Lecture 20: Tensor Decomposition Techniques, Method of Moments, Latent Variable Models

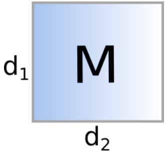
- Introduction to tensors
- Method of moments
- LVM: single topic model, gaussian mixture model, multiview model
- MoM for LVMs using tensor decomposition techniques

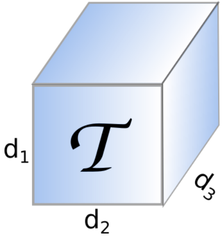
$\Rightarrow$ Consistent learning algorithms!

Spectral Methods (high-level recap)

- Spectral methods are an alternative to EM to learn latent variable models (e.g. HMMs in the previous lecture, single-topic/Gaussian mixtures models in this one).
- Spectral methods usually achieve learning by extracting structure from observable quantities through eigen-decompositions/tensor decompositions.
- Advantages of spectral methods:
  - computationally efficient,
  - consistent,
  - no local optima.

Tensors



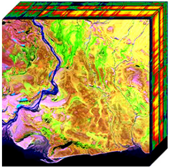


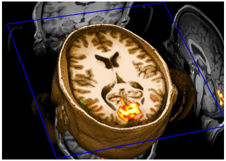
$\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}$
$\mathbf{M}_{ij} \in \mathbb{R}$ for
$i \in [d_1], j \in [d_2]$

$\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$
$(\mathcal{T}_{ijk}) \in \mathbb{R}$ for $i \in [d_1], j \in [d_2], k \in [d_3]$

Tensors and Machine Learning

(i) Data has a tensor structure: color image, video, multivariate time series...



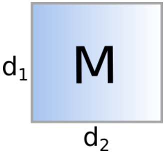


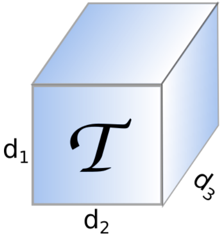
(ii) Tensor as parameters of a model: weighted tree automata, factorization machines...

$$f(\mathbf{x}) = \sum_{i,j,k} \mathbf{W}_{i,j,k} \mathbf{x}_i \mathbf{x}_j \mathbf{x}_k$$

(iii) Tensors as tools: tensor method of moments, system of polynomial equations, layer compression in neural networks...

Tensors





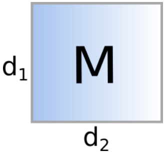
$\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}$

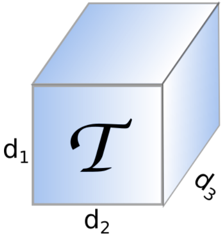
$\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$

$\mathbf{M}_{ij} \in \mathbb{R}$ for $i \in [d_1], j \in [d_2]$

$(\mathcal{T}_{ijk}) \in \mathbb{R}$ for $i \in [d_1], j \in [d_2], k \in [d_3]$

Tensors





$$\mathbf{M} \in \mathbb{R}^{d_1 \times d_2} \qquad \mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$$

$$\mathbf{M}_{ij} \in \mathbb{R} \text{ for } i \in [d_1], j \in [d_2] \quad (\mathcal{T}_{ijk}) \in \mathbb{R} \text{ for } i \in [d_1], j \in [d_2], k \in [d_3]$$

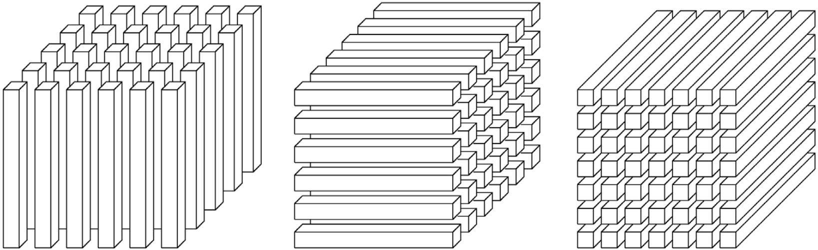
- Outer product. If $\mathbf{u} \in \mathbb{R}^{d_1}$, $\mathbf{v} \in \mathbb{R}^{d_2}$, $\mathbf{w} \in \mathbb{R}^{d_3}$:

$$\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^\top \in \mathbb{R}^{d_1 \times d_2} \qquad (\mathbf{u} \otimes \mathbf{v})_{i,j} = \mathbf{u}_i \mathbf{v}_j$$

$$\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w} \in \mathbb{R}^{d_1 \times d_2 \times d_3} \qquad (\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w})_{i,j,k} = \mathbf{u}_i \mathbf{v}_j \mathbf{w}_k$$

Tensors: mode-n fibers

- Matrices have rows and columns, tensors have fibers[1]:



(a) Mode-1 (column) fibers: $\mathbf{x}_{:jk}$ (b) Mode-2 (row) fibers: $\mathbf{x}_{i:k}$ (c) Mode-3 (tube) fibers: $\mathbf{x}_{ij:}$

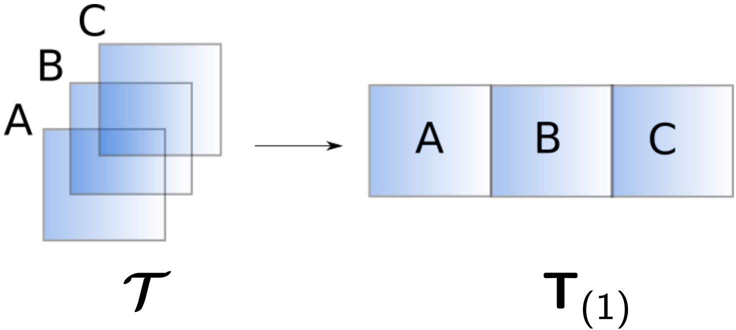
Fig. 2.1 *Fibers of a 3rd-order tensor.*

[1] fig. from [Kolda and Bader, *Tensor decompositions and applications*, 2009].

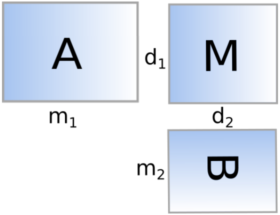
Tensors: Matricizations

- $\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ can be reshaped into a matrix as

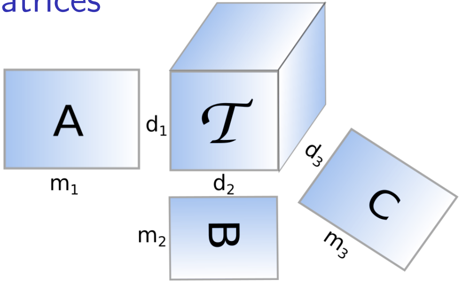
$$\mathbf{T}_{(1)} \in \mathbb{R}^{d_1 \times d_2 d_3}$$
$$\mathbf{T}_{(2)} \in \mathbb{R}^{d_2 \times d_1 d_3}$$
$$\mathbf{T}_{(3)} \in \mathbb{R}^{d_3 \times d_1 d_2}$$



Tensors: Multiplication with Matrices



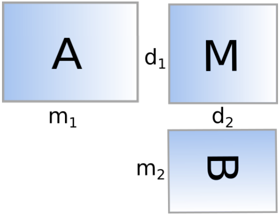
$$\mathbf{AMB}^\top \in \mathbb{R}^{m_1 \times m_2}$$



