TENSOR METHODS FOR ML

→ COMPRESS NEURAL NETWORKS

→ LEARNING LATENT VARIABLE MODELS

(I) EFFICIENT COMPUTATIONS WITH TENSOR NETWORKS

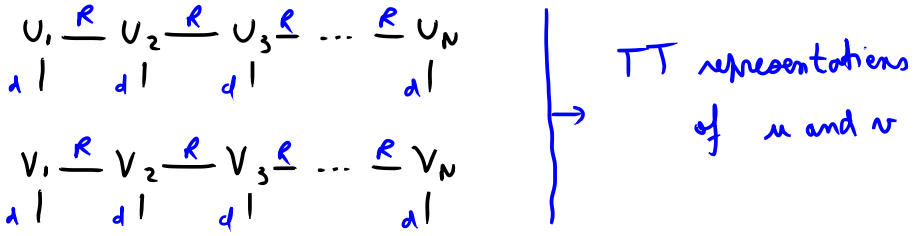
- **Inner product**: $u, v \in \mathbb{R}^{d^N}$

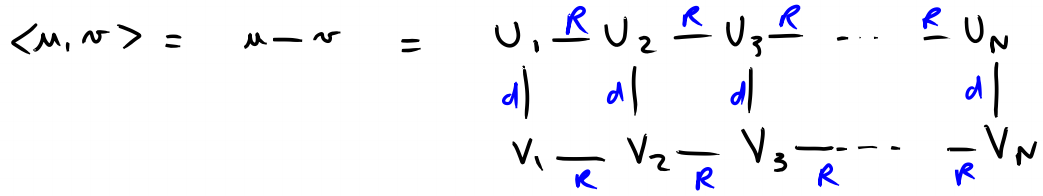
  What is the computational cost for $\langle u, v \rangle = \sum_{i=1}^{d^N} u_i v_i$?

  ↳ $\mathcal{O}(d^N)$

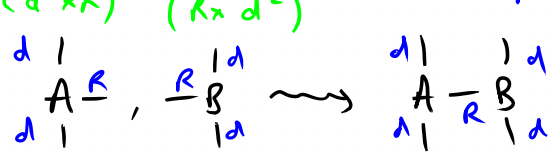
Suppose u and v are in the TT format:

$$u, v \in \mathbb{R}^{d^N} \cong \mathbb{R}^{\underbrace{d \times d \times \cdots \times d}_{N \text{ times}}}$$



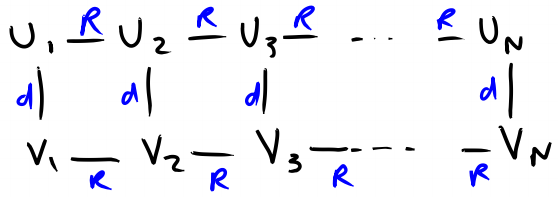


Recall: Computing the matrix product AB has cost $\mathcal{O}(mnp)$ ($m \times n$, $n \times p$)

For ex, Computing  can be done

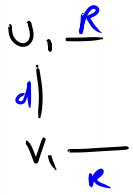
with a matrix product between matrices of shapes $d^2 \times R$ and $R \times d^2$. Hence the complexity is $\mathcal{O}(Rd^4)$.

We want to compute



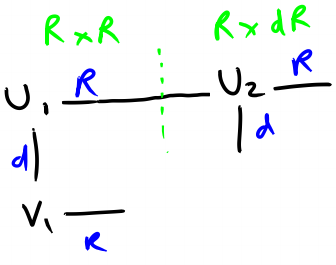
COMPLEXITY

① Compute



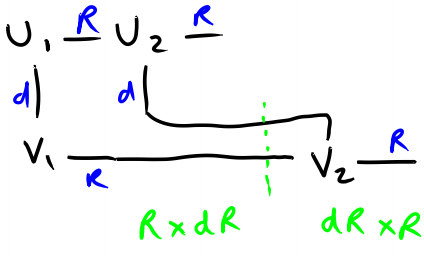
$\rightsquigarrow R \times d \quad d \times R$ $\quad O(dR^2)$

