Recovering Joint PMF from Pairwise Marginals

Shahana Ibrahim, Xiao Fu

School of EECS
Oregon State University, Corvallis

Virtual Presentation at Asilomar 2020
November 1-5, 2020

Joint Probability Mass Function (PMF) Learning

- Many ML tasks boil down to learning joint PMF of RVs.

  - ☐ **recommender systems**
  - ☐ **data classification**
  - ☐ **survey/database completion**
  - ☐ **language modeling**

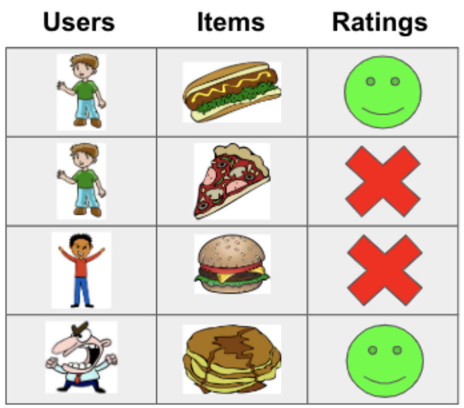
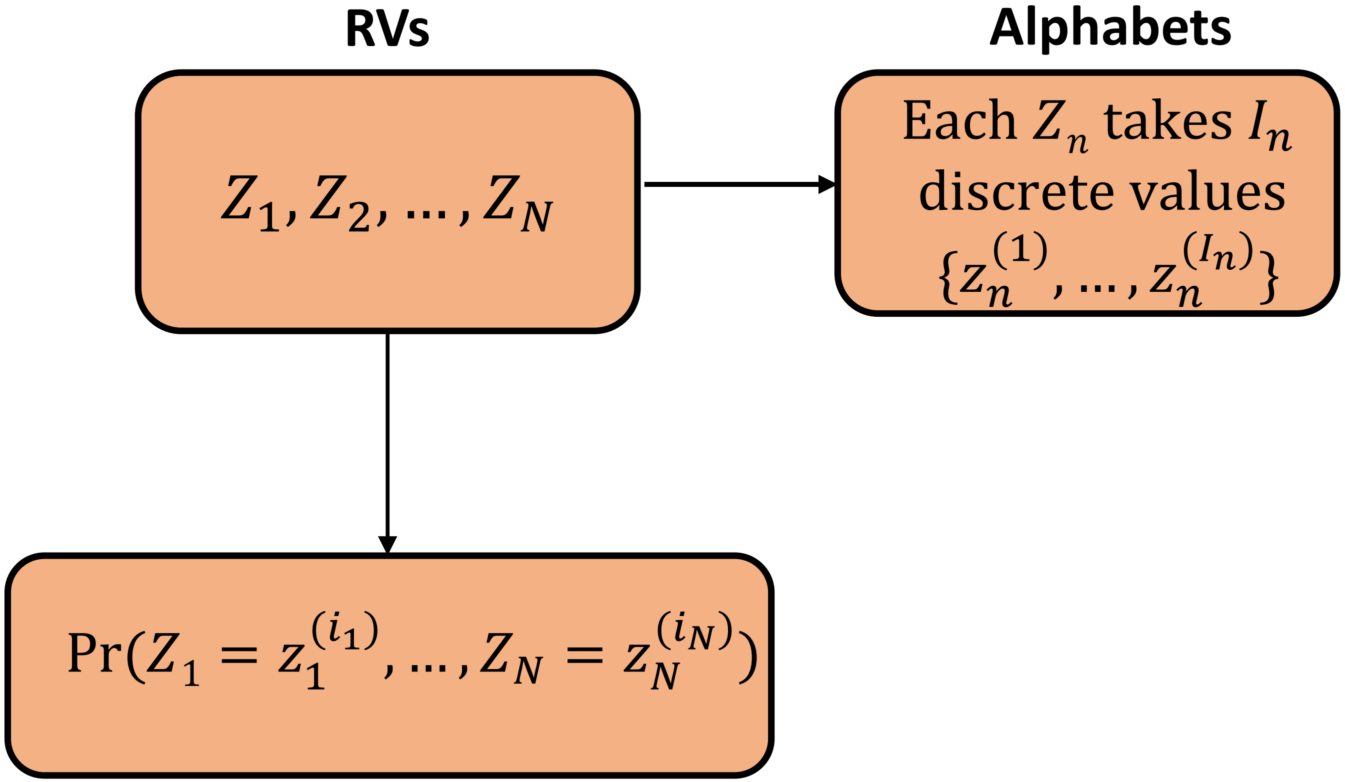


Image source : Google

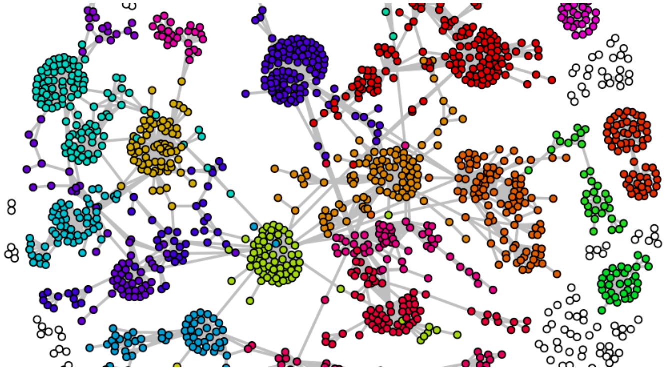
- Knowing joint PMF allows us to construct certain optimal predictors, e.g., MAP and MMSE.