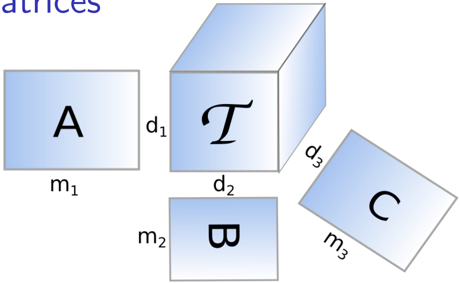
$$\mathcal{T} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$$

For vectors, we write $\mathcal{T} \bullet_n \mathbf{v} = \mathcal{T} \times_n \mathbf{v}^\top$

Tensors: Multiplication with Matrices



$$\mathbf{AMB}^\top \in \mathbb{R}^{m_1 \times m_2}$$



$$\mathcal{T} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$$

For vectors, we write $\mathcal{T} \bullet_n \mathbf{v} = \mathcal{T} \times_n \mathbf{v}^\top$

ex: If $\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ and $\mathbf{B} \in \mathbb{R}^{m_2 \times d_2}$, then $\mathcal{T} \times_2 \mathbf{B} \in \mathbb{R}^{d_1 \times m_2 \times d_3}$ and

$$(\mathcal{T} \times_2 \mathbf{B})_{i_1, i_2, i_3} = \sum_{k=1}^{d_2} \mathcal{T}_{i_1, k, i_3} \mathbf{B}_{i_2, k} \text{ for all } i_1 \in [d_1], i_2 \in [m_2], i_3 \in [d_3].$$

Tensors are not easy...

MOST TENSOR PROBLEMS ARE NP HARD

CHRISTOPHER J. HILLAR AND LEK-HENG LIM

ABSTRACT. The idea that one might extend numerical linear algebra, the collection of matrix computational methods that form the workhorse of scientific and engineering computing, to *numerical multilinear algebra*, an analogous collection of tools involving hypermatrices/tensors, appears very promising and has attracted a lot of attention recently. We examine here the computational tractability of some core problems in numerical multilinear algebra. We show that tensor analogues of several standard problems that are readily computable in the matrix (i.e. 2-tensor) case are NP hard. Our list here includes: determining the feasibility of a system of bilinear equations, determining an eigenvalue, a singular value, or the spectral norm of a 3-tensor, determining a best rank-1 approximation to a 3-tensor, determining the rank of a 3-tensor over $\mathbb{R}$ or $\mathbb{C}$. Hence making tensor computations feasible is likely to be a challenge.

[Hillar and Lim, *Most tensor problems are NP-hard*, Journal of the ACM, 2013.]

Tensors vs. Matrices: Rank

- The rank of a matrix $\mathbf{M}$ is:
  - the number of linearly independent columns of $\mathbf{M}$
  - the number of linearly independent rows of $\mathbf{M}$
  - the smallest integer R such that $\mathbf{M}$ can be written as a sum of R rank-one matrix:
$$\mathbf{M} = \sum_{i=1}^{R} \mathbf{u}_i \mathbf{v}_i^\top.$$

Tensors vs. Matrices: Rank

- The rank of a matrix $\mathbf{M}$ is:
  - the number of linearly independent columns of $\mathbf{M}$
  - the number of linearly independent rows of $\mathbf{M}$
  - the smallest integer R such that $\mathbf{M}$ can be written as a sum of R rank-one matrix:

$$\mathbf{M} = \sum_{i=1}^{R} \mathbf{u}_i \mathbf{v}_i^\top .$$

- The multilinear rank of a tensor $\mathcal{T}$ is a tuple of integers (R_1, R_2, R_3) where R_n is the number of linearly independent mode-n fibers of $\mathcal{T}$:

$$R_n = \mathrm{rank}(\mathbf{T}_{(n)})$$

Tensors vs. Matrices: Rank

- The rank of a matrix $\mathbf{M}$ is:
  - the number of linearly independent columns of $\mathbf{M}$
  - the number of linearly independent rows of $\mathbf{M}$
  - the smallest integer R such that $\mathbf{M}$ can be written as a sum of R rank-one matrix:

$$\mathbf{M} = \sum_{i=1}^{R} \mathbf{u}_i \mathbf{v}_i^\top .$$

- The multilinear rank of a tensor $\boldsymbol{\mathcal{T}}$ is a tuple of integers (R_1, R_2, R_3) where R_n is the number of linearly independent mode-n fibers of $\boldsymbol{\mathcal{T}}$:

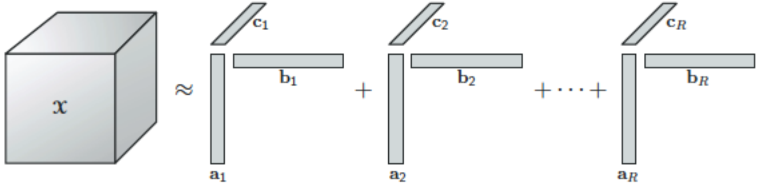
$$R_n = \mathrm{rank}(\mathbf{T}_{(n)})$$

- The CP rank of $\boldsymbol{\mathcal{T}}$ is the smallest integer R such that $\boldsymbol{\mathcal{T}}$ can be written as a sum of R rank-one tensors:

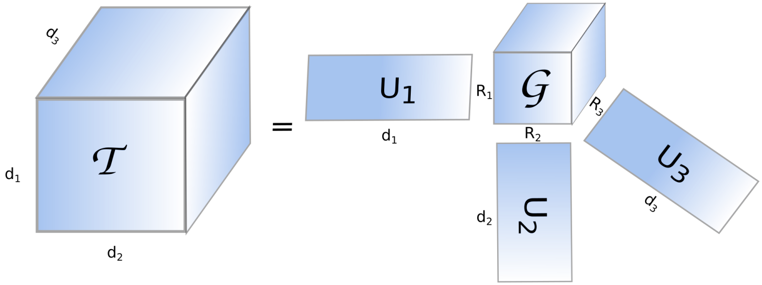
$$\boldsymbol{\mathcal{T}} = \sum_{i=1}^{R} \mathbf{u}_i \otimes \mathbf{v}_i \otimes \mathbf{w}_i .$$

CP and Tucker decomposition

- CP decomposition[2]:



- Tucker decomposition:



[2] fig. from [Kolda and Bader, *Tensor decompositions and applications*, 2009].

Hardness results

- Those are all NP-hard for tensor of order ≥ 3 in general:
  - Compute the CP rank of a given tensor
  - Find the best approximation with CP rank R of a given tensor
  - Find the best approximation with multilinear rank $(R_1, \cdots, R_p)$ of a given tensor (*)
  - ...
- On the positive side:
  - Computing the multilinear rank is easy and efficient algorithms exist for (*).
  - Under mild conditions, the CP decomposition is unique (modulo scaling and permutations).

  $\Rightarrow$ Very relevant for model identifiability...