②



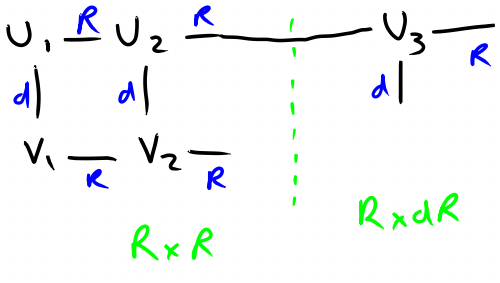
$O(dR^3)$

③



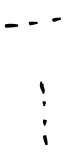
$O(dR^3)$

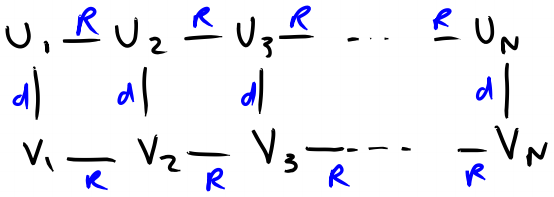
④



$O(dR^3)$

⑤





this is linear in N

TOTAL COMPLEXITY $O(NdR^3)$

↳ in contrast with $O(d^N)$!

exp. in N

- **Matrix vector product**

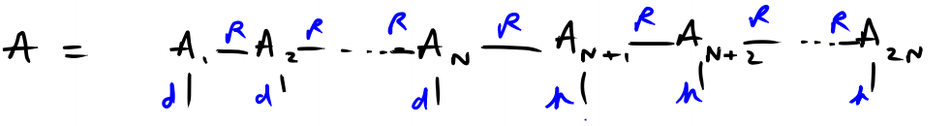
$A \in \mathbb{R}^{d^N \times n^N}$, $x \in \mathbb{R}^{n^N}$: we want to compute $b = Ax \in \mathbb{R}^{d^N}$.

In fact, we want to compute b in the TT format.

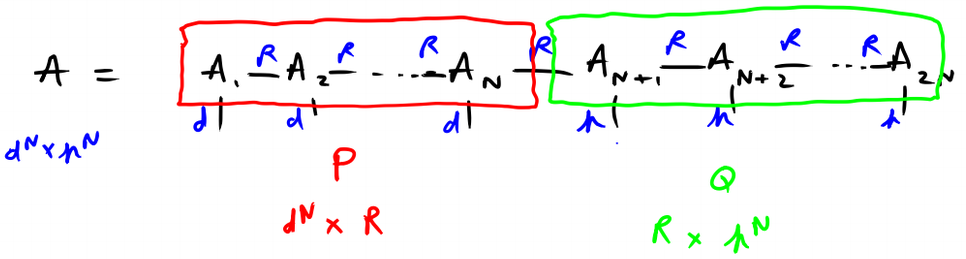
- TT representation of matrices :

$$A \in \mathbb{R}^{d^N \times n^N} \cong \mathbb{R}^{\underbrace{d \times d \times \dots \times d}_{N \text{ times}} \times \underbrace{n \times n \times \dots \times n}_{N \text{ times}}}$$

① First solution :

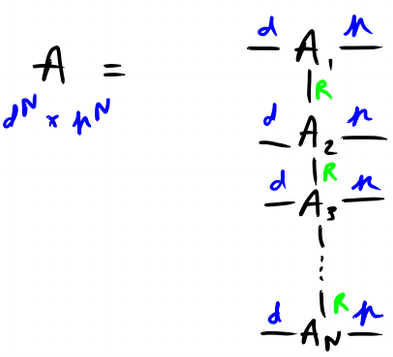


↳ PBM : A will be of rank at most R



↳ $A = PQ$ → Rank R factorization of A, hence $\text{rank}(A) \leq R$.

