{d_1} t_{ij_0k_0} \boldsymbol{e}_i \in A, \ \boldsymbol{t}_{i_0:k_0} = \sum_{j=1}^{d_2} t_{i_0jk_0} \boldsymbol{e}_j \in B, \ \boldsymbol{t}_{i_0j_0:} = \sum_{k=1}^{d_3} t_{i_0j_0k} \boldsymbol{e}_k \in C.$$

These vectors are called mode-1, mode-2 and mode-3 fibres, respectively.

Fibres arise from contracting certain indices with fixed standard basis vectors. In wiring calculus,

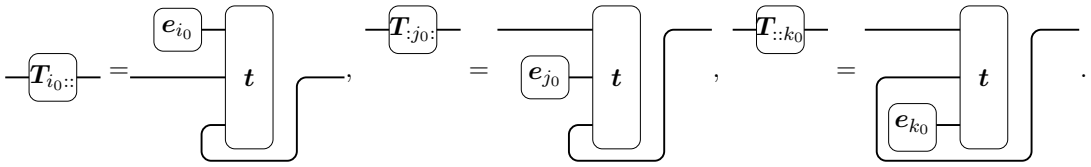


Example 2.3. For matrices (order two tensors) mode-1 fibers are column vectors and mode-2 fibers are row vectors.

Definition 2.4 (slice)**.** Slices are 2-dimensional sections of a tensor $[t_{ijk}] \in A \otimes B \otimes C$ and we treat them as matrices. Fix $1 \leq i_0 \leq d_1$, $1 \leq j_0 \leq d_2$, $1 \leq k_0 \leq d_3$ and define

$$\begin{aligned}
\boldsymbol{T}_{i_0::} &= \sum_{j,k} t_{i_0jk} \boldsymbol{e}_j \otimes \boldsymbol{e}_k \simeq \sum_{j,k} t_{i_0jk} \boldsymbol{e}_j \boldsymbol{e}_k^T \in \mathbb{R}^{d_2 \times d_3}, \\
\boldsymbol{T}_{:j_0:} &= \sum_{i,k} t_{ij_0k} \boldsymbol{e}_i \otimes \boldsymbol{e}_k \simeq \sum_{i,k} t_{ij_0k} \boldsymbol{e}_i \boldsymbol{e}_k^T \in \mathbb{R}^{d_1 \times d_3}, \\
\boldsymbol{T}_{::k_0} &= \sum_{i,j} t_{ijk_0} \boldsymbol{e}_i \otimes \boldsymbol{e}_j \simeq \sum_{imj} t_{ijk_0} \boldsymbol{e}_i \boldsymbol{e}_j^T \in \mathbb{R}^{d_1 \times d_2}.
\end{aligned}$$

Slices arise from contracting one index with a fixed standard basis vector and turning the remaining order-two tensor into a matrix (bend one index):



Example 2.5 (the Schur-product)**.** Set $d_1 = d_2 = d_3 = d$. Then, the tensor associated with the Hadamard product $\boldsymbol{x} \odot \boldsymbol{y}$ is $\boldsymbol{h} = \sum_{i=1}^{d} \boldsymbol{e}_i \otimes \boldsymbol{e}_i \otimes \boldsymbol{e}_i$. This corresponds to a "data cube" with ones on the super-diagonal and zeros everywhere else.

Example 2.6 (Matrix multiplication for 2×2 matrices)**.** Associate the standard matrix basis for 2×2 matrices with four standard basis vectors: $\boldsymbol{e}_i \boldsymbol{e}_j^T \to \boldsymbol{e}_i \otimes \boldsymbol{e}_j = \boldsymbol{e}_{(i,j)}$. Then, matrix multiplication can be viewed as a tensor in $\mathbb{R}^4 \times \mathbb{R}^4 \times \mathbb{R}^4$:

$$\langle 2, 2, 2 \rangle = \sum_{i,j,k=1}^{2} \boldsymbol{e}_{(i,j)} \otimes \boldsymbol{e}_{(j,k)} \otimes \boldsymbol{e}_{(k,i)}.$$

The associated data cube is depicted in Figure 1.

2.3 Matricization: transforming tensors into a matrix

Matricization, also knownas *unfolding* or *flattening*, is the process of reordering the elements of a tensor into a matrix. It is easiest understood in wiring notation. Consider an order three tensor

$$\boldsymbol{T} = \quad \boldsymbol{T} \quad \in A \otimes B \otimes C$$

There is an easy standard procedure to convert this tensor into a matrix. Leave one index line unchanged, group the other two together and bend them. For order three

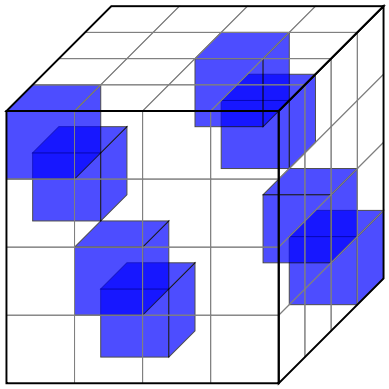
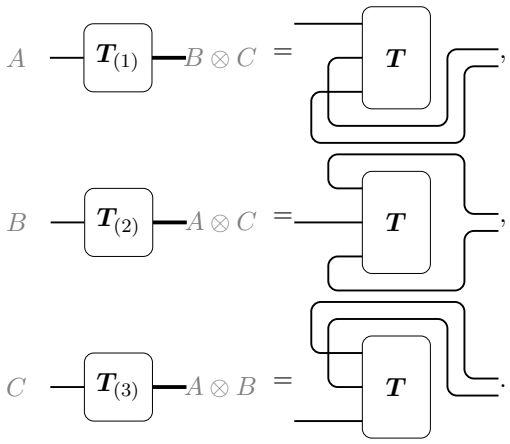


Figure 1 Data cube visualization of 2×2 matrix multiplication viewed as a tensor in $\mathbb{R}^4 \otimes \mathbb{R}^4 \otimes \mathbb{R}^4$.

tensors, there are $\binom{3}{2} = 3$ possible ways to achieve this goal:



The subscript indicates which tensor factor remain unchanged. These matriciations are called mode-k unfoldings, where k refers to the tensor factor that remains unchanged.

Example 2.7. Consider $\boldsymbol{T} \in \mathbb{R}^3 \otimes \mathbb{R}^4 \otimes \mathbb{R}^2 \simeq \mathbb{R}^{3\times4\times2}$ with frontal slices

$$\boldsymbol{T}_{::1} = \begin{pmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{pmatrix}, \quad \boldsymbol{T}_{::2} = \begin{pmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{pmatrix}.$$

Then,

$$\boldsymbol{T}_{(1)} = \begin{pmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{pmatrix} \in \mathbb{R}^{3\times(4\times 2)},$$

$$\boldsymbol{T}_{(2)} = \begin{pmatrix} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{pmatrix} \in \mathbb{R}^{4\times(3\times 2)},$$

$$\boldsymbol{T}_{(3)} = \begin{pmatrix} 1 & 2 & 3 & \cdots & 11 & 12 \\ 13 & 14 & 15 & \cdots & 23 & 24 \end{pmatrix} \in \mathbb{R}^{2\times(4\times 3)}.$$

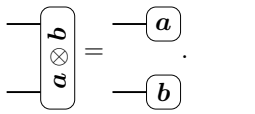
3 Useful product operation on vectors and matrices

3.0.1 Kronecker product

Fix $\boldsymbol{a} = (a_1, \ldots, a_{d_1})^T \in \mathbb{R}^{d_1}$ and $\boldsymbol{b} = (b_1, \ldots, b_{d_2})^T \in \mathbb{R}^{d_2}$. We define the *Kronecker product*

$$\boldsymbol{a} \otimes \boldsymbol{b} = \begin{pmatrix} a_1\boldsymbol{b} \\ \vdots \\ a_{d_1}\boldsymbol{b} \end{pmatrix} \in \mathbb{R}^{d_1\times d_2} \simeq \mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2}.$$

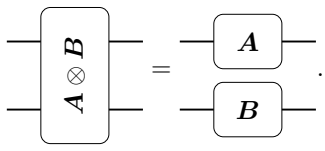
This is a concrete realization of the tensor product that has accompanied us throughout the corse of this lecture. It is basis independent and in wiring notation, we denote it by



This product operation readily extends to matrices. For $\boldsymbol{A} \in \mathbb{R}^{d_1\times d_2}$ and $\boldsymbol{B} \in \mathbb{R}^{d_3\times d_4}$ we set

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{pmatrix} a_{11}\boldsymbol{B} & \cdots & a_{1d_2}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{d_11}\boldsymbol{B} & \cdots & a_{d_1d_2}\boldsymbol{B} \end{pmatrix} \in \mathbb{R}^{(d_1\times d_3)\times(d_2\times d_4)} \simeq \mathbb{R}^{d_1\times d_2} \otimes \mathbb{R}^{d_3\times d_4}.$$

In wiring notation, this corresponds to arranging operators in parallel:



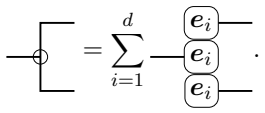
3.0.2 Hadamard product

The Hadamard product only makes sense for vectors (matrices) with equal dimensions.

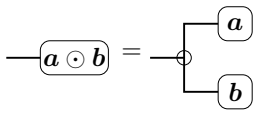
Fix $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$ and define their entry-wise product:

$$\boldsymbol{a} \odot \boldsymbol{b} = \begin{pmatrix} a_1 b_1 \\ \vdots \\ a_d b_d \end{pmatrix} \in \mathbb{R}^{d_1}.$$

Note that the coefficients of this vector correspond to a certain (symmetric) sub-selection of entries in the Kronecker product. We introduce the following wiring "gadget" to make this explicit:



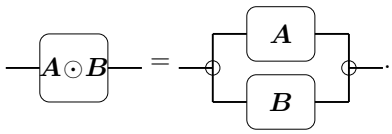
Then,



The Hadamard product can be extended to matrices of equal dimension. For $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{d_1 \times d_2}$ we define

$$\boldsymbol{A} \odot \boldsymbol{B} = \begin{pmatrix} a_{11} b_{11} & \cdots & a_{1d_2} b_{1d_2} \\ \vdots & \ddots & \vdots \\ a_{d_1 1} b_{d_1 1} & \cdots & a_{d_1 d_2} b_{d_1 d_2} \end{pmatrix} \in \mathbb{R}^{d_1 \times d_2}.$$

Similar to the vector case, this matrix product arises from sub-selecting certain entries of the Kronecker product $\boldsymbol{A} \otimes \boldsymbol{B}$. In wiring notation, this sub-selection is achieved by applying the Hadamard gadget twice:



Finally, note that we recover the Hadamard tensor $\boldsymbol{h} \in \mathbb{R}^{d_1} \otimes \mathbb{R}^{d_1} \otimes \mathbb{R}^{d_1}$ from Example 2.5 by bending the first and third indicex of the Hadamard gadget to the left:

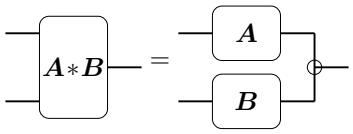
$$\boldsymbol{h} = \sum_{i=1}^{d_1} \boldsymbol{e}_i \otimes \boldsymbol{e}_i \otimes \boldsymbol{e}_i = \tag{2}$$

3.0.3 Khatri-Rao product

Suppose that $\boldsymbol{A} \in \mathbb{R}^{d_1 \times d_2}$ and $\boldsymbol{B} \in \mathbb{R}^{d_3 \times d_2}$ have the same number of columns. Then, we can define the following matrix product:

$$\boldsymbol{A} * \boldsymbol{B} = [\boldsymbol{a}_1 \cdots \boldsymbol{a}_{d_2}] * [\boldsymbol{b}_1 \cdots \boldsymbol{b}_{d_2}] = [\boldsymbol{a}_1 \otimes \boldsymbol{b}_1 \cdots \boldsymbol{a}_{d_2} \otimes \boldsymbol{b}_{d_2}] \in \mathbb{R}^{(d_1 \times d_3) \times d_2}.$$

This is called the *Khatri-Rao product*. It results from a one-sided application of the Hadamard gadget.



This graphical notation underlines the intermediary nature of this product. It is – in a precise sense – half way between the general Kronecker product and the highly structured Hadamard product.

3.0.4 Useful identities

Let $\boldsymbol{A}^\dagger$ denote the Moore-Penrose inverse. Then, persistence of the Kronecker product for operators readily implies

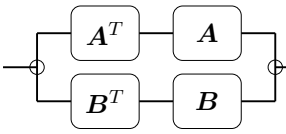
$$(\boldsymbol{A}\otimes\boldsymbol{B})(\boldsymbol{C}\otimes\boldsymbol{D}) = \boldsymbol{AC}\otimes\boldsymbol{BD} \quad \text{and} \quad (\boldsymbol{A}\otimes\boldsymbol{B})^\dagger = \boldsymbol{A}^\dagger\otimes\boldsymbol{B}^\dagger.$$

Although not obvious from the entry-wise definition, the Khatri-Rao product is associative:

$$(\boldsymbol{A}*\boldsymbol{B})*\boldsymbol{C} = \boldsymbol{A}*(\boldsymbol{B}*\boldsymbol{C}) =: \boldsymbol{A}*\boldsymbol{B}*\boldsymbol{C}.$$

This readily follows from the wiring definition and specific features of the Hadamard gadget: this gadget corresponds to a 3-way Kronecker delta. It is only non-zero if all of the three indices coincide. Like ordinary Kronecker products, this 3-way generalization is associative. The wiring formalism also allows for readily establishing the following useful identity between Khatri-Rao and Hadamard product:

$$(\boldsymbol{A}*\boldsymbol{B})^T(\boldsymbol{A}*\boldsymbol{B}) =$$



$$= (\boldsymbol{A}\boldsymbol{A}^T)\odot(\boldsymbol{B}^T\boldsymbol{B}).$$

This identity provides a simple expression for the Moore-Penrose inverse:

$$(\boldsymbol{A}*\boldsymbol{B})^\dagger = \Big((\boldsymbol{A}^T\boldsymbol{A})\odot(\boldsymbol{B}^T\boldsymbol{B})\Big)^\dagger(\boldsymbol{A}*\boldsymbol{B})^T. \tag{3}$$

4 The CP decomposition

4.1 Recapitulation: minimal rank decompositions

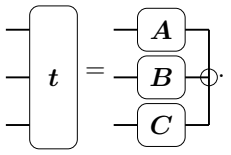
Recall that we may express any tensor in as a sum of rank-one elements. For order three tensors, we obtain the following decomposition:

$$\boldsymbol{t} = \sum_{i=1}^{r} \boldsymbol{a}_i\otimes\boldsymbol{b}_i\otimes\boldsymbol{c}_i \in \mathbb{R}^{d_1}\otimes\mathbb{R}^{d_2}\otimes\mathbb{R}^{d_3}. \tag{4}$$

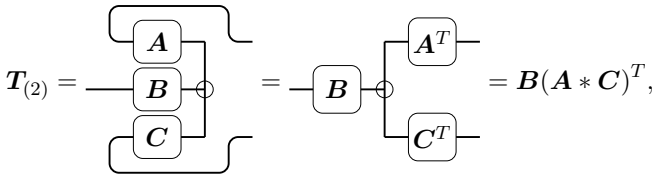
Here $\boldsymbol{a}_1,\dots,\boldsymbol{a}_r\in A$, $\boldsymbol{b}_1,\dots,\boldsymbol{b}_r\in B$ and $\boldsymbol{c}_1,\dots,\boldsymbol{c}_r\in C$. Set $\boldsymbol{A}=[\boldsymbol{a}_1\cdots\boldsymbol{a}_r]\in\mathbb{R}^{d_1\times r}$, $\boldsymbol{B}=[\boldsymbol{b}_1\cdots\boldsymbol{b}_r]\in\mathbb{R}^{d_2\times r}$ and $\boldsymbol{C}=[\boldsymbol{c}_1\cdots\boldsymbol{c}_r]\in\mathbb{R}^{d_3\times r}$. Then, Kolda advocates the following notation:

$$\boldsymbol{t} = [\![\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}]\!]. \tag{5}$$

The matrices $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{C}$ are called *factor matrices*. The Hadamard gadget (2) allows us to convert Kolda's notation into a simple wiring diagram. Equation (5) is equivalent to



Factor matrices are closely related to matriciations. For instance,

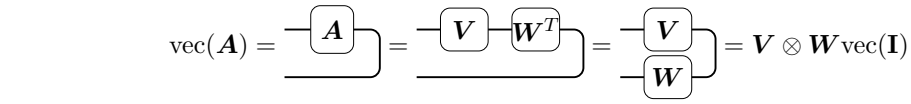


where we have identified the transpose of the Khatri-Rao product. Similar relation hold true for other the other mode-k unfoldings. Each matriciation singles out another factor matrix on the right:

$$\boldsymbol{T}_{(1)} = \boldsymbol{A}(\boldsymbol{B} * \boldsymbol{C})^T, \quad \boldsymbol{T}_{(2)} = \boldsymbol{B}(\boldsymbol{A} * \boldsymbol{C})^T \quad \text{and} \quad \boldsymbol{T}_{(3)} = \boldsymbol{C}(\boldsymbol{A} * \boldsymbol{B})^T. \tag{6}$$

4.2 CP decomposition

Minimal rank decompositions (4) of tensors closely resemble matrix factorizations. Indeed, suppose that $\boldsymbol{A} \in \mathbb{R}^{d_1 \times d_2}$ may be decomposed into a product of smaller matrices: $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{W}^T$, where $\boldsymbol{V} \in \mathbb{R}^{d_1 \times r}$ and $\boldsymbol{W} \in \mathbb{R}^{d_2 \times r}$. Then,



The (modified) Hadamard gadget $\boldsymbol{h} = \sum_{i=1}^{r} \boldsymbol{e}_i \otimes \boldsymbol{e}_i \otimes \boldsymbol{e}_i$ may be viewed as a natural extension of the vectorized identity $\mathrm{vec}(\mathbf{I}) = \sum_{i=1}^{r} \boldsymbol{e}_i \otimes \boldsymbol{e}_i$. One of the most classical tensor factorizations is based on this correspondence:

Definition 4.1 (CP decompostion)**.** A decomposition (4) of a tensor $\boldsymbol{t}$ into a sum of R rank-one elements is called a *CP decomposition* with rank R.