Joint PMF of N RVs



- Short hand notation for $\Pr(Z_1 = z_1^{(i_1)}, \ldots, Z_N = z_N^{(i_N)})$ is $\Pr(i_1, \ldots, i_N)$.

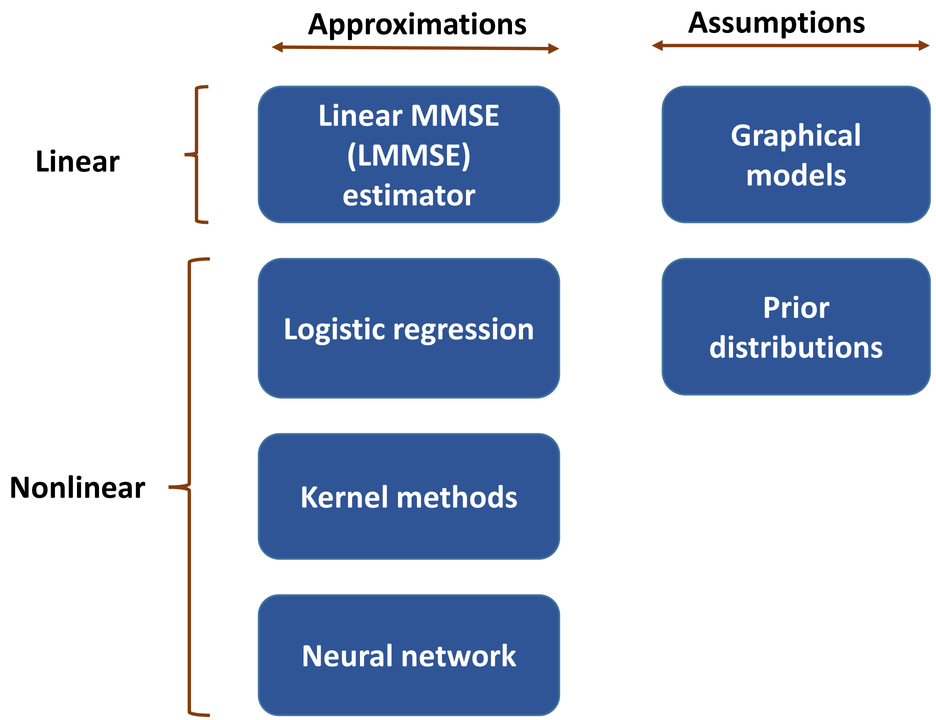
Curse of Dimensionality



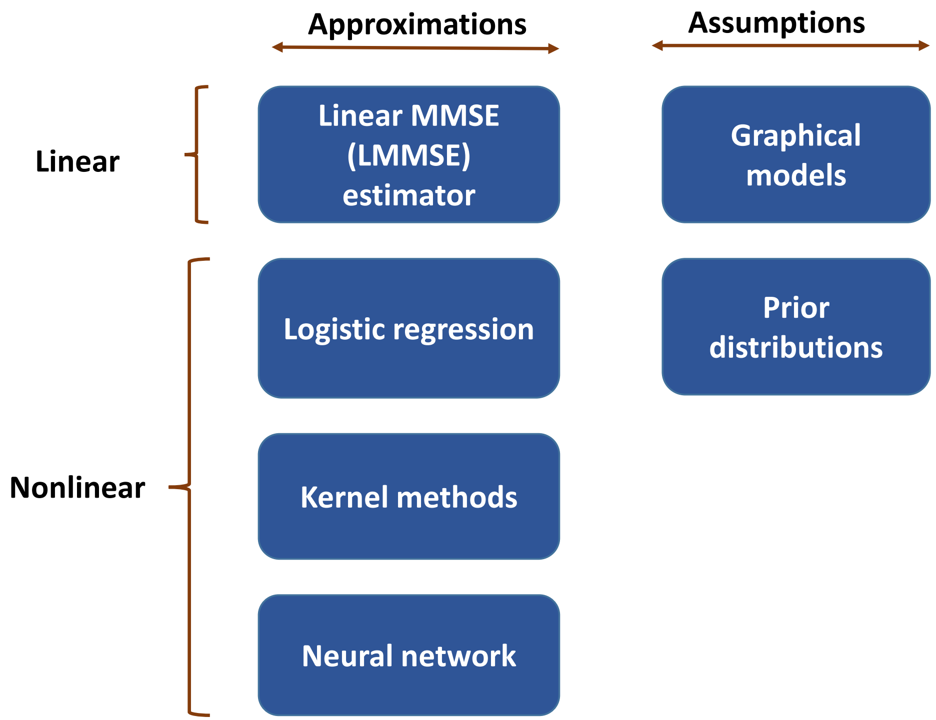
- Consider $N = 10$ RVs each taking $I_n = 10$ different values:
  - **joint PMF has 10^{10} entries** to learn!!!

