CS 224 Fall 2023
11 September 2023

Scribes: Oam Patel, Ben Schiffer, Kevin Luo

Lecture 2: Tensor Decomposition, Jennrich's Algorithm, and Applications

1 Historical Motivation

Charles Spearman (1863 - 1945) posited there are two types of intelligence: mathematical and verbal. Any assessment then tested these two intelligences in different quantities. Hence, if one constructed a matrix $\mathbf{M} \in \mathbb{R}^{n \times m}$, where $\mathbf{M}_{ij}$ is the test score of the ith student on the jth test, he believed that $\mathbf{M}$ would admit a low rank decomposition as $\mathbf{M} \approx \mathbf{U}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{n \times k}$ and $\mathbf{V} \in \mathbb{R}^{m \times k}$, with $k = 2$. Specifically, i-th row of $\mathbf{U}$, $[U_{i,1}, U_{i,2}]^\top$, would consist of the math and verbal intelligence of the i-th student, and the j-th row of $\mathbf{V}$, $[\mathbf{V}_{j,1}, \mathbf{V}_{j,2}]$, would contain the amount of math and verbal testing of the jth test, such that $M_{ij} = U_{i,1}V_{j,1} + U_{i,2}V_{j,2}$.

Unfortunately for Spearman, $\mathbf{U}$ and $\mathbf{V}$ cannot be uniquely determined – we can rotate both by an orthogonal matrix. However, under very mild assumptions, we can solve this identifiability issue using tensors.

1.1 Solution to the rotation problem

In the Spearman's problem, he essentially wished to decompose M as $M = \sum_{i=1}^{k} u_i v_i^\top$, where $\{u_i\}$ and $\{v_i\}$ can be just viewed as the columns of $\mathbf{U}$ and $\mathbf{V}$. It turns out that, under some mild conditions, if we instead try to add auxiliary information to our problem, and decompose an order 3 tensor M as $M = \sum_{i=1}^{k} u_i \otimes v_i \otimes w_i$, where these $\{w_i\}$ can be thought of as experimental conditions, the rotation problem disappears under mild conditions on $\{u_i\}, \{v_i\}, \{w_i\}$.

2 Tensor basics

An order 3 tensor $T \in \mathbb{R}^{r \times s \times t}$ is an array of numbers indexed as T_{ijk}, with $i, j, k \in [r] \times [s] \times [t]$.

Definition 1 (Tensor rank)**.** *The rank of an order 3 tensor T is the smallest k for which T*

admits a decomposition of the form

$$T = \sum_{i=1}^{k} u_i \otimes v_i \otimes w_i.$$

Writing T in this way is known as CP decomposition.

Order 2 tensors are just matrices, in which case the idea of rank corresponds to our understanding of rank for matrices.

Like matrices, one can define notions of an operator norm, eigenvectors, and eigenvalues, but it turns out that, while for matrices we can compute these quantities through efficient algorithms like SVD, for tensors of order 3 or higher, they become NP-Hard [HL09].

3 Tensor Decomposition

The following algorithm is attributed to Jennrich, but in fact the history behind the name is murky, [Mat].

Theorem 1 (Jennrich)**.** *Given $T \in \mathbb{R}^{d\times d\times d}$ of the form*

$$T = \sum_{i=1}^{k} u_i^{\otimes 3}$$

for $u_1, \ldots, u_k$ linearly independent, there is a poly(d) -time algorithm for recovering $u_1, \ldots, u_k$ exactly.

Note that $k \leq d$ due to the linear independence condition, while the maximum rank of a tensor is d^2. In fact, this applies to a more general form of tensor, with looser conditions:

Theorem 2. *Given $T \in \mathbb{R}^{d_1\times d_2\times d_3}$ of the form*

$$T = \sum_{i=1}^{k} u_i \otimes v_i \otimes w_i$$

where $\{u_i\}$, $\{v_i\}$, $\{w_i\}$ satisfy

1. *$u_1, \ldots, u_k$ are linearly independent*
2. *$v_1, \ldots, v_k$ are linearly independent*

3. *$d_3 \geq 2$ and no two w_i, w_j are collinear (note that $d_3 = 1$ is the matrix case and hence this restriction makes sense, also note that the $\{w_i\}$ need not be linearly independent, but just none can be multiples of another)*

then there exists a poly(d) time algorithm to recover $\{u_i\}, \{v_i\}, \{w_i\}$.

