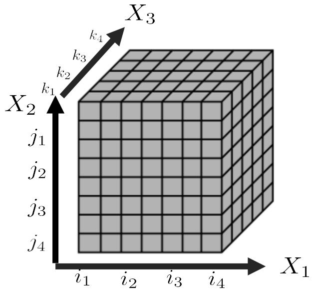


University of Virginia

Tensors & Probability

Nonparametric Multivariate Density Estimation: A Low-Rank Characteristic Function Approach

Nicholas D. Sidiropoulos

(joint work with M. Amiridi and N. Kargas)

December, 2020

December, 2020

``Καὶ γνώσεσθε τὴν ἀλήθειαν, καὶ ἡ ἀλήθεια ἐλευθερώσει ὑμᾶς”

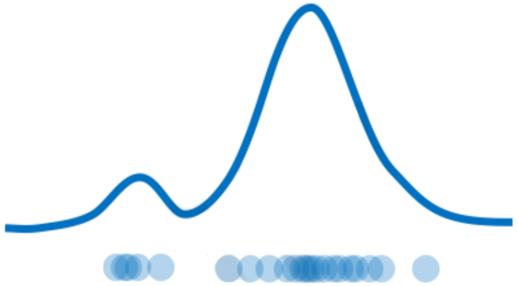
“And ye shall know the truth, and the truth shall make you free.”

December, 2020

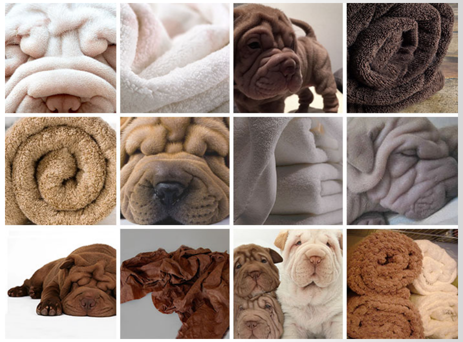
Modeling high-dimensional distributions

- **Unsupervised learning**
  - We need unsupervised models to deal with **uncertainty**
  - Discover hidden structure in the data
  - **Probability Density Function Estimation** (PDF) is a fundamental problem in unsupervised ML
    - ✓ **Goal:** Given training samples, learn the data generating distribution

PDF Estimation



Training Data



Training Data

$P(Dog) = ?$

Modeling high-dimensional distributions

- **PDF Estimation** is a fundamental problem in unsupervised ML
  - Given a dataset $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$, where $\boldsymbol{x}_i \in \mathbf{R}^N$
  - We assume the data has been drawn iid from an unknown data generating distribution: $\boldsymbol{x}_i \sim f_{\boldsymbol{X}}(\boldsymbol{x}_i)$

$$\boxed{f_{\boldsymbol{X}}(\cdot)} \; ?$$

$$\downarrow$$

$$\boxed{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M}$$

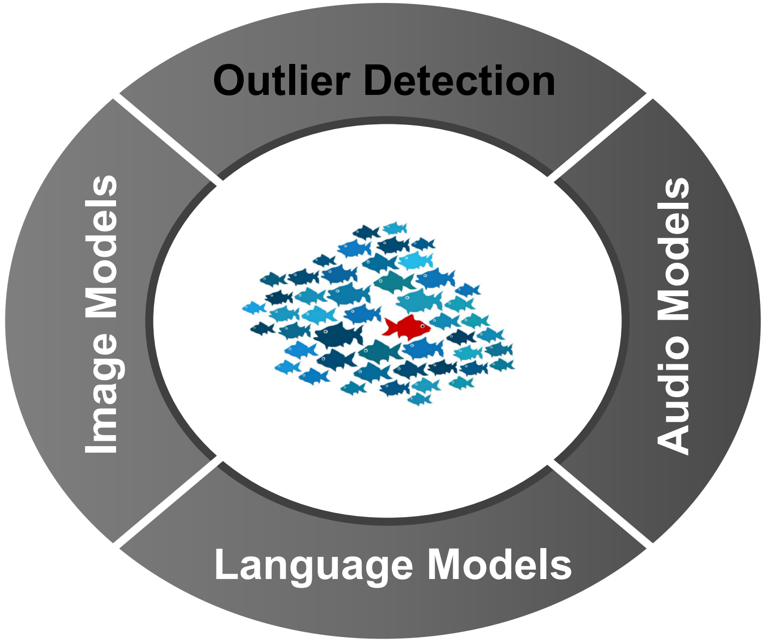
- **Goal**: Estimate $f_{\boldsymbol{X}}(\cdot)$
- **Why?** If we can learn high-dimensional **joint PDFs**, we can address ML problems using principled methods
  - Estimating any marginal or conditional distribution, expectation
  - Computing the most likely value of a subset of features conditioned on others
  - Deriving optimal estimators, classifiers