- The 'naive' approach is to count the occurrences of the joint variable realizations:
  - **the number of examples $S \gg \Omega(10^{10})$ to achieve reasonable accuracy.**

Existing Alternatives for Joint PMF Learning

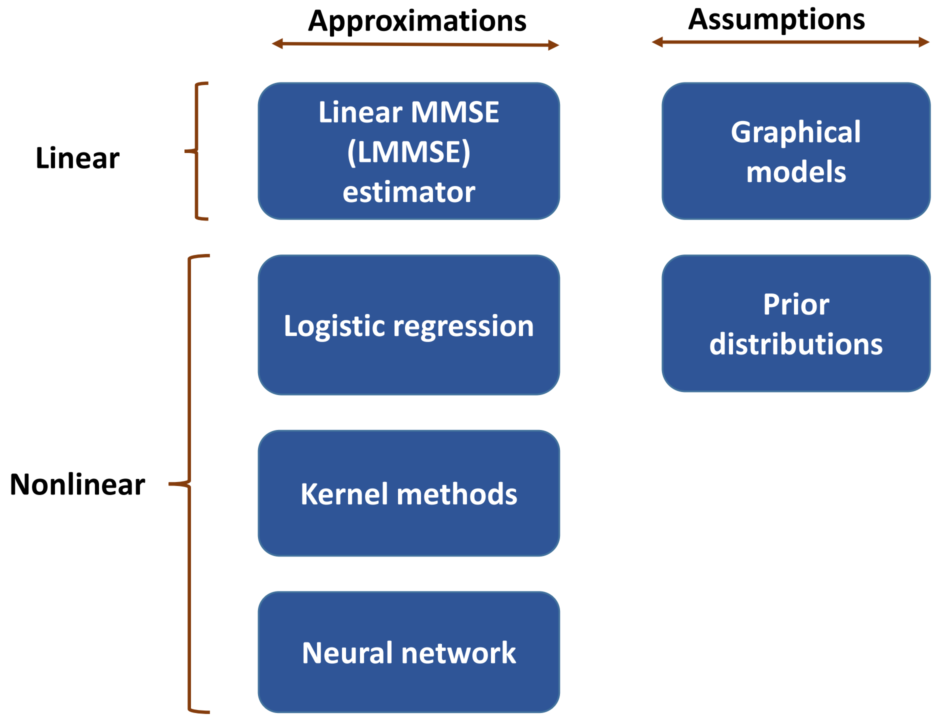


Existing Alternatives for Joint PMF Learning



These methods do not directly learn joint PMF.

Existing Alternatives for Joint PMF Learning



These methods do not directly learn joint PMF.

Can we reliably learn the joint PMF given limited data without any structural assumptions?

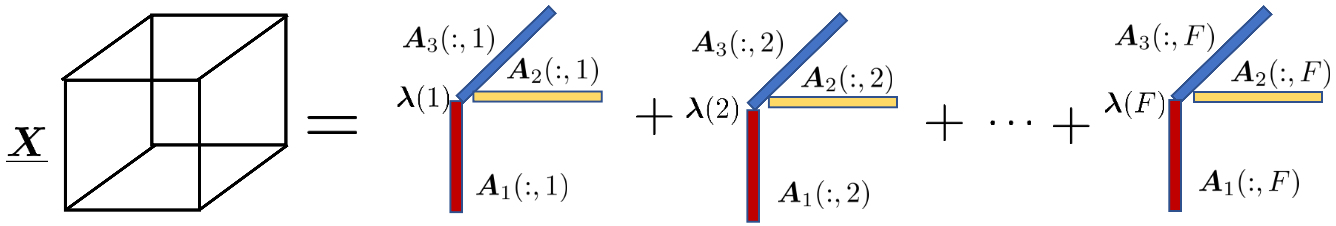
Joint PMF Learning via Tensor CPD [Kargas et al., 2018]

- Joint PMF $\Pr(i_1, \ldots, i_N)$ can be represented as an N-th order tensor:

$$\underline{\boldsymbol{X}}(i_1, \ldots, i_N) = \Pr(i_1, \ldots, i_N), \quad \underline{\boldsymbol{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_N}.$$

- An N-th order tensor $\underline{\boldsymbol{X}}$ admits Canonical Polyadic Decomposition (CPD) with rank F:

$$\underline{\boldsymbol{X}}(i_1, \ldots, i_N) = \sum_{f=1}^{F} \boldsymbol{\lambda}(f) \prod_{n=1}^{N} \boldsymbol{A}_n(i_n, f), \quad \boldsymbol{A}_n \in \mathbb{R}^{I_n \times F}, \ \boldsymbol{\lambda} \in \mathbb{R}^F.$$

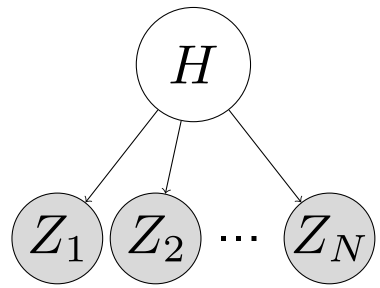


Canonical Polyadic Decomposition (CPD)

Joint PMF Learning via Tensor CPD [Kargas et al., 2018]