The name is a synergy of CANDECOMP (canonical decomposition) and PARAFAC (parallel factors). CP decompositions always exist. Indeed, we have defined the tensor product space $\mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \mathbb{R}^{d_3}$ as the linear hull of all elementary tensor products $\boldsymbol{a}_i \otimes \boldsymbol{b}_i \otimes \boldsymbol{c}_i$. The CP decomposition seeks to achieve the opposite: decompose a general tensor into a sum of R elementary tensor products. However, finding them is in general very challenging for the following reasons:

1. Finding a CP decomposition with rank R equal to the tensor rank r would implicitly identify the tensor rank. However, we know that the problem of determining the rank of a tensor is NP-hard.
2. CP decompositions are unique, but not invariant under linear transformations. Therefore, we cannot assume that the columns of $\boldsymbol{A}$, $\boldsymbol{B}$, or $\boldsymbol{C}$ are orthogonal.
3. The existence of border rank shows that CP decompositions need not be stable.

At the same time, CP decompositions are highly valuable in concrete applications. As already pointed out, the name CP is a snythesis of two different names that have a long history:

1. CANDECOMP was introduced for analyzing multiple similarity/dissimilarity matrices from a variety of subjects. Simple averaging of all subjects annihilates different points of view. Example: vowel sound date from different individuals (mode 1) spoke different vowels (mode 2) and the format (pitch, frequency pattrn) was measured (mode 3). This point of view was adopted by many research groups in different fields ranging from chemometrics, neuroscience to telecommunication.
2. PARAFAC was introduced, because tensor methods eliminate ambiguities associated with traditional PCA. In contrast to matrices, tensor factorizations are almost always unique.

4.3 An alternating least squares algorithm for approximating the CP decomposition

As already pointed out, there are complexity-theoretic obstructions towards computing the CP decomposition exactly. This however, does not mean that we cannot come up with approximation heuristics. Here, we present one such heuristic that is based on alternating least squares. It is designed to compute individual iterations as quickly as possible. The hope is then that many simple iterations ultimately converge to something useful.

The concrete goal is to compute an accurate rank-R approximation of a given tensor $\boldsymbol{t} \in \mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \mathbb{R}^{d_3}$. The extended inner product provides us with a natural distance measure – the extended Euclidean norm (1) – that quantifies approximation accuracy. Moreover, we shall consider the approximation rank R as a free input parameter to our algorithm. For given R, we aim to solve

$$\text{minimize} \quad \|\boldsymbol{t} - [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]\| \quad \text{subject to} \quad \boldsymbol{A} \in \mathbb{R}^{d_1 \times R},\ \boldsymbol{B} \in \mathbb{R}^{d_2 \times R},\ \boldsymbol{C} \in \mathbb{R}^{d_3 \times r}.$$

Solving this problem requires simultaneous optimization over three different matrix variables. This is typically a very challenging problem. *Alternating least squares* (ALS) is a popular approach for iteratively solving such problems heuristically. First, fix $\boldsymbol{B}, \boldsymbol{C}$, solve for $\boldsymbol{A}$ and update $\boldsymbol{A}$ to be the optimal solution of the single-variable problem. Then, fix $\boldsymbol{A}, \boldsymbol{C}$, solve for $\boldsymbol{B}$ and update accordingly. Keeping $\boldsymbol{A}, \boldsymbol{B}$ fixed and updating $\boldsymbol{C}$ in a similar fashion completes one ALS cycle. This cycle is repeated many times until some stopping condition is reached. Prominent stopping conditions are (i) very little

input : tensor $\boldsymbol{t} \in \mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \mathbb{R}^{d_3}$, rank R
Initialize $\boldsymbol{A} \in \mathbb{R}^{d_1 \times R}$, $\boldsymbol{B} \in \mathbb{R}^{d_2 \times R}$, $\boldsymbol{C} \in \mathbb{R}^{d_3 \times R}$, $N_{\max}$ (max nr. of iterations);
while $i < N$ **do**
  $\boldsymbol{A} \leftarrow \boldsymbol{T}_{(1)}(\boldsymbol{B} * \boldsymbol{C})\Big(\boldsymbol{B}^T\boldsymbol{B} \odot \boldsymbol{C}^T\boldsymbol{C}\Big)^{\dagger}$;
  $\boldsymbol{B} \leftarrow \boldsymbol{T}_{(2)}(\boldsymbol{A} * \boldsymbol{C})\Big(\boldsymbol{A}^T\boldsymbol{A} \odot \boldsymbol{C}^T\boldsymbol{C}\Big)^{\dagger}$;
  $\boldsymbol{C} \leftarrow \boldsymbol{T}_{(3)}(\boldsymbol{A} * \boldsymbol{B})\Big(\boldsymbol{A}^T\boldsymbol{A} \odot \boldsymbol{B}^T\boldsymbol{B}\Big)^{\dagger}$;
  (Break loop if a certain stopping condition is reached);
  $i \leftarrow i + 1$;
end
output : matrix factors $\boldsymbol{A} \in \mathbb{R}^{d_1 \times R}$, $\boldsymbol{B} \in \mathbb{R}^{d_2 \times R}$, $\boldsymbol{C}^{d_3 \times R}$

Algorithm 1: Alternating least squares (ALS) algorithm for approximating CP decompositions.

improvement in the objective function, (ii) very little change in the factor matrices, (iii) the objective value (norm to target tensor) is close to zero, or (iv) a pre-fixed maximum number of cycle repetitions is exhausted.

We emphasize that initialization also plays a very important role in such ALS-type algorithms. The initial choices of $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ may affect the performance considerably. A naive initialization would correspond to populate all three matrices with random entries. This often works well in practice, because random initialization may avoid "hard problem instances". Smarter initialization techniques use spectral information about matriciations of $\boldsymbol{t}$ to reduce the distance of the initialization to the final target tensor. This may limit the number of cycle repetitions required for convergence. However, we emphasize that we have barely scratched the surface here.

Algorithm 1 summarizes pseudo-code for an ALS approach to computing CP decompositions. The individual updates are optimized to require as few resources as possible. To understand their working, let us focus on the first sub-iteration. Fix $\boldsymbol{B}, \boldsymbol{C}$ and optimize the norm distance over $\boldsymbol{A} \in \mathbb{R}^{d_1 \times R}$. The Euclidean norm has an interesting feature. It is invariant under re-arranging tensor indices. This in particular includes matricitations. This together with the first identity in Eq. (6) allows for isolating the contribution of $\boldsymbol{A}$ to the norm difference:

$$\|\boldsymbol{t} - [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]\| = \Big\|\boldsymbol{T}_{(1)} - \boldsymbol{A}(\boldsymbol{B} * \boldsymbol{C})^T\Big\|.$$

Since $\boldsymbol{B}, \boldsymbol{C}$ and $\boldsymbol{T}_{(1)}$ are fixed, minimizing this (Frobenius) norm distance reduces to a simple least squares problem. The optimal solution is well-known and corresponds to

$$\boldsymbol{A}_{\sharp} = \boldsymbol{T}_{(1)}\Big((\boldsymbol{B}_0 * \boldsymbol{C})^T\Big)^{\dagger},$$

where $\dagger$ denotes the Moore-Penrose pseudo-inverse. Computing this pseudo-inverse is the most expensive step in the update $\boldsymbol{A} \to \boldsymbol{A}^{\sharp}$. The cost scales polynomially in

the dimension $d_2 d_3 R$ of $\boldsymbol{B} * \boldsymbol{C}$. The identity (3) allows for significantly reducing this dimension:

$$\boldsymbol{A}_\sharp = \boldsymbol{T}_{(1)}(\boldsymbol{B} * \boldsymbol{C})\Big(\boldsymbol{B}^T\boldsymbol{B} \odot \boldsymbol{C}^T\boldsymbol{T}\Big)^\dagger. \tag{7}$$

This re-formulation only requires computing the pseudo-inverse of $\boldsymbol{B}^T\boldsymbol{B} \odot \boldsymbol{C}^T\boldsymbol{C}$ – a $R \times R$ matrix.

After we have updated $\boldsymbol{A}$ according to (7), we move on to optimizing over $\boldsymbol{B}$ exclusively. Choosing a different matriciation – $\boldsymbol{T}_{(2)}$ in this case – allows for isolation $\boldsymbol{B}$ and repeating the least squares arguments from before. A similar analysis extends to the optimization over $\boldsymbol{C}$ exclusively, where the right matriciation is $\boldsymbol{T}_{(3)}$.

Lecture 14: The Tucker decomposition for tensors

Scribe: Richard Kueng

ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp
May 15, 2019

1 Agenda

1. Prelude: different views on matrix factorization
2. The Tucker decomposition
   (a) Definition and Motivation
   (b) Algorithms
   (c) Specifications

2 Prelude: different views on matrix factorization

Matrix factorizations are an indispensable tool for both analytical and numerical linear algebra. The key idea is to decompose a large matrix into smaller constituents that are easier to analyze and work with. As such, matrix factorizations feature prominently in data analysis.

In a nutshell, there are two different views on matrix factorization that are somewhat dual to each other. What is more, both approaches are ultimately based on the singular value decomposition and therefore yield comparable results. This equivalence, however is broken for higher order tensors.

One view gives rise to the CP decomposition, while the other motivatates the Tucker decomposition.

2.1 Independent component analysis (ICA)

Let $\mathbb{R}^{n\times m}$ denote the linear space of real-valued $n \times m$ matrices. This space is the linear hull of all outer products (rank-one matrices):

$$\mathbb{R}^{n\times m} = \left\{ \sum_{i=1}^{r} \boldsymbol{x}_i \boldsymbol{y}_i^T \,:\, \boldsymbol{x}_1, \dots, \boldsymbol{x}_r \in \mathbb{R}^n,\ \boldsymbol{y}_1, \dots, \boldsymbol{y}_r \in \mathbb{R}^m,\ r \in \mathbb{N} \right\}.$$

Outer products are the elementary elements that generate the matrix space. A natural approach to matrix factorization is to find the best elementary fit to a given matrix, remove its contribution and iterate. More precisely, fix $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ and execute the following iterative procedure.

Firstly, identify the best rank-one fit to $\boldsymbol{X}$. This can be obtained my maximizing the Rayleigh quotient:

$$\underset{\boldsymbol{a}\in\mathbb{R}^n, \boldsymbol{b}\in\mathbb{R}^m}{\text{maximize}} \quad \frac{\langle \boldsymbol{a}, \boldsymbol{X}\boldsymbol{b}\rangle}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|} \tag{1}$$

Secondly, $\boldsymbol{X}_{(1)} = \lambda_\sharp \boldsymbol{a}_\sharp \boldsymbol{b}_\sharp^T$ and remove this leading contribution from $\boldsymbol{X}$. I.e. update $\boldsymbol{X} \mapsto \boldsymbol{X} - \boldsymbol{X}_{(1)}$. This step is often called deflation.

This two-step procedure can be repeated r times to obtain a sequence of outer products that approximates $\boldsymbol{X}$ ever more accurately:

$$\boldsymbol{X} \simeq \sum_{i=1}^{r} \boldsymbol{X}_{(i)} = \sum_{i=1}^{r} \lambda_i \boldsymbol{a}_i \boldsymbol{b}_i^T.$$

As r increases, this approximation becomes more accurate and exactly reproduces $\boldsymbol{X}$ once $r = \operatorname{rank}(\boldsymbol{X})$. In many concrete applications, $r \ll \min\{n, m\}$ already provides an excelent approximation.

However, the order of contributions is also important. By construction, Importantly, the relevance of each contribution diminishes with each iteration: $\lambda_i \geq \lambda_{i+1}$ for all $i = 1, \ldots, r-1$.

Vectorization provides a straightforward mapping of this procedure to order-two tensors:

$$\operatorname{vec}(\boldsymbol{X}) \simeq \sum_{i=1}^{r} \lambda_i \boldsymbol{a}_i \otimes \boldsymbol{b}_i \in \mathbb{R}^n \otimes \mathbb{R}^m.$$

We approximate $\boldsymbol{t} = \operatorname{vec}(\boldsymbol{X}) \in \mathbb{R}^n \otimes \mathbb{R}^m$ by a sequence of elementary tensor products $\lambda_i \boldsymbol{a}_i \otimes \boldsymbol{b}_i$. The *CP decomposition* is a natural extension of this factorization approach to higher order tensors. For instance,

$$\boldsymbol{t} \simeq \sum_{i=1}^{r} \lambda_i \boldsymbol{a}_i \otimes \boldsymbol{b}_i \otimes \boldsymbol{c}_i \in \mathbb{R}^n \otimes \mathbb{R}^m \otimes \mathbb{R}^l,$$

where $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_r \in \mathbb{R}^n$, $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_r \in \mathbb{R}^m$, $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_r \in \mathbb{R}^l$ and $\lambda_1, \ldots, \lambda_r \in \mathbb{R}$.

2.2 Principal component analysis (PCA)

Independent component analysis treats $\mathbb{R}^{n \times m}$ as a vector space. It decomposes $\boldsymbol{X}$ into a linear combination of distinguished elements, namely rank-one matrices. However, $\mathbb{R}^{n \times m}$ is more than just a vector space. We can also multiply matrices. This is the starting point for another approach to matrix factorization: approximate $\boldsymbol{X}$ by a product of (smaller) matrices:

$$\boldsymbol{X} \simeq \boldsymbol{V}\boldsymbol{W}^T \quad \text{where} \quad \boldsymbol{V} \in \mathbb{R}^{n \times r},\ \boldsymbol{W} \in \mathbb{R}^{m \times r},\ \boldsymbol{C} \in \mathbb{R}^{r \times r}. \tag{2}$$

Ideally, one chooses $r \ll \min\{n, m\}$ to expose latent structures.

This approach has a long and proud tradition in statistics that dates back to Pearson in 1901. It is called *principal component analysis*. A data matrix $\boldsymbol{X}$ is decomposed into *factors* – the columns of $\boldsymbol{V}$ – and *loadings* – the columns of $\boldsymbol{W}$. The factors isolate core features of the data, while the loadings highlight how these core features need to be combined.

Concrete approximations require a notion of distance on $\mathbb{R}^{n \times m}$. Typically, one chooses the Frobenius norm. PCA then corresponds to choosing a value for r and solving the following optimization problem:

$$\underset{\boldsymbol{V} \in \mathbb{R}^{n \times r},\, \boldsymbol{W} \in \mathbb{R}^{n \times r}}{\text{minimize}} \quad \left\| \boldsymbol{X} - \boldsymbol{V}\boldsymbol{W}^T \right\|.$$

In contrast to the previous factorization approach, it is not so clear how to extend this method to tensors. Tensor product spaces do not have a natural algebra structure – it is not clear how to multiply them.

The following re-interpretation of PCA helps to overcome this challenge, but is also insightful by itself. Matrices can be interpreted as either elements of a vector space, or concrete realizations of a linear operator:

$$\boldsymbol{X} \in \mathbb{R}^{n\times m} \quad \text{vs.} \quad \boldsymbol{X} \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m).$$

We can interpret $\boldsymbol{V}$ in Eq. (2) as a linear operator from a small space $\mathbb{R}^r$ to a much larger space $\mathbb{R}^n$. In contrast, we treat $\boldsymbol{W}^T$ as an element of the (small) vector space $\mathbb{R}^{m\times r}$. This interpretation can be readily extended to order-two tensors:

$$\mathrm{vec}(\boldsymbol{X}) = \underbrace{\boldsymbol{A} \otimes \mathbf{I}}_{\text{blow-up}} \underbrace{\mathrm{vec}(\boldsymbol{V})}_{\text{tensor}}.$$

The core information of a concrete data table is contained in a small matrix – the loadings which determine interactions among final rows and columns. The factors correspond to a blow-up that embed these interactions in a much larger column space.

2.3 Relation between both approaches

For matrices, ICA and PCA are closely related. In fact it is useful to view them as primal and dual approaches to solve the same problem. This close relation is due to the singular value decomposition (SVD) – the royal emperor of all matrix factorizations. This single decomposition solves both ICA and PCA at once. Fix $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ and apply a SVD:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \sum_{i=1}^{\mathrm{rank}(\boldsymbol{X})} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T.$$

Assume that the singular values are arranged in non-increasing order and the vectors $\boldsymbol{u}_i \in \mathbb{R}^n$, $\boldsymbol{v}_i \in \mathbb{R}^m$ are orthogonal and normalized. Then,

$$\langle \boldsymbol{u}_1, \boldsymbol{X}\boldsymbol{v}_1 \rangle = \sum_{i=1}^{\mathrm{rank}(\boldsymbol{X})} \sigma_i \langle \boldsymbol{u}_1, \boldsymbol{u}_i \rangle \langle \boldsymbol{v}_i, \boldsymbol{v}_1 \rangle = \sigma_1.$$

which is the maximum Rayleigh quotient value achievable:

$$\frac{\langle \boldsymbol{a}, \boldsymbol{X}\boldsymbol{b} \rangle}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|} \leq \|\boldsymbol{X}\|_\infty = \sigma_1.$$

This highlights that the rank-one matrix $\sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T$ provides the best rank-one approximation to $\boldsymbol{X}$. Subtracting this contribution and iterating the procedure reveals additional SVD triples $(\sigma_i, \boldsymbol{u}_i, \boldsymbol{v}_i)$. Their order is dictated by the size of the singular values. For r iterations, we obtain the following approximation accuracy:

$$\|\boldsymbol{X} - \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T\|^2 = \sum_{i=r+1}^{\mathrm{rank}(\boldsymbol{X})} \sigma_i^2.$$

The SVD also provides an optimal solution for PCA. Set $\boldsymbol{V}_\sharp = [\sigma_1 \boldsymbol{u}_1, \dots, \sigma_r \boldsymbol{u}_r] \in \mathbb{R}^{n \times r}$ and $\boldsymbol{W}_\sharp = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_r]$. Then,

$$\|\boldsymbol{X} - \boldsymbol{V}_\sharp \boldsymbol{W}_\sharp^T\|^2 = \|\boldsymbol{X} - \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T\|^2 = \sum_{i=r+1}^{\operatorname{rank}(\boldsymbol{X})} \sigma_i^2.$$

Not only, does this approximation accuracy exactly coincide with ICA-value. The Eckart-Young-Mirski Theorem asserts that this value is optimal and cannot be further improved.