Modeling high-dimensional distributions

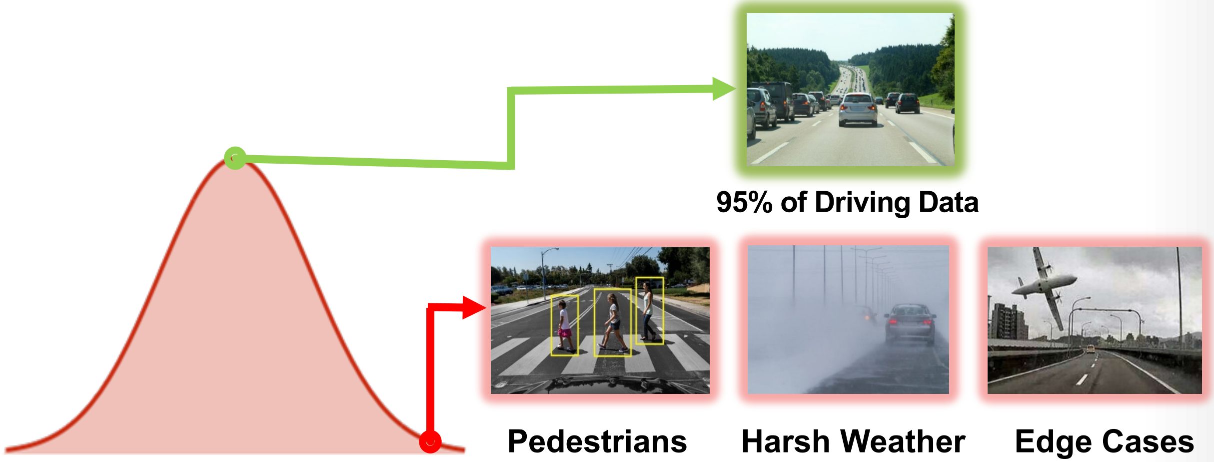
- **PDF Estimation** is a fundamental problem in unsupervised ML
  - Given a dataset $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$, where $\boldsymbol{x}_i \in \mathbf{R}^N$
  - We assume the data has been drawn iid from an unknown data generating distribution: $\boldsymbol{x}_i \sim f_{\boldsymbol{X}}(\boldsymbol{x}_i)$

- **Goal**: Estimate $f_{\boldsymbol{X}}(\cdot)$
- **Challenges**
  1. Curse of Dimensionality:
     - Modern datasets are high-dimensional and complex, we often operate in the sample-starved regime
  2. Incomplete realizations
  3. Model identifiability?
  4. Expressivity - tractability trade-off
  5. Sample complexity

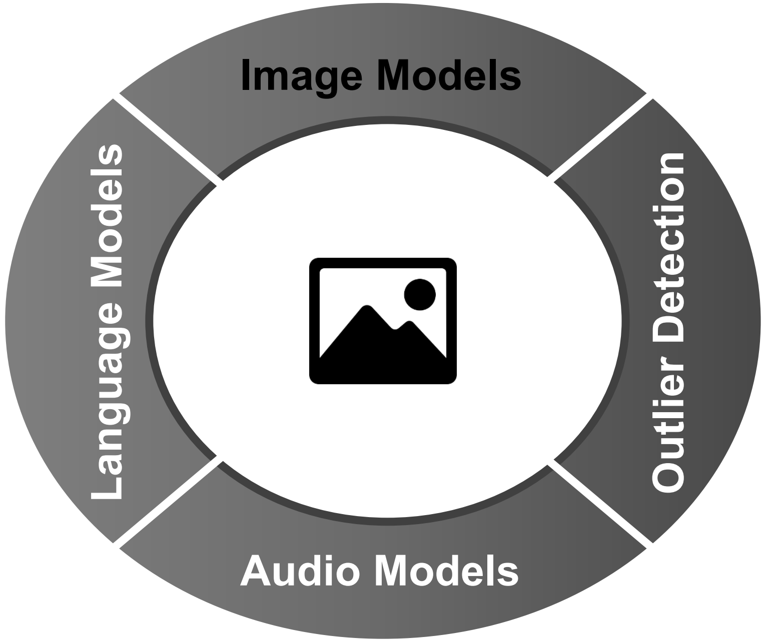
Probabilistic modeling: Applications



- **Multidimensional probabilistic models have many applications in ML**
  - **Problem**: Detect new or rare events!
    - e.g. Fraud detection: Legitimate financial transactions vs fraudulent transactions
  - **Strategy**: Leverage statistical models, detect outliers in the distribution
    - e.g. self driving cars: Use outliers to train more robust models