② TT matrix representation :



↳ Remark • If $A = \underset{d^2 \times n^2}{\;}$ $\begin{matrix} A_1 \\ A_2 \end{matrix}$ and $R = 1$,

then $A = \underset{d^2 \times n^2}{\;}$ $\begin{matrix} A_1 \\ A_2 \end{matrix}$ $= A_1 \otimes A_2$ (⊗ : KRONECKER PRODUCT)

$d \times n \quad d \times n$

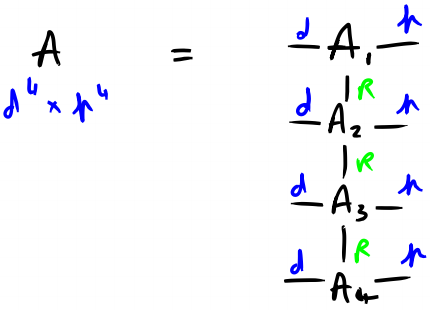
• In a TN, an edge of dimension 1 is equivalent to having no edge :

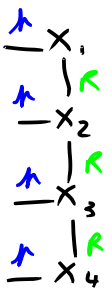
$$\left(\underset{m}{B} \; \underset{1}{C} \right)_{ij} = \sum_{k=1}^{1} B_{ik} C_{kj} = B_{i1} C_{1j} = b_i c_j$$

B : $m \times 1$ vector b, C : $1 \times m$ vector c^T

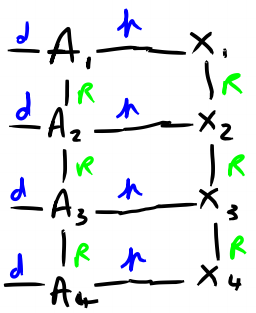
$= b \, c^T$

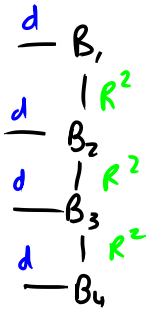
Back to matrix vector product:

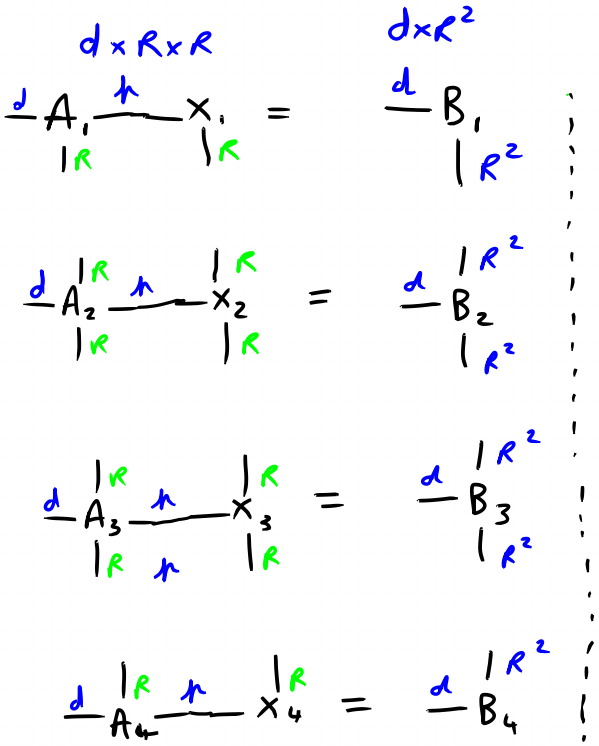
$A \in \mathbb{R}^{d^4 \times n^4}$ = 

$x \in \mathbb{R}^{n^4}$ = 

We want a TT representation of $b = Ax$

$b \in \mathbb{R}^{d^4}$ = 

b = 



$\Rightarrow$ If $A \in \mathbb{R}^{d^N \times n^N}$ given as a TT matrix with rank R

and $x \in \mathbb{R}^{n^N}$ given as a TT vector with rank R

then a rank R^2 TT representation of $b = Ax$ can be computed in time $\mathcal{O}(NR^4 d n)$.

↳ in contrast with $\mathcal{O}(d^N n^N)$

- $u, v \in \mathbb{R}^{d^N}$, given as rank R TT-vector, a rank $2R$ TT representation of $u + v$ can be computed in $\mathcal{O}(NR^3 d)$.

- $u, v \in \mathbb{R}^{d^N}$, given as rank R TT-vector, a rank R^2 TT representation of $u \circledast v$ can be computed in $\mathcal{O}(NR^3 d)$.
  ↳ component-wise product.