- **Any joint PMF admits a naive Bayes model representation** with respect to a single hidden variable H;

$$\underline{\boldsymbol{X}}(i_1,\ldots,i_N) = \mathsf{Pr}(Z_1 = i_1,\ldots,Z_N = i_N),$$
$$= \sum_{f=1}^{F} \mathsf{Pr}(H=f) \prod_{n=1}^{N} \mathsf{Pr}(Z_n = i_n | H = f).$$



$$\underline{\boldsymbol{X}}(i_1,\ldots,i_N) = \sum_{f=1}^{F} \boldsymbol{\lambda}(f) \prod_{n=1}^{N} \boldsymbol{A}_n(i_n, f). \longleftarrow \text{CPD}$$

Decomposition of joint PMF tensor can identify the latent factors $\boldsymbol{A}_n$'s and $\boldsymbol{\lambda}$:

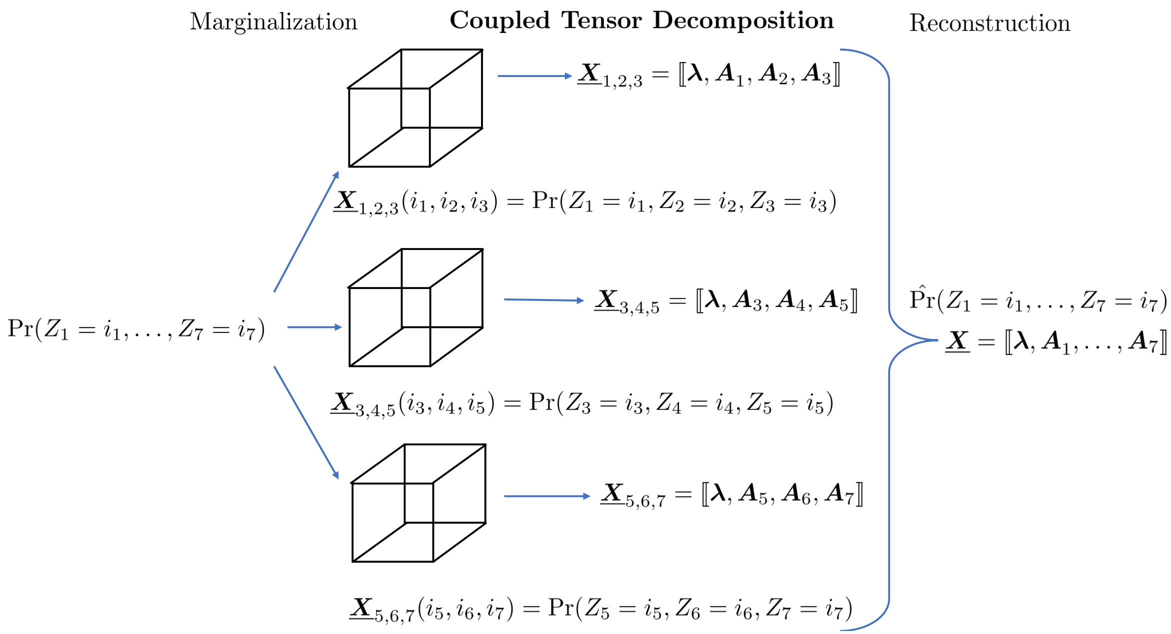
$$\boldsymbol{A}_n(i_n, f) := \mathsf{Pr}(Z_n = i_n | H = f), \quad \boldsymbol{\lambda}(f) := \mathsf{Pr}(H = f).$$

- **However, $\underline{\boldsymbol{X}}$ is not available. How do we identify $\boldsymbol{A}_n$'s and $\boldsymbol{\lambda}$ then?**

Joint PMF learning via Three-dimensional marginals

- If F is small, joint PMF can be provably recovered through **a coupled tensor decomposition using only three-dimensional marginals**, i.e, $\Pr(i_j, i_k, i_\ell)$ for different j, k, ℓ [Kargas et al., 2018].

- Many joint PMFs in real-world data are relatively low-rank tensors—since RV's are often "reasonably dependent" [Kargas et al., 2018].



Challenges in Existing Approaches

- The result in [Kargas et al., 2018] is inspiring, however, some challenges exist:
  - **High sample complexity:** Estimating $\Pr(i_j, i_k, i_\ell)$'s is not easy, since one needs many co-occurrences of three RVs.
  - **High computational complexity:** CPD is an NP-hard problem [Hillar and Lim, 2013]—and the optimization involves many tensors.

Challenges in Existing Approaches

- The result in [Kargas et al., 2018] is inspiring, however, some challenges exist:
  - **High sample complexity:** Estimating $\Pr(i_j, i_k, i_\ell)$'s is not easy, since one needs many co-occurrences of three RVs.
  - **High computational complexity:** CPD is an NP-hard problem [Hillar and Lim, 2013]—and the optimization involves many tensors.

- The work in [Yeredor and Haardt, 2019] takes an ML perspective:
  - Directly estimates $\boldsymbol{A}_n$'s and $\boldsymbol{\lambda}$ using an EM algorithm—scalable approach.
  - **Unclear theoretical guarantees**: EM algorithm's convergence guarantee and estimation accuracy are unclear.

