Low Rank Joint Probability Tensor Recovery from 1D Marginals

Jian Vora

Department of Electrical Engineering, IIT Bombay

jianvora@iitb.ac.in

December 8, 2020

Overview

Problem Statement

The ever classical problem of given N random variables $X_1, X_2, ..., X_N$, and their realisations as samples, we need to estimate $p(X_1, X_2, ..., X_N)$

Lots of work in this domain like Gaussian mixture models, Bayesian Nets, Markov Random Fields, Variational Autoencoders ...

Can we estimate the joint density from the marginals? Does transforming the data help in the above goal? Do we have any prior knowledge of naturally occurring probability densities?

Low Rank of the Joint Probability

Q. Do we have any prior knowledge of joint probability densities? Yes!

- For real world data, random variables are reasonably (in)dependent
- For example, in the case of images, we know neighbouring pixels values are highly correlated while value of far off pixels are almost independent.

This implies that the joint probability density $p(X_1, X_2, ..., X_N)$ has a *low-rank*. But what does it mean for a tensor to have a low-rank?

We shall now explore a model which helps in capturing this low-rankness of a tensor which will be used to model the joint density throughout this talk.

Canonical Polyadic Decomposition of a Tensor

$N-$way tensor $\boldsymbol{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \ldots \times I_N}$ admits a Canonical Polyadic Decomposition(CPD) if it can be decomposed as a sum of F rank-1 tensors. Let $a \otimes b$ denote the outerproduct of two vectors, then formally the CPD model can be stated as follows:

$$\boldsymbol{X} = \sum_{f=1}^{f=F} \boldsymbol{\lambda}(f) \boldsymbol{A}_1(:, f) \otimes \boldsymbol{A}_2(:, f) \otimes \boldsymbol{A}_3(:, f) \otimes \ldots \otimes \boldsymbol{A_N}(:, f)$$

where $\boldsymbol{\lambda} \in \mathbb{R}^F$ and $\boldsymbol{A_i} \in \mathbb{R}^{I_i \times F}$.

The joint probability density $p(X_1, X_2, ..X_N)$ is an $N-$way tensor(referred to as $\boldsymbol{X}$ henceforth), more specifically

$$\boldsymbol{X}(x_1, x_2, \ldots x_N) = p(X_1 = x_1, X_2 = x_2, ..X_N = x_N)$$

Canonical Polyadic Decomposition of a Tensor(Notation)

- $\{\boldsymbol{A_i}\}_{i=1}^{N}$ are referred to as mode latent factors
- Elements of $\boldsymbol{\lambda}$ are referred to as component mixing weights
- $\boldsymbol{X}$ is the core pmf tensor which we aim to recover
- F is the rank of the tensor

For a tensor $\boldsymbol{X}$ permitting a Canonical Polyadic Decomposition, we refer $\boldsymbol{X} = [\boldsymbol{\lambda}, \boldsymbol{A}_1, \boldsymbol{A}_2, ..., \boldsymbol{A_N}]$. Thus there exists a bijection between $\boldsymbol{X}$ and the set $[\boldsymbol{\lambda}, \boldsymbol{A}_1, \boldsymbol{A}_2, ..., \boldsymbol{A_N}]$ and recovering the core tensor is equivalent to recovering the mode latent factors and the mixing weights.

This drastically reduces the number of parameters to be estimated which were previously exponential in N!

Canonical Polyadic Decomposition of a Tensor

The CPD model can be considered as a naive bayes model with the hidden state H taking a bounded number of states(= the rank F)

$$Pr(X_1 = i_1, X_2 = i_2, .., X_N = i_n) = \sum_{f=1}^{f=F} Pr(H = f) \prod_{n=1}^{n=N} Pr(X_n = i_n | H = f)$$

with $\boldsymbol{\lambda}(f) = Pr(H = f)$ and $\boldsymbol{A}_n(i_n, f) = Pr(X_n = i_n | H = f)$

Uniqueness of the CPD Model

For a tensor $\boldsymbol{X}$ of rank F, we say that a decomposition $\boldsymbol{X} = [\boldsymbol{\lambda}, \boldsymbol{A}_1, \boldsymbol{A}_2, .., \boldsymbol{A}_N]$ is essentially unique if the factors are unique up to a common permutation and scaling / counter-scaling of columns of the mode latent factors.