- TT ROUNDING:

  Given a rank R representation of $u \in \mathbb{R}^{d^N}$, the TT rounding operation returns a rank $\hat{R}$ TT representation of u which is "close" to the original representation (in time $\mathcal{O}(NR^3 d)$).

FOR MORE DETAILS:

SIAM J. SCI. COMPUT.
Vol. 33, No. 5, pp. 2295–2317

TENSOR-TRAIN DECOMPOSITION*

I. V. OSELEDETS†

Abstract. A simple nonrecursive form of the tensor decomposition in d dimensions is presented. It does not inherently suffer from the curse of dimensionality, it has asymptotically the same number of parameters as the canonical decomposition, but it is stable and its computation is based on low-rank approximation of auxiliary *unfolding matrices*. The new form gives a clear and convenient way to implement all basic operations efficiently. A fast rounding procedure is presented, as well as basic linear algebra operations. Examples showing the benefits of the decomposition are given, and the efficiency is demonstrated by the computation of the smallest eigenvalue of a 19-dimensional operator.

(11) COMPRESSING NEURAL NETWORKS

Tensorizing Neural Networks

(NeurIPS, 2015)

Alexander Novikov[1,4] **Dmitry Podoprikhin**[1] **Anton Osokin**[2] **Dmitry Vetrov**[1,3]
[1]Skolkovo Institute of Science and Technology, Moscow, Russia
[2]INRIA, SIERRA project-team, Paris, France
[3]National Research University Higher School of Economics, Moscow, Russia
[4]Institute of Numerical Mathematics of the Russian Academy of Sciences, Moscow, Russia
novikov@bayesgroup.ru podoprikhin.dmitry@gmail.com
anton.osokin@inria.fr vetrovd@yandex.ru

Fully connected layer: $y = \sigma(Wx + b)$, $y \in \mathbb{R}^M$, $W \in \mathbb{R}^{M \times N}$, $b \in \mathbb{R}^M$, $x \in \mathbb{R}^N$, σ = activation function

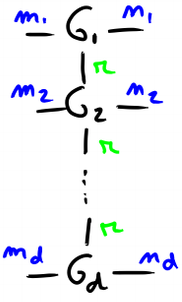
size: $M \times N$

For sake of simplicity, we ignore the bias term: $y = \sigma(Wx)$

IDEA: Represent W as a TT matrix.

$M = m_1 \times m_2 \times \cdots \times m_d$

$N = n_1 \times n_2 \times \cdots \times n_d$ (d is the order of the tensor)

W ($M \times N$) = 

TT layer

↳ parameters are $G_1, G_2, \ldots, G_d$

COMPLEXITY:

Operation	Time	Memory
FC forward pass	$O(MN)$	$O(MN)$
TT forward pass	$O(dr^2 m \max\{M, N\})$	$O(r \max\{M, N\})$
FC backward pass	$O(MN)$	$O(MN)$
TT backward pass	$O(d^2 r^4 m \max\{M, N\})$	$O(r^3 \max\{M, N\})$

Table 1: Comparison of the asymptotic complexity and memory usage of an $M \times N$ TT-layer and an $M \times N$ fully-connected layer (FC). The input and output tensor shapes are $m_1 \times \ldots \times m_d$ and $n_1 \times \ldots \times n_d$ respectively ($m = \max_{k=1\ldots d} m_k$) and r is the maximal TT-rank.

EXPERIMENT ON CIFAR 10:

INPUT (3072) → TT-layer ($3072 \to 2^{18}$, $\sim 250{,}000$) → TT-layer ($2^{18} \to 2^{12}$, 4096) → FC layer ($2^{12} \to 10$) → OUTPUT (10)

- COMPUTING GRADIENTS OF TENSOR NETWORKS

↳ See TN-gradients.pdf

Tensor Decomposition for Learning Latent Variable Models (LVM)

Journal of Machine Learning Research 15 (2014) 2773-2832 Submitted 2/13; Revised 3/14; Published 8/14

Tensor Decompositions for Learning Latent Variable Models

Animashree Anandkumar A.ANANDKUMAR@UCI.EDU
Electrical Engineering and Computer Science
University of California, Irvine
2200 Engineering Hall
Irvine, CA 92697