This equivalence breaks down for tensors of higher order. To illustrate the main problem, let us consider the following seemingly trivial identity:

$$\sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \otimes \boldsymbol{v}_i = \boldsymbol{U}\boldsymbol{\Sigma} \otimes \boldsymbol{V} \sum_{i=1}^{r} \boldsymbol{e}_i \otimes \boldsymbol{e}_i = \boldsymbol{U}\boldsymbol{\Sigma} \otimes \boldsymbol{V} \operatorname{vec}(\mathbf{I}) = \operatorname{vec}\Big(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\Big) = \operatorname{vec}\Big(\boldsymbol{V}_\sharp \boldsymbol{W}_\sharp^T\Big).$$

The left hand side is the (vectorized) ICA, while the right hand side is the (vectorized) PCA. The steps in between, however, break down for tensors of higher order. We cannot move around operators for free anymore. Consequently, the two approaches obtain a rather different flavor and give rise to the oldest and most classical tensor decompositions:

1. ICA gives rise to the *CP decompostion* – the topic of Lecture 13.
2. PCA gives rise to the *Tucker decomposition* – today's topic.

3 The Tucker decomposition

Once more we shall focus our attention of order three tensors: $\boldsymbol{t} \in \mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \mathbb{R}^{d_3}$. A generalization to higher orders is straightforward, but becomes more involved notation-wise.

The key idea behind the Tucker decomposition is to view $\boldsymbol{t}$ as a blow-up of another order-three tensor that lives in a much smaller space.

Definition 3.1. A tensor $\boldsymbol{t} \in \mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \mathbb{R}^{d_3}$ admits a *Tucker decomposition* of local dimensions (R_1, R_2, R_3) if there are matrices $\boldsymbol{A} \in \mathbb{R}^{d_1 \times R_1}$, $\boldsymbol{B} \in \mathbb{R}^{d_2 \times R_2}$, $\boldsymbol{C} \in \mathbb{R}^{d_3 \times R_3}$ and a tensor $\boldsymbol{g} \in \mathbb{R}^{R_1} \otimes \mathbb{R}^{R_2} \otimes \mathbb{R}^{R_3}$ such that

$$\boldsymbol{t} = \boldsymbol{A} \otimes \boldsymbol{B} \otimes \boldsymbol{C}\boldsymbol{g}.$$

This representation becomes interesting if the local dimensions are much smaller than the ambient dimensions: $r_i \ll d_i$ for $i = 1, 2, 3$. In this case, the *core tensor* $\boldsymbol{g}$ is much smaller than the original tensor $\boldsymbol{t}$. The latter is recovered by blowing up $\boldsymbol{g}$ in different directions. Importantly, these blow-ups are assumed to be independent:

$$\boldsymbol{A} \otimes \boldsymbol{B} \otimes \boldsymbol{C} \in \mathcal{L}\Big(\mathbb{R}^{R_1}, \mathbb{R}^{d_1}\Big) \otimes \mathcal{L}\Big(\mathbb{R}^{R_2}, \mathbb{R}^{d_2}\Big) \otimes \mathcal{L}\Big(\mathbb{R}^{R_3}, \mathbb{R}^{d_3}\Big).$$

It is instructive, to expand the Tucker decomposition further. Write $\boldsymbol{A} = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_{R_1}]$, $\boldsymbol{B} = [\boldsymbol{b}_1, \dots, \boldsymbol{b}_{R_2}]$, $\boldsymbol{C} = [\boldsymbol{c}_1, \dots, \boldsymbol{c}_{R_3}]$ and interpret $\boldsymbol{g} \in \mathbb{R}^{R_1} \otimes \mathbb{R}^{R_2} \otimes \mathbb{R}^{R_3}$ as a ternary array: $[g_{i,j,k}] \in \mathbb{R}^{R_1 \times R_2 \times R_3}$. Then,

$$\boldsymbol{t} = \sum_{i=1}^{R_1} \sum_{j=1}^{R_2} \sum_{k=1}^{R_3} g_{ijk} \boldsymbol{a}_i \otimes \boldsymbol{b}_j \otimes \boldsymbol{c}_k.$$

Remark 3.2. The *CP decomposition* is a special case of a Tucker decomposition. Set $R_1 = R_2 = R_3 = R$ and fix $\boldsymbol{g} = \boldsymbol{h}_r = \sum_{i=1}^{r} \boldsymbol{e}_i \otimes \boldsymbol{e}_i \otimes \boldsymbol{e}_i$ – the Hadamard tensor. Then,

$$\boldsymbol{A} \otimes \boldsymbol{B} \otimes \boldsymbol{C}\boldsymbol{h}_R = \sum_{i=1}^{r} \boldsymbol{a}_i \otimes \boldsymbol{b}_i \otimes \boldsymbol{c}_i.$$

The above remark highlights that the Tucker decomposition is more flexible than the CP decomposition. It also has a natural interpretation in terms of data compression. This becomes interesting when choosing $r \ll d_1$, $r \ll d_2$ and $R_3 \ll d_3$ still results in an accurate approximation of the original tensor. The "big tensor" $\boldsymbol{t} \in \mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \mathbb{R}^{d_3}$ arises from blowing up a much smaller tensor that mediates interactions between different factors. Hence, $\boldsymbol{g}$ may be viewed as a tensor generalization of the loadings matrix in PCA. The individual factors $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ describe how these original interactions are embedded in a much larger space.

Although not necessary, one typically assumes that the blow-ups $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ are isometric embeddings. In other words, they correspond to matrices with orthogonal columns normalized to unit length.

Example 3.3 (Tucker decomposition for matrices)**.** Identify $\mathbb{R}^{d_1 \times d_2}$ with the tensor product $\mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2}$. The exact correspondence is provided by vectorization and its inverse. Applying this inverse to a tucker decomposition of an order two-tensor yields

$$\boldsymbol{X} = \mathrm{vec}^{-1}(\boldsymbol{A} \otimes \boldsymbol{B}\boldsymbol{g}) = \boldsymbol{A}\mathrm{vec}(\boldsymbol{g})^{-1}\boldsymbol{B}^T = \boldsymbol{A}\boldsymbol{G}\boldsymbol{V}^T.$$

Importantly the core matrix $\boldsymbol{G} = \mathrm{vec}^{-1}(\boldsymbol{g}) \in \mathbb{R}^{r_1 \times R_2}$ need not be diagonal. This additional flexibility may allow for achieving accurate approximation with even smaller internal degrees of freedom than the SVD (there, the core matrix $\boldsymbol{\Sigma}$ is necessarily diagonal).

3.1 Specifications

The Tucker decomposition is rather general and flexible. Several specifications of the Tucker decomposition have become popular in the data analysis literature.

3.1.1 Tucker2 decomposition

Set one of the blow-ups to be the identity matrix. For instance, choose $R_3 = d_3$ and fix $\boldsymbol{C} = \mathbf{I}$. A Tucker2 decomposition of a tensor is

$$\boldsymbol{t} = \boldsymbol{A} \otimes \boldsymbol{B} \otimes \mathbf{I}\boldsymbol{g}.$$

The tensor $\boldsymbol{g}$ itself mediates interactions between the first two tensor factors and the third. This may be viewed as a PCA with an additional tensor product constraint on the factor matrix. To see this, consider a matriciation that leaves the third tensor factor invariant:

$$\boldsymbol{t}_{(3)} = \boldsymbol{g}_{(3)}\boldsymbol{A}^T \otimes \boldsymbol{B}^T \quad \text{and consequently} \quad \boldsymbol{t}_{(3)}^T = \boldsymbol{A} \otimes \boldsymbol{B}\boldsymbol{g}_{(3)}^T.$$

As advertised, $\boldsymbol{g}_{(3)}^T \in \mathbb{R}^{d_1 d_2 \times d_3}$ is a matrix that mediates correlations between the first two tensor factors and the final one.

3.1.2 Tucker1 decomposition

The Tucker1 decomposition is an even more radical extension of the previous restriction. Set two of the blow-ups to be the identity. For instance, choose $R_2 = d_2$, $R_3 = d_3$ and fix $\boldsymbol{B} = \mathbf{I}$, as well as $\boldsymbol{C} = \mathbf{I}$. This specification recovers the traditional PCA for $d_1 \times d_2 d_3$ matrices. Choose a matriciation for the first component and observe

$$\boldsymbol{t}_{(1)} = \boldsymbol{A}\boldsymbol{g}_{(1)} \quad \text{where} \quad \boldsymbol{A} \in \mathbb{R}^{d_1 \times R_1},\ \boldsymbol{g}_{(1)} \in \mathbb{R}^{R_1 \times d_2 d_3}.$$

3.2 Computing the Tucker decomposition

It should not come as a surprise that the problem of computing Tucker decompositions is challenging. We will now describe an alternating least squares algorithm (ALS) that attempts to find good Tucker approximations of a given order-three tensor. The approximation quality is measured in the Euclidean norm induced by the extended standard inner product:

$$\|\boldsymbol{t}\|^2 = \langle \boldsymbol{t}, \boldsymbol{t} \rangle = \sum_{i,i'} \sum_{j,j'} \sum_{k,k'} t_{ijk} t_{i'j'k'} \langle \boldsymbol{e}_i, \boldsymbol{e}_{i'} \rangle \langle \boldsymbol{e}_j, \boldsymbol{e}_{j'} \rangle \langle \boldsymbol{e}_k, \boldsymbol{e}_{k'} \rangle = \sum_{i,j,k} t_{ijk}^2.$$

ALS-type algorithms are a common heuristic to address complicated, multi-objective optimization problems. The core idea is to fix all but one variable and optimize over the remaining variable while expending as few computational resources as possible. Sweeping across all different variables results in an update for each contribution and the cheapness of each step allows for iterating this procedure many times.

On first sight, the problem of finding a Tucker decomposition is similar to computing the CP decomposition of a tensor. However, there are two core differences: (i) the core tensor $\boldsymbol{g} \in \mathbb{R}^{r_1} \otimes \mathbb{R}^{R_2} \otimes \mathbb{R}^{R_3}$ is a new optimization variable. (ii) the blow-ups $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are assumed to be isometries.

Nonetheless, the final ALS algorithm looks rather similar to the one we developed for the CP decomposition. Pseudo-code for it is provided in Algorithm1.

Let us start the discussion of the algorithm with the update for the core tensor $\boldsymbol{g}$. Suppose that $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ are fixed isometries. Isometric invariance of the Euclidean norm readily implies

$$\|\boldsymbol{t} - \boldsymbol{A} \otimes \boldsymbol{B} \otimes \boldsymbol{C}\boldsymbol{g}\| = \Big\|\boldsymbol{A}^T \otimes \boldsymbol{B}^T \otimes \boldsymbol{C}^T\boldsymbol{t} - \boldsymbol{g}\Big\|.$$

Clearly, this expression is minimized if we choose

$$\boldsymbol{g} = \boldsymbol{A}^T \otimes \boldsymbol{B}^T\boldsymbol{C}^T\boldsymbol{t} \in \mathbb{R}^{r_1} \otimes \mathbb{R}^{R_2} \otimes \mathbb{R}^{R_3}.$$

input : tensor $\boldsymbol{t} \in \mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \mathbb{R}^{d_3}$, inner dimensions (R_1, R_2, R_3)
Initialize isometries $\boldsymbol{A} \in \mathbb{R}^{d_1 \times R_1}$, $\boldsymbol{B} \in \mathbb{R}^{d_2 \times R_1}$, $\boldsymbol{C} \in \mathbb{R}^{d_3 \times R_3}$ and $N_{\max}$;
while $i < N_{\max}$ **do**
  $\boldsymbol{g} \leftarrow \boldsymbol{A}^T \otimes \boldsymbol{B}^T \otimes \boldsymbol{C}^T \boldsymbol{t}$;
  Compute SVD of $\boldsymbol{T}_{(1)}\boldsymbol{B} \otimes \boldsymbol{C}$, extract top left singular vectors and update $\boldsymbol{A} \leftarrow [\boldsymbol{u}_1, \dots, \boldsymbol{u}_{R_1}]^T$;
  Compute SVD of $\boldsymbol{T}_{(2)}\boldsymbol{A} \otimes \boldsymbol{C}$,extract top left singular vectors and update $\boldsymbol{B} \leftarrow [\boldsymbol{u}_1, \dots, \boldsymbol{u}_{R_2}]^T$;
  Compute SVD of $\boldsymbol{T}_{(3)}\boldsymbol{A} \otimes \boldsymbol{B}$, extract top left singular vectors and update $\boldsymbol{C} \leftarrow [\boldsymbol{u}_1, \dots, \boldsymbol{u}_{R_3}]^T$;
  (Break loop if a certain stopping condition is reached);
  $i \leftarrow i + 1$;
end
output : matrix factors $\boldsymbol{A} \in \mathbb{R}^{d_1 \times R}$, $\boldsymbol{B} \in \mathbb{R}^{d_2 \times R}$, $\boldsymbol{C}^{d_3 \times R}$

Algorithm 1: Alternating least squares (ALS) algorithm for approximating the Tucker decomposition.

This simple update rule fully takes care of the core tensor and we can restrict our attention to optimizing the isometries individually. Suppose that $\boldsymbol{g}$ is of the form (3.2). Then, the fact that $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are isometries implies

$$\begin{aligned}\|\boldsymbol{t} - \boldsymbol{A} \otimes \boldsymbol{B} \otimes \boldsymbol{C}\boldsymbol{g}\|_2^2 =& \langle \boldsymbol{t}, \boldsymbol{t} \rangle - 2\langle \underbrace{\boldsymbol{A}^T \otimes \boldsymbol{B}^T \otimes \boldsymbol{C}^T}_{\boldsymbol{g}} \boldsymbol{t}, \boldsymbol{g} \rangle + \langle \boldsymbol{g}, \underbrace{\boldsymbol{A}^T\boldsymbol{A}}_{\mathbf{I}} \otimes \underbrace{\boldsymbol{B}^T\boldsymbol{B}}_{\mathbf{I}} \otimes \underbrace{\boldsymbol{C}^T\boldsymbol{C}}_{\mathbf{I}} \boldsymbol{g} \rangle \\ =& \|\boldsymbol{t}\|^2 - 2\langle \boldsymbol{g}, \boldsymbol{g} \rangle + \langle \boldsymbol{g}, \boldsymbol{g} \rangle = \|\boldsymbol{t}\|^2 - \langle \boldsymbol{g}, \boldsymbol{g} \rangle.\end{aligned}$$

The first contribution $\|\boldsymbol{t}\|^2$ is fixed and constant. In turn, minimizing the Euclidean distance to $\boldsymbol{t}$ is equivalent to maximizing the norm of the core tensor $\boldsymbol{g} = \boldsymbol{A}^T \otimes \boldsymbol{B}^T \otimes \boldsymbol{C}^T \boldsymbol{t}$. If we keep $\boldsymbol{B}$,$\boldsymbol{C}$ fixed, the single-objective optimization over the remaining isometry becomes

$$\max_{\boldsymbol{A} \in \mathbb{R}^{d_1 \times r_1}} \left\| \boldsymbol{A}^T \otimes \boldsymbol{B}^T \otimes \boldsymbol{C}^T \boldsymbol{t} \right\|^2 \quad \text{subject to} \quad \boldsymbol{A}^T\boldsymbol{A} = \mathbf{I}.$$

We can isolate the contribution of $\boldsymbol{A}^T$ by choosing a particular matriciation (recall that the Euclidean norm remains invariant under re-grouping of indicies):

$$\left\| \boldsymbol{A}^T \otimes \boldsymbol{B}^T \otimes \boldsymbol{C}^T \boldsymbol{t} \right\|^2 = \left\| \boldsymbol{A}^T \boldsymbol{T}_{(1)} \boldsymbol{B} \otimes \boldsymbol{C} \right\|_F^2$$

Here, $\|\cdot\|_F$ denotes the Frobenius norm (the natural Euclidean norm for matrices). Principal component analysis tells us how me maximize this expression over all isometries. Simply compute an SVD $\boldsymbol{t} \otimes \boldsymbol{B} \otimes \boldsymbol{B} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \in \mathbb{R}^{d_1 \times R_2 R_3}$, collect the leading R_1 singular vectors and transpose:

$$\boldsymbol{A}_\sharp^T = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_{R_1}] \in \mathbb{R}^{R_1 \times d_1}.$$

The computational cost of this update is governed by the SVD. We emphasize that our approach to the problem results in the SVD of a $d_1 \times R_2 R_3$-matrix. The (potentially large) dimensions d_2 and d_3 do not feature at all.