4 Jennrich's Algorithm

4.1 Tensor basics

Let $T \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ be defined as

$$T = \sum_{i=1}^{k} u_i \otimes v_i \otimes w_i.$$

Definition 2 (Tensor Contraction)**.** *Let $(z_1, z_2, z_3) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \times \mathbb{R}^{d_3}$. Then the* contraction *of T along z_1, z_2, z_3 is defined as*

$$T(z_1, z_2, z_3) = \sum_{a \in [d_1], b \in [d_2], c \in [d_3]}^{d} T_{a,b,c}(z_1)_a (z_2)_b (z_3)_c.$$

Every tensor T has an associated polynomial $p : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \times \mathbb{R}^{d_3} \to \mathbb{R}$ defined as

$$p(z_1, z_2, z_3) = \sum_{a,b,c} T_{abc}(z_1)_a (z_2)_b (z_3)_c$$

Definition 3 (Partial Contraction)**.** *The partial contraction, denoted $T(:, :, z) : \mathbb{R}^{d_3} \to \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$, is a matrix-valued function, defined as follows on rank-1 tensors, and extended in the natural way to higher ranks. For $\delta = u \otimes v \otimes w$,*

$$\delta(:, :, z)_{ab} = u_a \cdot v_b \cdot \langle z, w \rangle$$

In general, one has that

$$T(:, :, z)_{ab} = T(e_a, e_b, z)$$

where e_a is the standard basis vector with 1 in the a-th coordinate, and likewise for e_b.

In particular, if T were order-2 (so a matrix), one has that

$$T(:, z) = Tz.$$

5 The Algorithm

Algorithm 1: JENNRICH(T)

Input: $T \in \mathbb{R}^{d_1 \times d_2 \times d_3}$

Output: Determines $\{u_i\}$, $\{v_i\}$, $\{w_i\}$ such that $T = \sum_i u_i \otimes v_i \otimes w_i$

$z, z' \leftarrow_{i.i.d.} \mathrm{Unif}(S^{d-1})$

$M_z \leftarrow T(:,:,z)$

$M_{z'} \leftarrow T(:,:,z')$

$(\lambda_i, u_i)_{i=1}^d \leftarrow \textsc{EigenDecompose}(M_z M_{z'}^+)$

// A^+ denotes pseudo-inverse; λ_i eigenvalues; u_i corresponding eigenvectors

$(\mu_j, v_j)_{j=1}^d \leftarrow \textsc{EigenDecompose}((M_z^+ M_{z'})^\top)$

// Match u_i and v_i by fact eigenvalues should be reciprocal

$\{(u_i, v_i)\}_{i=1}^k \leftarrow \{(u_i, v_j) \mid \lambda_i \mu_j = 1\}$ // exactly k such pairs

// now we solve for the w's with a linear system

$\boldsymbol{\lambda}_{(a,b),c} = (u_c)_a (v_c)_b$ // $\boldsymbol{\lambda} \leftarrow \mathbb{R}^{d_1 d_2 \times k}$

$\mathbf{T}_{\text{matrix}} = \text{reshape}(T, (d_1 d_2, d_3))$ // $\mathbf{T}_{\text{matrix}} \in \mathbb{R}^{d_1 d_2 \times d_3}$

// Let $\mathbf{W}$ be the matrix with w_i as its i-row, meaning $\mathbf{W} \in \mathbb{R}^{k \times d_3}$ and $\mathbf{T}_{\text{matrix}} = \boldsymbol{\lambda}\mathbf{W}$

$\mathbf{W} = \boldsymbol{\lambda}^+ \mathbf{T}_{\text{matrix}}$

return $\{(u_i, v_i, w_i)\}_{i=1}^d$

The algorithm above, also known as simultaneous diagonalization, was not actually the algorithm Jennrich proposed (alternating least squares), as discussed in the link above. Instead, this algorithm seems drawn from [LRA93].