Rong Ge RONGGE@MICROSOFT.COM
Microsoft Research
One Memorial Drive
Cambridge, MA 02142

Daniel Hsu DJHSU@CS.COLUMBIA.EDU
Department of Computer Science
Columbia University
1214 Amsterdam Avenue, #0401
New York, NY 10027

Sham M. Kakade SKAKADE@MICROSOFT.COM
Microsoft Research
One Memorial Drive
Cambridge, MA 02142

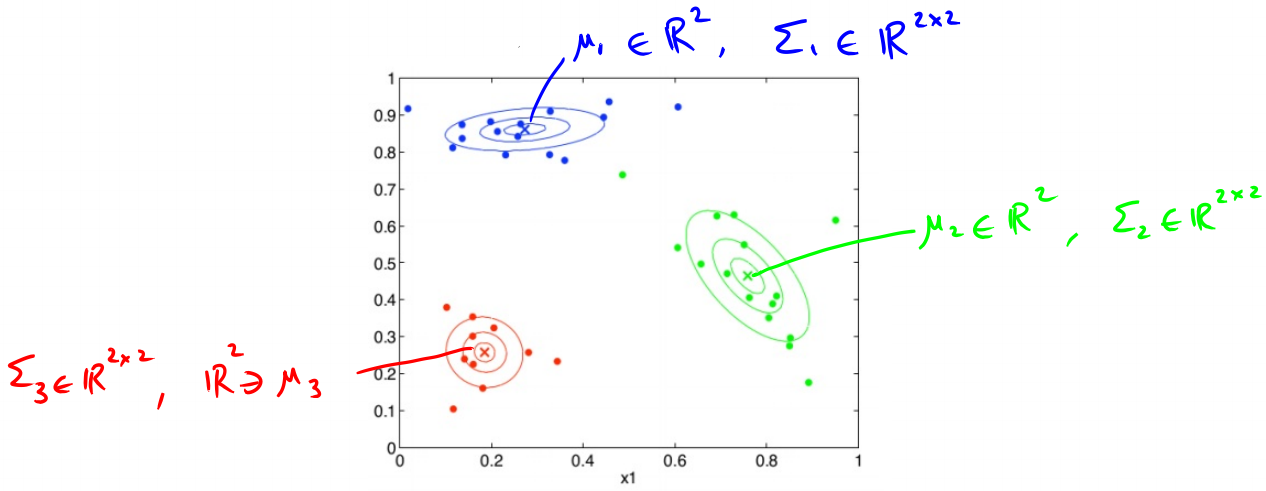
Matus Telgarsky MTELGARS@CS.UCSD.EDU
Department of Statistics
Rutgers University
110 Frelinghuysen Road
Piscataway, NJ 08854

Editor: Benjamin Recht

(JMLR, 2014)

1) Examples of LVM

- Gaussian mixture model (GMM)



GMM with k components in $\mathbb{R}^d$:

- parameters: $\mu_1, \dots, \mu_k \in \mathbb{R}^d$ (centers)
  $\Sigma_1, \dots, \Sigma_k \in \mathbb{R}^{d\times d}$ ← p.s.d. matrices (covariance matrices)
  $h_1, \dots, h_k \in \mathbb{R}_+$, $0 \le h_i \le 1$, $\sum_{i=1}^{k} h_i = 1$ (mixing probabilities)
- data generating distribution

Draw $x \sim$ GMM ←
(i) Draw a Gaussian h with probabilities $\mathbb{P}(h = i) = h_i$, $i = 1, \dots, k$
(ii) Draw $x \sim \mathcal{N}(\mu_i, \Sigma_i)$

- **Single topic model**

(Latent variable is the topic of a document)

Topic: discrete random variable taking values $1, 2, \dots, k$

Vocabulary of size d (d different words).

→ We want to model the distribution over the vocabulary given a topic.