Density Estimation: Learning from Data

$$\begin{tikzcd}
f \arrow[d] \\
\mathcal{S} = \{x_1, \cdots, x_n\} \arrow[d] \\
\widehat{f}
\end{tikzcd}$$

Learning from Data: Gaussian

$$\mathcal{N}(x; \mu, \sigma^2)$$

$$\downarrow$$

$$\mathcal{S} = \{x_1, \cdots, x_n\}$$

$$\downarrow$$

$$\begin{cases} \mathbb{E}[x] & = & \mu & \simeq \frac{1}{n}\sum_{i=1}^{n} x_i \\ \mathbb{E}[x^2] & = & \sigma^2 + \mu^2 & \simeq \frac{1}{n}\sum_{i=1}^{n} x_i^2 \end{cases}$$

$$\downarrow$$

$$\widehat{\mu}, \widehat{\sigma}^2$$

Learning from Data: Method of Moments (Pearson, 1894)

$$f(x; \theta_1, \cdots, \theta_k)$$

$$\downarrow$$

$$\mathcal{S} = \{x_1, \cdots, x_n\}$$

$$\downarrow$$

$$\begin{cases} \mathbb{E}[x] & = g_1(\theta_1, \cdots, \theta_k) & \simeq \frac{1}{n}\sum_{i=1}^n x_i \\ \mathbb{E}[x^2] & = g_2(\theta_1, \cdots, \theta_k) & \simeq \frac{1}{n}\sum_{i=1}^n x_i^2 \\ & \vdots & \\ \mathbb{E}[x^k] & = g_k(\theta_1, \cdots, \theta_k) & \simeq \frac{1}{n}\sum_{i=1}^n x_i^k \end{cases}$$

$$\downarrow$$

$$\widehat{\theta}_1, \cdots \widehat{\theta}_k$$

Method of Moments: Gaussian distribution

- If X follows a normal distribution $\mathcal{N}(\mu, \sigma^2)$:

$$\mathbb{E}[X] = \mu$$
$$\mathbb{E}[X^2] = \sigma^2 + \mu^2$$

- If $S = \{X_1, \cdots X_n\}$ are i.i.d. from $\mathcal{N}(\mu, \sigma^2)$, by the law of large numbers:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \to_n \mu$$
$$\hat{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n} X_i^2 - \hat{\mu}^2 \quad \to_n \sigma^2$$

Method of Moments: Gaussian distribution

- If X follows a normal distribution $\mathcal{N}(\mu, \sigma^2)$:

$$\mathbb{E}[X] = \mu$$
$$\mathbb{E}[X^2] = \sigma^2 + \mu^2$$

- If $S = \{X_1, \cdots X_n\}$ are i.i.d. from $\mathcal{N}(\mu, \sigma^2)$, by the law of large numbers:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \rightarrow_n \mu$$
$$\hat{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n} X_i^2 - \hat{\mu}^2 \quad \rightarrow_n \sigma^2$$

$\Rightarrow$ Here MoM and ML estimators are equal but this is not always the case (e.g. uniform distribution).

Method of Moments: Binomial distribution

- If X follows a binomial distribution $\mathcal{B}(k, p)$, then $X = \sum_{i=1}^{k} B_i$ where each B_i follows a Bernoulli with parameter p. Hence,

$$\mathbb{E}[X] = \mathbb{E}[B_1 + \cdots + B_k] = \sum_{i=1}^{k} \mathbb{E}[B_i] = kp$$

$$\mathbb{E}[X^2] = \mathbb{E}[(B_1 + \cdots + B_k)^2] = k^2 p^2 + kp(1-p)$$

Method of Moments: Binomial distribution

- If X follows a binomial distribution $\mathcal{B}(k, p)$, then $X = \sum_{i=1}^{k} B_i$ where each B_i follows a Bernoulli with parameter p. Hence,

$$\mathbb{E}[X] = \mathbb{E}[B_1 + \cdots + B_k] = \sum_{i=1}^{k} \mathbb{E}[B_i] = kp$$
$$\mathbb{E}[X^2] = \mathbb{E}[(B_1 + \cdots + B_k)^2] = k^2p^2 + kp(1-p)$$

- If $S = \{X_1, \cdots X_n\}$ are i.i.d. from $\mathcal{B}(k, p)$, by the LLN:

$$\hat{k} = m_1^2/(m_1^2 + m_1 - m_2) \qquad \to_n k$$
$$\hat{p} = (m_1^2 + m_1 - m_2)/m_1 \qquad \to_n p$$

where $m_1 = \frac{1}{n}\sum_i X_i$ and $m_2 = \frac{1}{n}\sum_i X_i^2$.

Method of Moments: Binomial distribution

- If X follows a binomial distribution $\mathcal{B}(k, p)$, then $X = \sum_{i=1}^{k} B_i$ where each B_i follows a Bernoulli with parameter p. Hence,

$$\mathbb{E}[X] = \mathbb{E}[B_1 + \cdots + B_k] = \sum_{i=1}^{k} \mathbb{E}[B_i] = kp$$
$$\mathbb{E}[X^2] = \mathbb{E}[(B_1 + \cdots + B_k)^2] = k^2p^2 + kp(1-p)$$

- If $S = \{X_1, \cdots X_n\}$ are i.i.d. from $\mathcal{B}(k, p)$, by the LLN:

$$\hat{k} = m_1^2/(m_1^2 + m_1 - m_2) \qquad \rightarrow_n k$$
$$\hat{p} = (m_1^2 + m_1 - m_2)/m_1 \qquad \rightarrow_n p$$

where $m_1 = \frac{1}{n}\sum_i X_i$ and $m_2 = \frac{1}{n}\sum_i X_i^2$.

- $0 \leq \hat{p} \leq 1$ but $\hat{k}$ may not be an integer.

Method of Moments: Multivariate case

- What if the random variable $\mathbf{x}$ takes its values in $\mathbb{R}^d$?

Method of Moments: Multivariate case

- What if the random variable $\mathbf{x}$ takes its values in $\mathbb{R}^d$?
- Let's look at the multivariate normal. If $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the first and second moments are

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu} \quad \text{and} \quad \mathbb{E}[\mathbf{x}\mathbf{x}^\top] = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^\top$$

Method of Moments: Multivariate case

- What if the random variable $\mathbf{x}$ takes its values in $\mathbb{R}^d$?
- Let's look at the multivariate normal. If $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the first and second moments are

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu} \quad \text{and} \quad \mathbb{E}[\mathbf{x}\mathbf{x}^\top] = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^\top$$

- What if we need higher order moments? The second order moment is $\mathbb{E}[\mathbf{x}\mathbf{x}^\top]$, but what is e.g. the third order moment?