5.1 Proof of Correctness

As an aside, first note that if a matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ has linearly independent columns, then $\mathbf{A}^+ \mathbf{A} = \mathbf{I}_{m \times m}$.

Lemma 1. *In the notation above,*

$$M_z \cdot M_{z'}^+ = U D_z D_{z'}^{-1} U^+$$

and

$$M_z^+ \cdot M_{z'} = (V^\top)^+ D_z D_{z'}^{-1} V^\top$$

Proof. First note that

$$\begin{aligned} M_z &= \sum_{i=1}^{k} (u_i \otimes v_i \otimes w_i)(:,:,z) \\ &= \sum_{i=1}^{k} u_i \otimes v_i \cdot \langle w_i, z \rangle \\ &= U D_z V^\top \end{aligned}$$

Likewise, $M_{z'} = U D_{z'} V^\top$, where

$$\begin{array}{rr} U \in \mathbb{R}^{d \times k} & U = [u_1, u_2, \ldots, u_k] \\ V \in \mathbb{R}^{d \times k} & V = [v_1, v_2, \ldots, v_k] \\ D_z \in \mathbb{R}^{k \times k} & D_z = \mathrm{diag}(\langle w_1, z \rangle, \ldots, \langle w_k, z \rangle) \\ D_{z'} \in \mathbb{R}^{k \times k} & D_{z'} = \mathrm{diag}(\langle w_1, z' \rangle, \ldots, \langle w_k, z' \rangle) \end{array}.$$

Using this representation, we get that

$$\begin{aligned} M_z \cdot M_{z'}^+ &= U D_z V^\top (U D_{z'} V^\top)^+ \\ &= U D_z V^\top (V^\top)^+ D_{z'}^{-1} U^+ \\ &= U D_z ((V)^+ V)^\top D_{z'}^{-1} U^+ \\ &= U D_z D_{z'}^{-1} U^+ \end{aligned}$$

where the final equality comes from the fact that V has linearly independent columns. The same holds for $M_z^+ \cdot M_{z'}$ by a symmetric argument.

$\square$

The lemma above show that $M_z M_{z'}^+$ admits a diagonalization, and in particular, its eigenvectors with nonzero eigenvalue are exactly the columns of U, which are just $\{u_i\}_{i=1}^k$, where u_i has eigenvalue $\frac{\langle w_i, z \rangle}{\langle w_i, z' \rangle}$. Likewise, this shows that the eigenvectors with nonzero eigenvalue of $(M_z^+ M_{z'})^\top$ are the columns of V, which are the $\{v_i\}_{i=1}^k$, now with eigenvalue $\frac{\langle w_i, z' \rangle}{\langle w_i, z \rangle}$.

Thus, calculating the eigendecomposition of those two matrices, we obtain the $\{u_i\}_{i=1}^k$ and $\{v_i\}_{j=1}^k$ up to permutation. We can then pair up them up appropriately by using the fact that the corresponding eigenvalues of u_i and v_i are reciprocals of one another. Note that the non collinearity condition of the w_i necessitates that there will be no duplicate nonzero eigenvalues.

Now it remains to compute the w_i. This is done by setting up a linear system. Define vectors $\lambda^{ab} \in \mathbb{R}^k$ componentwise as $\lambda_i^{ab} = (u_i)_a (v_i)_b$. Define the matrix $\mathbf{W} = [w_1^\top, w_2^\top, \ldots, w_k^\top]^\top$. Now observe that $T_{abc} = \langle \lambda^{ab}, W^{(c)} \rangle$, where $W^{(c)} =$

$((w_1)_c, (w_2)_c, \ldots, (w_k)_c)$ denotes the c-th column of W. This is now just some linear system, where the unknowns are the w_i. note that we can write this linear system as $\boldsymbol{\lambda}$

To see that the solution to this is unique, we will summarize these constraints as a matrix equation. Wrap the λ^{ab} into a matrix, letting $\boldsymbol{\lambda} \in \mathbb{R}^{d_1 d_2 \times k}$ have rows which are just the λ^{ab}. Likewise, reshape T into $\mathbf{T}_{\text{matrix}} \in \mathbb{R}^{d_1 d_2 \times d_3}$; done consistently, this yields $\mathbf{T}_{\text{matrix}} = \boldsymbol{\lambda}\mathbf{W}$.