- Parameters:

$\mathbb{P}(\text{topic} = i) = \pi_i$ $\quad \pi_1, \pi_2, \dots, \pi_k \in \mathbb{R}$ (topic probabilities) $\quad 0 \le \pi_i \le 1$, $\sum_{i=1}^{k} \pi_i = 1$

$\mathbb{P}(\text{word} = j \mid \text{topic} = i) = (\mu_i)_j$ $\quad \mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^d$ (voc. distrib. given a topic)

for $i = 1, \dots, k$: $\quad 0 \le (\mu_i)_j \le 1$, $\sum_{j=1}^{n} (\mu_i)_j = 1$

- Data generating distribution

Draw a document of length ℓ from the single topic model.

- Draw a topic h with probabilities $\mathbb{P}(h = i) = \pi_i$
- Draw independently ℓ words with probabilities $\mathbb{P}(\text{word } j \mid \text{topic} = h) = (\mu_i)_j$

2) Moments and Latent variable models

Def: • If x is a random variable taking its values in $\mathbb{R}$, its n^{th} order moment is $\mathbb{E}[x^n]$.

• If x is a random variable taking its values in $\mathbb{R}^d$, its n^{th} order moment is $\mathbb{E}[x^{\otimes n}] = \mathbb{E}[\underbrace{x \otimes x \otimes \cdots \otimes x}_{n \text{ times}}] \in \mathbb{R}^{\underbrace{d \times d \times \cdots \times d}_{n \text{ times}}}$

↳ . 3rd moment : $\mathbb{E}[x \otimes x \otimes x]_{i,j,k} = \mathbb{E}[x_i x_j x_k]$

. 1st moment : $\mathbb{E}[x] \rightarrow$ mean

. 2nd moment : $\mathbb{E}[x \otimes x] = \mathbb{E}[x x^T]$

↳ if $\mathbb{E}[x] = 0$, this is the covariance matrix.

↳ $\mathbb{V}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2$

$Cov[x] = \mathbb{E}[xx^T] - \mathbb{E}[x]\mathbb{E}[x]^T$

→ the method of moments give the same estimators as the maximum likelihood for a Gaussian distribution.

If $x \sim \mathcal{N}(\mu, \sigma^2)$ $\quad \mathbb{E}[x] = \mu$, $\mathbb{E}[x^2] = \sigma^2 + \mu^2$ $\quad \rightarrow \quad \mu = \mathbb{E}[x] \approx \hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} x_i$, $\sigma^2 = \mu^2 - \mathbb{E}[x^2] \approx \hat{\mu}^2 - \frac{1}{m}\sum_{i=1}^{m} x_i^2$

Single Topic Model

parameters: $\theta = \{\pi_1, \pi_2, \ldots, \pi_k, \mu_1, \ldots, \mu_k\}$ ($\pi_h \in \mathbb{R}$, $\mu_h \in \mathbb{R}^d$)

$$\mathbb{P}(\text{word } i,j \mid \text{topic } h) = \mathbb{P}(\text{word } i \mid \text{topic } h)\,\mathbb{P}(\text{word } j \mid \text{topic } h)$$
$$= (\mu_h)_i\,(\mu_h)_j$$
$$= (\mu_h \otimes \mu_h)_{i,j}$$

$$\mathbb{P}(\text{word } i,j) = \sum_{h=1}^{k} \mathbb{P}(\text{topic } h)\,\mathbb{P}(\text{word } i,j \mid \text{topic } h)$$
$$= \sum_{h=1}^{k} \pi_h\,(\mu_h \otimes \mu_h)_{i,j}$$

$$\mathbb{P}(\text{word } i_1, i_2, i_3) = \sum_{h=1}^{k} \pi_h\,(\mu_h \otimes \mu_h \otimes \mu_h)_{i_1,i_2,i_3}$$

Note that if x_1, x_2 and x_3 are the one-encodings of the first 3 words in a document, then

$$\mathbb{E}[x_1 \otimes x_2]_{i_1,i_2} = \mathbb{P}(\text{word } i_1, i_2)$$
$$\mathbb{E}[x_1 \otimes x_2 \otimes x_3]_{i_1,i_2,i_3} = \mathbb{P}(\text{word } i_1, i_2, i_3)$$