Method of Moments: Multivariate case

- What if the random variable $\mathbf{x}$ takes its values in $\mathbb{R}^d$?
- Let's look at the multivariate normal. If $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the first and second moments are

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu} \quad \text{and} \quad \mathbb{E}[\mathbf{x}\mathbf{x}^\top] = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^\top$$

- What if we need higher order moments? The second order moment is $\mathbb{E}[\mathbf{x}\mathbf{x}^\top]$, but what is e.g. the third order moment?

$$\mathbb{E}[\mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}]$$

Tensor Decomposition for Learning Latent Variable Models

Latent Variable Model:

$$f(\mathbf{x}) = \sum_{i=1}^{k} w_i f_i(\mathbf{x}; \boldsymbol{\mu}_i)$$

$$\Big\downarrow$$

$$\mathcal{S} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \subset \mathbb{R}^d$$

$$\Big\downarrow \text{ Structure in the Low Order Moments}$$

$$\begin{cases} \mathbb{E}[\mathbf{x} \otimes \mathbf{x}] & = g_1\left(\sum_{i=1}^{k} w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i\right) \\ \mathbb{E}[\mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}] & = g_2\left(\sum_{i=1}^{k} w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i\right) \end{cases}$$

$$\Big\downarrow \text{ Tensor Power Method}$$

$$\widehat{w}_i, \widehat{\boldsymbol{\mu}}_i$$

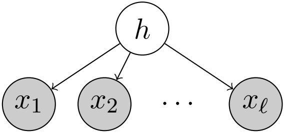
Single Topic Model

- Documents modeled as bags of words:
  - Vocabulary of d words
  - k different topics
  - ℓ words per document
- Documents are drawn as follows:
  (1) Draw a topic h randomly with probability $\mathbb{P}[h = j] = w_j$ for $j \in [k]$
  (2) Draw ℓ word independently according to the distribution $\boldsymbol{\mu}_h \in \Delta^{d-1}$

Single Topic Model

- Documents modeled as bags of words:
  - Vocabulary of d words
  - k different topics
  - ℓ words per document
- Documents are drawn as follows:
  (1) Draw a topic h randomly with probability $\mathbb{P}[h = j] = w_j$ for $j \in [k]$
  (2) Draw ℓ word independently according to the distribution $\boldsymbol{\mu}_h \in \Delta^{d-1}$

$\Rightarrow$ Words are independent given the topic:



Single Topic Model

- Documents modeled as bags of words:
  - Vocabulary of d words
  - k different topics
  - ℓ words per document
- Documents are drawn as follows:
  (1) Draw a topic h randomly with probability $\mathbb{P}[h = j] = w_j$ for $j \in [k]$
  (2) Draw ℓ word independently according to the distribution $\boldsymbol{\mu}_h \in \Delta^{d-1}$
- Using one-hot encoding for the words $\mathbf{x}_1, \cdots, \mathbf{x}_\ell \in \mathbb{R}^d$ in a document we have

$$
\begin{aligned}
(\mathbb{E}[\mathbf{x}_1])_i &= \mathbb{P}[\text{1st word} = i] \\
(\mathbb{E}[\mathbf{x}_1 \otimes \mathbf{x}_2])_{i,j} &= \mathbb{P}[\text{1st word} = i, \text{2nd word} = j] \\
&\vdots \\
(\mathbb{E}[\mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \cdots \otimes \mathbf{x}_\ell])_{i_1,\cdots,i_\ell} &= \mathbb{P}[\text{1st word} = i_1, \text{2nd word} = i_2, \cdots, \\
&\qquad \ell\text{-th word} = i_\ell]
\end{aligned}
$$

Single Topic Model

- Documents are drawn as follows:
  (1) Draw a topic h randomly with probability $\mathbb{P}[h = j] = w_j$ for $j \in [k]$
  (2) Draw ℓ word independently according to the distribution $\boldsymbol{\mu}_h \in \Delta^{d-1}$
- Using one-hot encoding for the words $\mathbf{x}_1, \cdots, \mathbf{x}_\ell \in \mathbb{R}^d$ in a document we also have

$$\mathbb{E}[\mathbf{x}_1 \mid h = j] = \boldsymbol{\mu}_j$$
$$\mathbb{E}[\mathbf{x}_1 \otimes \mathbf{x}_2 \mid h = j] = \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j$$
$$\mathbb{E}[\mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \mid h = j] = \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j$$

Single Topic Model

- Documents are drawn as follows:
  (1) Draw a topic h randomly with probability $\mathbb{P}[h=j]=w_j$ for $j \in [k]$
  (2) Draw ℓ word independently according to the distribution $\boldsymbol{\mu}_h \in \Delta^{d-1}$
- Using one-hot encoding for the words $\mathbf{x}_1, \cdots, \mathbf{x}_\ell \in \mathbb{R}^d$ in a document we also have

$$\mathbb{E}[\mathbf{x}_1 \mid h=j] = \boldsymbol{\mu}_j$$
$$\mathbb{E}[\mathbf{x}_1 \otimes \mathbf{x}_2 \mid h=j] = \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j$$
$$\mathbb{E}[\mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \mid h=j] = \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j$$

  From which we can deduce

$$\mathbb{E}[\mathbf{x}_1 \otimes \mathbf{x}_2] = \sum_{j=1}^{k} w_j \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j$$

$$\mathbb{E}[\mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3] = \sum_{j=1}^{k} w_j \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j$$

Mixture of Spherical Gaussians

- Mixture of k d-dimensional Gaussians ($k \leq d$) *with the same variance* $\sigma^2\mathbf{I}$:
  (1) Draw a Gaussian h randomly with $\mathbb{P}[h = j] = w_j$ for $j \in [k]$
  (2) Draw $\mathbf{x}$ from the multivariate normal $\mathcal{N}(\boldsymbol{\mu}_h, \sigma^2\mathbf{I})$

Mixture of Spherical Gaussians

- Mixture of k d-dimensional Gaussians ($k \le d$) *with the same variance* $\sigma^2 \mathbf{I}$:
  (1) Draw a Gaussian h randomly with $\mathbb{P}[h = j] = w_j$ for $j \in [k]$
  (2) Draw $\mathbf{x}$ from the multivariate normal $\mathcal{N}(\boldsymbol{\mu}_h, \sigma^2 \mathbf{I})$
- The first three moments are:

$$\mathbb{E}[\mathbf{x}] = \sum_{j=1}^{k} w_j \boldsymbol{\mu}_j$$

$$\mathbb{E}[\mathbf{x} \otimes \mathbf{x}] = \sigma^2 \mathbf{I} + \sum_{j=1}^{k} w_j \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j$$

$$\mathbb{E}[\mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}] = \sum_{j=1}^{k} w_j \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j + \sigma^2 \sum_{i=1}^{d} (\mathbb{E}[\mathbf{x}] \otimes \mathbf{e}_i \otimes \mathbf{e}_i + \mathbf{e}_i \otimes \mathbb{E}[\mathbf{x}] \otimes \mathbf{e}_i + \mathbf{e}_i \otimes \mathbf{e}_i \otimes \mathbb{E}[\mathbf{x}])$$

Mixture of Spherical Gaussians

- Mixture of k Gaussians *with the same variance* $\sigma^2\mathbf{I}$:
  (1) Draw a Gaussian h randomly with probability $\mathbb{P}[h=j]=w_j$ for $j\in[k]$
  (2) Draw $\mathbf{x}$ from the multivariate normal $\mathcal{N}(\boldsymbol{\mu}_h,\sigma^2\mathbf{I})$
- Hence

$$\mathbf{M}_2=\mathbb{E}[\mathbf{x}\otimes\mathbf{x}]-\sigma^2\mathbf{I}=\sum_{j=1}^{k}w_j\boldsymbol{\mu}_j\otimes\boldsymbol{\mu}_j$$

$$\begin{aligned}\boldsymbol{\mathcal{M}}_3&=\mathbb{E}[\mathbf{x}\otimes\mathbf{x}\otimes\mathbf{x}]-\sigma^2\sum_{i=1}^{d}(\mathbb{E}[\mathbf{x}]\otimes\mathbf{e}_i\otimes\mathbf{e}_i+\mathbf{e}_i\otimes\mathbb{E}[\mathbf{x}]\otimes\mathbf{e}_i+\cdots)\\&=\sum_{j=1}^{k}w_j\boldsymbol{\mu}_j\otimes\boldsymbol{\mu}_j\otimes\boldsymbol{\mu}_j\end{aligned}$$