Challenges in Existing Approaches

- The result in [Kargas et al., 2018] is inspiring, however, some challenges exist:
  - **High sample complexity:** Estimating $\Pr(i_j, i_k, i_\ell)$'s is not easy, since one needs many co-occurrences of three RVs.
  - **High computational complexity:** CPD is an NP-hard problem [Hillar and Lim, 2013]—and the optimization involves many tensors.

- The work in [Yeredor and Haardt, 2019] takes an ML perspective:
  - Directly estimates $\boldsymbol{A}_n$'s and $\boldsymbol{\lambda}$ using an EM algorithm—scalable approach.
  - **Unclear theoretical guarantees**: EM algorithm's convergence guarantee and estimation accuracy are unclear.

How do we address these issues?

Proposed Approach

- We propose a **pairwise marginal-based** approach.
  - With the same amount of data, the second-order statistics can be estimated with much higher accuracy, compared to the third-order ones [Han et al., 2015].
- Pairwise marginal of Z_j and Z_k: $\mathsf{Pr}(i_j, i_k) = \sum_{f=1}^{F} \mathsf{Pr}(f)\mathsf{Pr}(i_j|f)\mathsf{Pr}(i_k|f)$

Proposed Approach

- We propose a **pairwise marginal-based** approach.
  - With the same amount of data, the second-order statistics can be estimated with much higher accuracy, compared to the third-order ones [Han et al., 2015].

- Pairwise marginal of Z_j and Z_k: $\underbrace{\Pr(i_j, i_k)}_{:=\boldsymbol{X}_{jk}(i_j,i_k)} = \sum_{f=1}^{F} \underbrace{\Pr(f)}_{:=\boldsymbol{\lambda}(f)} \underbrace{\Pr(i_j|f)}_{:=\boldsymbol{A}_j(i_j|f)} \Pr(i_k|f)$.

$$\boxed{\boldsymbol{X}_{jk} = \boldsymbol{A}_j \boldsymbol{D}(\boldsymbol{\lambda}) \boldsymbol{A}_k^\top,} \text{ where } \boldsymbol{D}(\boldsymbol{\lambda}) = \mathrm{Diag}(\boldsymbol{\lambda}).$$

Proposed Approach

- We propose a **pairwise marginal-based** approach.
  - With the same amount of data, the second-order statistics can be estimated with much higher accuracy, compared to the third-order ones [Han et al., 2015].

- Pairwise marginal of Z_j and Z_k: $\underbrace{\mathsf{Pr}(i_j, i_k)}_{:=\boldsymbol{X}_{jk}(i_j,i_k)} = \sum_{f=1}^{F} \underbrace{\mathsf{Pr}(f)}_{:=\boldsymbol{\lambda}(f)} \underbrace{\mathsf{Pr}(i_j|f)}_{:=\boldsymbol{A}_j(i_j|f)} \mathsf{Pr}(i_k|f)$.

$$\boxed{\boldsymbol{X}_{jk} = \boldsymbol{A}_j \boldsymbol{D}(\boldsymbol{\lambda}) \boldsymbol{A}_k^\top,} \text{ where } \boldsymbol{D}(\boldsymbol{\lambda}) = \mathrm{Diag}(\boldsymbol{\lambda}).$$

The challenge is to identify $\boldsymbol{A}_j$'s and $\boldsymbol{\lambda}$ from pairwise marginals $\boldsymbol{X}_{jk}$'s.

Identifiability of Matrix Factorization

- Key idea in [Kargas et al., 2018]: **Tensors admit unique CPD, under mild conditions**.

- Pairwise distributions $\boldsymbol{X}_{jk} = \boldsymbol{A}_j \boldsymbol{D}(\boldsymbol{\lambda}) \boldsymbol{A}_k^\top$ are matrices, and low-rank matrix decomposition is in general *nonunique*.

$$\boldsymbol{X}_{jk} = \boldsymbol{A}_j \boldsymbol{D}(\boldsymbol{\lambda}) \boldsymbol{Q} (\boldsymbol{A}_k \boldsymbol{Q}^{-\top})^\top, \text{for any nonsingular } \boldsymbol{Q} \in \mathbb{R}^{F \times F}.$$

- Most natural way: apply **NMF (nonnegative matrix factorization)**:

$$\boldsymbol{X}_{jk} = \underbrace{\boldsymbol{A}_j}_{\boldsymbol{W} \in \mathbb{R}^{I_j \times F}} \underbrace{\boldsymbol{D}(\boldsymbol{\lambda}) \boldsymbol{A}_k^\top}_{\boldsymbol{H}^\top \in \mathbb{R}^{F \times I_k}}$$

- In many cases, $F \gg \min\{I_j, I_k\} \implies$ NMF tools cannot be directly applied.