The same idea allows for updating the other isometries $\boldsymbol{B}$ and $\boldsymbol{C}$ in an analogous fashion. The only difference is that other matriciations are used to isolate the contributions of a given isometry ($\boldsymbol{T}_{(2)}$ for $\boldsymbol{B}$ and $\boldsymbol{T}_{(3)}$ for $\boldsymbol{C}$).

Lecture 15: Tensor train decomposition I

Scribe: Richard Kueng

ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp
May 20, 2019

1 Agenda

1. Motivation: the Schmidt decomposition for order-two tensors
2. Tensor train decomposition: keep applying the Schmidt decomposition sequentially
3. Examples of tensor train representations

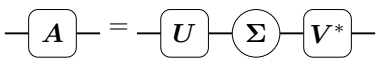
2 Motivation: Schmidt decompositions of order-2 tensors

2.1 Recapitulation: The SVD

Let $\boldsymbol{A} \in \mathcal{L}(\mathbb{C}^{d_2}, \mathbb{C}^{d_1})$ be an operator, or equivalently: a complex-valued $d_1 \times d_2$ matrix. Every such operator admits a singular value decomposition:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*.$$

Here, $\boldsymbol{\Sigma} \in \mathbb{R}^{r\times r}$ is a diagonal matrix that collects the singular values and $\boldsymbol{U} \in \mathbb{C}^{d_1\times r}$, $\boldsymbol{V} \in \mathbb{C}^{d_2\times r}$ are isometries. In wiring notation, we write



and use a round box to notationally underline the diagonal nature of $\boldsymbol{\Sigma}$.

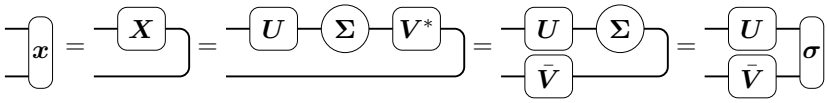
The SVD by itself may already be a compression of the original matrix. Suppose that $\boldsymbol{A}$ has rank $r \ll \min\{d_1, d_2\}$. Then, the inner lines denote indices that live in $\mathbb{C}^r$, rather than $\mathbb{C}^{d_1}$ (left), or $\mathbb{C}^{d_2}$ (right). Even if this is not the case, we may obtain a compressed approximation by truncating the inner index dimension from $r = \operatorname{rank}(\boldsymbol{A})$ to $R \leq r$. Define $\boldsymbol{\Sigma}_{(R)} = \operatorname{diag}(\sigma_1, \ldots, \sigma_R, 0, \ldots, 0)$ and set $\boldsymbol{A}_{(R)} = \boldsymbol{U}\boldsymbol{\Sigma}_{(R)}\boldsymbol{V}^*$. Isometric invariance of the Frobenius norm then implies

$$\left\|\boldsymbol{A} - \boldsymbol{A}_{(R)}\right\|_F^2 = \left\|\boldsymbol{U}\Big(\boldsymbol{\Sigma} - \boldsymbol{\Sigma}_{(R)}\Big)\boldsymbol{V}^*\right\|_F^2 = \sum_{i=R+1}^{r} \sigma_i(\boldsymbol{A})^2. \tag{1}$$

The Eckart-Young-Mirski Theorem asserts that this rank-R approximation is optimal.

2.2 The Schmidt decomposition of order two tensors

The SVD and its truncated version readily extend to tensors of order two. Recall that we may identify the space of complex-valued $d_1 \times d_2$ matrices with $\mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}$. The precise correspondence is provided by *vectorization* ($\boldsymbol{x}\boldsymbol{y}^T \mapsto \boldsymbol{x} \otimes \boldsymbol{y}$) and the outer product representation ($\boldsymbol{x} \otimes \boldsymbol{y} \mapsto \boldsymbol{x}\boldsymbol{y}^T$). Identify $\boldsymbol{x} \in \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}$ with a $d_1 \times d_2$ matrix $\boldsymbol{X}$ and apply an SVD:



Here, we have implicitly defined $\boldsymbol{\sigma} = \mathrm{vec}(\boldsymbol{\Sigma}) = \sum_{i=1}^{r} \sigma_i \boldsymbol{e}_i \otimes \boldsymbol{e}_i \in \mathbb{C}^r \otimes \mathbb{C}^r$. It is worthwhile to formulate this decomposition formula explicitly without wiring diagrams. Set $r = \mathrm{rank}(\boldsymbol{X})$ and decompose the isometries into columns: $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r] \in \mathbb{C}^{d_1 \times r}$, $\bar{\boldsymbol{V}} = [\bar{\boldsymbol{v}}_1, \ldots, \bar{\boldsymbol{v}}_r] \in \mathbb{C}^{d_2 \times r}$. Then, the above decomposition reads

$$\boldsymbol{x} = \boldsymbol{U} \otimes \bar{\boldsymbol{V}} \mathrm{vec}(\boldsymbol{\Sigma}) = \sum_{i=1}^{r} \boldsymbol{U} \otimes \bar{\boldsymbol{V}} \sum_{i=1}^{r} \sigma_i \boldsymbol{e}_i \otimes \boldsymbol{e}_i = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \otimes \bar{\boldsymbol{v}}_i \in \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}. \tag{2}$$

Definition 2.1. The decomposition (2) is called a *Schmidt decomposition*. The parameter $r = \mathrm{rank}(\mathrm{vec}^{-1}(\boldsymbol{x}))$ is called the *Schmidt-rank*.

Schmidt decompositions are an important tool in quantum information theory. They features prominently in the study of bi-partite entanglement. The following properties follow directly from the SVD.

Proposition 2.2. *The Schmidt decomposition* (2) *has many desirable features:*

1. *the weights $\sigma_1, \ldots, \sigma_r$ are strictly positive,*
2. *$\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r\}$ is a set of r orthonormal vectors in $\mathbb{C}^{d_1}$,*
3. *$\{\bar{\boldsymbol{v}}_1, \ldots, \bar{\boldsymbol{v}}_r\}$ is a set of r orthonormal vectors in $\mathbb{C}^{d_2}$.*

Another important feature of Schmidt-decomposition is compressed approximation. The following claim is an immediate consequence of Equation (1).

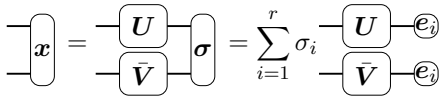
Corollary 2.3. *Fix $\boldsymbol{x} \in \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}$ and $R \in \mathbb{N}$. Then, the truncated Schmidt decomposition $\boldsymbol{x}_{(R)} = \sum_{i=1}^{R} \sigma_i \boldsymbol{u}_i \otimes \bar{\boldsymbol{v}}_i$ is the best (tensor) rank-R approximation of $\boldsymbol{x}$. It achieves*

$$\|\boldsymbol{x} - \boldsymbol{x}_{(R)}\|^2 = \sum_{i=R+1}^{r} \sigma_i^2.$$

where, $\|\cdot\|$ denotes the Euclidean norm on $\mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}$.

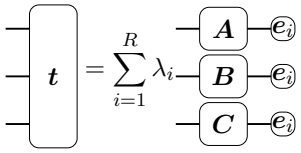
2.3 Context: relation between Schmidt and CP+Tucker decompositions

The Schmidt decomposition (2) is a natural starting point for cornering tensor decompositions. For order-two tensors it reads



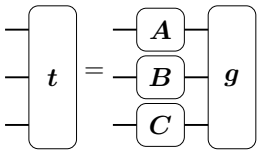
(3)

The *CP decomposition* generalizes the final expression to higher order tensors:



$$= \sum_{i=1}^{R} \lambda_i \boldsymbol{a}_i \otimes \boldsymbol{b}_i \otimes \boldsymbol{c}_i \in \mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \mathbb{R}^{d_3}.$$

The *Tucker decomposition* is based on a generalization of the second expression in Eq. (3):



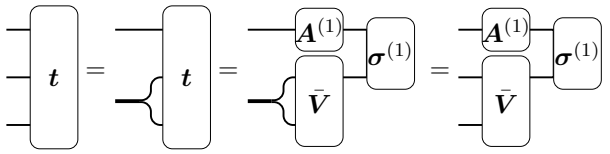
$$= \sum_{i=1}^{R_1} \sum_{j=1}^{R_2} \sum_{k=1}^{R_3} g_{ijk} \boldsymbol{a}_i \otimes \boldsymbol{b}_j \otimes \boldsymbol{c}_k \in \mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \mathbb{R}^{d_3}.$$

While both decomositions coincide for order-two tensors – see Eq. (3) – they obtain a unique genuine tensor flavor when extended to higher order. Moreover, each generalization comes at a price: Corollary 2.3 does not generalize. It is not clear how to truncate CP and Tucker decompositions in an optimal fashion. The tensor train decomposition – the main topic of the remaining lectures – is designed to preserve this optimal compression property.

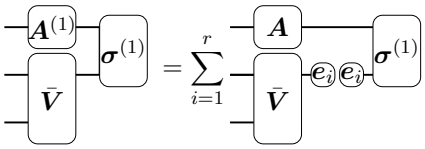
3 The tensor train decomposition

3.1 Derivation of tensor train representations

The Schmidt decomposition (2) provides a way to factorize order-two tensors into sums of elementary tensors. We can naively apply it to an order-three tensor $\boldsymbol{t} \in \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2} \otimes \mathbb{C}^{d_3}$ by grouping the second and third tensor factor together: Interpret $\mathbb{C}^{d_2} \otimes \mathbb{C}^{d_3}$ as a single complex vector space $\mathbb{C}^{d_2 d_3}$ of much larger dimension. In wiring notation,

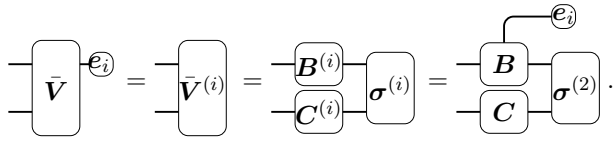


where we have set $\boldsymbol{A}^{(1)} = \boldsymbol{U} \in \mathbb{C}^{d_1 \times r}$. Here, r denotes the Schmidt rank of $\boldsymbol{t}$ viewed as an order-two tensor. A single Schmidt decomposition allows for decoupling the first tensor factor from the rest. But nothing stops us from repeating this procedure. Insert a resolution of the identity to obtain



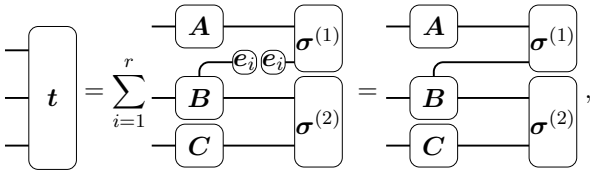
(4)

and apply a Schmidt decomposition to each $\bar{\boldsymbol{V}} \boldsymbol{e}_i \in \mathbb{C}^{d_2} \otimes \mathbb{C}^{d_3}$ individually:

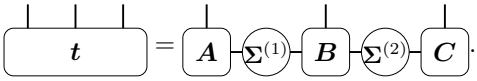


(5)

The last reformulation is a mathematical trick. We absorb the index dependence into an additional tensor degree of freedom: $\boldsymbol{B}$ corresponds to an order-three tensor that implicitly takes care of the labeling. Inserting Eq. (5) into Eq. (4) yields

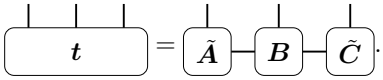


where we have absorbed the resolution of the identity. We can now bend indices and use $\boldsymbol{\sigma}^{(i)} = \mathrm{vec}(\boldsymbol{\Sigma}^{(i)})$ to represent the wiring diagram on the right hand side in a more symmetric fashion. To further increase readability, we also rotate wiring diagrams by 90 degrees:



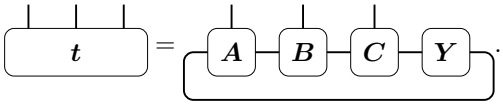
(6)

This is a *tensor train* decomposition. We can further pinpoint the underlying tensor structure by absorbing the diagonal matrices $\boldsymbol{\Sigma}^{(1)}$ and $\boldsymbol{\Sigma}^{(2)}$ into one of the other tensor components:

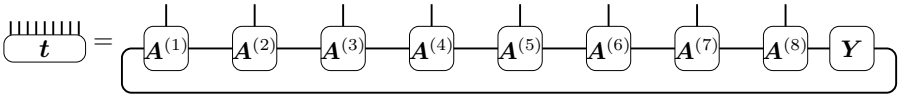


(7)

The *tensor train* decomposition factorizes a general order-three tensor into a "train" of three individual tensors: the boundary is comprised of order two-tensors $\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{C}}$ while the center corresponds to an order three tensor $\boldsymbol{B}$. This asymmetry can be defined away by another minor modification. Include an additional tensor factor on each end of the train and combine them with a trace operation:



The matrix $\boldsymbol{Y}$ provides an additional degree of freedom. Choosing an outer product $\boldsymbol{Y} = \boldsymbol{y}_l \boldsymbol{y}_r^*$ effectively recovers the asymmetric tensor train from (7). An extension of this decomposition to tensors of arbitrary order is now straightforward, e.g. for order-8 tensors we obtain



(8)

A single order three-tensor – a "wagon"– represents each tensor factor. These wagons are connected by an internal line that represents a virtual degree of freedom. This line connects all the wagons as well as the single matrix $\boldsymbol{Y}$ at the top – the "locomotive".

3.2 Definition of tensor trains and key features

Definition 3.1. A tensor train (TT) is a tensor $\boldsymbol{t} \in \mathbb{C}^{d_1} \otimes \cdots \otimes \mathbb{C}^{d_n}$ that is fully characterized by an array of N order three tensors $\left\{\boldsymbol{A}^{(i)}\right\} \in \mathbb{C}_i^D \otimes \mathbb{C}^{D_{i+1}} \otimes \mathbb{C}^{d_i}$ and a single matrix $\boldsymbol{Y} \in \mathbb{C}^{D_{N+1} \times D_1}$:

$$\boldsymbol{t} = \sum_{i_1=1}^{d_1} \cdots \sum_{i_N=1}^{d_N} \operatorname{tr}\Big(\boldsymbol{A}_{::i_1}^{(1)} \boldsymbol{A}_{::i_2}^{(2)} \cdots \boldsymbol{A}_{::i_N}^{(N)} \boldsymbol{Y}\Big) \boldsymbol{e}_{i_1} \otimes \boldsymbol{e}_{i_2} \otimes \cdots \otimes \boldsymbol{e}_{i_N}. \tag{9}$$

Here, $\boldsymbol{A}_{::i_k}^{(k)} \in \mathbb{C}^{D_i \times D_{i+1}}$ denote the frontal matrix slices of $\boldsymbol{A}^{(k)}$ with respect to the third factor.

Note that we may express each order three tensor $\boldsymbol{A}^{(k)}$ completely by its frontal slices:

$$\boldsymbol{A}_{i_k}^{(k)} = \boldsymbol{A}_{::i_k}^{(k)} \quad \text{for} \quad 1 \leq i_k \leq d_k.$$

This re-formulation is not only convenient notation-wise. It also highlights the origin of an alternative name for tensor trains.

Remark 3.2 (Alternative name: matrix product state (MPS)). According to Eq. (9), each expansion coefficient of $\boldsymbol{t}$ with respect to the extended standard basis is a trace of a product of matrices: $\boldsymbol{t}_{i_1 \cdots i_N} = \operatorname{tr}\Big(\boldsymbol{A}_{i_1}^{(1)} \cdots \boldsymbol{A}_{i_N}^{(N)} \boldsymbol{Y}\Big)$. In quantum mechanics, vectors are typically associated with pure states: Up to normalization, $\boldsymbol{\rho} = \boldsymbol{t}\boldsymbol{t}^*$ describes a joint pure state of N quantum mechanical systems. These two features together explain an alternative nomenclature from quantum mechanics: *matrix product states*.

It is instructive to introduce the following summary parameters for external and internal dimensions:

$$d = \max_{1 \leq i \leq N} d_i \quad \text{and} \quad D = \max_{1 \leq i \leq N+1} D_i.$$

The maximum internal dimension D is called the *bond dimension*. The dimension of the tensor product $\mathbb{C}^{d_1} \otimes \cdots \otimes \mathbb{C}^{d_N}$ is then roughly d^N, while the number of degrees of freedom in a TT is

$$\deg(\mathrm{TT}) = NdD^2 + D^2 = (Nd+1)D^2. \tag{10}$$

The tensor train representation is complete: every tensor $\boldsymbol{t} \in \mathbb{C}^{d_1} \otimes \cdots \otimes \mathbb{C}^{d_N}$ can be represented as a TT. This expressiveness, however, does not come cheap. In general, the internal dimensions D must scale *exponentially* in the number of tensor factors:

$$D \simeq \frac{d^{(N-1)/2}}{\sqrt{N}}.$$

This relation readily follows from comparing (10) to the overall tensor space dimension d^N. It should not come as a surprise – dimensions of tensor products grow very quickly. Nonetheless, such a general scaling is prohibitively expensive. Why would we want

to represent individual expansion coefficients of an order N tensor as the trace of a product of N $d^{N/2} \times d^{N/2}$ matrices?

The true advantage of the TT formalism stems from manually trimming the internal dimension to a much smaller value. Suppose that D only scales *polynomially* in the number of tensor factors N. Then, the associated TT is described by

$$\deg(\mathrm{TT}) = C_d \mathrm{poly}(N)$$

degrees of freedom – an exponential compression. What is more, the connection to the SVD exactly tells us how we have to trim a general tensor train: isolate the diagonal singular value matrices between the trains, see Eq. (6), and truncate them to only contain the D largest singular values. The Eckart-Young-Mirski Theorem ensures that this compression is optimal – even for large tensor products. The incurred approximation error is bounded by the singular values that we cut out.