Uniqueness of the CPD

We clearly do not have the scaling ambiguity if we the non-negativity and sum to one constraints on the columns of the latent factors. Let $\boldsymbol{X} = [\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_2]$, where $\boldsymbol{A}_1 \in \mathbb{R}^{I_1 \times F}$, $\boldsymbol{A}_2 \in \mathbb{R}^{I_2 \times F}$, and $\boldsymbol{A}_3 \in \mathbb{R}^{I_3 \times F}$ with $I_1 \leq I_2 \leq I_3$ without loss of generality, then:

Lemma on Uniqueness

If $\min(I_1, I_2) \geq 3$ and $F \leq I_3$, then, $\text{rank}(\boldsymbol{X}) = F$ and the decomposition of $\boldsymbol{X}$ is essentially unique, almost surely, if and only if $F \leq (I_1 - 1)(I_2 - 1)$

Thus, instead of F being $I_1 I_2 I_3$ which would imply a full rank of the core pmf tensor, we can control the low-rankness of the joint density using a single tunable hyperparameter F

`[Kargas et.al, IEEE TSP 2018]`

Random Linear Projections

For a given vector $X \in \mathbb{R}^N$, we define the following operation as random linear projections:

$$Y = \Phi X$$

where $\Phi \in \mathbb{R}^{M \times N}$ is composed of entries drawn i.i.d from a standard normal Gaussian or a bernoulli $\{-1, 1\}$ distribution.

Used a lot in compressed sensing, dimensionality reduction, GMM learning

Radon and Inverse Radon Transform

The Radon transform of a 2-D function $f(x, y)$ is defined as:

$$R(r, \theta)[f] = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y)\delta(r - xcos\theta - ysin\theta)dxdy$$

The above operation is invertible and is referred to as the inverse radon transform. Filtered backprojection(FBP) is one of the common algorithms used for inverse radon transform defined as:

$$\hat{f}(x, y) = \int_{0}^{\pi} \int_{-\infty}^{+\infty} R(v, \theta)H(v)dvd\theta$$

$H(v)$ is the filter transfer function with some common choices being Ram-Lak, Cosine, Shepp Logan. θ is the key parameter which determines the direction of projection.

Density Estimation from Random Projections

Consider a random vector $X \in \mathbb{R}^N$ and another random vector $\theta \in \mathbb{R}^N$

$\theta^T X$ is a scalar variable and hence we can estimate $p(\theta^T X)$ which is a 1D density estimate and is fairly easy to do using either histogramming or kernel density estimate.

We can we this for various angles θ and hence we can accumulate a set of 1D densities of the form $p(\theta_i^T X)$. But do these mean anything in our original goal of estimating the joint density $p(X_1, X_2, ..., X_N)$?

We can infact relate this 1D densities of random projections to the core pmf tensor $\boldsymbol{X}$. It is tomography which comes to our rescue!

Density Estimation from Random Projections

Let $\theta = [\theta_1, \theta_2, ..., \theta_N]$ and $X = [X_1, X_2, ..., X_N]$, then

$$p(\theta^T X = t) = p(\sum_i \theta_i X_i = t)$$

$$p(\theta^T X = t) = \sum_{a_1} \sum_{a_2} .. \sum_{a_{N-1}} p(\theta_1 X_1 = a_1, \theta_2 X_2 = a_2, ... \theta_N X_N = t - \sum_{i=1}^{i=N-1} a_i)$$

Thus the 1D density estimate of a projection of a data vector is infact the radon transform of the joint probability tensor taken at an angle θ.

Thus performing an inverse radon transform approach on these densities projected along various angles gives us the original core tensor!

`[Sullivan et.al, Journal of the Royal Statistical Society, 1993]`

Empirical Results

We perform preliminary experiments on recovering just 2 dimensional densities from these 1D projections. Inverting the radon transform becomes costly as the dimensionality increases.
Evaluation Metric: Jensen-Shannon Divergence

$$JS(P, Q) = \frac{KL(P||(P+Q)/2) + KL(Q||(P+Q)/2)}{2}$$

Data Generation Process : We take a truncated 2D gaussian (from -2 to +2 in either directions) with an identity covariance matrix and 0 mean. This is put into 10 bins as cummulative interval measures to get the PDF matrix. PDF of $\theta_i X$ is generated using **histograms**.

Empirical Results

Number of samples	JSD Radon	JSD Histogramming
100	0.0608	0.1902
500	0.0478	0.0529
1000	0.0221	0.0203
10000	0.0107	0.0019

Table: Reconstruction Comparisons with 1000 projections for Radon

Clearly, the radon approach for density estimation does a better job in the low-sample regime. For the higher samples, we incur some error as the inverse radon operator is lossy in nature.

We now look at reconstruction errors for a variety of Gaussian mixture models as a function of both the number of projections and samples.