Proposed Virtual NMF-based Approach

- Consider a splitting of the indices of the N variables, i.e.,

$$\mathcal{S}_1 = \{\ell_1, \dots, \ell_M\}, \quad \mathcal{S}_2 = \{\ell_{M+1}, \dots, \ell_N\},$$
$$\mathcal{S}_1 \cup \mathcal{S}_2 = \{1, \dots, N\}, \quad \mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset.$$

- We construct the following matrix:

$$\widetilde{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{X}_{\ell_1 \ell_{M+1}} & \dots & \boldsymbol{X}_{\ell_1 \ell_N} \\ \vdots & \ddots & \vdots \\ \boldsymbol{X}_{\ell_M \ell_{M+1}} & \dots & \boldsymbol{X}_{\ell_M \ell_N} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{A}_{\ell_1} \\ \vdots \\ \boldsymbol{A}_{\ell_M} \end{bmatrix}}_{\boldsymbol{W}} \underbrace{\boldsymbol{D}(\boldsymbol{\lambda})[\boldsymbol{A}_{\ell_{M+1}}^\top, \dots, \boldsymbol{A}_{\ell_N}^\top]}_{\boldsymbol{H}^\top}.$$

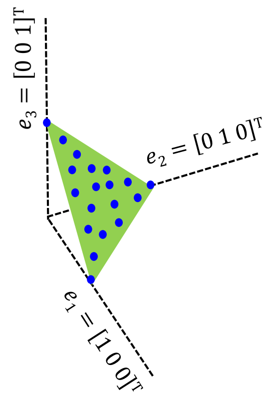
Idea: construct $\widetilde{\boldsymbol{X}}$ such that $F \leq \min\{MI, (N-M)I\}$ so that $\boldsymbol{W}$ and $\boldsymbol{H}$ are more likely to satisfy certain NMF identifiability conditions.

Separability - A Celebrated NMF Tool

Separability [Donoho and Stodden, 2003] and ε-separability: If $\boldsymbol{H} \geq \boldsymbol{0}$, and $\boldsymbol{\Lambda} = \{l_1, \ldots, l_F\}$ such that $\boldsymbol{H}(\boldsymbol{\Lambda}, :) = \boldsymbol{\Sigma} = \mathrm{Diag}(\alpha_1, \ldots, \alpha_F)$ and $\alpha_f > 0$, then, $\boldsymbol{H}$ satisfies *separability*. When $\boldsymbol{\Lambda} = \{l_1, \ldots, l_F\}$ satisfies $\|\boldsymbol{H}(l_f, :) - \boldsymbol{e}_f\|_2 \leq \varepsilon$ for $f = 1, \ldots, F$, $\boldsymbol{H}$ is called ε-separable.

NMF Model : $\widetilde{\boldsymbol{X}} = \boldsymbol{W}\boldsymbol{H}^\top$

Under separability on $\boldsymbol{H}$, estimation of $\boldsymbol{W}$ is an index identification task: $\boldsymbol{W}\boldsymbol{\Sigma} = \widetilde{\boldsymbol{X}}(\boldsymbol{\Lambda}, :)$.



- **Successive projection algorithm (SPA)** from the NMF literature [Gillis and Vavasis, 2014] can be employed.
  - ☐ very scalable - a Gram-Schmitt-like algorithm
  - ☐ robust to noise and slight violation of separability

Scalable Algorithm - CNMF-SPA

- $\boldsymbol{A}_{\ell_n} \in \mathbb{R}^{I_{\ell_n} \times F}, n \in \{1, \ldots, M\}$ can be identified upto column permutations ($\widehat{\boldsymbol{A}}_{\ell_n} = \boldsymbol{A}_{\ell_n}\boldsymbol{\Pi}$) since

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{A}_{\ell_1} \\ \vdots \\ \boldsymbol{A}_{\ell_M} \end{bmatrix}, \boldsymbol{1}^\top \boldsymbol{A}_k = \boldsymbol{1}^\top, \ \boldsymbol{A}_k \geq \boldsymbol{0}.$$

- $\boldsymbol{A}_{\ell_n}$ for $n \in \{M+1, \ldots, N\}$ can be identified upto column permutations, since $\boldsymbol{H}$ matrix can be estimated using (constrained) least squares, $\arg\min_{\boldsymbol{H} \geq 0} \|\widetilde{\boldsymbol{X}} - \boldsymbol{W}\boldsymbol{H}^\top\|_F^2$.
- $\boldsymbol{\lambda}$ can be identified as $\widehat{\boldsymbol{\lambda}} = (\widetilde{\boldsymbol{H}} \odot \boldsymbol{W})^\dagger \text{vec}(\widetilde{\boldsymbol{X}}) = \boldsymbol{\Pi\lambda}$, since

$$\widetilde{\boldsymbol{X}} = \underbrace{\begin{bmatrix} \boldsymbol{A}_{\ell_1} \\ \vdots \\ \boldsymbol{A}_{\ell_M} \end{bmatrix}}_{\boldsymbol{W}} \boldsymbol{D}(\boldsymbol{\lambda}) \underbrace{[\boldsymbol{A}_{\ell_{M+1}}^\top, \ldots, \boldsymbol{A}_{\ell_N}^\top]}_{\widetilde{\boldsymbol{H}}^\top}.$$

The method is very scalable - a good choice as an initialization algorithm.

Performance Analysis of CNMF-SPA

- What are the key elements in characterizing the performance?
  - ☐ S - The number of available joint realizations of N RVs
  - ☐ p - Probability of observing each variable.
  - ☐ ε - Deviation from separability condition.