Note now that left multiplication by $\boldsymbol{\lambda}^+$ now yields $\mathbf{W}$, provided $\boldsymbol{\lambda}$ has linearly independent columns, meaning it has column rank k. Therefore, if $\boldsymbol{\lambda}$ has full column rank, then $\mathbf{W} = \boldsymbol{\lambda}^+\mathbf{T}_{\text{matrix}}$. We conclude the proof with a lemma giving that desired result.

Lemma 2. $\boldsymbol{\lambda}$ *has full column rank.*

Proof. Assume otherwise. Note that the i-th column of $\boldsymbol{\lambda}$, denoted $\boldsymbol{\lambda}^{(i)}$, is

$$(\lambda_i^{ab})_{(a,b)\in[d_1]\times[d_2]} = ((u_i)_a(v_i)_b)_{(a,b)\in[d_1]\times[d_2]},$$

so $\boldsymbol{\lambda}^{(i)} = \operatorname{vec}(u_i \otimes v_i) = \operatorname{vec}(u_i v_i^\top)$. Then if there exists some linear dependence among the rows, we have that there exist some constants c_i, not all zero, such that

$$0 = \sum_{i=1}^{k} c_i \boldsymbol{\lambda}^{(i)}$$
$$0 = \sum_{i=1}^{k} c_i u_i v_i^\top$$

Since the u_i are linearly independent, we can find some vector x which is orthogonal to $u_2, \ldots, u_k$, but not u_1. Then

$$\sum_{i=1}^{k} c_i x^\top u_i v_i^\top = 0$$
$$c_1 \langle u_1, x\rangle v_1 = 0$$

which implies $c_1 = 0$. We can repeat this for any index, yielding that $c_i = 0$ for every i, and hence no such dependence exists.

□

6 Applications

6.1 Mixture of Gaussians

6.1.1 A Historical Aside

Study of Gaussian mixtures began when (in)famous statistician Karl Pearson wanted to study crabs on an island. He believed that there were some number of species of crabs, existing in different relative proportions, each of which possessed some mean characteristics, and his observations of the crabs on the island were draws from this mixture distribution. He modeled this as a classic mixture of Gaussians, and wanted to estimate the distribution over the classes, as well as the mean characteristic of each class. More is detailed here, [Moo].

6.1.2 Method of Moments and Tensor Decomposition

The approach described below is attributed to [HK13].

Consider the classic mixture of Gaussians setting. Let $\lambda_1, ..., \lambda_k \in [0, 1]$ such that $\sum_{i=1}^{k} \lambda_i = 1$ and $\mu_1, ..., \mu_k \in \mathbb{R}^d$. Note that there are a total of $2k$ unknowns. Now suppose we obtain distributions from a Gaussian mixture distribution of k normal random variables represented as

$$q = \sum_{i=1}^{k} \lambda_i \mathsf{N}(\mu_i, \mathrm{Id})$$

Formally, the samples from q are drawn with the following steps:

- Draw $i \in [1 : k]$ with probability λ_i
- Sample $g \sim \mathsf{N}(0, \mathrm{Id})$
- Output $\mu_i + g$

The goal is, given many samples from q, to estimate $\{\mu_i\}, \{\lambda_i\}$ up to small errors. We will use the **Method of Moments** combined with tensors to achieve this goal. First, note that the first moment (expectation) of a sample x drawn from q can be written as

$$\mathbb{E}[x] = \sum_{i=1}^{k} \lambda_i \mathbb{E}[\mu_i + g] = \sum_{i=1}^{k} \lambda_i \mu_i$$

Now (unsurprisingly) we will find the expectations of a tensor.