Empirical Results on GMMs

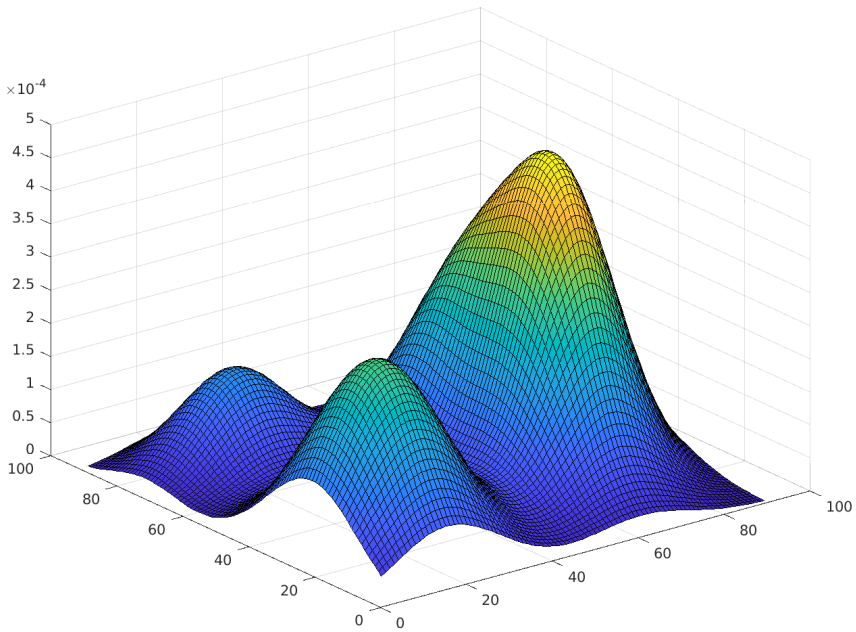

Figure: GMM 1 (4 components)

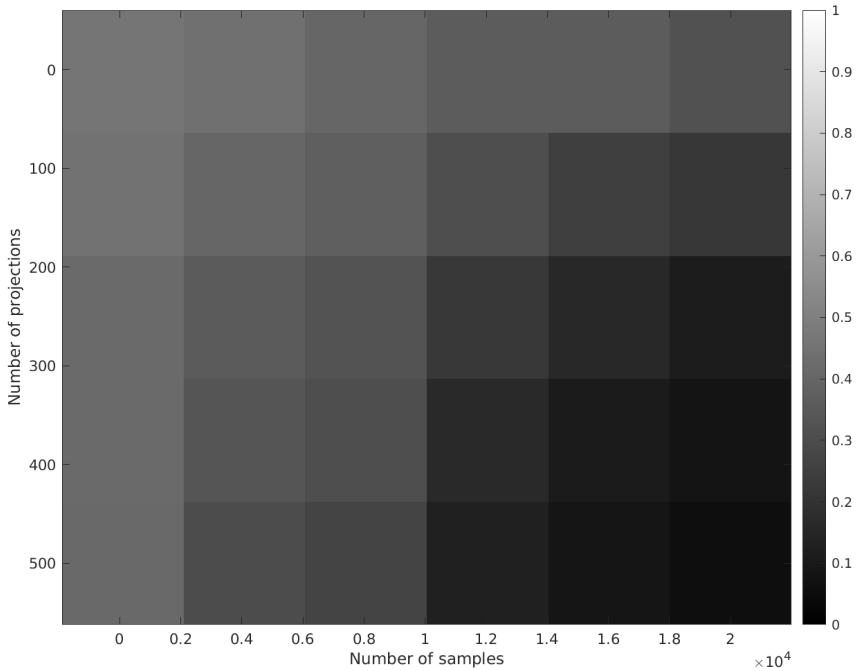

Figure: GMM 1 Errors

Empirical Results on GMMs

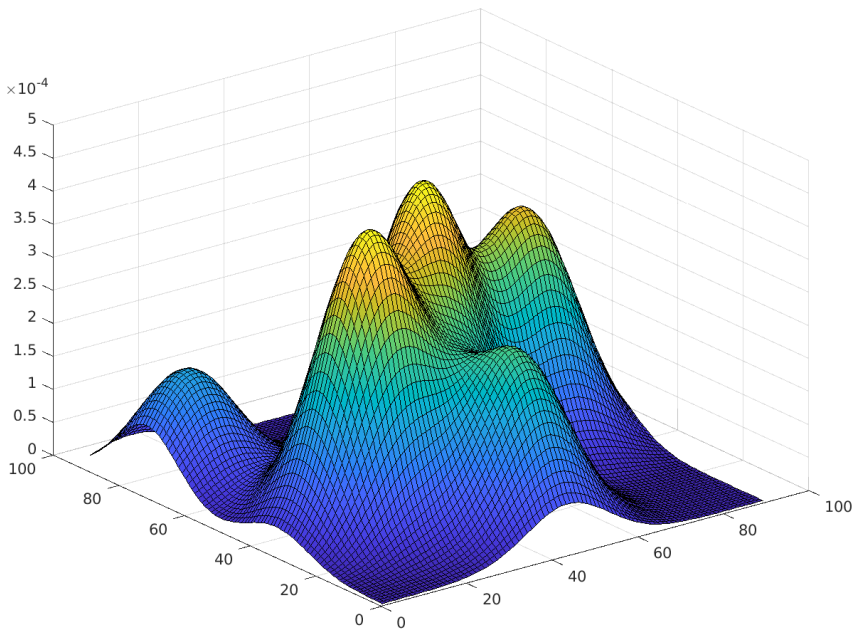

Figure: GMM 2 (9 components)