- Splitting: $\mathcal{S}_1 = \{1, \ldots, M\}$ and $\mathcal{S}_2 = \{M+1, \ldots, N\}$.
  - Testing all combinations for separability is not feasible.

- **Assumption 1:** The rows of $\boldsymbol{A}_m$'s are generated from the $(F-1)$-probability simplex uniformly at random.

$$\widetilde{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{X}_{\ell_1 \ell_{M+1}} & \cdots & \boldsymbol{X}_{\ell_1 \ell_N} \\ \vdots & \vdots & \vdots \\ \boldsymbol{X}_{\ell_M \ell_{M+1}} & \cdots & \boldsymbol{X}_{\ell_M \ell_N} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{A}_{\ell_1} \\ \vdots \\ \boldsymbol{A}_{\ell_M} \end{bmatrix}}_{\boldsymbol{W}} \underbrace{\boldsymbol{D}(\boldsymbol{\lambda})[\boldsymbol{A}_{\ell_{M+1}}^\top, \ldots, \boldsymbol{A}_{\ell_N}^\top]}_{\boldsymbol{H}^\top}.$$

$$\boxed{\textbf{Intuition : More rows in } \boldsymbol{H} \rightarrow \textbf{better chance to satisfy separability}}$$

Theorem 1: CNMF-SPA Performance Characterization (Informal)

Assume that $M \geq F/I$, $\longrightarrow$ Low rank condition for $\widetilde{\boldsymbol{X}}$

$$p = \Omega\left(\frac{1}{\sqrt{S}}\log(1/\delta)\right), \longrightarrow$$ Prob. of observing each RV needs to be above certain threshold

$$S = \Omega\left(\frac{FI\log(1/\delta)}{p^2}\right), \longrightarrow$$ More no. of joint realizations are needed for larger F and I

$$N = M + \Omega\left(\frac{\varepsilon^{-2F}}{FI}\log\left(\frac{F}{\delta}\right)\right), \longrightarrow$$ Larger N implies more rows in $\boldsymbol{H}$

for sufficiently small $0 \leq \varepsilon \leq 1$. Under Assumption 1, `CNMF-SPA` outputs $\widehat{\boldsymbol{A}}_m, m \in \mathcal{S}_1$ with probability at least $1-\delta$ such that

$$\min_{\boldsymbol{\Pi}:\ \text{permuation}} \|\widehat{\boldsymbol{A}}_m\boldsymbol{\Pi} - \boldsymbol{A}_m\|_2 = O\left(\max\ \underbrace{(\sigma_{\max}(\boldsymbol{W})\sqrt{F}\varepsilon}_{\text{deviation from separability}}, \underbrace{\frac{M\sqrt{IF\log(1/\delta)}}{p\sqrt{S}})}_{\text{error due to finite samples}}\right).$$

CNMF-SPA - In a Nutshell

☑ **Scalable algorithm**

☑ **Lower sample complexity**

☑ **Provable joint PMF recovery**

CNMF-SPA - In a Nutshell

☑ **Scalable algorithm**

☑ **Lower sample complexity**

☑ **Provable joint PMF recovery**

Can we further enhance the performance of CNMF-SPA?

EM Algorithm Meets CNMF-SPA

- Recall the joint PMF model $\mathsf{Pr}(i_1, i_2, \ldots, i_N) = \sum_{f=1}^{F} \mathsf{Pr}(f) \prod_{n=1}^{N} \mathsf{Pr}(i_n|f)$.

- Yeredor and Haardt [2019] proposed an EM algorithm for maximizing the log-likelihood of the joint PMF by iterating over:

  E-step: $\widehat{q} \leftarrow$ estimated using observed realizations and current estimates $\widehat{\boldsymbol{A}}_n$ and $\widehat{\boldsymbol{\lambda}}$.

  M-step: $\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{\lambda}} \leftarrow$ estimated using observed realizations and current value of $\widehat{q}$.

- EM algorithm exhibits promising performance and scalability.
  - ☐ How to understand its performance?
  - ☐ Yeredor and Haardt [2019] noticed EM converges to undesired solutions if randomly initialized. How to properly initialize?

Performance Analysis of EM

- We define two key parameters $\overline{D}_1$ and $\overline{D}_2$:

  ☐ $\overline{D}_1$ – measuring the average KL divergence between the columns of $\boldsymbol{A}_n$.
  ☐ $\overline{D}_2$ – measuring the deviation of $\boldsymbol{\lambda}$ from the uniform distribution.