Theorem 3.1 (Anandkumar et al., 2012c) *If*

$$\begin{aligned} M_2 &:= \mathbb{E}[x_1 \otimes x_2] \\ M_3 &:= \mathbb{E}[x_1 \otimes x_2 \otimes x_3], \end{aligned}$$

then

$$\begin{aligned} M_2 &= \sum_{i=1}^{k} w_i \, \mu_i \otimes \mu_i \\ M_3 &= \sum_{i=1}^{k} w_i \, \mu_i \otimes \mu_i \otimes \mu_i. \end{aligned}$$

$(w_i = h_i)$

GMM

parameters : $\theta = \{ \mu_1, \dots, \mu_k, \Sigma_1, \dots, \Sigma_k, w_1, \dots, w_k \}$

$\mu_1, \dots, \mu_k$: centers $\in \mathbb{R}^d$

$\Sigma_1, \dots, \Sigma_k$: covariance matrices $\in \mathbb{R}^{d \times d}$

$w_1, \dots, w_k$: mixing probabilities

↳ Here we assume each $\Sigma_i = \sigma_i^2 I$ for each $i = 1, \dots, k$

↳ $\sigma_i \in \mathbb{R}$

Theorem 3.2 (Hsu and Kakade, 2013) *Assume $d \geq k$. The variance σ^2 is the smallest eigenvalue of the covariance matrix $\mathbb{E}[x \otimes x] - \mathbb{E}[x] \otimes \mathbb{E}[x]$. Furthermore, if*

$$\begin{aligned} M_2 &:= \mathbb{E}[x \otimes x] - \sigma^2 I \\ M_3 &:= \mathbb{E}[x \otimes x \otimes x] - \sigma^2 \sum_{i=1}^{d} (\mathbb{E}[x] \otimes e_i \otimes e_i + e_i \otimes \mathbb{E}[x] \otimes e_i + e_i \otimes e_i \otimes \mathbb{E}[x]), \end{aligned}$$

then

$$\begin{aligned} M_2 &= \sum_{i=1}^{k} w_i \, \mu_i \otimes \mu_i \\ M_3 &= \sum_{i=1}^{k} w_i \, \mu_i \otimes \mu_i \otimes \mu_i. \end{aligned}$$

$\sigma = \sigma_1 = \sigma_2 = \dots = \sigma_k$

(e_i : i-th vector of the canonical basis)

Summary : In both cases, there exists $M_2 \in \mathbb{R}^{d \times d}$ and a tensor $M_3 \in \mathbb{R}^{d \times d \times d}$ (which can be estimated from data) satisfying

$$\begin{aligned} M_2 &= \sum_{i=1}^{k} w_i \, \mu_i \otimes \mu_i \\ M_3 &= \sum_{i=1}^{k} w_i \, \mu_i \otimes \mu_i \otimes \mu_i. \end{aligned}$$

↳ CP decomposition of M_3 (under mild conditions, this decomposition unique)

Decomposition of M_2 and M_3:

$$M_2 = \sum_{i=1}^{k} w_i \, \mu_i \otimes \mu_i$$

$$M_3 = \sum_{i=1}^{k} w_i \, \mu_i \otimes \mu_i \otimes \mu_i.$$

Observation : • The problem of orthogonal tensor decomposition,

$$T = \sum_{i=1}^{k} w_i \, \mu_i \otimes \mu_i \otimes \mu_i \quad \text{where} \quad \langle \mu_i, \mu_j \rangle = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{o.w.} \end{cases}$$

is easy to solve.

• We can use M_2 to "orthogonalize" M_3.

$M_3 \longrightarrow \tilde{M}_3 \longrightarrow \tilde{w}_i, \tilde{\mu}_i \longrightarrow w_i, \mu_i$

$M_3 \longrightarrow \tilde{M}_3$: using M_2

$\tilde{M}_3 \longrightarrow \tilde{w}_i, \tilde{\mu}_i$: orthogonal tensor decomposition

$$\tilde{M}_3 = \sum_{i=1}^{k} \tilde{w}_i \, \tilde{\mu}_i \otimes \tilde{\mu}_i \otimes \tilde{\mu}_i$$