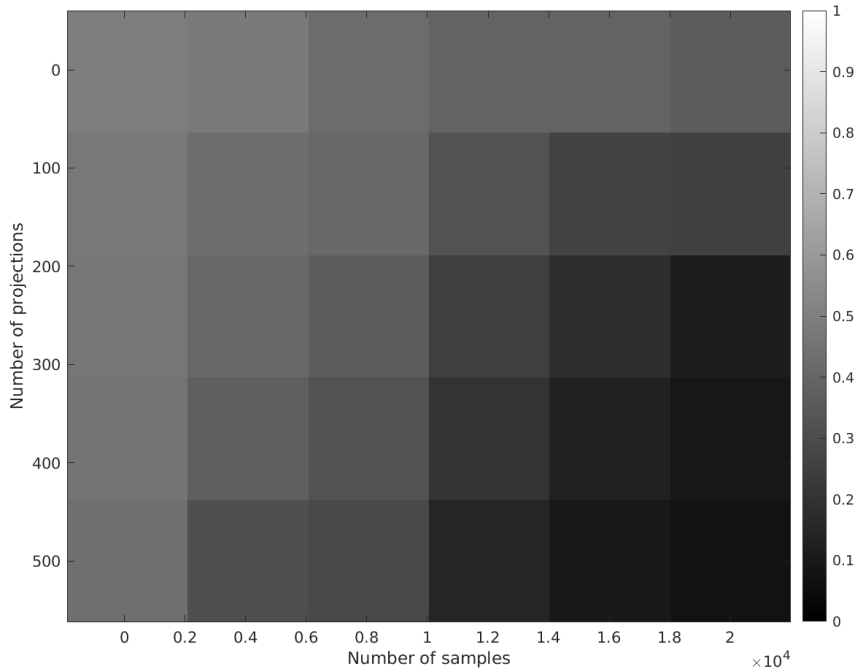

Figure: GMM 2 Errors

Empirical Results on GMMs

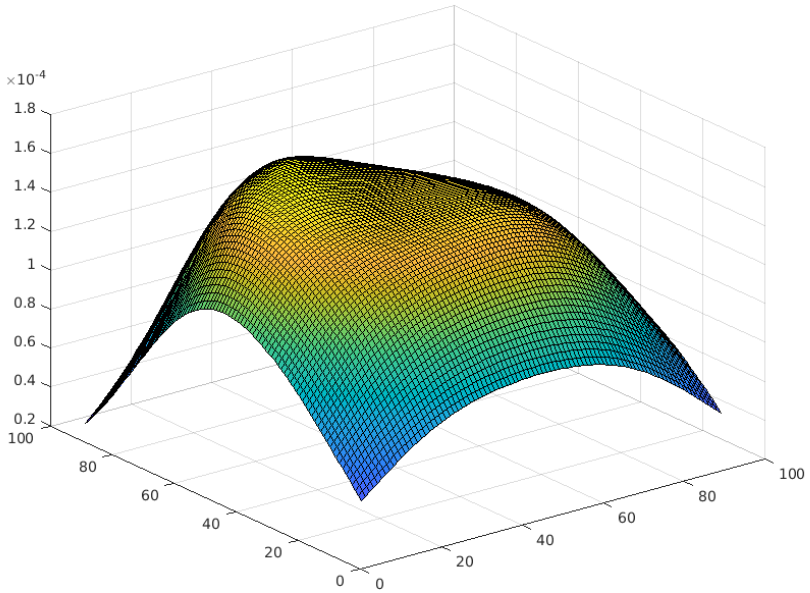

Figure: GMM 3 (4 components)

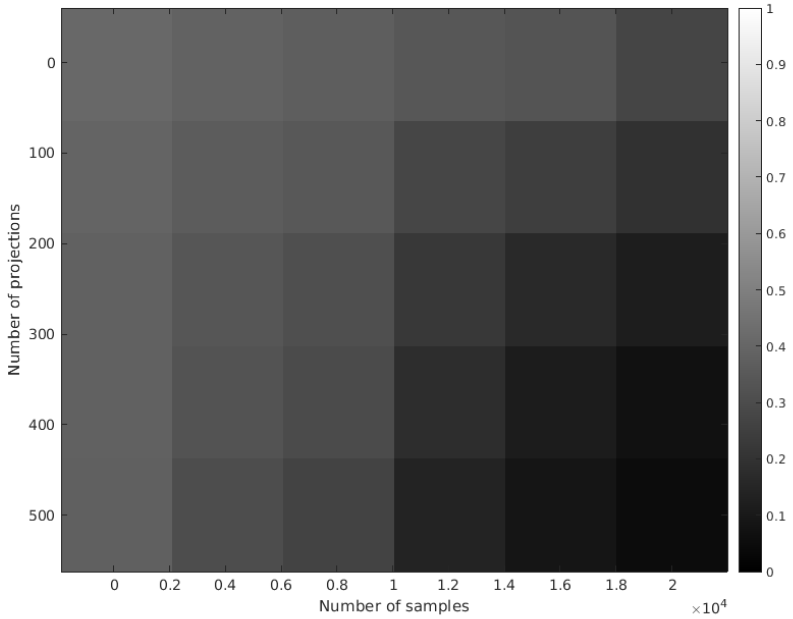

Figure: GMM 3 Errors

Joint Recovery from 3-way marginals

This was the first work concerning density estimation from marginals by [Kargas et.al, IEEE TSP 2018]