Mixture of Spherical Gaussians

- Mixture of k Gaussians *with the same variance* $\sigma^2 \mathbf{I}$:
  (1) Draw a Gaussian h randomly with probability $\mathbb{P}[h = j] = w_j$ for $j \in [k]$
  (2) Draw $\mathbf{x}$ from the multivariate normal $\mathcal{N}(\boldsymbol{\mu}_h, \sigma^2 \mathbf{I})$
- Hence

$$\mathbf{M}_2 = \mathbb{E}[\mathbf{x} \otimes \mathbf{x}] - \sigma^2 \mathbf{I} = \sum_{j=1}^{k} w_j \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j$$

$$\begin{aligned} \boldsymbol{\mathcal{M}}_3 &= \mathbb{E}[\mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}] - \sigma^2 \sum_{i=1}^{d} (\mathbb{E}[\mathbf{x}] \otimes \mathbf{e}_i \otimes \mathbf{e}_i + \mathbf{e}_i \otimes \mathbb{E}[\mathbf{x}] \otimes \mathbf{e}_i + \cdots) \\ &= \sum_{j=1}^{k} w_j \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j \end{aligned}$$

- How can we estimate σ^2?

Mixture of Spherical Gaussians

- Mixture of k Gaussians *with the same variance* $\sigma^2 \mathbf{I}$.
- σ^2 is the smallest eigenvalue of the covariance matrix!

proof: Let $\bar{\boldsymbol{\mu}} = \mathbb{E}[\mathbf{x}] = \sum_j w_j \boldsymbol{\mu}_j$, we have

$$\mathbf{S} = \mathbb{E}[(\mathbf{x} - \bar{\boldsymbol{\mu}}) \otimes (\mathbf{x} - \bar{\boldsymbol{\mu}})] = \sum_{j=1}^{k} w_j (\boldsymbol{\mu}_j - \bar{\boldsymbol{\mu}}) \otimes (\boldsymbol{\mu}_j - \bar{\boldsymbol{\mu}}) + \sigma^2 \mathbf{I}$$

Let $\mathbf{A} = \sum_{j=1}^{k} w_j (\boldsymbol{\mu}_j - \bar{\boldsymbol{\mu}}) \otimes (\boldsymbol{\mu}_j - \bar{\boldsymbol{\mu}})$. $\mathbf{A}$ is p.s.d. and has rank $r \leq k - 1 < d$. Hence if $\mathbf{U}$ diagonalizes $\mathbf{A}$ we have $\mathbf{U}\mathbf{A}\mathbf{U}^\top = \mathbf{D}$ where $\mathbf{D}$ is diagonal with its first $d - r$ diagonal entries equal to 0. The results follow from observing that $\mathbf{U}\mathbf{S}\mathbf{U}^\top = \mathbf{D} + \sigma^2 \mathbf{I}$.

Mixture of Spherical Gaussians

- When each spherical Gaussians has its own variance σ_j^2 we have the following result:

Theorem (D. Hsu and D. Kakade, ITCS, 2013.)

- *The average variance* $\bar{\sigma}^2 = \sum_{i=1}^{k} w_i \sigma_i^2$ *is the smallest eigenvalue of the covariance matrix* $\mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^\top]$.
- *Let* $\mathbf{v}$ *be any unit-norm eigenvector corresponding to* $\bar{\sigma}^2$ *and let*

$$\mathbf{m}_1 = \mathbb{E}[\mathbf{x}(\mathbf{v}^\top(\mathbf{x} - \mathbb{E}[\mathbf{x}]))^2], \qquad \mathbf{M}_2 = \mathbb{E}[\mathbf{x} \otimes \mathbf{x}] - \bar{\sigma}^2 \mathbf{I}, \quad \textit{and}$$

$$\boldsymbol{\mathcal{M}}_3 = \mathbb{E}[\mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}] - \sum_{i=1}^{n} [\mathbf{m}_1 \otimes \mathbf{e}_i \otimes \mathbf{e}_i + \mathbf{e}_i \otimes \mathbf{m}_1 \otimes \mathbf{e}_i + \mathbf{e}_i \otimes \mathbf{e}_i \otimes \mathbf{m}_1]$$

where $\mathbf{e}_1, \cdots, \mathbf{e}_n$ *is the coordinate basis of* $\mathbb{R}^n$. *Then,*

$$\mathbf{m}_1 = \sum_{i=1}^{k} w_i \sigma_i^2 \boldsymbol{\mu}_i, \quad \mathbf{M}_2 = \sum_{i=1}^{k} w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \ , \ \textit{and} \quad \boldsymbol{\mathcal{M}}_3 = \sum_{i=1}^{k} w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i$$

Structure in the Low-Order Moments of Latent Variable Models

- For single topic models and spherical Gaussian mixtures, we showed that the tensors $\mathbf{M}_2 = \sum_{i=1}^{k} w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i$ and $\boldsymbol{\mathcal{M}}_3 = \sum_{i=1}^{k} w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i$ can be expressed as functions of the 2nd and 3rd order moments.
- Similar results can be shown for hidden Markov models, latent Dirichlet allocation, independent component analysis and multiview models[3].
- $\mathbf{M}_2$ and $\boldsymbol{\mathcal{M}}_3$ can be estimated from data, now it remains to recover the parameters w_i, $\boldsymbol{\mu}_i$ from $\mathbf{M}_2$ and $\boldsymbol{\mathcal{M}}_3$.

[3] see [Anandkumar et al. *Tensor decompositions for learning latent variable models*, JMLR 2014].

Tensor Decomposition for Learning Latent Variable Models

Latent Variable Model:

$$f(\mathbf{x}) = \sum_{i=1}^{k} w_i f_i(\mathbf{x}; \boldsymbol{\mu}_i)$$

$$\Big\downarrow$$

$$\mathcal{S} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \subset \mathbb{R}^d$$

$$\Big\downarrow \text{ Structure in the Low Order Moments}$$

$$\begin{cases} \mathbb{E}[\mathbf{x} \otimes \mathbf{x}] & = g_1\left(\sum_{i=1}^{k} w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i\right) \\ \mathbb{E}[\mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}] & = g_2\left(\sum_{i=1}^{k} w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i\right) \end{cases}$$

$$\Big\downarrow \text{ Tensor Power Method}$$

$$\widehat{w}_i, \widehat{\boldsymbol{\mu}}_i$$

Method of Moments with Tensor Decomposition

$$\begin{cases} \widehat{\mathbf{M}}_2 & \simeq \sum_{i=1}^{k} w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \\ \widehat{\boldsymbol{\mathcal{M}}}_3 & \simeq \sum_{i=1}^{k} w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \end{cases}$$

$$\Big\downarrow ?$$

$$\widehat{w}_i, \widehat{\boldsymbol{\mu}}_i$$

- $k \leq d$
- $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k \in \mathbb{R}^d$ are linearly independent
- $w_1, \cdots, w_k \in \mathbb{R}$ are strictly positive real numbers

Method of Moments with Tensor Decomposition

- Under which conditions can we recover the weights w_j and vectors $\boldsymbol{\mu}_j$ for $j \in [k]$ from $\mathbf{M}_2 = \sum_j w_j \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j$?