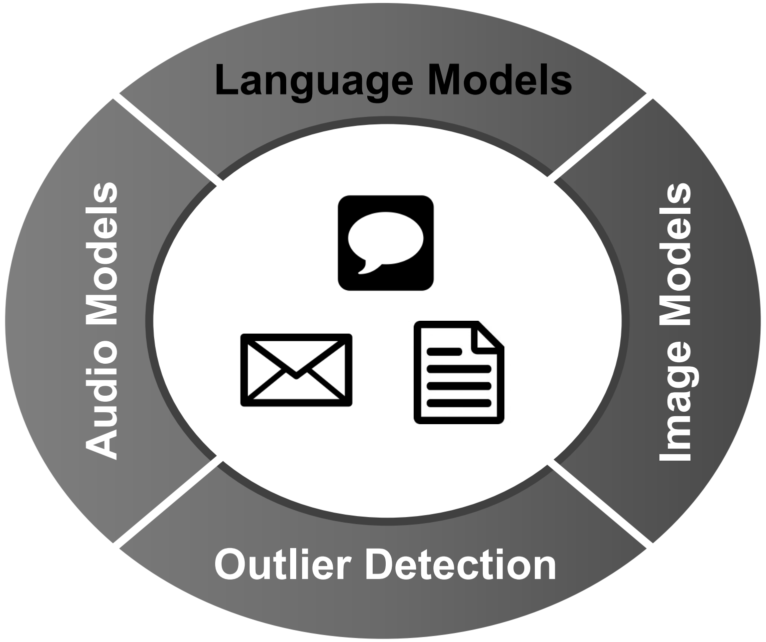
Probabilistic modeling: Applications



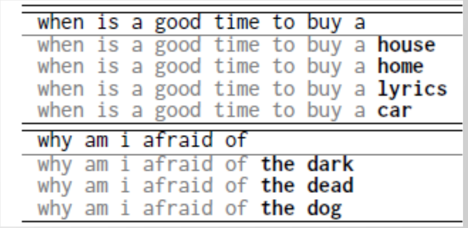
- **Multidimensional probabilistic models have many applications in ML**
  - **Generate high-fidelity images**
    - Create realistic and pleasing artwork -- **Zach Monge, CycleGAN, Zhu et al.**

    - Which face is artificially generated? -- **Philip Wang, Flickr-Faces-HQ dataset**

Probabilistic modeling: Applications

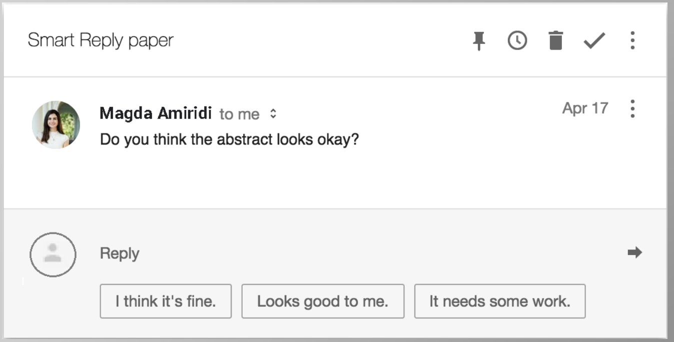


- **Multidimensional probabilistic models have many applications in ML**
  - **Text synthesis**
    - Generate new Wikipedia-like articles, Smart Reply, Autocomplete