Procedure: Joint PMF Recovery From Triples

1. Estimate $\boldsymbol{X}_{j,k,l}$ from the data
2. Jointly factor $\boldsymbol{X}_{j,k,l} = [\boldsymbol{\lambda}, \boldsymbol{A_j}, \boldsymbol{A_k}, \boldsymbol{A_l}]$ using the CPD model of rank F
3. Once all the latent factors are available using the joint factorisation, reconstruct using

$$\boldsymbol{X} = \sum_{f=1}^{f=F} \boldsymbol{\lambda}(f)\boldsymbol{A}_1(:,f) \otimes \boldsymbol{A}_2(:,f) \otimes \boldsymbol{A}_3(:,f) \otimes ... \otimes \boldsymbol{A_N}(:,f)$$

For solving the joint factorisation problem, we use the following least squares optimization problem using alternating minimization:

$$\min_{\{\boldsymbol{A_n}\},\boldsymbol{\lambda}} \sum_{j} \sum_{k>j} \sum_{l>k} \|\boldsymbol{X}_{j,k,l} - [\boldsymbol{\lambda}, \boldsymbol{A_j}, \boldsymbol{A_k}, \boldsymbol{A_l}]\|_F^2$$

subject to appropriate normalization of $\boldsymbol{\lambda}$ and $\boldsymbol{A_n}, n = 1, 2, 3, ..., N$

Joint Recovery using Expectation Maximization

Work by [Yeredor et.al., Signal Process. Letters, 2019] tries to find the parameters by maximising the likelihood of observing the data

$$\log P(y[1], y[2], ..., y[T]) = \sum_{t=1}^{t=T} \log \sum_{f=1}^{f=F} \lambda_f \prod_{n=1}^{n=N} \boldsymbol{A_n}(y[t], f)$$

Thus, we formulate the following optimization problem for minimizing the negative log-likelihood:

$$\min_{\{\boldsymbol{A_n}\},\boldsymbol{\lambda}} - \sum_{t=1}^{t=T} \log \sum_{f=1}^{f=F} \lambda_f \prod_{n=1}^{n=N} \boldsymbol{A_n}(y[t], f)$$

subject to appropriate normalization of $\boldsymbol{\lambda}$ and $\boldsymbol{A_n}, n = 1, 2, 3, ..., N$

Joint Recovery using 2-way marginals

Work by [Ibrahim et.al., https://arxiv.org/abs/2006.16912]
Construct the 2-way marginals $\boldsymbol{X}_{j,k} = \boldsymbol{A_j} \boldsymbol{D}(\lambda) \boldsymbol{A_k^T}$ using histogramming

Split the N variables into 2 sets with M being the index of the split. Thus we have 2 splits namely $\{1, 2, ..., M\}$ and $\{M+1, M+2, ..., N\}$

Arrange the 2-way marginals in the form a matrix $\boldsymbol{A} \in \mathbb{R}^{MI \times (N-M)I}$ where the block $\boldsymbol{A_{i,j}} = \boldsymbol{X_{i,M+i}}$ where NMF can be applied to get the factors

Use the SPA algorithm for NMF, followed by refining the estimates by running EM with the inital conditions as the output of the SPA algorithm

Algorithm for high-dimensional density estimation

1. For i, j such that $1 \leq i < j \leq N$, take random linear projections of data with variables X_i and X_j. For each such pair, if we take M projections, then in total we will have $M.N(N-1)/2$ scalar numbers.
2. Using the above projections, estimate densities of the form $p(X_i, X_j)$ using the inverse radon transform method.
3. Once we have all the pair wise marginals, then reconstruct the core probability tensor using the SPA algorithm.
4. Refine the estimates of the mode latent factors by running the EM algorithm on the estimates obtained.

Experimental Conditions

Mode latent factors were generated with values of each column from a sine wave to ensure the underlying core pmf tensor is smooth. The amplitude, frequency and phase were chosen at random for each column of the mode latent factors.

The mixing ratios were chosen randomly with no such smoothness prior.