3.3 Examples and and a non-example

3.3.1 Elementary tensor product

Consider the elementary tensor product $\boldsymbol{t} = \boldsymbol{e}_1 \otimes \cdots \otimes \boldsymbol{e}_1 \in \mathbb{C}^d \otimes \cdots \otimes \mathbb{C}^d$. Then, its expansion coefficients with respect to the extended standard basis are

$$[\boldsymbol{t}]_{i_1,\ldots,i_N} = \delta_{i_1,1} \cdots \delta_{i_N,1}.$$

These admit a particularly concise tensor train decomposition. Choose $D = 1$ (trivial bond dimension) and set $\boldsymbol{A}_j = \delta_{1,j} \in \mathbb{C} \simeq \mathbb{C}^{1\times 1}$ for all $1 \leq j \leq d$ and $\boldsymbol{Y} = 1 \in \mathbb{C} = \mathbb{C}^{1\times 1}$. Then,

$$\mathrm{tr}(\boldsymbol{A}_{i_1} \boldsymbol{A}_{i_2} \cdots \boldsymbol{A}_{i_N} \boldsymbol{Y}) = \delta_{i_1,1} \cdots \delta_{i_N,1} = [\boldsymbol{t}]_{i_1,\ldots,i_N}.$$

More general elementary tensor products $\boldsymbol{x}_1 \otimes \cdots \otimes \boldsymbol{x}_N$ can be constructed in a similar fashion.

3.3.2 The GHZ state

Define the following tensor $\boldsymbol{A} \in \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$ ($d = D = 2$) via its frontal slices

$$\boldsymbol{A}_1 = \boldsymbol{A}_{::1} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \boldsymbol{A}_2 = \boldsymbol{A}_{::2} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

and set $\boldsymbol{Y} = \mathbf{I} \in \mathbb{C}^{2\times 2}$. Then, this collection defines a tensor train on $(\mathbb{C}^d)^{\otimes N}$ with bond dimension D. Note that $\boldsymbol{A}_i^2 = \boldsymbol{A}_i$ for $i = 1, 2$ and $\boldsymbol{A}_1 \boldsymbol{A}_0 = \mathbf{0} \in \mathbb{C}^{2\times 2}$. Therefore,

$$\begin{aligned} \boldsymbol{t} &= \sum_{i_1=1}^{2} \cdots \sum_{i_N=1}^{2} \mathrm{tr}(\boldsymbol{A}_{i_1} \cdots \boldsymbol{A}_{i_N} \mathbf{I}) \boldsymbol{e}_{i_1} \otimes \cdots \otimes \boldsymbol{e}_{i_N} \\ &= \sum_{i_1=1}^{2} \cdots \sum_{i_N=1}^{2} \delta_{i_1=\cdots=i_N} \boldsymbol{e}_{i_1} \otimes \cdots \otimes \boldsymbol{e}_{i_N} = \boldsymbol{e}_1^{\otimes N} + \boldsymbol{e}_2^{\otimes N}. \end{aligned}$$

This describes a highly structured tensor product vector that is associated to a prominent pure quantum state $\boldsymbol{\rho} = \boldsymbol{t}\boldsymbol{t}^*/2$ – the Greenberger–Horne–Zeilinger (GHZ) state.

3.3.3 The W-state

Define the following tensor $\boldsymbol{A} \in \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$ ($d = D = 2$) via its frontal slices:

$$\boldsymbol{A}_1 = \boldsymbol{A}_{::1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \boldsymbol{I},\ \boldsymbol{A}_2 = \boldsymbol{A}_{::2} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad \text{and set} \quad \boldsymbol{Y} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Note that $\boldsymbol{A}_1^2 = \boldsymbol{A}_1$, $\boldsymbol{A}_1\boldsymbol{A}_2 = \boldsymbol{A}_2$ and $\boldsymbol{A}_2\boldsymbol{A}_2 = \boldsymbol{0}$. Moreover, $\operatorname{tr}(\boldsymbol{A}_1\boldsymbol{Y}) = 0$ and $\operatorname{tr}(\boldsymbol{A}_2\boldsymbol{X}) = 1$. These elementary relations fully characterize the following tensor train with bond dimension D on $(\mathbb{C}^2)^{\otimes N}$:

$$\begin{aligned}\boldsymbol{t} =& \sum_{i_1=1}^{2} \cdots \sum_{i_N=1}^{2} \operatorname{tr}(\boldsymbol{A}_{i_1} \cdots \boldsymbol{A}_{i_N}\boldsymbol{Y}) \boldsymbol{e}_{i_1} \otimes \cdots \otimes \boldsymbol{e}_{i_N} \\ =& \boldsymbol{e}_2 \otimes \boldsymbol{e}_1^{\otimes(N-1)} + \boldsymbol{e}_1 \otimes \boldsymbol{e}_2 \otimes \boldsymbol{e}_1^{\otimes(N-2)} + \cdots + \boldsymbol{e}_1^{\otimes(N-1)} \otimes \boldsymbol{e}_1.\end{aligned}$$

Up to normalization, this TT produces a pure quantum state that is an equal superposition of all possible permutations of $\boldsymbol{e}_2 \otimes \boldsymbol{e}_1^{\otimes(N-1)}$:

$$\boldsymbol{t} = N! \boldsymbol{P}_{\vee N} \boldsymbol{e}_2 \otimes \boldsymbol{e}_1 \otimes \cdots \otimes \boldsymbol{e}_1.$$

The associated (pure) quantum state is called the W-state. It features prominently in the study of multi-partite entanglement.

3.4 A random element of $(\mathbb{C}^d)^{\otimes N}$

Set $d_1 = \ldots = d_N = d$ and consider a Haar-random unit vector in $(\mathbb{C}^d)^{\otimes N} \simeq \mathbb{C}^{Nd}$:

$$\boldsymbol{u} \sim \mathbb{S}\Big(\mathbb{C}^{Nd}\Big).$$

The parameter counting argument from (10) suggests that an exponentially large bond dimension $D \simeq d^{N/2}$ is required to accurately approximate this generic tensor. This is indeed the case. To see this, divide the N tensor factors into two families. The first N_1 factors are grouped into family A, while the remaining $N - N_1$ factors belong to family B. Haar integration implies the following concentration identity:

$$\mathbb{E}\left\| \operatorname{tr}_B(\boldsymbol{u}\boldsymbol{u}^*) - \frac{1}{d^{N_1}} \boldsymbol{I} \right\|_2 < d^{-(N-N_1)/2}.$$

We refer to Homework Sheet II for details. Moreover, Levi's Lemma asserts that the norm deviation of any concrete realization of $\boldsymbol{u}$ will concentrate sharply around this expected value. Next, choose $N_1 = N/3$. This ensures $d^{N_1} \geq d^{(N-N_1/2)}$ and in turn,

$$\left\| d^{N_1} \operatorname{tr}_B(\boldsymbol{u}\boldsymbol{u}^*) - \mathbf{I} \right\|_2 < 1 \quad \text{with overwhelming probability.}$$

Next, note that we may express $\mathbf{I}$ as a sum of $\dim(A) = d^{N_1}$ outer products: $\mathbf{I} = \sum_i \boldsymbol{v}_i \boldsymbol{v}_i^*$. Inserting this into the norm bound demands

$$\left\| d^{N_1} \operatorname{tr}_B(\boldsymbol{u}\boldsymbol{u}^*) - \sum_{i=1}^{d^{N_1}} \boldsymbol{v}_i \boldsymbol{v}_i^* \right\|_2 < 1 \quad \text{with overwhelming probability.}$$

This has profound consequences. The partial trace $\mathrm{tr}_B(\boldsymbol{u}\boldsymbol{u}^*)$ must approximate each of the d^{N_1} outer products to accuracy strictly larger than one. This is only possible, if the TT representation gives rise to at least d^{N_1} different outer products when taking the partial trace. In turn, this imposes a lower bound on the bond dimension that connects the system A with the system B:

$$D_{N_1,N_1+1} \geq d^{N_1} = d^{N/3}.$$

We thus conclude that the largest bond dimension for expressing a generic vector must grow exponentially in the number of tensor factors. This argument may be readily extended to lower bound *any* virtual degree of freedom: $D_i \geq d^{N/3}$ for all $1 \leq i \leq N$. Indeed, a random vector $\boldsymbol{u} \in \mathbb{S}^{d^N}$ does not care about the specific ordering of tensor factors and we may permute them at will.

Lecture 16: Tensor train decomposition II

Scribe: Richard Kueng

ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp
May 22, 2019

1 Agenda

1. Recapitulation: tensor trains (TT)/matrix product states (MPS)
2. TT properties and symmetries
3. Exponential decay of correlations accross a tensor

2 Recapitulation: tensor trains/ matrix product states

2.1 Definition of tensor trains

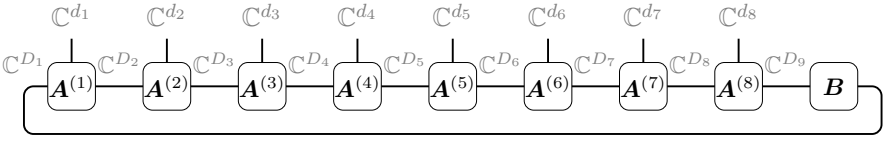
Tensor trains (TT) are a decomposition of order-N tensors. They arise from sequentially applying Schmidt decompositions to separate individual tensor factors one at a time. This approach is rather different from other prominent tensor decompositions and has very unique features. Over the last decades, TT have become a highly useful tool in quantum physics, as well as machine learning.

Definition 2.1 (tensor train (TT))**.** A tensor train representation of $\boldsymbol{t} \in \mathbb{C}^{d_1} \otimes \cdots \otimes \mathbb{C}^{d_N}$ is characterized by an array of N order-three tensors $\boldsymbol{A}^{(1)} \in \mathbb{C}^{D_1} \otimes \mathbb{C}^{D_2} \otimes \mathbb{C}^{d_1}, \ldots, \boldsymbol{A}^{(N)} \in \mathbb{C}^{D_N} \otimes \mathbb{C}^{D_{N+1}} \otimes \mathbb{C}^{d_N}$, and a single matrix $\boldsymbol{Y} \in \mathbb{C}^{D_{N+1} \times D_1}$:

$$\boldsymbol{t} = \sum_{i_1=1}^{d_1} \cdots \sum_{i_N=1}^{d_N} \operatorname{tr}\Big(\boldsymbol{A}^{(1)}_{::i_1} \boldsymbol{A}^{(2)}_{::i_2} \cdots \boldsymbol{A}^{(N)}_{::i_N} \boldsymbol{Y}\Big) \boldsymbol{e}_{i_1} \otimes \cdots \otimes \boldsymbol{e}_{i_N} \in \mathbb{C}^{d_1} \otimes \cdots \otimes \mathbb{C}^{d_N}. \quad (1)$$

The elements $\boldsymbol{A}^{(k)}_{::i_k} \in \mathbb{C}^{D_k \times D_{k+1}}$ denote the frontal slices of the tensor $\boldsymbol{A}^{(k)}$, and $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{d_k}$ denotes the standard basis of $\mathbb{C}^{d_k}$.

The TT decomposition becomes exceptionally clear in wiring notation. Here is an example of a general TT for $N = 8$ factors:



The internal dimensions $D_1, \ldots, D_{N+1}$ are often called *bond dimensions*. Their size does not feature in the final tensor expressions. Viewed from this angle, the bond dimensions are "virtual" degrees of freedom. In contrast, the dimensions $d_1, \ldots, d_N$ are fixed and in one-to-one correspondence with the tensor product space $\mathbb{C}^{d_1} \otimes \cdots \otimes \mathbb{C}^{D_N}$ – the actual object of interest. As such, these dimensions are often called *physical dimensions*, because they carry an actual meaning.

Remark 2.2 (Alternative name: matrix product states (MPS)). The expansion (1) is characterized by the trace of a product of N matrices – the frontal slices associated with the tensor trains. Moreover, the outer product $\boldsymbol{t}\boldsymbol{t}^*$ is proportional to a pure state of a joint quantum system. For these reasons, tensor trains are typically called *matrix product states* in the physics literature.

TT are only valuable if the bond dimension scales moderately in the number of tensor factors. A simple parameter counting argument highlights that this is a substantial restriction. A generic tensor will require an exponentially large bond dimension for accurate representation.

In summary: TT can only efficiently represent a tiny fraction of all elements in a very large tensor space. However, this small fraction contains tensor products with very reasonable behavior that reflects our intuition about how physically meaningful tensors should behave. One such feature – exponential decay of correlations – will be covered later in today's lecture.

2.2 Additional definitions and Physics jargon

The first definition addresses tensor product spaces, independent of concrete tensor representations.

Definition 2.3 (thermodynamic limit for tensor products)**.** Set $d_1 = \cdots = d_N = d$. The *thermodynamic limit* is the limit of infinitely large tensor products of $\mathbb{C}^d$:

$$\lim_{N\to\infty} \left(\mathbb{C}^d\right)^{\otimes N}.$$

The thermodynamic limit may be viewed as a mathematical idealization. The study of many tensor-related properties – as well as tensors themselves – often becomes easier. A concrete example are moments of random variables and, more generally, polynomials. We know from earlier lectures that both are related to the symmetric subspace of $(\mathbb{C}^d)^{\otimes N}$. High order moments tend to become more and more well-behaved and regular. Likewise, polynomials of very high degrees accurately approximate smooth functions (Taylor's theorem) which are often easier to work with.

Within physics, the thermodynamic limit arises naturally when one tries to approximate infinite dimensional Hilbert spaces in a discrete fashion. Informally speaking, it marks the transition between matrix analysis and functional analysis. Traditional quantum mechanics is phrased in this language. Interestingly, the first introduction of tensor trains / matrix product states was phrased in this language (Fannes, Nachtergaele, Werner, *Finitely correlated states on quantum spin chains*, 1992). Only later, Frank Verstraete (then at Caltech) and others discretized this observation and popularized the TT/MPS framework in its current finite-dimensional form.

Definition 2.4 (Translation invariant tensor trains)**.** A tensor train $\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(N)}, \boldsymbol{Y}$ is *translation invariant* if all order-three tensors are the same

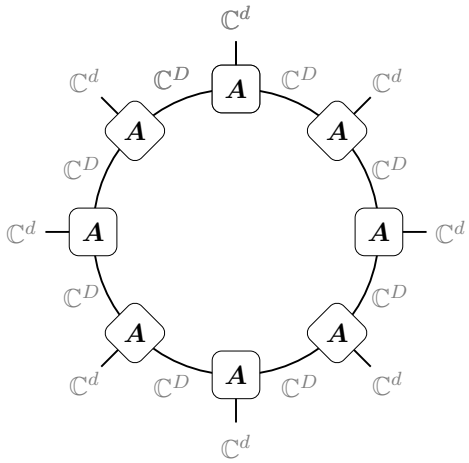
$$\boldsymbol{A}^{(i)} = \boldsymbol{A}^{(j)} \in \mathbb{C}^D \otimes \mathbb{C}^D \otimes \mathbb{C}^d \quad \text{for all} \quad 1 \leq i, j \leq d.$$

This in particular implies $d_1 = d_2 = \cdots d_N$ and $D_1 = D_2 = \cdots = D_{N+1} = D$.

Translation invariance substantially reduces the complexity of a TT decomposition. It naturally arises in applications, where the individual tensor factors are indistinguishable, e.g. a quantum system describing N identical particles on a line. Symmetric tensors $\boldsymbol{t} \in \bigvee^N \left(\mathbb{C}^d\right)$ are another promising candidate for such a substantial simplification. Such tensors arise naturally when one considers the moment distribution of data – e.g. frequencies of words in topic related texts.

Definition 2.5 (Periodic boundary conditions)**.** A tensor train $\boldsymbol{A}^{(1)}, \dots, \boldsymbol{A}^{(N)}, \boldsymbol{Y}$ is said to have *periodic boundary conditions* if $D_{N+1} = D_1$ and $\boldsymbol{Y} = \mathbf{I}$.

This nomenclature has a geometric origin. The left and right-most constituents of a TT are connected by a trace. If $\boldsymbol{B} = \mathbf{I}$, there is no discontinuity when moving from the N-th train to the first. Effectively, the trains form a circle, not a line. Periodic boundary conditions go well with translation invariance. Combining both assumptions results in a circle of identical trains with physical indices pointing outwards:



2.3 Uniqueness of tensor train decompositions

Recall that tensor factorizations are typically unique. Kruskal's theorem provides a rather mild condition that ensures that the minimal rank decomposition of a tensor – i.e the optimal *CP decomposition* – is unique up to trivial ambiguities. This is not the case for matrix factorizations. If $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^*$ is a matrix factorization, then so is $\boldsymbol{U}\boldsymbol{R}\boldsymbol{R}^{-1}\boldsymbol{V}^*$ for any invertible matrix $\boldsymbol{R} \in \mathbb{C}^{r \times r'}$ $(r' \geq r)$. Tensor train decompositions behave in a similar fashion. This lack of uniqueness should not come as a surprise. After all, we developed tensor train decomposition by sequentially applying Schmidt decompositions. The latter simple correspond to matriciating tensors and applying a matrix SVD.