Method of Moments with Tensor Decomposition

- Under which conditions can we recover the weights w_j and vectors $\boldsymbol{\mu}_j$ for $j \in [k]$ from $\mathbf{M}_2 = \sum_j w_j \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j$?
  (i) If the $\boldsymbol{\mu}_j$ are orthonormal and the w_j are distinct, they are the unit eigenvectors of $\mathbf{M}_2$ and the weights are its eigenvalues.
  $\rightarrow$ We would still need to recover the signs of the $\boldsymbol{\mu}_j$...
  (ii) Otherwise, this is not possible!

Method of Moments with Tensor Decomposition

- Under which conditions can we recover the weights w_j and vectors $\boldsymbol{\mu}_j$ for $j \in [k]$ from $\boldsymbol{\mathcal{M}}_3 = \sum_j w_j \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j$?

Method of Moments with Tensor Decomposition

- Under which conditions can we recover the weights w_j and vectors $\boldsymbol{\mu}_j$ for $j \in [k]$ from $\boldsymbol{\mathcal{M}}_3 = \sum_j w_j \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j$?
  - $\rightarrow$ We can recover $\pm w_j^{1/3} \boldsymbol{\mu}_j$ if the $\boldsymbol{\mu}_j$ are linearly independent using Jennrich's algorithm (this is sufficient for e.g. single topics model)

Method of Moments with Tensor Decomposition

- Under which conditions can we recover the weights w_j and vectors $\boldsymbol{\mu}_j$ for $j \in [k]$ from $\boldsymbol{\mathcal{M}}_3 = \sum_j w_j \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j$?
  - $\rightarrow$ We can recover $\pm w_j^{1/3} \boldsymbol{\mu}_j$ if the $\boldsymbol{\mu}_j$ are linearly independent using Jennrich's algorithm (this is sufficient for e.g. single topics model)
  - $\rightarrow$ For any vector $\mathbf{v} \in \mathbb{R}^d$ we have

$$\boldsymbol{\mathcal{M}}_3 \bullet_1 \mathbf{v} = \sum_{j=1}^{k} w_j (\mathbf{v}^\top \boldsymbol{\mu}_j) \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^\top.$$

thus if the $\boldsymbol{\mu}_j$ are orthonormal we can recover the $\boldsymbol{\mu}_j$ as eigenvectors and the w_j by solving the linear equation $\lambda_j = w_j (\mathbf{v}^\top \boldsymbol{\mu}_j)$.
(No more ambiguity for the signs of the $\boldsymbol{\mu}_j$ since the w_j are positive.)

Method of Moments with Tensor Decomposition

- Under which conditions can we recover the weights w_j and vectors $\boldsymbol{\mu}_j$ for $j \in [k]$ from $\boldsymbol{\mathcal{M}}_3 = \sum_j w_j \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j$?
  - → We can recover $\pm w_j^{1/3} \boldsymbol{\mu}_j$ if the $\boldsymbol{\mu}_j$ are linearly independent using Jennrich's algorithm (this is sufficient for e.g. single topics model)
  - → For any vector $\mathbf{v} \in \mathbb{R}^d$ we have

$$\boldsymbol{\mathcal{M}}_3 \bullet_1 \mathbf{v} = \sum_{j=1}^{k} w_j (\mathbf{v}^\top \boldsymbol{\mu}_j) \boldsymbol{\mu}_j \otimes \boldsymbol{\mu}_j = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^\top.$$

thus if the $\boldsymbol{\mu}_j$ are orthonormal we can recover the $\boldsymbol{\mu}_j$ as eigenvectors and the w_j by solving the linear equation $\lambda_j = w_j (\mathbf{v}^\top \boldsymbol{\mu}_j)$.
(No more ambiguity for the signs of the $\boldsymbol{\mu}_j$ since the w_j are positive.)
idea: Use $\mathbf{M}_2$ to whiten the tensor $\boldsymbol{\mathcal{M}}_3$, then recover the parameters using eigen-decomposition or tensor power method.

Jennrich's algorithm. [Harshman, 1970]

Let $\mathcal{T} = \sum_{j=1}^{k} \mathbf{v}_j \otimes \mathbf{v}_j \otimes \mathbf{v}_j$ where the $\mathbf{v}_j$ are linearly independent (this implies $k \leq d$).

- For any $\mathbf{x} \in \mathbb{R}^d$, we have

$$\mathbf{T}_{\mathbf{x}} := \mathcal{T} \bullet_1 \mathbf{x} = \mathbf{U}\mathbf{D}_{\mathbf{x}}\mathbf{U}^\top$$

where $\mathbf{U} = [\mathbf{v}_1, \cdots, \mathbf{v}_k] \in \mathbb{R}^{d \times k}$ and $\mathbf{D}_{\mathbf{x}}$ is the diagonal matrix with entries $\mathbf{v}_j^\top \mathbf{x}$.

Jennrich's algorithm. [Harshman, 1970]

Let $\mathcal{T} = \sum_{j=1}^{k} \mathbf{v}_j \otimes \mathbf{v}_j \otimes \mathbf{v}_j$ where the $\mathbf{v}_j$ are linearly independent (this implies $k \leq d$).

- For any $\mathbf{x} \in \mathbb{R}^d$, we have

$$\mathbf{T}_\mathbf{x} := \mathcal{T} \bullet_1 \mathbf{x} = \mathbf{U}\mathbf{D}_\mathbf{x}\mathbf{U}^\top$$

  where $\mathbf{U} = [\mathbf{v}_1, \cdots, \mathbf{v}_k] \in \mathbb{R}^{d \times k}$ and $\mathbf{D}_\mathbf{x}$ is the diagonal matrix with entries $\mathbf{v}_j^\top \mathbf{x}$.

- If we draw two unit vectors $\mathbf{x}, \mathbf{y}$ at random in $\mathbb{R}^d$ we have

$$\mathbf{T}_\mathbf{x}(\mathbf{T}_\mathbf{y})^+ = \mathbf{U}\mathbf{D}_\mathbf{x}(\mathbf{D}_\mathbf{y})^{-1}\mathbf{U}^+.$$

  By drawing $\mathbf{x}$ and $\mathbf{y}$ at random we ensure that, with probability one,

  - $\mathbf{D}_\mathbf{y}$ is invertible
  - the diagonal entries of $\mathbf{D}_\mathbf{x}(\mathbf{D}_\mathbf{y})^{-1}$ are distinct

Jennrich's algorithm. [Harshman, 1970]

Let $\mathcal{T} = \sum_{j=1}^{k} \mathbf{v}_j \otimes \mathbf{v}_j \otimes \mathbf{v}_j$ where the $\mathbf{v}_j$ are linearly independent (this implies $k \leq d$).