200 projections were taken for the radon approach.

$$\text{MSE} = \|\lambda - \hat{\lambda}\|_2^2/F + \sum_i \|A_i - \hat{A}_i\|_2^2/NF$$

Algorithms compared for performance

CNMF-SPA: Using the SPA algorithm to predict the mode latent factors `[Ibrahim et.al.]`

SPA-EM: Running EM after the CNMF-SPA algorithm to refine the mode latent factors `[Ibrahim et.al.]`

CNMF-SPA-R: Using the SPA algorithm to predict the mode latent factors but with the initial 2-way marginals coming from the radon transform approach `[Our Approach]`

SPA-EM-R: Running EM after the CNMF-SPA-R algorithm to refine the mode latent factors `[Our Approach]`

RAND-EM: Not estimating and marginals, run EM directly with radnom initialiaztions of the mode latent factors and λ `[Yeredor et.al.]`

Selected Empirical Results

	1000	5000	10000	30000
SPA	0.0911	0.0543	0.0459	0.0497
SPA-EM	0.0737	0.0721	0.0358	**0.0422**
SPA-R	**0.0607**	**0.0469**	0.0376	0.0433
SPA-R-EM	0.0668	0.0579	**0.0353**	0.0452
RAND-EM	0.0713	0.0473	0.0537	0.0598

Table: MSE Comparison for F = 10, I = 50, N = 5

	1000	5000	10000	30000
SPA	0.0821	0.0963	0.0654	0.0497
SPA-EM	0.0814	0.0518	0.0452	0.0473
SPA-R	0.0611	0.0543	**0.0442**	0.0475
SPA-R-EM	0.0642	**0.0512**	0.0494	**0.0468**
RAND-EM	**0.0473**	0.0521	0.0489	0.0719