- **Assumption 2**: Assume that $\boldsymbol{A}_n, \boldsymbol{\lambda}$ and the initial estimates $\widehat{\boldsymbol{A}}_n^0, \widehat{\boldsymbol{\lambda}}^0$ satisfy

$$\boldsymbol{A}_n(i,f) \geq \rho_1, \quad \boldsymbol{\lambda}(f) \geq \rho_2,$$

$$\underbrace{|\widehat{\boldsymbol{A}}_n^0(i,f) - \boldsymbol{A}_n(i,f)| \leq \delta_1 := \frac{4}{\rho_1(4+\overline{D})}}_{\text{Initial estimation errors of } \boldsymbol{A}_n\text{'s are bounded}}, \quad \underbrace{|\widehat{\boldsymbol{\lambda}^0}(f) - \boldsymbol{\lambda}(f)| \leq \delta_2 := \frac{4}{\rho_2(4+N\overline{D})}}_{\text{Initial estimation error of } \boldsymbol{\lambda} \text{ is bounded}}$$

Theorem 2: EM Convergence (Informal)

Let $\delta_{\min} = \min(\delta_1, \delta_2)$, $\overline{D} = {(\overline{D}_1+\overline{D}_2)}/{2}$. Assume that the following hold:

$$N = \Omega\left(\frac{\log(SF^2/(p\rho_2\mu))}{\rho_1\overline{D}}\right), \longrightarrow \boxed{\text{No. of RVs is above certain threshold}}$$

$$S = \Omega\left(\frac{F^2\log(NFI/\mu)}{p^2\rho_2^2\delta_{\min}^2}\right), \longrightarrow \boxed{\text{More no. of joint realizations are needed for larger } N, F, I}$$

Then, under Assumption 2, the EM algorithm in [Yeredor and Haardt, 2019] outputs the below with a probability at least $1-\mu$:

$$\left.\begin{array}{l} |\widehat{\boldsymbol{A}}_n(i,f) - \boldsymbol{A}_n(i,f)|^2 = O\left(\frac{\log(NFI/\mu)}{Sp}\right) \leq \delta_1^2, \\ |\widehat{\boldsymbol{\lambda}}(f) - \boldsymbol{\lambda}(f)|^2 = O\left(\frac{F^2\log(NFI/\mu)}{S}\right) \leq \delta_2^2. \end{array}\right\} \longrightarrow \boxed{\text{est. error decreases from intial error}}$$

Insight : CNMF-SPA Initialized EM is a scalable approach with theoretical guarantees.

Experiments: Data Classification

- Data: **UCI** datasets (https://archive.ics.uci.edu/ml/datasets.php).
- Training:Validation:Testing $= 70\%{:}10\%{:}20\%$.
- We estimate the **joint PMF of the features and the label** using training set and then predict the labels on the testing data by constructing an MAP predictor.

Table 1: UCI Dataset Car ($N = 7, I_{\text{avg}} = 4, F = 4$)

Algorithm	**Avg. Accuracy (%)**	**Time (s)**
CNMF-SPA [Proposed]	69.26±2.28	0.007
CNMF-SPA-EM [Proposed]	**86.61±1.76**	0.018
CTD [Kargas et al., 2017]	83.47±2.34	0.845
CTD-EM [Yeredor and Haardt, 2019]	85.72±1.88	0.955
SVM	83.65±1.58	0.147
Linear Regression	80.68±1.61	0.029
Neural Net	85.00±3.22	0.193
SVM-RBF	76.22±3.93	0.793
Naive Bayes	83.42±2.15	0.026

Experiments: Data Classification

Table 2: UCI Dataset Mushroom ($N = 22, I_{\text{avg}} = 6, F = 2$)

Algorithm	Avg. Accuracy (%)	Time (sec.)
CNMF-SPA [Proposed]	92.23+/-6.15	0.025
CNMF-SPA-EM [Proposed]	**99.47+/-0.80**	0.242
CTD [Kargas et al., 2017]	96.40+/-0.59	13.695
CTD-EM [Yeredor and Haardt, 2019]	97.18+/-1.21	13.931
SVM	97.47+/-0.46	37.213
Linear Regression	93.38+/-0.59	0.040
Neural Net	98.98+/-1.97	1.036
SVM-RBF	98.89+/-0.34	2.291
Naive Bayes	94.84+/-0.55	0.048

Conclusion

- A new framework for recovering joint PMF is proposed.
  - □ **two-dimensional marginals-based method**
  - □ **reduced sample complexity and computational burden**
  - □ **scalable NMF based algorithm**
  - □ **effective under finite samples and sparse data**

- An EM algorithm is shown to provably improve the output of our approach.
  - □ **appealing joint PMF recovery accuracy**

Thank You!!

*

References

D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in Neural Information Processing Systems*, pages 1141–1148, 2003.

N. Gillis and S.A. Vavasis. Fast and robust recursive algorithms for separable nonnegative matrix factorization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(4): 698–714, April 2014.

Y. Han, J. Jiao, and T. Weissman. Minimax estimation of discrete distributions under ℓ_1 loss. *IEEE Trans. Info. Theory*, 61(11):6343–6354, 2015.

Christopher J Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):45, 2013.

Nikos Kargas, Nicholas D Sidiropoulos, and Xiao Fu. Tensors, learning, and'kolmogorov extension'for finite-alphabet random vectors. *arXiv preprint arXiv:1712.00205*, 2017.

Nikos Kargas, Nicholas D. Sidiropoulos, and Xiao Fu. Tensors, learning, and kolmogorov extension for finite-alphabet random vectors. *IEEE Trans. Signal Process.*, 66:4854–4868, Jul 2018.

A. Yeredor and M. Haardt. Maximum likelihood estimation of a low-rank probability mass tensor from partial observations. *IEEE Signal Process. Lett.*, 26(10):1551–1555, Oct 2019.