$$
\begin{aligned}
\mathbb{E}[x^{\otimes 3}] &= \sum_{i=1}^{k} \lambda_i \mathbb{E}\left[(\mu_i + g)^{\otimes 3}\right] \\
&= \sum_{i=1}^{k} \lambda_i \mathbb{E}\Big[\mu_i^{\otimes 3} + g^{\otimes 3} + \mu_i \otimes \mu_i \otimes g + \mu_i \otimes g \otimes \mu_i + \\
&\qquad\qquad g \otimes \mu_i \otimes \mu_i + \mu_i \otimes g \otimes g + g \otimes g \otimes \mu + g \otimes \mu_i \otimes g\Big]
\end{aligned}
$$

We can use the following lemma to simplify the above equation

Lemma 3 (Moments)**.** *With the variables defined as above, we have that*

$$
\mathbb{E}[g^{\otimes 3}] = \mathbb{E}[\mu_i \otimes \mu_i \otimes g] = \mathbb{E}[\mu_i \otimes g \otimes \mu_i] = \mathbb{E}[g \otimes \mu_i \otimes \mu_i] = 0
$$

and

$$
\sum_{i=1}^{k} \lambda_i \mathbb{E}[\mu_i \otimes g \otimes g + g \otimes g \otimes \mu_i + g \otimes \mu_i \otimes g] = \left(\sum_{i=1}^{k} \lambda_u \mu_i\right) \otimes_3 \mathit{Id}
$$

where we define

$$
z \otimes_3 \mathit{Id} := \sum_{a=1}^{d} z \otimes e_a \otimes e_a + e_a \otimes z \otimes e_a + e_a \otimes e_a \otimes z
$$

Proof. To note the first result, we observe that the odd moments of a Gaussian distribution have expectation 0 by symmetry. For the second result, we can see that $\mathbb{E}[g \otimes g]$ is the identity matrix by construction, as it is the variance of a standard multivariate normal g. This observation exactly simplifies the LHS to be equal to the RHS. □

Returning to the proof of the algorithm correctness, applying the above lemma to simplify the expression $\mathbb{E}[x^{\otimes 3}]$ gives

$$
\mathbb{E}[x^{\otimes 3}] = \sum_{i=1}^{k} \lambda_i \mu_i^{\otimes 3} + \left(\sum_{i=1}^{k} \lambda_i \mu_i\right) \otimes_3 \mathrm{Id}
$$

Now, recalling that $\mathbb{E}[x] = \sum_{i=1}^{k} \lambda_i \mu_i$, we can rearrange terms and get that

$$
\mathbb{E}[x^{\otimes 3}] - \mathbb{E}[x] \otimes_3 \mathrm{Id} = \sum_{i=1}^{k} \lambda_i \mu_i^{\otimes 3}
$$

Applying Jennrich's algorithm to our empirical estimation of the LHS then allows us to recover $v_i = \lambda_i^{1/3}\mu_i^{\otimes 3}$. It remains to compute the λ_i. We set up a linear system to do this.

$$\begin{aligned}\mathbb{E}[x] &= \sum_{i=1}^{k} \lambda_i \mu_i \\ &= \sum_{i=1}^{k} \lambda_i^{2/3} v_i\end{aligned}$$

If the μ_i are all linearly independent, since $k \leq d$ (to even run Jennrich's), this system has a unique solution. The reference for this algorithm assumes that the means are in general linear position, meaning there are no linear dependences, but it would seem that even if there were multiple solutions, there would probably be only one satisfying $\sum \lambda_i = 1$.

6.2 Mixture of Exponentials

Reference for this section: [HK15].

Recall the setup from lecture 1, where we are given access to a function that, for any ω where $||\omega|| \leq 1$,

$$G : \omega \to \frac{1}{k}\sum_{j=1}^{k} e^{2\pi i \langle \omega, \mu_j \rangle}$$

The goal in this problem is to (as in the Gaussian mixture case) recover the $\mu_1, ..., \mu_k$. The algorithm is as follows. Note the choice of constants is to ensure that we are applying G to ω satisfying $||\omega|| \leq 1$.