Table: MSE Comparison for F = 10, I = 200, N = 4

Selected Empirical Results

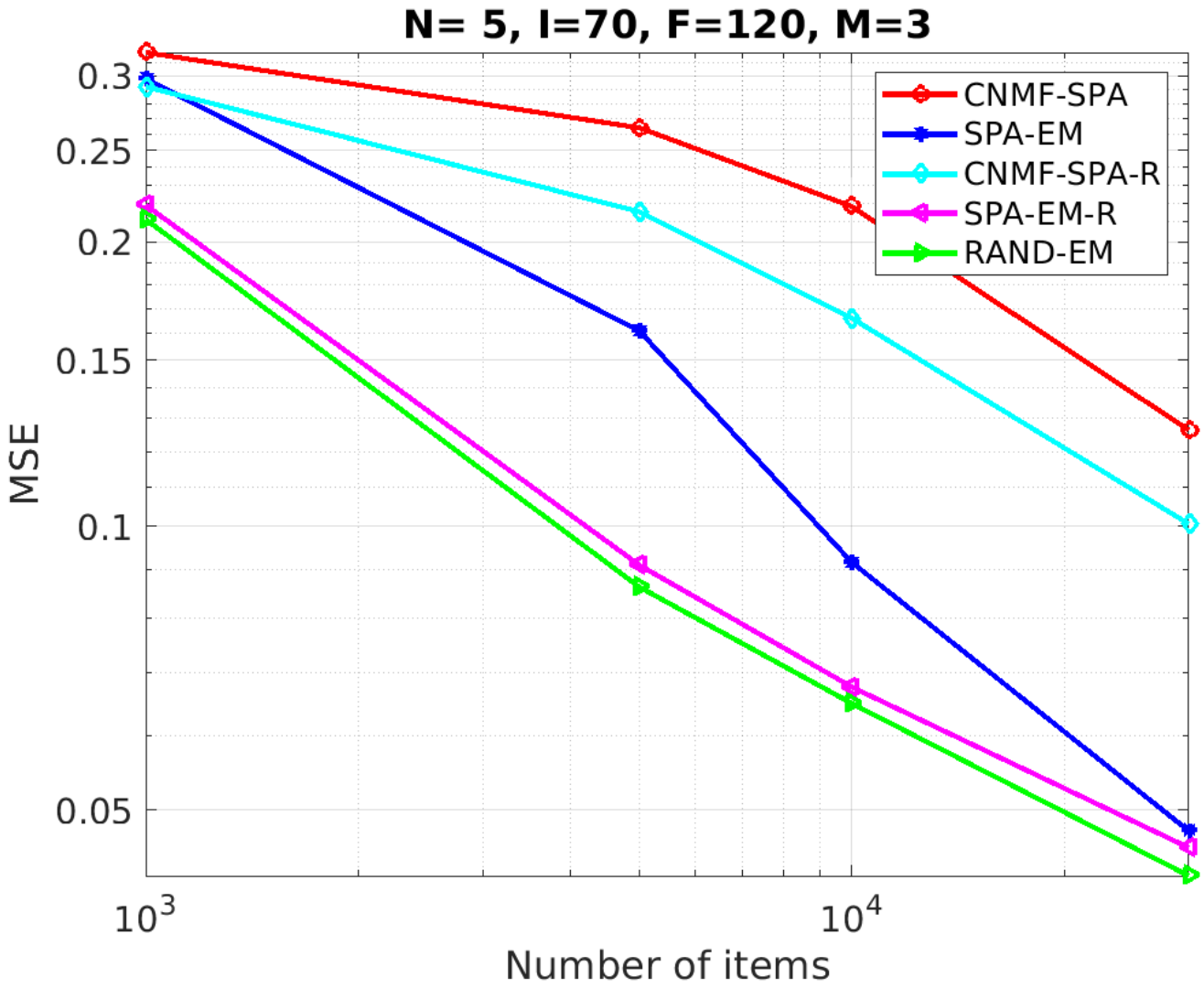

Figure: Errors averaged over 5 different randomly chosen densities

Selected Empirical Results

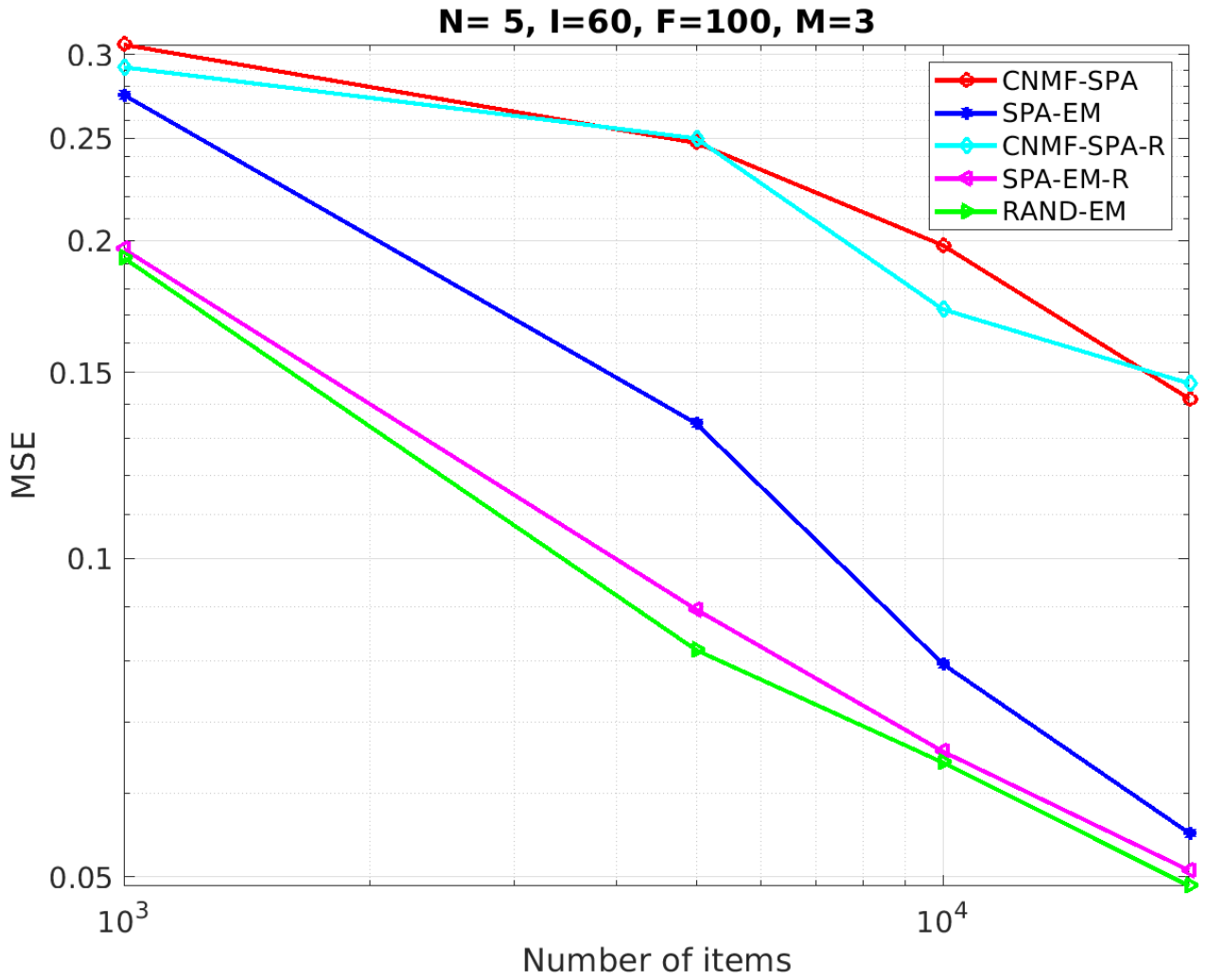

Figure: Errors compared in a different setting

Empirical Results for Missing Data

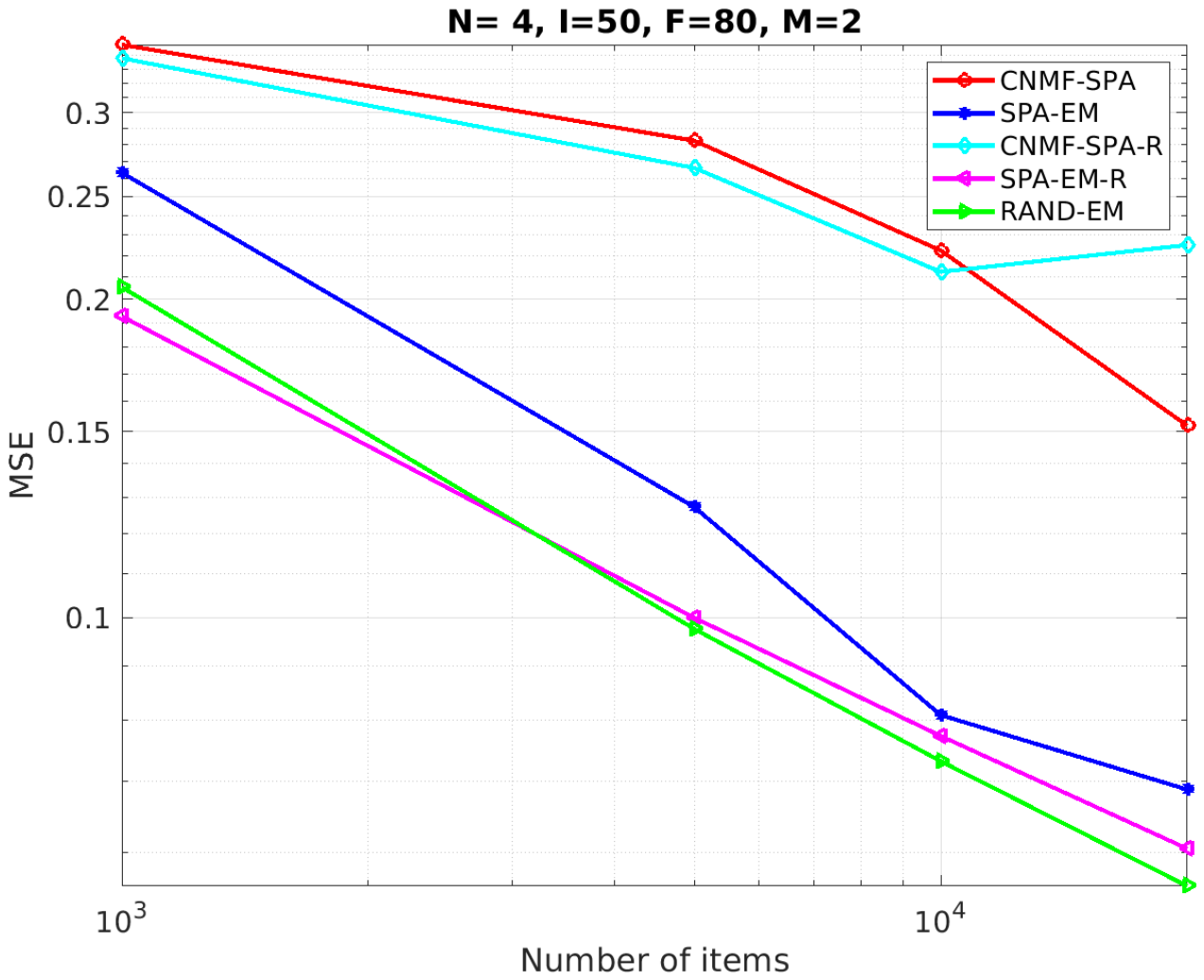

Figure: Errors with **missing data** with probability 0.2

Empirical Results for Higher Dimensions

	1000	10000	50000
SPA	0.182	0.158	0.083
SPA-EM	0.217	0.126	0.085
SPA-R	0.142	0.117	0.103
SPA-R-EM	**0.102**	**0.081**	0.078
RAND-EM	0.113	0.084	**0.076**

Table: MSE Comparison for F = 200, I = 90, N = 12

Future Work

1. Try the algorithm on real-world datasets for doing some probabilistic inferences like classification or regression
2. Remove the constraint of the rank of the tensor being strictly F and rather pose the problem as a low-rank tensor recovery where we penalize some metric to penalise the rank of the tensor. This would in general lead to a more expressive model and reduce a tunable parameter from the system.