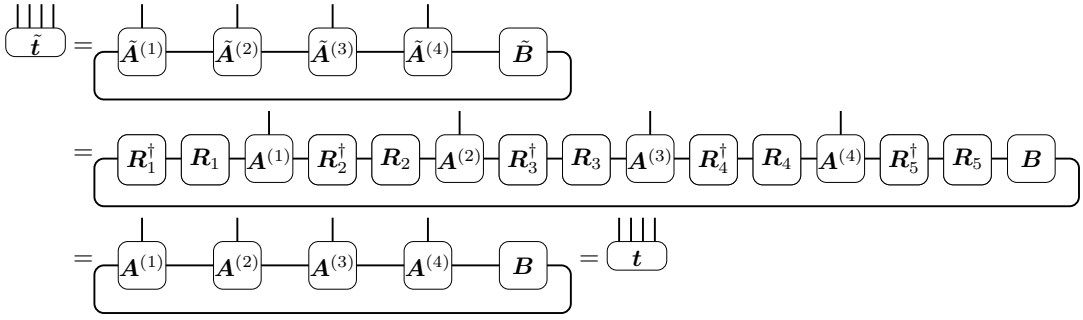
Theorem 2.6 (Gauge freedom)**.** *Let $\boldsymbol{A}^{(1)}, \dots, \boldsymbol{A}^{(N)}, \boldsymbol{Y}$ be a TT with physical dimensions $d_1, \dots, d_N$ and bond dimensions $D_1, \dots, D_{N+1}$. Choose $a_1 \geq D_1, \dots, a_{N+1} \geq D_{N+1}$ and $\boldsymbol{R}_1 \in \mathbb{C}^{a_1 \times D_1}, \dots, \boldsymbol{R}_{N+1} \in \mathbb{C}^{r_{N+1} \times D_{N+1}}$ such that each matrix admits a left inverse,*

i.e. $\boldsymbol{R}_i^\dagger \boldsymbol{R}_i = \boldsymbol{I} \in \mathbb{C}^{D_i \times D_i}$. *Then, the transformations*

$$\begin{aligned} \boldsymbol{A}^{(k)}_{::i_k} &\mapsto \boldsymbol{R}_k \boldsymbol{A}^{(k)}_{::i_k} \boldsymbol{R}_{k+1}^\dagger \quad \textit{for all} \quad 1 \leq i_k \leq d_k, \\ \boldsymbol{Y} &\mapsto \boldsymbol{R}_{N+1} \boldsymbol{Y} \boldsymbol{R}_1^\dagger \end{aligned}$$

do not affect the associated tensor $\boldsymbol{t} \in \mathbb{C}^{d_1} \otimes \ldots \otimes \mathbb{C}^{d_N}$.

Proof. With wiring diagrams. We will address the case $N = 4$, but the proof readily generalizes. Let $\tilde{\boldsymbol{A}}^{(1)}, \ldots, \tilde{\boldsymbol{A}}^{(4)}$ and $\boldsymbol{B}$ denote the descriptions that result from such a transformation. Then,



□

Physicists call this freedom a *gauge freedom*. It is exploited by several state-of-the art algorithms that use tensor trains. It allows for converting the order three tensors $\boldsymbol{A}^{(k)}$ into a standard form that greatly reduces the cost of computing tensor contractions.

3 Exponential decay of correlations within tensor trains

We have already alluded to physically well-motivated structures that seem to be hidden within the TT formalism. Chief among them is the following feature: Correlations between individual tensor factors decay exponentially with their mutual distance.

3.1 Conditional expectation value

In order to rigorously state this claim, we need a bit of additional notation.

Definition 3.1 (Conditional expectation value)**.** Fix a tensor $\boldsymbol{t} \in \mathbb{C}^{d_1} \otimes \cdots \otimes \mathbb{C}^{d_N}$ and an operator $\boldsymbol{A}$ acting on this tensor product space. The *conditional expectation value* of $\boldsymbol{A}$ with respect to $\boldsymbol{t}\boldsymbol{t}^*$ is

$$\langle \boldsymbol{A} \rangle_{\boldsymbol{t}\boldsymbol{t}^*} = \frac{\operatorname{tr}(\boldsymbol{A}\boldsymbol{t}\boldsymbol{t}^*)}{\langle \boldsymbol{t}, \boldsymbol{t} \rangle} = \frac{\langle \boldsymbol{t}, \boldsymbol{A}\boldsymbol{t} \rangle}{\langle \boldsymbol{t}, \boldsymbol{t} \rangle}.$$

The origin of this notion hails from quantum mechanics. The normalized outer product $\boldsymbol{t}\boldsymbol{t}^*/\langle \boldsymbol{t}, \boldsymbol{t} \rangle$ describes a pure quantum state of the joint system comprised of N (potentially different) quantum mechanical systems.

The operator $\boldsymbol{A}$ may correspond to an element in a quantum mechanical measurement: $\boldsymbol{A} = \boldsymbol{H}_{\lambda_k} \succeq \boldsymbol{0}$, where $\sum_k \boldsymbol{H}_{\lambda_k} = \mathbf{I}$. In this case, the conditional expectation value tells us the probability for obtaining outcome λ_k when measuring $\boldsymbol{\rho} = \boldsymbol{t}\boldsymbol{t}^*$.

3.2 Aside: quantum measurements vs. observables

Many physicists like to combine a quantum measurement (resolution of the identity) with the associated outcomes to obtain a single hermitian operator. This operator is called an *observable*:

$$\boldsymbol{O} = \sum_k \lambda_k \boldsymbol{H}_k.$$

The conditional expectation value of an observable then corresponds to the expected measurement outcome:

$$\langle \boldsymbol{O} \rangle_{\boldsymbol{t}\boldsymbol{t}^*} = \mathrm{tr}\left(\sum_k \lambda_k \boldsymbol{H}_k \boldsymbol{t}\boldsymbol{t}^* \right) = \sum_k \lambda_k \mathrm{tr}(\boldsymbol{H}_k \boldsymbol{t}\boldsymbol{t}^*) = \sum_k \lambda_k \mathrm{Pr}[\lambda_k | \boldsymbol{t}\boldsymbol{t}^*].$$

Observables often have a concrete physical interpretation, like energy, or spin. The conditional expectation value of an observable corresponds to the expected size of the associated physical quantity. Single-shot measurement results may differ from this expectation.

3.3 2-point correlators

In quantum mechanics, tensor products arise naturally when studying joint quantum systems. Every tensor factor corresponds to a microscopic quantum system. Resource constraints typically prevent us from performing joint measurements on all N systems simultaneously. Instead, we may restrict our measurement effort to few select systems and ignore (marginalize over) the rest. The following short-hand notation captures this feature:

$$\begin{aligned}\hat{\boldsymbol{O}}_j :=& \boldsymbol{I}_{d_1 \times d_1} \otimes \boldsymbol{I}_{d_2 \times d_2} \cdots \otimes \boldsymbol{I}_{d_{j-1} \times d_{j-1}} \otimes \boldsymbol{O}_j \otimes \boldsymbol{I}_{d_{j+1} \times d_{j+1}} \otimes \cdots \otimes \boldsymbol{I}_{d_N \times d_N}, \\ \hat{\boldsymbol{O}}_k :=& \boldsymbol{I}_{d_1 \times d_1} \otimes \boldsymbol{I}_{d_2 \times d_2} \cdots \otimes \boldsymbol{I}_{d_{k-1} \times d_{k-1}} \otimes \boldsymbol{O}_k \otimes \boldsymbol{I}_{d_{k+1} \times d_{k+1}} \otimes \cdots \otimes \boldsymbol{I}_{d_N \times d_N}\end{aligned}$$

Each of these operators only acts non-trivially on the j-th and k-th tensor factor, respectively. For $j \neq k$, the product $\hat{\boldsymbol{O}}_j \hat{\boldsymbol{O}}_k = \hat{\boldsymbol{O}}_k \hat{\boldsymbol{O}}_j$ commutes and acts non-trivially on both the j-th and the k-th tensor factor.

One of the most fascinating aspects of quantum mechanics is that measurements necessarily affect the quantum system in question. An interaction – e.g. a measurement – with the j-th system may affect the joint quantum state of all constituents. The following definition provides a way to probe this effect:

Definition 3.2 (2-point correlator)**.** For a tensor $\boldsymbol{t} \in \mathbb{C}^{d_1} \otimes \cdots \otimes \mathbb{C}^{d_N}$ and operators $\boldsymbol{O}_j \in \mathbb{C}^{d_j \times d_j}$, $\boldsymbol{O}_k \in \mathbb{C}^{d_k \times d_k}$ we define the *2-point correlator*:

$$\langle \hat{\boldsymbol{O}}_j \hat{\boldsymbol{O}}_k \rangle_{\boldsymbol{t}\boldsymbol{t}^*} - \langle \hat{\boldsymbol{O}}_j \rangle_{\boldsymbol{t}\boldsymbol{t}^*} \langle \hat{\boldsymbol{O}}_k \rangle_{\boldsymbol{t}\boldsymbol{t}^*}$$

Two point correlators allow us to probe the spread of local perturbations within the joint quantum system. Suppose that we poke the j-th system and are interested in estimating how severely the k-th system is affected by this interaction. Then, $\langle \hat{\boldsymbol{O}}_j \hat{\boldsymbol{O}}_k \rangle_{\boldsymbol{t}\boldsymbol{t}^*}$ is a meaningful measure to address this question. It vanishes if and only if $\hat{\boldsymbol{O}}_j$ has no influence on the conditional expectation value $\langle \hat{\boldsymbol{O}}_k \rangle_{\boldsymbol{t}\boldsymbol{t}^*}$. Otherwise, it is strictly larger and upper-bounds the strength of correlations between the j-th and the k-th subsystem

3.4 Tensor trains and exponential decay of 2-point correlators

Theorem 3.3. *Let $\boldsymbol{t} \in \mathbb{C}^{d_1} \otimes \cdots \otimes \mathbb{C}^{d_N}$ be a tensor that admits a translationally invariant TT representation with periodic boundary conditions that is also* injective[1]*. Then, 2-point correlators decay exponentially with distance:*

$$\left| \langle \hat{\boldsymbol{O}}_j \hat{\boldsymbol{O}}_k \rangle_{\boldsymbol{t}\boldsymbol{t}^*} - \langle \hat{\boldsymbol{O}}_j \rangle_{\boldsymbol{t}\boldsymbol{t}^*} \langle \hat{\boldsymbol{O}}_k \rangle_{\boldsymbol{t}\boldsymbol{t}^*} \right| \leq \mathrm{poly}(d, D, \|\boldsymbol{O}_j\|_\infty, \|\boldsymbol{O}_k\|_\infty) \mathrm{e}^{-c|j-k|}. \tag{2}$$

Note that this result is only meaningful if the bond dimension is moderate, i.e. $D = C_d \mathrm{poly}(N)$. Otherwise, the pre-factor could absorb any exponential decay in distance. Before proving this result, it is worthwhile to point out a strong converse.

Theorem 3.4 (Brandao, Horodecki; 2015)**.** *Let $\boldsymbol{\rho} = \boldsymbol{t}\boldsymbol{t}^* / \langle \boldsymbol{t}, \boldsymbol{t} \rangle$ be a pure joint quantum state of N "identical" systems. Suppose that all 2-point correlators decay exponentially in the sense of Eq. (2). Then, $\boldsymbol{t} \in \mathbb{C}^d \otimes \cdots \otimes \mathbb{C}^d$ is well-approximated by a TT with polynomial bond dimension $D = C_d \mathrm{poly}(N)$.*

3.5 Proof of Exponential decay of correlations in tensor trains

We will present a self-contained proof of Theorem 3.3 that is based on several simplifying assumptions:

1. Translation invariance: $\boldsymbol{A}^{(i)} = \boldsymbol{A}^{(j)}$ for all $1 \leq i, j \leq N$. This also ensures $d_1 = \cdots = d_N = d$ and $D_1 = \cdots = D_{N+1} = D$.

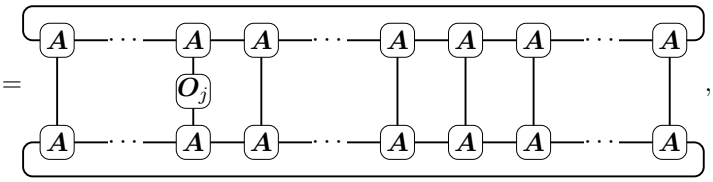
2. Periodic boundary conditions: $\boldsymbol{B} = \mathbf{I}$,

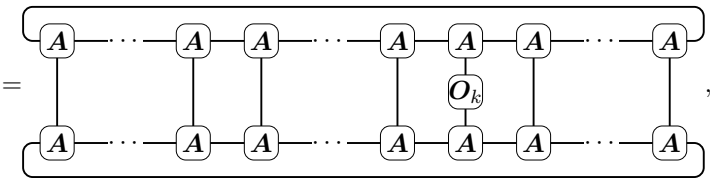
3. Thermodynamic limit: we will assume N to be "very big" (think $N \to \infty$), but refrain from a rigorous limit analysis.

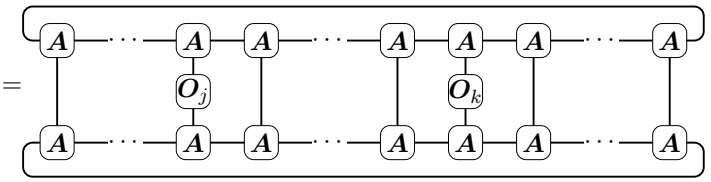
We will also need a fourth assumption – injectivity. We will introduce it later, once we need it. Also, we assume without loss that $\boldsymbol{t}$ is normalized: $\langle \boldsymbol{t}, \boldsymbol{t} \rangle = 1$. For now, let us

[1] We refer to Definition 3.5 below for a precise definition of this term.

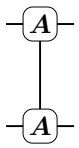
rewrite the expressions of interest in wiring notation:

$\langle \hat{\boldsymbol{O}}_j \rangle_{\boldsymbol{t}\boldsymbol{t}^*} =$  ,

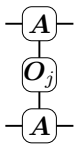
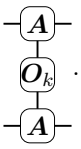
$\langle \hat{\boldsymbol{O}}_j \rangle_{\boldsymbol{t}\boldsymbol{t}^*} =$  , (3)

$\langle \hat{\boldsymbol{O}}_j \hat{\boldsymbol{O}}_k \rangle_{\boldsymbol{t}\boldsymbol{t}^*} =$  (4)

Cyclicity of the trace allows us to assume without loss $j < k$ – as indicated in the diagrams above. If we read these expressions from left to right, they reveal a lot of structure. We are dealing with traces of N operators that act on $\mathbb{C}^D \otimes \mathbb{C}^D$. What is more, translation invariance ensures that almost all of them are the same. This central building block is called a *transfer matrix*:

$\boldsymbol{T} =$  $= \sum_{i=1}^{d} \boldsymbol{A}_{::i} \otimes \boldsymbol{A}^*_{::i} \in \mathcal{L}\Big(\mathbb{C}^D \otimes \mathbb{C}^D\Big)$

All but two (one) constituents are such transfer matrices. The remaining two operators are

$\boldsymbol{T}_{\boldsymbol{O}_j} =$  $\in \mathcal{L}\Big(\mathbb{C}^D \otimes \mathbb{C}^D\Big)$ and $\boldsymbol{T}_{\boldsymbol{O}_j} =$  .

This new notation considerably simplifies the conditional expectation values. Inserting these matrix definitions into the wiring diagram expressions readily yields

$$
\begin{aligned}
\langle \hat{\boldsymbol{O}}_j \rangle_{\boldsymbol{t}\boldsymbol{t}^*} &= \mathrm{tr}\Big(\boldsymbol{T}^{j-1} \boldsymbol{T}_{\boldsymbol{O}_j} \boldsymbol{T}^{N-j-1}\Big) = \mathrm{tr}\Big(\boldsymbol{T}^{N-1} \boldsymbol{T}_{\boldsymbol{O}_j}\Big), \\
\langle \hat{\boldsymbol{O}}_k \rangle_{\boldsymbol{t}\boldsymbol{t}^*} &= \mathrm{tr}\Big(\boldsymbol{T}^{k-1} \boldsymbol{T}_{\boldsymbol{O}_k} \boldsymbol{T}^{N-k-1}\Big) = \mathrm{tr}\Big(\boldsymbol{T}^{N-1} \boldsymbol{T}_{\boldsymbol{O}_k}\Big), \\
\langle \hat{\boldsymbol{O}}_j \hat{\boldsymbol{O}}_k \rangle_{\boldsymbol{t}\boldsymbol{t}^*} &= \mathrm{tr}\Big(\boldsymbol{T}^{j-1} \boldsymbol{T}_{\boldsymbol{O}_j} \boldsymbol{T}^{k-j-1} \boldsymbol{T}_{\boldsymbol{O}_k} \boldsymbol{T}^{N-k-1}\Big) = \mathrm{tr}\Big(\boldsymbol{T}^{N-2-(k-j)} \boldsymbol{T}_{\boldsymbol{O}_j} \boldsymbol{T}^{k-j-1} \boldsymbol{T}_{\boldsymbol{O}_k}\Big).
\end{aligned}
$$

Here, we have also used cyclicity of the trace to further simplify these expressions. We are now ready to phrase our fourth assumption:

4 Injectivity: The transfer matrix $\boldsymbol{T}$ is diagonalizable and it has a unique largest eigenvalue of one. All other eigenvalues are smaller in modulus.

The assumption that $\lambda_{\max} = 1$ readily follows from normalization. The power method demands for $N \to \infty$ (thermodynamic limit)

$$1 = \langle \boldsymbol{t}, \boldsymbol{t} \rangle = \mathrm{tr}\Big(\boldsymbol{T}^N\Big) \simeq \lambda_{\max}^N \quad \text{which ensures} \quad \lambda_{\max} = 1.$$

In contrast, uniqueness of the largest eigenvalue is a more severe demand that is essential for exponential decay of correlations.

Definition 3.5. A translationally invariant TT is *injective* if the transfer matrix is diagonalizable and has a unique largest eigenvalue.