- For any $\mathbf{x} \in \mathbb{R}^d$, we have

$$\mathbf{T_x} := \mathcal{T} \bullet_1 \mathbf{x} = \mathbf{U}\mathbf{D_x}\mathbf{U}^\top$$

  where $\mathbf{U} = [\mathbf{v}_1, \cdots, \mathbf{v}_k] \in \mathbb{R}^{d \times k}$ and $\mathbf{D_x}$ is the diagonal matrix with entries $\mathbf{v}_j^\top \mathbf{x}$.

- If we draw two unit vectors $\mathbf{x}, \mathbf{y}$ at random in $\mathbb{R}^d$ we have

$$\mathbf{T_x}(\mathbf{T_y})^+ = \mathbf{U}\mathbf{D_x}(\mathbf{D_y})^{-1}\mathbf{U}^+.$$

  By drawing $\mathbf{x}$ and $\mathbf{y}$ at random we ensure that, with probability one,
  - $\mathbf{D_y}$ is invertible
  - the diagonal entries of $\mathbf{D_x}(\mathbf{D_y})^{-1}$ are distinct

- Since $\mathbf{U}$ has rank k we have $\mathbf{U}^+\mathbf{U} = \mathbf{I}$ and the $\mathbf{v}_j$'s can be recovered as eigenvectors of $\mathbf{T_x}(\mathbf{T_y})^+$ (up to the signs).

Tensor Power Method / (Simultaneous) Diagonalization

We want to solve the following system of equations in w_i,$\boldsymbol{\mu}_i$:

$$\begin{cases} \mathbf{M}_2 & = \sum_{i=1}^k w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \\ \boldsymbol{\mathcal{M}}_3 & = \sum_{i=1}^k w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \end{cases}$$

Overview:

1. Use $\mathbf{M}_2$ to transform the tensor $\boldsymbol{\mathcal{M}}_3$ into an orthogonally decomposable tensor: i.e. find $\mathbf{W} \in \mathbb{R}^{k \times d}$ such that

$$\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{M}}_3 \times_1 \mathbf{W} \times_2 \mathbf{W} \times_3 \mathbf{W} = \sum_{i=1}^{k} \tilde{w}_i \tilde{\boldsymbol{\mu}}_i \otimes \tilde{\boldsymbol{\mu}}_i \otimes \tilde{\boldsymbol{\mu}}_i$$

   where the $\tilde{\boldsymbol{\mu}}_i \in \mathbb{R}^k$ are orthonormal.
2. Use (simultaneous) diagonalization or the tensor power method to recover the weights $\tilde{w}_i$ and vectors $\tilde{\boldsymbol{\mu}}_i$.
3. Recover the original weights w_i and vectors $\boldsymbol{\mu}_i$ by 'reverting' the transformation from step 1.

Orthonormalization

$$\begin{cases} \mathbf{M}_2 & = \sum_{i=1}^k w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \\ \mathcal{M}_3 & = \sum_{i=1}^k w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \end{cases}$$

- $\mathbf{M}_2 = \sum_{i=1}^k w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i = \mathbf{U}\mathbf{D}\mathbf{U}^\top$ eigendecomposition of $\mathbf{M}_2$.
- $\mathbf{W} = \mathbf{D}^{-1/2}\mathbf{U}^\top \in \mathbb{R}^{k \times d}$ and $\widetilde{\boldsymbol{\mu}}_i = \sqrt{w_i}\mathbf{W}\boldsymbol{\mu}_i \in \mathbb{R}^k$.

Orthonormalization

$$\begin{cases} \mathbf{M}_2 & = \sum_{i=1}^k w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \\ \mathcal{M}_3 & = \sum_{i=1}^k w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \end{cases}$$

- $\mathbf{M}_2 = \sum_{i=1}^k w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i = \mathbf{U}\mathbf{D}\mathbf{U}^\top$ eigendecomposition of $\mathbf{M}_2$.
- $\mathbf{W} = \mathbf{D}^{-1/2}\mathbf{U}^\top \in \mathbb{R}^{k \times d}$ and $\widetilde{\boldsymbol{\mu}}_i = \sqrt{w_i}\mathbf{W}\boldsymbol{\mu}_i \in \mathbb{R}^k$.
- We have $\widetilde{\boldsymbol{\mu}}_i^\top \widetilde{\boldsymbol{\mu}}_j = \delta_{ij}$ for all i, j, because

$$\mathbf{I} = \mathbf{W}\mathbf{M}_2\mathbf{W}^\top = \mathbf{W}\left(\sum_{i=1}^k w_i \boldsymbol{\mu}_i \boldsymbol{\mu}_i^\top\right)\mathbf{W}^\top = \sum_{i=1}^k \widetilde{\boldsymbol{\mu}}_i \widetilde{\boldsymbol{\mu}}_i^\top$$

$$\Rightarrow \mathcal{T} = \mathcal{M}_3 \times_1 \mathbf{W} \times_2 \mathbf{W} \times_3 \mathbf{W} = \sum_{i=1}^k \frac{1}{\sqrt{w_i}} \widetilde{\boldsymbol{\mu}}_i \otimes \widetilde{\boldsymbol{\mu}}_i \otimes \widetilde{\boldsymbol{\mu}}_i.$$

Orthonormalization

$$\begin{cases} \mathbf{M}_2 & = \sum_{i=1}^k w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \\ \mathcal{M}_3 & = \sum_{i=1}^k w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \end{cases}$$

- $\mathbf{M}_2 = \sum_{i=1}^k w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i = \mathbf{UDU}^\top$ eigendecomposition of $\mathbf{M}_2$.
- $\mathbf{W} = \mathbf{D}^{-1/2}\mathbf{U}^\top \in \mathbb{R}^{k \times d}$ and $\widetilde{\boldsymbol{\mu}}_i = \sqrt{w_i}\mathbf{W}\boldsymbol{\mu}_i \in \mathbb{R}^k$.
- We have $\widetilde{\boldsymbol{\mu}}_i^\top \widetilde{\boldsymbol{\mu}}_j = \delta_{ij}$ for all i, j, because

$$\mathbf{I} = \mathbf{W}\mathbf{M}_2\mathbf{W}^\top = \mathbf{W}\left(\sum_{i=1}^k w_i \boldsymbol{\mu}_i \boldsymbol{\mu}_i^\top\right)\mathbf{W}^\top = \sum_{i=1}^k \widetilde{\boldsymbol{\mu}}_i \widetilde{\boldsymbol{\mu}}_i^\top$$

$\Rightarrow$ $\mathcal{T} = \mathcal{M}_3 \times_1 \mathbf{W} \times_2 \mathbf{W} \times_3 \mathbf{W} = \sum_{i=1}^k \frac{1}{\sqrt{w_i}} \widetilde{\boldsymbol{\mu}}_i \otimes \widetilde{\boldsymbol{\mu}}_i \otimes \widetilde{\boldsymbol{\mu}}_i$.

- Since $\mathbf{UU}^\top \boldsymbol{\mu}_i = \boldsymbol{\mu}_i$ for all i we have $\mathbf{W}^+ \tilde{\boldsymbol{\mu}}_i = \sqrt{w_i}\boldsymbol{\mu}_i$.