3. Scale the algorithm to very high dimensional data sets(for example 500) which is not possible in the current framework because of the limitations of the SPA algorithm
4. Try estimating the actual continuous density by spline interpolation on the mode latent factors.

References

1. K. P. Murphy, Machine learning: a probabilistic perspective. MIT press, 2012.
2. N. Kargas, N. D. Sidiropoulos, and X. Fu, "Tensors, learning, and 'Kolmogorov extension'for finite-alphabet random vectors," IEEE Trans. Signal Process., vol. 66, no. 18, pp. 4854–4868, 2018.
3. A. Yeredor and M. Haardt, "Maximum likelihood estimation of a lowrank probability mass tensor from partial observations," IEEE Signal Process. Lett., vol. 26, no. 10, pp. 1551–1555, Oct 2019
4. S. Ibrahim, X. Fu, Recovering Joint Probability of Discrete Random Variables from Pairwise Marginals - https://arxiv.org/abs/2006.16912
5. Tensors and Probability: An Intriguing Union - N. Sidiropoulos, N. Kargas, X. Fu, GlobalSIP 2018 Keynote
6. Finbarr O'Sullivan and Yudi Pawitan, Multidimensional Density Estimation by Tomography, Journal of the Royal Statistical Society. Series B (Methodological) , 1993, Vol. 55, No. 2 (1993), pp. 509-521

Questions?