Apply an eigenvalue decomposition $\boldsymbol{T} = \boldsymbol{W}\boldsymbol{D}\boldsymbol{W}^{-1}$, where $\boldsymbol{D} = \mathrm{diag}(\lambda_{\max}, \lambda_2, \ldots, \lambda_{D^2})$ and $|\lambda_i| < \lambda_{\max} = 1$ for all $2 \leq i \leq D^2$. We can again imply the power method to conclude

$$\begin{aligned}\lim_{N\to\infty} \boldsymbol{T}^N = \lim_{N\to\infty} \boldsymbol{W}\boldsymbol{D}^N\boldsymbol{W}^{-1} = \boldsymbol{W} \lim_{N\to\infty} \mathrm{diag}\Big(1^N, \lambda_2^N, \ldots, \lambda_{D^2}^N\Big)\boldsymbol{W}^{-1} \\ = \boldsymbol{W}\mathrm{diag}(1, 0, \ldots, 0)\boldsymbol{W}^{-1} = \boldsymbol{w}_l\boldsymbol{w}_r^*,\end{aligned}$$

where we have implicitly defined $\boldsymbol{w}_l = \boldsymbol{W}\boldsymbol{e}_1 \in \mathbb{C}^D$ and $\boldsymbol{w}_r = (\boldsymbol{W}^{-1})^*\boldsymbol{e}_1 \in \mathbb{C}^d$. Injectivity together with the thermodynamic limit ensure that high powers of the transfer matrix with itself approach an outer product.

This insight allows us to considerably simplify the 2-point correlator:

$$\begin{aligned}\langle \hat{\boldsymbol{O}}_j\hat{\boldsymbol{O}}_k\rangle_{\boldsymbol{t}\boldsymbol{t}^*} - \langle \hat{\boldsymbol{O}}_j\rangle_{\boldsymbol{t}\boldsymbol{t}^*}\langle \hat{\boldsymbol{O}}_k\rangle_{\boldsymbol{t}\boldsymbol{t}^*} =& \mathrm{tr}\Big(\boldsymbol{T}^{N-2-(k-j)}\boldsymbol{T}_{\boldsymbol{O}_j}\boldsymbol{T}^{k-j-1}\boldsymbol{T}_{\boldsymbol{O}_k}\Big) - \mathrm{tr}\Big(\boldsymbol{T}^{N-1}\boldsymbol{T}_{\boldsymbol{O}_j}\Big)\mathrm{tr}\Big(\boldsymbol{T}^{N-1}\boldsymbol{T}_{\boldsymbol{O}_k}\Big) \\ \simeq& \mathrm{tr}\Big(\boldsymbol{T}^{\infty}\boldsymbol{T}_{\boldsymbol{O}_j}\boldsymbol{T}^{k-j-1}\boldsymbol{T}_{\boldsymbol{O}_k}\Big) - \mathrm{tr}\Big(\boldsymbol{T}^{\infty}\boldsymbol{T}_{\boldsymbol{O}_j}\Big)\mathrm{tr}(\boldsymbol{T}^{\infty}\boldsymbol{T}_{\boldsymbol{O}_k}) \\ =& \boldsymbol{w}_r^*\boldsymbol{T}_{\boldsymbol{O}_j}\boldsymbol{T}^{k-j-1}\boldsymbol{T}_{\boldsymbol{O}_k}\boldsymbol{w}_l - \boldsymbol{w}_r^*\boldsymbol{T}_{\boldsymbol{O}_j}\boldsymbol{w}_l\boldsymbol{w}_r^*\boldsymbol{T}_{\boldsymbol{O}_k}\boldsymbol{w}_l \\ =& \boldsymbol{w}_r^*\boldsymbol{T}_{\boldsymbol{O}_j}\boldsymbol{T}^{k-j-1}\boldsymbol{T}_{\boldsymbol{O}_k}\boldsymbol{w}_l - \boldsymbol{w}_r^*\boldsymbol{T}_{\boldsymbol{O}_j}\boldsymbol{T}^{\infty}\boldsymbol{T}_{\boldsymbol{O}_k}\boldsymbol{w}_l \\ =& \boldsymbol{w}_r^*\boldsymbol{T}_{\boldsymbol{O}_j}\Big(\boldsymbol{T}^{k-j-1} - \boldsymbol{T}^{\infty}\Big)\boldsymbol{T}_{\boldsymbol{O}_k}\boldsymbol{w}_l.\end{aligned}$$

We are now almost done. Exponential decay readily follows from the following observation:

$$\begin{aligned}\boldsymbol{T}^{k-j-1} - \boldsymbol{T}^{\infty} =& \boldsymbol{W}\Big(\boldsymbol{D}^{k-j-1} - \boldsymbol{D}^{\infty}\Big)\boldsymbol{W}^{-1} = \boldsymbol{W}\mathrm{diag}\Big(1^{k-j-1} - 1, \lambda_2^{k-j-1}, \ldots, \lambda_{D^2}^{k-j-1}\Big)\boldsymbol{W}^{-1} \\ =& \boldsymbol{W}\mathrm{diag}\Big(0, \lambda_2^{k-j-1}, \ldots, \lambda_{D^2}^{k-j-1}\Big)\boldsymbol{W}^{-1}.\end{aligned}$$

Since $|\lambda_i| < 1$ for all $2 \leq i \leq D^2$, this matrix difference decays exponentially in any matrix norm. A quick look at the expression above highlights that control of the

operator norm suffices:

$$\begin{aligned}\left|\langle \hat{\boldsymbol{O}}_j \hat{\boldsymbol{O}}_k \rangle_{\boldsymbol{t}\boldsymbol{t}^*} - \langle \hat{\boldsymbol{O}}_j \rangle_{\boldsymbol{t}\boldsymbol{t}^*} \langle \hat{\boldsymbol{O}}_k \rangle_{\boldsymbol{t}\boldsymbol{t}^*}\right| \leq & \left| \boldsymbol{e}_1^* \boldsymbol{W}^{-1} \boldsymbol{T}_{\boldsymbol{O}_j} \boldsymbol{W} \left(\boldsymbol{D}^{k-j-1} - \boldsymbol{D}^{\infty} \right) \boldsymbol{W}^{-1} \boldsymbol{T}_{\boldsymbol{O}_k} \boldsymbol{W} \boldsymbol{e}_1 \right| \\ \leq & \|\boldsymbol{W}^{-1} \boldsymbol{T}_{\boldsymbol{O}_j} \boldsymbol{W} \boldsymbol{e}_1\|_{\ell_2} \|\boldsymbol{W}^{-1} \boldsymbol{T}_{\boldsymbol{O}_k} \boldsymbol{W} \boldsymbol{e}_1\|_{\ell_2} \left\| \boldsymbol{D}^{k-j-1} - \boldsymbol{D}^{\infty} \right\|_{\infty} \\ \leq & \mathrm{poly}(D, d, \|\boldsymbol{O}_j\|_{\infty}, \|\boldsymbol{O}_k\|_{\infty}) \max_{2 \leq i \leq D^2} |\lambda_i|^{k-j-1}.\end{aligned}$$

This establishes the relation advertised in Theorem 3.3.

Lecture 17: Tensor train algorithms (DMRG lite)

Scribe: Richard Kueng

ACM 270-1, Spring 2019
Richard Kueng & Joel Tropp
May 29, 2019

1 Agenda

1. Problem statement: compute ground state energies of joint quantum systems
2. Concrete examples
3. Tensor train ansatz
4. Matrix product operators
5. Alternate tensor train minimization (DMRG1)
6. Extension (DMRG2) and rigorous convergence guarantees

2 Problem statement: compute ground state energies of joint quantum systems

2.1 Recapitulation: quantum states

Consider the set of joint quantum states on N identical systems – each with local dimension d:

$$\mathsf{S}\Big(\mathbb{C}^d)^{\otimes N}\Big) = \Big\{\boldsymbol{\rho} \in \mathcal{L}(H^{\otimes N}) : \ \boldsymbol{\rho} \succeq \mathbf{0}, \ (\mathbf{I}, \boldsymbol{\rho}) = 1\Big\}.$$

Here, $(\boldsymbol{X}, \boldsymbol{Y}) = \mathrm{tr}(\boldsymbol{X}\boldsymbol{Y}$ denotes the Frobenius inner product on $\mathcal{L}((\mathbb{C}^d)^{\otimes N})$ This is a convex subset of the (real-valued) space of $d^N \times d^N$ hermitian matrices. The extreme points of this set correspond to pure states:

$$\boldsymbol{\rho} = \boldsymbol{u}\boldsymbol{u}^* \quad \text{with} \quad \boldsymbol{u} \in (\mathbb{C}^d)^{\otimes N}, \langle \boldsymbol{u}, \boldsymbol{u} \rangle = 1.$$

A measurement is a resolution of the identity:

$$\boldsymbol{H}_{\lambda_1}, \dots, \boldsymbol{H}_{\lambda_m} : \boldsymbol{H}_{\lambda_k} \succeq \mathbf{0}, \sum_{k=1}^{m} \boldsymbol{H}_{\lambda_k} = \mathbf{I}.$$

Born's rule asserts

$$\Pr[\lambda_k | \boldsymbol{\rho}] = (\boldsymbol{H}_{\lambda_k}, \boldsymbol{\rho})$$

An important sub-class of measurements are *projective measurements*: Each $\boldsymbol{H}_{\lambda_k}$ is an orthogonal projection $\boldsymbol{P}_{\lambda_k}$. This ensures that the psd constraint is met by default. Moreover, a set of orthogonal projectors forms a resolution of the identity if and only if the projectors project onto mutually orthogonal subspaces whose union spans all of $(\mathbb{C}^d)^{\otimes N}$.

2.2 Quantum mechanical observables and the ground state problem

Let $\boldsymbol{P}_{\lambda_1}, \ldots, \boldsymbol{P}_{\lambda_m}$ be a projective quantum measurement. Suppose that the measurement outcomes are real-valued numbers (e.g. energy, or spin). Then, we can combine measurement and outcomes to a single Hermitian matrix:

$$\boldsymbol{O} = \sum_{k=1}^{m} \lambda_k \boldsymbol{P}_{\lambda_k}.$$

This object is called an *observable*. The associated measurements arise from a spectral decomposition. Note that

$$\langle \boldsymbol{O} \rangle_{\boldsymbol{\rho}} = (\boldsymbol{O}, \boldsymbol{\rho}) = \sum_{k=1}^{m} \lambda_k (\boldsymbol{P}_{\lambda_k}, \boldsymbol{\rho}) = \sum_{k=1}^{m} \lambda_k \mathrm{Pr}[\lambda_k | \boldsymbol{\rho}] = \mathbb{E}_{\boldsymbol{\rho}}[\lambda].$$

This conditional expectation value measures the expected physical quantity achieved by a quantum state $\boldsymbol{\rho}$. Arguably, the most important physical quantity of any system is energy.

Definition 2.1. The observable associated with energy is caled a *Hamiltonian* and is denoted by $\boldsymbol{H} \in \mathcal{L}((\mathbb{C}^d)^{\otimes N})$. Its smallest eigenvalue $\lambda_{\min}$ is called the *ground state energy*.

2.3 The ground state problem

One of the most fundamental questions in quantum physics and chemistry is: Given a Hamiltonian $\boldsymbol{H}$, find the smallest expected energy achievable and – ideally – a quantum state that achieves this minimal value.

Definition 2.2. Let $\boldsymbol{H}$ be a Hamiltonian. A quantum state $\boldsymbol{\rho}_\sharp$ is said to be *in the ground state* if $\langle \boldsymbol{H} \rangle_{\boldsymbol{\rho}_\sharp} = \min_{\boldsymbol{\rho}} \langle \boldsymbol{H} \rangle_{\boldsymbol{\rho}}$.

The following immediate consequence of convexity allows for substantially reducing the complexity of the ground state problem.

Lemma 2.3. *For any Hamiltonian $\boldsymbol{H}$, there always exists a pure state $\boldsymbol{\rho} = \boldsymbol{u}\boldsymbol{u}^*$ that achieves the ground state energy: $\langle \boldsymbol{H} \rangle_{\boldsymbol{u}\boldsymbol{u}^*} = \min_{\boldsymbol{\rho}} \langle \boldsymbol{H} \rangle_{\boldsymbol{\rho}}$.*

We note in passing that this ground state need not be unique. There might be other pure states that achieve the same energy in expectation. Linearity then implies that any convex mixture of such pure ground states is also a ground state.

Proof. Fix $\boldsymbol{H}$ and note that the function $(\boldsymbol{H}, \boldsymbol{\rho})$ is linear in $\boldsymbol{\rho}$ and therefore also concave. We minimize this concave function over the set of all quantum states which is convex. A fundamental result from convex optimization states that a concave function achieves its minimum over a convex set at the boundary. This boundary corresponds to the set of all pure quantum states. □

Problem statement Let $\boldsymbol{H} \in \mathcal{L}((\mathbb{C}^d)^{\otimes N})$ be a Hamiltonian. The ground state problem corresponds to solving the following Rayleigh quotient:

$$\underset{\boldsymbol{u} \in (\mathbb{C}^d)^{\otimes N}}{\text{minimize}} \quad \frac{\langle \boldsymbol{u}, \boldsymbol{H}\boldsymbol{u} \rangle}{\langle \boldsymbol{u}, \boldsymbol{u} \rangle}.$$

This problem is not difficult in its own right. A "simple" eigenvalue decomposition of $\boldsymbol{H}$ would readily solve it. The challenge stems from the curse of dimensionality associated with tensor product spaces: $\boldsymbol{u}$ lives in a d^N dimensional space. Even for moderate N, this exponential growth renders a full eigenvalue decomposition of $\boldsymbol{H}$ impractical. The associated runtime would be $\mathcal{O}\Big(d^{3N}\Big)$.

3 The ground state problem for spin chains

Stated as it is, the ground state problem might seem strange at first. The glaring difficulty stems from very large dimensions. But how do such high dimensional Hamiltonians– very large-dimensional hermitian matrix – arise in the first place? This is a feature of many body physics, and – more general – the study of emergent phenomena (e.g. swarm behavior in certain animal species). Already very simple, structured interactions between N players can give rise to a very intricate global interaction patterns. This is in particular true for interactions among N simple quantum mechanical systems. Understanding such phenomena may help to explain effects that we can measure in the lab. Recall the following short-hand notation for operators on $\mathcal{L}((\mathbb{C}^d)^{\otimes N})$:

$$\hat{\boldsymbol{O}}_j = \mathbf{I}^{\otimes(j-1)} \otimes \boldsymbol{O} \otimes \mathbf{I}^{\otimes(N-j-1)} \quad \text{for} \quad \boldsymbol{O} \in \mathcal{L}(\mathbb{C}^d),\ 1 \leq j \leq N.$$

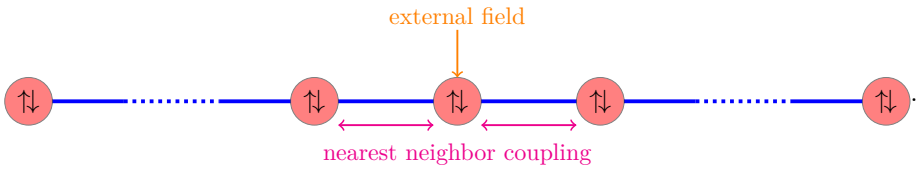
Also, recall the Pauli matrices

$$\boldsymbol{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ \boldsymbol{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},\ \boldsymbol{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

The spin is a 2-dimensional degree of freedom and loosely resembles a magnetic moment (think of an electric current that passes through a closed ring). The Pauli matrices play a crucial role in the study of spin. They correspond to observables that measure the orientation of the spin along the three different axes in space.

Now, suppose that we have prepared N quantum mechanical systems along a line. And we have isolated them sufficiently from the environment and each other such that only their spin degree of freedom matters. This allows us to accurately approximate each system with a 2-dimensional quantum state – the spin state. The joint system is described by an enormous density matrix in $\mathcal{L}((\mathbb{C}^2)^{\otimes N})$. Since spin resembles a magnetic moment, the individual systems are affected by external electric/magnetic fields and can also interact with each other. Physicists have come up with various toy models that reflect these types of interactions. The result is typically a big Hamiltonian

$\boldsymbol{H}$ that is comprised of many simple terms:



For each individual particle, the energy contributions are very simple. They feature spin interactions among nearest neighbors and a contribution from an external field. Since energy is additive, the full Hamiltonian then corresponds to a sum of N simple terms:

$$\boldsymbol{H} = -\sum_{i=1}^{N-1} \left(\hat{\boldsymbol{C}}_i \hat{\boldsymbol{C}}_{i+1} + \hat{\boldsymbol{B}}_i \right) \in \mathcal{L}\left((\mathbb{C}^2)^{\otimes N} \right).$$

Example 3.1 (Ising model)**.** The Ising model is arguably the simplest interesting spin chain model. The nearest neighbor interactions are mediated by Pauli-$\boldsymbol{X}$ matrices, while the external (magnetic) field contributes a Pauli-$\boldsymbol{Z}$ term each:

$$\boldsymbol{H}_{\text{Ising}} = -J \sum_{i=1}^{N-1} \hat{\boldsymbol{X}}_i \hat{\boldsymbol{X}}_{i+1} - h \sum_{i=1}^{N} \boldsymbol{Z}_i. \tag{1}$$

The parameters J (coupling strength) and h (external field strengths) are the parameters of the model. Ising actually could provide an analytic solution to this ground state problem. The Ising model can be readily extended to higher dimensions (e.g. lattices). The higher dimensional Ising ground state problem can be solved efficiently (i.e. in time polynomial in N) for for planar graphs (e.g. a 2D lattice), but is NP-complete for non-planar graphs (e.g. a lattice in 3D).

Example 3.2 (Heisenberg model)**.** The Heisenberg model expands on Ising by considering nearest neighbor interactions along all possible spin directions:

$$\boldsymbol{H}_{\text{Heisenberg}} = -J_X \sum_{i=1}^{N-1} \hat{\boldsymbol{X}}_i \hat{\boldsymbol{X}}_{i+1} - J_Y \sum_{i=1}^{N-1} \hat{\boldsymbol{Y}}_i \hat{\boldsymbol{Y}}_{i+1} - J_Z \sum_{i=1}^{N-1} \hat{\boldsymbol{Z}}_i \hat{\boldsymbol{Z}}_{i+1} - h \sum_{i=1}^{N} \boldsymbol{Z}_i. \tag{2}$$

This quantum mechanical model is used in the study of critical points and phase transitions of magnetic systems. Note that the Ising model is a specification of the Heisenberg model, where $J_Y = J_Z = 0$.