Orthogonal Tensor Decomposition

Using $\mathbf{M}_2$ we've reduced the problem of solving

$$\begin{cases} \mathbf{M}_2 & = \sum_{i=1}^{k} w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \\ \mathcal{M}_3 & = \sum_{i=1}^{k} w_i \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \end{cases}$$

into the problem of finding an orthogonal decomposition of the tensor

$$\mathcal{T} = \mathcal{M}_3 \times_1 \mathbf{W} \times_2 \mathbf{W} \times_3 \mathbf{W} = \sum_{i=1}^{k} \tilde{w}_i \tilde{\boldsymbol{\mu}}_i \otimes \tilde{\boldsymbol{\mu}}_i \otimes \tilde{\boldsymbol{\mu}}_i.$$

Orthogonal Tensor Decomposition via Diagonalization

- We want to find the orthogonal decomposition

$$\mathcal{T} = \sum_{i=1}^{k} \tilde{w}_i \tilde{\boldsymbol{\mu}}_i \otimes \tilde{\boldsymbol{\mu}}_i \otimes \tilde{\boldsymbol{\mu}}_i \quad \in \mathbb{R}^{k \times k \times k}$$

(where the $\tilde{\boldsymbol{\mu}}_i$ are unit norm orthogonal vectors)

$\Rightarrow$ The $\tilde{w}_j$'s and $\tilde{\boldsymbol{\mu}}_j$'s can be recovered as eigenvalues/vectors of any projection $\mathcal{T} \bullet_1 \mathbf{v}$:

Orthogonal Tensor Decomposition via Diagonalization

- We want to find the orthogonal decomposition

$$\mathcal{T} = \sum_{i=1}^{k} \tilde{w}_i \tilde{\boldsymbol{\mu}}_i \otimes \tilde{\boldsymbol{\mu}}_i \otimes \tilde{\boldsymbol{\mu}}_i \quad \in \mathbb{R}^{k \times k \times k}$$

(where the $\tilde{\boldsymbol{\mu}}_i$ are unit norm orthogonal vectors)

$\Rightarrow$ The $\tilde{w}_j$'s and $\tilde{\boldsymbol{\mu}}_j$'s can be recovered as eigenvalues/vectors of any projection $\mathcal{T} \bullet_1 \mathbf{v}$:

- For any vector $\mathbf{v}$ we have

$$\mathcal{T} \bullet_1 \mathbf{v} = \sum_{j=1}^{k} \tilde{w}_j (\mathbf{v}^\top \tilde{\boldsymbol{\mu}}_j) \tilde{\boldsymbol{\mu}}_j \otimes \tilde{\boldsymbol{\mu}}_j = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^\top$$

  with $\mathbf{U} = [\tilde{\boldsymbol{\mu}}_1 \cdots \tilde{\boldsymbol{\mu}}_k]$ and $\boldsymbol{\Lambda}_{j,j} = \tilde{w}_j (\mathbf{v}^\top \tilde{\boldsymbol{\mu}}_j)$.
- $\mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^\top$ is the eigendecomposition of $\mathcal{T} \bullet_1 \mathbf{v}$ (since $\mathbf{U}^\top \mathbf{U} = \mathbf{I}$).

Orthogonal Tensor Decomposition via Diagonalization

- We want to find the orthogonal decomposition

$$\mathcal{T} = \sum_{i=1}^{k} \tilde{w}_i \tilde{\boldsymbol{\mu}}_i \otimes \tilde{\boldsymbol{\mu}}_i \otimes \tilde{\boldsymbol{\mu}}_i \quad \in \mathbb{R}^{k \times k \times k}$$

(where the $\tilde{\boldsymbol{\mu}}_i$ are unit norm orthogonal vectors)

$\Rightarrow$ The $\tilde{w}_j$'s and $\tilde{\boldsymbol{\mu}}_j$'s can be recovered as eigenvalues/vectors of any projection $\mathcal{T} \bullet_1 \mathbf{v}$:

  - For any vector $\mathbf{v}$ we have

$$\mathcal{T} \bullet_1 \mathbf{v} = \sum_{j=1}^{k} \tilde{w}_j (\mathbf{v}^\top \tilde{\boldsymbol{\mu}}_j) \tilde{\boldsymbol{\mu}}_j \otimes \tilde{\boldsymbol{\mu}}_j = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$$

    with $\mathbf{U} = [\tilde{\boldsymbol{\mu}}_1 \cdots \tilde{\boldsymbol{\mu}}_k]$ and $\boldsymbol{\Lambda}_{j,j} = \tilde{w}_j (\mathbf{v}^\top \tilde{\boldsymbol{\mu}}_j)$.
  - $\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ is the eigendecomposition of $\mathcal{T} \bullet_1 \mathbf{v}$ (since $\mathbf{U}^\top \mathbf{U} = \mathbf{I}$).

- This may be sensitive to noise. Performing simultaneous diagonalization of several random projections is a more robust approach [Kuleshov et al., AISTATS 2015].

Tensor Power Method

- Extension to orthogonal tensors of the power method (which computes the dominant eigenvector of a matrix):

Theorem (Anandkumar et al., JMLR, 2014)

Let $\mathcal{T} \in \bigotimes^3 \mathbb{R}^d$ *have an orthonormal decomposition*

$$\mathcal{T} = \sum_{i=1}^{k} \tilde{w}_i \widetilde{\boldsymbol{\mu}}_i \otimes \widetilde{\boldsymbol{\mu}}_i \otimes \widetilde{\boldsymbol{\mu}}_i .$$

Let $\boldsymbol{\theta}_0 \in \mathbb{R}^d$, *suppose that* $|\tilde{w}_1 . \widetilde{\boldsymbol{\mu}}_1^\top \boldsymbol{\theta}_0| > |\tilde{w}_j . \widetilde{\boldsymbol{\mu}}_j^\top \boldsymbol{\theta}_0| > 0$ *for all* $j > 1$. *For* $t = 1, 2, \cdots$, *define*

$$\boldsymbol{\theta}_t = \frac{\mathcal{T} \bullet_1 \boldsymbol{\theta}_{t-1} \bullet_2 \boldsymbol{\theta}_{t-1}}{\|\mathcal{T} \bullet_1 \boldsymbol{\theta}_{t-1} \bullet_2 \boldsymbol{\theta}_{t-1}\|} \quad \textit{and} \quad \lambda_t = \mathcal{T} \bullet_1 \boldsymbol{\theta}_t \bullet_2 \boldsymbol{\theta}_t \bullet_3 \boldsymbol{\theta}_t$$

Then, $\boldsymbol{\theta}_t \to \widetilde{\boldsymbol{\mu}}_1$ *and* $\lambda_t \to \tilde{w}_1$.

Conclusion

- For a wide class of latent variable models, the method of moments can be implemented by exploiting the tensor structure in the low order moments.
- This approach relies on extracting an orthogonal decomposition of a symmetric 3rd order tensor.
- Although tensor decomposition are usually intractable, orthogonal decompositions can be computed efficiently (*when the number of components is less than the dimension*).
- The estimators obtained for the parameters of the LVM are consistent (in contrast with the EM estimator).
- Both sample complexity and computational complexity are polynomial.