Algorithm 2: AIRYDISCJENNRICH(G)

Input: $G : \omega \to \frac{1}{k}\sum_{j=1}^{k} e^{2\pi i \langle \omega, \mu_j \rangle}$

Output: Distribution Parameters

1 $\omega_1, ..., \omega_m \in \mathbb{R}^2$ random from $B(0.49)$

    // Note that $B(0.49)$ is defined as the ball of radius 0.49 around the origin in $\mathbb{R}^2$

2 $x_1 = (0.02, 0)$ and $x_2 = (0, 0.02)$

3 $T_{abc} = G(\omega_a + \omega_b + x_c)$

4 Apply Jennrich's algorithm to this low rank tensor and decompose into $T = \frac{1}{k}\sum_{j=1}^{k} u_j \otimes u_j \otimes w_j$ (Lemma 4)

5 Use the w_j to reconstruct the desired parameters.

The key steps in the above algorithm are the last two, which rely on the specific tensor decomposition of T that can be used to reconstruct our system parameters. This is formalized in the following lemma.

Lemma 4. *Define T as in the above algorithm, where*

$$T_{abc} = G(\omega_a + \omega_b + x_c)$$

Then the tensor decomposition of T can be rewritten as

$$T = \frac{1}{k}\sum_{j=1}^{k} u_j \otimes u_j \otimes w_j$$

where $(u_j)_a = e^{2\pi i\langle \mu_j, \omega_a\rangle}$ and $(w_j)_c = e^{2\pi i\langle \mu_j, x_c\rangle}$.

Proof. Examining a single element of T, we have that by construction,

$$T_{abc} = G(\omega_a + \omega_b + x_c) = \frac{1}{k}\sum_{j=1}^{k} e^{2\pi i\langle \mu_j, \omega_a\rangle} e^{2\pi i\langle \mu_j, \omega_b\rangle} e^{2\pi i\langle \mu_j, x_c\rangle} = \frac{1}{k}\sum_{j=1}^{k} (u_j)_a (u_j)_b (w_j)_c$$

where u_j, w_j are defined as in the lemma statement.

Therefore, we have exactly rewritten the T_{abc} as a sum of products of elements of vectors, and therefore the T_{abc} can be rewritten as

$$T_{abc} = \frac{1}{k}\sum_{j=1}^{k} u_j \otimes u_j \otimes w_j$$

□

With this lemma, there is only one additional step to showing the correctness of the above algorithm. Namely, once Jennrich's algorithm gives us u_j, w_j, we need to reconstruct the desired parameters. This follows from the fact that u_j, w_j give us a system of equations containing the μ_j that can be directly solved for the desired parameters.

Note that the application of Jennrich's algorithm relies on $\{u_j\}$ to be linearly independent and w_i to be non-collinear, which is true for non-degenerate choices of $\{\mu_j\}$. What are we really doing compared to last time? Last time, we formed matrices UU^T and UDU^T, cleverly chosen Hankel matrices by applying G to a grid. Now, we are finding matrices UD_ZU^T and $UD_{Z'}U^T$. So all we changed was we used two different matrices to diagonalize and find the columns of U. So Jennrich's algorithm is in some sense a generalization of Matrix Pencil Method. Furthermore, this tensor approach works when the μ_j are arbitrary dimensions, unlike the matrix pencil method which applies to matrices.

References

[HK13] Daniel Hsu and Sham M. Kakade. Learning mixtures of spherical gaussians: Moment methods and spectral decompositions. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, page 11–20, New York, NY, USA, 2013. Association for Computing Machinery.

[HK15] Qingqing Huang and Sham M. Kakade. Super-resolution off the grid. *CoRR*, abs/1509.07943, 2015.

[HL09] Christopher J. Hillar and Lek-Heng Lim. Most tensor problems are NP hard. *CoRR*, abs/0911.1393, 2009.

[LRA93] S. E. Leurgans, R. T. Ross, and R. B. Abel. A decomposition for three-way arrays. *SIAM Journal on Matrix Analysis and Applications*, 14(4):1064–1083, 1993.

[Mat] Will the real jennrich's algorithm please stand up? `https://www.mathsci.ai/post/jennrich/`. Accessed: 2023-09-14.

[Moo] Pearson's polynomial. `http://blog.mrtz.org/2014/04/22/pearsons-polynomial.html`. Accessed: 2023-09-14.