4 Tensor Train Ansatz for solving ground state problems

4.1 Recapitulation: tensor trains

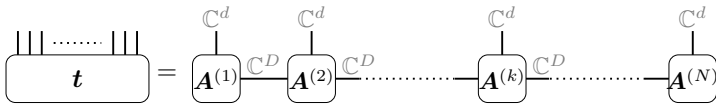
Every tensor $\boldsymbol{t} \in (\mathbb{C}^d)^{\otimes N}$ can be expanded as a tensor train. Each "wagon" corresponds to an order-three tensor $\boldsymbol{A}^{(k)} \in \mathbb{C}^{D_k} \otimes \mathbb{C}^{D_{k+1}} \otimes \mathbb{C}^d$ that is typically cut into frontal slices:

$$\boldsymbol{A}^{(k)}_i = \boldsymbol{A}^{(k)}_{::i} \in \mathbb{C}^{D_i \times D_{i+1}} \quad \text{for} \quad 1 \leq i \leq d.$$

These local tensors characterize the tensor $\boldsymbol{t}$ by means of the following expansion formula:

$$\boldsymbol{t} = \sum_{i_1=1,\dots,i_N=1}^{d} \operatorname{tr}\Big(\boldsymbol{A}^{(1)}_{i_1}\boldsymbol{A}^{(2)}_{i_2}\cdots\boldsymbol{A}^{(N)}_{i_N}\Big)\boldsymbol{e}_{i_1}\otimes\boldsymbol{e}_{i_2}\cdots\otimes\boldsymbol{e}_{i_N}.$$

The motivation for this representation (and its name) becomes exceptionally clear in wiring notation:

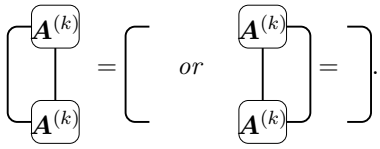


Remark 4.1. This is a slight modification of the TT framework introduced in previous lectures. The virtual index does not wrap around. In turn, the boundary tensors $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(2)}$ have a qualitatively different flavour from the other wagons: they are only order-two tensors. Although highly relevant in practice, we will ignore boundary effects/representations and entirely focus on tensors $\boldsymbol{A}^{(i)}$ in the center of the train.

Tensor trains approximate arbitrary tensors. They mediate correlations between individual factors by exploiting an additional degree of freedom that connects the wagons on a virtual level. The dimension of this auxiliar space D is called the bond dimension.

A large value of D – exponentially large in N – greatly increases the expressiveness of the TT model. Every tensor can be represented by a TT with bond dimension $D \simeq d^{N/2}$. Small values of D – polynomially large in N – facilitate actual tensor computations at the cost of expressiveness. Varying the parameter D interpolates between both regimes.

Fact 4.2 (Gauge transformations)**.** *Tensor train representations are never unique. We can apply arbitrary invertible linear transformations along the virtual degrees of freedom. This allows us to convert a given TT into either a* left- *or a* right- normal form*:*



4.2 Tensor train ansatz for the ground state problem

Recall the ground state problem for a given Hamiltonian $\boldsymbol{H} \in \mathcal{L}((\mathbb{C}^d)^{\otimes N})$:

$$\lambda_{\min} = \underset{\boldsymbol{u}\in(\mathbb{C}^d)^{\otimes N}}{\text{minimize}} \quad \frac{\langle \boldsymbol{u}, \boldsymbol{H}\boldsymbol{u}\rangle}{\langle \boldsymbol{u}, \boldsymbol{u}\rangle}.$$

The challenge in solving this problem does stem from the fact that $(\mathbb{C}^d)^{\otimes N}$ is a huge space with dimension d^N. A natural ansatz to approximate this problem is to restrict

the regime over which we optimize. Tensor trains with fixed bond dimension (b.d.) D are a natural candidate for such a restricted optimization:

$$\tilde{\lambda}_{\min}(D) = \underset{\boldsymbol{t} \text{ is TT with b.d. } D}{\text{minimize}} \frac{\langle \boldsymbol{t}, \boldsymbol{H}\boldsymbol{t} \rangle}{\langle \boldsymbol{t}, \boldsymbol{t} \rangle}. \tag{3}$$

This is an optimization over a strict subset of all tensors. What is more, a moment of thought reveals that TT with bond dimension D are included in the set of TT with bond dimension $D+1$:

$$\mathrm{TT}(1) \subseteq \mathrm{TT}(2) \subseteq \cdots \subseteq \mathrm{TT}(D) \subseteq \cdots \subseteq \mathrm{TT}(d^{N/2}) = (\mathbb{C}^d)^{\otimes N}.$$

In turn

$$\tilde{\lambda}_{\min}(1) \geq \tilde{\lambda}_{\min}(2) \geq \cdots \geq \tilde{\lambda}_{\max}(D) \geq \cdots \geq \tilde{\lambda}_{\min}(d^{N/2}) = \lambda_{\min},$$

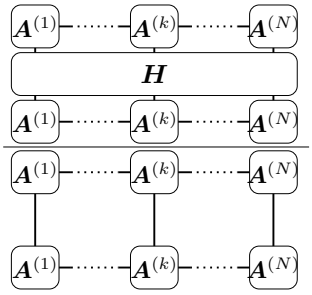
where the last equality follows from complete expressiveness of TT for sufficiently high bond dimension.

Although simple, this is a profound insight. While varying the bond dimension, we obtain ever more accurate approximations of the true ground state. However, there is a trade-off. The runtime of the underlying algorithm will scale polynomially in D.

4.3 Problem reformulation: Matrix product operators

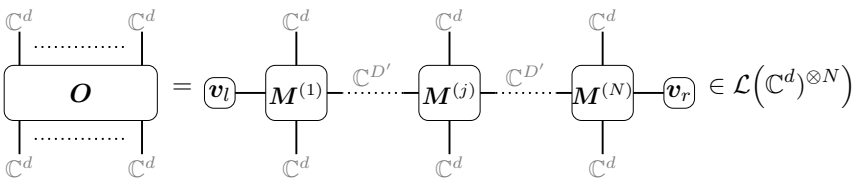
It is instructive to rewrite the objective function in (3) in wiring formalism:

$$\frac{\langle \boldsymbol{t}, \boldsymbol{H}\boldsymbol{t} \rangle}{\langle \boldsymbol{t}, \boldsymbol{t} \rangle} =$$



The denominator reveals a lot of structure. In particular, its form already suggests the potential benefits of transforming the individual $\boldsymbol{A}^{(k)}$'s into a suitable normal form. The big Hamiltonian in the enumerator, however, breaks this nice sequential structure. It seems highly advisable to decompose it further into a tensor expression that mimics the structure of tensor trains.

Definition 4.3 (Matrix product operator)**.** A *matrix product operator* (MPO) is fully characterized by collection of N order four tensors $\boldsymbol{M}^{(k)} \in \mathbb{C}^d \otimes \mathbb{C}^d \otimes \mathbb{C}^{D'} \otimes \mathbb{C}^{D'}$ (think operator-valued matrices of size $D' \times D'$) and two vectors $\boldsymbol{v}_l, \boldsymbol{v}_r \in \mathbb{C}^{D'}$:



Every operator acting on a tensor space may be represented as a MPO. The decomposition may be achieved in a fashion similar to the derivation of tensor trains. However, in general, the bond dimension must scale exponentially with the number of tensor factors: $D' \simeq d^N$ is necessary to accurately represent a generic operator.

However, the Hamiltonians we consider for the ground state problem are typically very far from being generic. Their simple structure manifests itself in a tiny bond dimension.

Example 4.4 (MPO for the Ising Hamiltonian)**.** The Ising Hamiltonian $\boldsymbol{H}_{\text{Ising}} \in \mathcal{L}((\mathbb{C}^2)^{\otimes N})$ (1) is fully characterized by a single MPO with bond dimension $D' = 3$:

$$\boldsymbol{M}^{(k)} = \begin{pmatrix} \mathbf{I} & 0 & 0 \\ \boldsymbol{X} & 0 & 0 \\ -h\boldsymbol{X} & -J\boldsymbol{X} & \mathbf{I} \end{pmatrix} \quad \text{and} \quad \boldsymbol{v}_l = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \ \boldsymbol{v}_r = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Example 4.5 (MPO for the Heisenberg Hamiltonian)**.** The Heisenberg Hamiltonian $\boldsymbol{H}_{\text{Heisenberg}} \in \mathcal{L}((\mathbb{C}^2)^{\otimes N})$ (2) is fully characterized by a single MPO with bond dimension $D' = 5$:

$$\begin{pmatrix} \mathbf{I} & 0 & 0 & 0 & 0 \\ \boldsymbol{X} & 0 & 0 & 0 & 0 \\ \boldsymbol{Y} & 0 & 0 & 0 & 0 \\ \boldsymbol{Z} & 0 & 0 & 0 & 0 \\ -h\boldsymbol{Z} & -J_X\boldsymbol{X} & -J_Y\boldsymbol{Y} & -J_Z\boldsymbol{Z} & \mathbf{I} \end{pmatrix} \quad \text{and} \quad \boldsymbol{v}_l = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \ \boldsymbol{v}_r = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

5 DMRG lite

5.1 Overview

We are now ready to discuss a simplified version of DMRG. The task is to approximately solve the ground state problem for a big Hamiltonian$\boldsymbol{H} \in \mathcal{L}\Big((\mathbb{C}^d)^{\otimes N}\Big)$. We will do a TT ansatz and use an alternating least squares heuristics to individually optimize $\boldsymbol{A}^{(k)}$ by keeping all other tensors fixed. One iteration consists of N independent optimizations – one for each tensor train wagon – and results in a global update of the TT approximation to the ground state. Repeating this "sweep" many times heuristically boosts convergence. The problem parameters are:

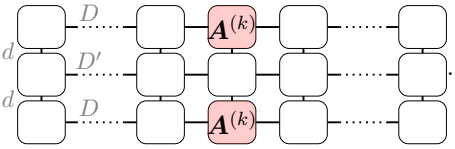
- local physical dimension d: typically this is small, e.g. $d = 2$ for spins.
- number of quantum systems N: we suppose that this number is very large.
- bond dimension D' of the expansion of $\boldsymbol{H}$ as a matrix product operator: this is fixed and typically small, e.g. $D = 3$ for the Ising model, or $D' = 5$ for the Heisenberg model.
- bond dimension D: this is a free parameter that we get to choose. We suppose that it scales moderately in the problem size: $D = \text{poly}(N)$.

Our ALS-type algorithm is based on a nifty sub-routine to individually optimize tensor $\boldsymbol{A}^{(k)}$ individually. Importantly, each optimization can be achieved in runtime

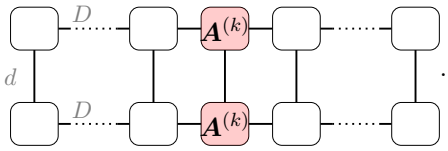
polynomial in d, D', N and the model parameter D. This is efficient, as long as we do not choose D to be too large. This results in a total runtime of $N = \text{poly}(d, D', N, D) = \text{poly}(d, D', N, D)$ for each iteration. This polynomial cost is cheap when compared to the exponentially large problem dimension d^N. This gain allows us to repeat these sequential updates many times to – hopefully – boost convergence to the the ground state energy.

5.2 ALS subroutine

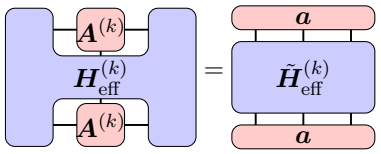
We focus on optimizing one tensor $\boldsymbol{A}^{(k)}$ while keeping all other elements in the TT fixed. Next, we insert an MPO representation of the Hamiltonian $\boldsymbol{H}$ in the enumerator of the objective energy function:

$\langle \boldsymbol{t}, \boldsymbol{H}\boldsymbol{t} \rangle =$ 

The MPO formalism ensures that this wiring diagram now looks very similar to the denominator:

$\langle \boldsymbol{t}, \boldsymbol{t} \rangle =$ 

We can now contract everything to the left of the $\boldsymbol{A}^{(k)}$ and everything to the right to get an effective environment tensor:

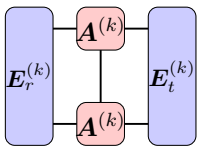
$\langle \boldsymbol{t}, \boldsymbol{H}\boldsymbol{t} \rangle =$ 

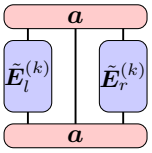
$=$

$= \langle \boldsymbol{a}, \tilde{\boldsymbol{H}}^{(k)}_{\text{eff}} \boldsymbol{a} \rangle, \quad \boldsymbol{a} = \text{vec}\Big(\boldsymbol{A}^{(k)}\Big) \in \mathbb{C}^{dD^2}.$

The last equation follows from vectorization and a re-arrangement of the tensor factors in $\boldsymbol{H}^{(k)}_{\text{eff}}$. It is easy to check that the runtime for constracting $\boldsymbol{H}^{(k)}_{\text{eff}}$ scales polynomially in D, D', d and linearly in N.

At a comparable cost, we can construct a similar environment tensor for the denominator:

$\langle \boldsymbol{t}, \boldsymbol{t} \rangle =$ 

$=$ 

$= \langle \boldsymbol{a}, \tilde{\boldsymbol{E}}^{(k)}_l \otimes \mathbf{I} \tilde{\boldsymbol{E}}^{(k)}_r \boldsymbol{a} \rangle.$

A smart normal form convention in the TT can further simplify this expression. If all tensors $\boldsymbol{A}^{(j)}$ left from $\boldsymbol{A}^{(k)}$ $(1 \leq j < k)$ are in left-normal form and all tensors right

from $\boldsymbol{A}^{(k)}$ ($k < j \leq N$) are in right-normal form, the effective tensors become trivial[1]: $\tilde{\boldsymbol{E}}_l^{(k)} = \mathbf{I}$ and $\tilde{\boldsymbol{E}}_r^{(k)} = \mathbf{I}$. Under these assumptions and reformulations, the optimization problem exactly resembles a Raiyleigh quotient:

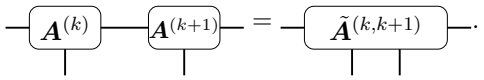
$$\underset{\boldsymbol{A}^{(k)}}{\text{minimize}} \quad \frac{\langle \boldsymbol{t}, \boldsymbol{H}\boldsymbol{t}\rangle}{\langle \boldsymbol{t}, \boldsymbol{t}\rangle} = \underset{\boldsymbol{a}\in\mathbb{C}^{D^2 d}}{\text{minimize}} \frac{\langle \boldsymbol{a}, \tilde{\boldsymbol{H}}_{\text{eff}}^{(k)}\boldsymbol{a}\rangle}{\langle \boldsymbol{a}, \boldsymbol{a}\rangle}.$$

It can be solved by computing the eigenvalue decomposition of $\tilde{\boldsymbol{H}}_{\text{eff}}^{(k)} \in \mathcal{L}\Big(\mathbb{C}^{D^2 d}\Big)$, extracting the smallest eigenvector $\boldsymbol{a}_\sharp$ and re-shaping it into a tensor. The runtime of a dense eigenvalue decomposition is of order $\mathcal{O}(dD^2)$. A subsequent conversion of the updated $\boldsymbol{A}^{(k)}$ into left- or right normal form comes at a similar cost.

5.3 Extensions and rigorous results

The above sweeping procedure is often called DMRG1. It is an iterative procedure, where the individual tensors in a TT representation are updated sequentially. The bond dimension D is a proper input to the heuristic – there is no easy way to change it within the algorithm.

This can be a severe drawback in practice. Once we start with a certain bond dimension value D, we must stick to it. DMRG2 is a conceptually simple refinement of DMRG1 that allows to adjust the bond dimension dynamically while the algorithm is running. The main idea is to group two tensors together and treat them as a single tensor:



Subsequently, apply DMRG1 to this coarse-grained tensor network. Once an update for $\tilde{\boldsymbol{A}}^{(k,k+1)}$ is obtained, we can subsequently apply a singular value decomposition to pull the two original tensors apart. The decay of the spectrum associated with this SVD provides us with valuable guidance on how to adjust bond dimensions dynamically.

Last but not least, we want to emphasize that a comparatively recent result provides a rigorous underpinning for tensor train approaches to solve the ground state problem. It applies to an iterative algorithm, that is similar in spirit to DMRG, but the details are somewhat different. The associated rigorous convergence guarantee applies to local Hamiltonians $\boldsymbol{H}$ of 1D-chains that have a *spectral gap*:

$$\lambda_{\min} < \lambda_k - \varepsilon \quad \text{for all} \quad \lambda_k \neq \lambda_{\min} \quad \text{and } \varepsilon > 0 \text{ is constant.}$$

Theorem 5.1 (Landau, Vazirani, Vidick; 2013)**.** *Let $\boldsymbol{H}$ be a local Hamiltonian of a 1D quantum system with a constant spectral gap. Then, there is a tensor train algorithm that accurately[2] approximates both the ground state (as a tensor train) and the ground state energy and runs in polynomial time.*

[1] Such a smart reformulation is achievable in practice: Start the first iteration by moving from left to right and convert all updated tensors into left normal form. Start the second iteration from right to left and convert all updated tensors into right normal form. Continue this sweeping procedure for subsequent iterations

[2] The accuracy is inverse polynomial in the problem parameters.