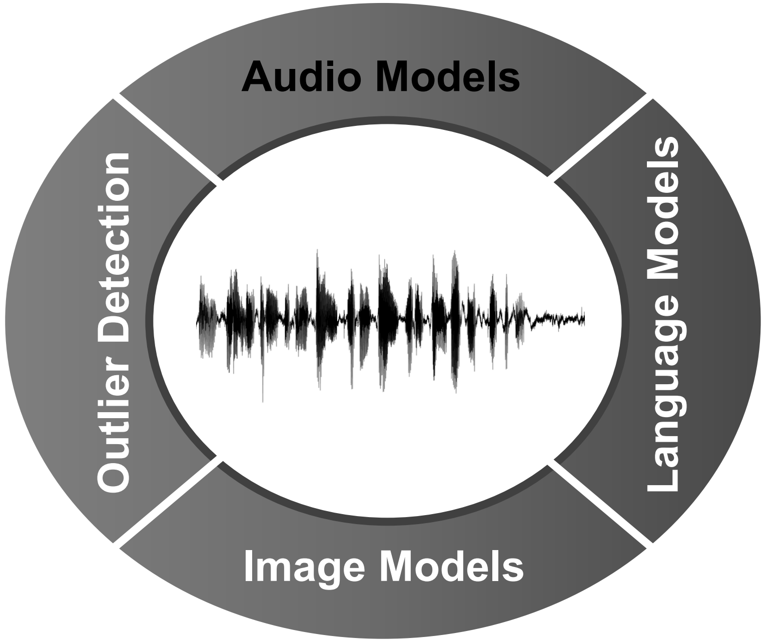
when is a good time to buy a
when is a good time to buy a **house**
when is a good time to buy a **home**
when is a good time to buy a **lyrics**
when is a good time to buy a **car**

why am i afraid of
why am i afraid of **the dark**
why am i afraid of **the dead**
why am i afraid of **the dog**



  - **Translation**
    - Model $p(y|x)$ to generate an English sentence y conditioned on the corresponding Chinese sentence x

Probabilistic modeling: Applications



- **Multidimensional probabilistic models have many applications in ML**
  - **Upsampling, Speech synthesis**
  - **Speech recognition**
    - Given a joint model of speech signals and language (text), we can infer spoken words from audio signals

Starting point

- **Categorical case:** joint PMF $f(i, j, k, \ell, \ldots)$

Every joint PMF of a finite-alphabet random vector can be represented by a naïve Bayes model with a finite number of latent states (rank).

- If the rank is low, the high dimensional joint PMF is almost surely identifiable from three-dimensional marginals under low-rank conditions

"Tensors, Learning, and Kolmogorov Extension for Finite-alphabet Random Vectors", Kargas, N. D. Sidiropoulos, X. Fu 2018

- **Extension to continuous random vectors** → joint PDF $f(x, y, z, v, \ldots)$ no longer a tensor!

Starting point

- **Categorical case:** joint PMF $f(i, j, k, \ell, \dots)$

Every joint PMF of a finite-alphabet random vector can be represented by a naïve Bayes model with a finite number of latent states (rank).

- If the rank is low, the high dimensional joint PMF is almost surely identifiable from three-dimensional marginals under low-rank conditions

"Tensors, Learning, and Kolmogorov Extension for Finite-alphabet Random Vectors", Kargas, N. D. Sidiropoulos, X. Fu 2018

- **Extension to continuous random vectors** → joint PDF $f(x, y, z, v, \dots)$ no longer a tensor!

  - One possibility: discretization
    - Coarse vs fine → discretization error vs statistical accuracy
    - How do we choose a discretization scheme?
    - Loss of identifiability
  - Is it possible to avoid discretization?
  - How can one represent a Probability Density Function through a tensor?

Sneak preview

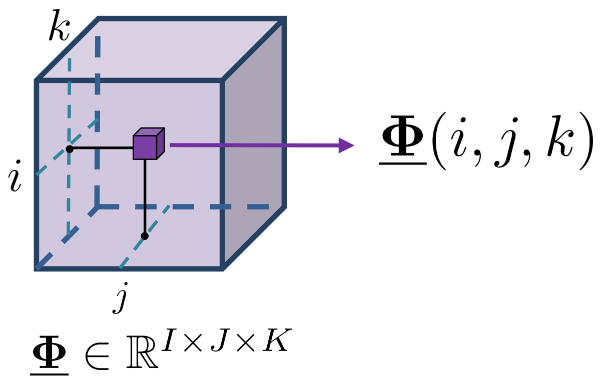
- **Tensors as universal PDF approximators**

  We will see that:

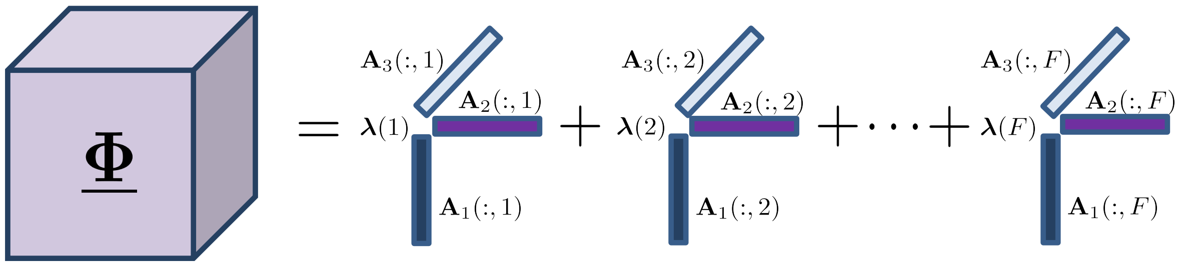
  - A finite mixture model (approximately) follows from
    1. **compactness of support**
    2. **continuous differentiability**
  - Assuming **low-rank in the Fourier domain,** a controllable approximation of the multivariate density is **identifiable**
  - High dimensional joint PDF **recovery by** observing **subsets** (triples) **of variables** is possible!

What are tensors? - Canonical Polyadic Decomposition

- An N-way tensor $\underline{\mathbf{\Phi}} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$ is a multidimensional array whose elements are indexed by N indices

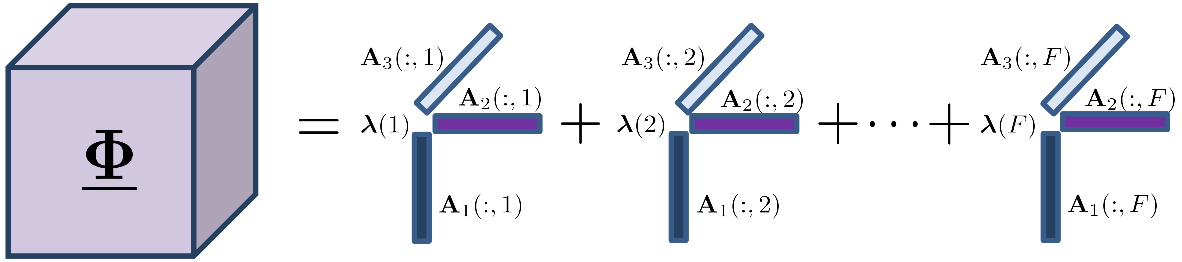


- Any tensor can be decomposed as a sum of F rank-1 tensors



What are tensors? - Canonical Polyadic Decomposition

- An N-way tensor $\underline{\mathbf{\Phi}} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$ is a multidimensional array whose elements are indexed by N indices



$$\underline{\mathbf{\Phi}} = \sum_{f=1}^{F} \mathbf{A}_1(:,f) \circ \mathbf{A}_2(:,f) \circ \cdots \circ \mathbf{A}_N(:,f)$$

- Any tensor can be decomposed as a sum of F rank-1 tensors
  - We use $\underline{\mathbf{\Phi}} = [\![\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N]\!]$ to denote the decomposition
  - Element-wise view: $\underline{\mathbf{\Phi}}(i_1, i_2, \ldots, i_N) = \sum_{f=1}^{F} \boldsymbol{\lambda}(f) \prod_{n=1}^{N} \mathbf{A}_n(i_n, f)$
- F is the smallest number for which such decomposition exists

Uniqueness of CPD

Essential Uniqueness

For a tensor $\underline{\mathbf{\Phi}}$ of rank *F*, we say that a decomposition $\underline{\mathbf{\Phi}} = [\![\mathbf{A}_1, \ldots, \mathbf{A}_N]\!]$ is essentially unique if the factors are unique up to a common permutation and scaling/counter-scaling of columns.

- This means that if there exists another decomposition $[\![\widehat{\mathbf{A}}_1, \ldots, \widehat{\mathbf{A}}_N]\!]$, then, there exists a permutation matrix and diagonal scaling matrices such that

$$\widehat{\mathbf{A}}_n = \mathbf{A}_n \mathbf{\Pi} \mathbf{\Lambda}_n \text{ and } \prod_{n=1}^{N} \mathbf{\Lambda}_n = \mathbf{I}$$

- There is no scaling ambiguity for the column-normalized representation $\underline{\mathbf{\Phi}} = [\![\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N]\!]$

Uniqueness of CPD

Essential Uniqueness

For a tensor $\underline{\mathbf{\Phi}}$ of rank *F*, we say that a decomposition $\underline{\mathbf{\Phi}} = [\![\mathbf{A}_1, \ldots, \mathbf{A}_N]\!]$ is essentially unique if the factors are unique up to a common permutation and scaling/counter-scaling of columns

Theorem (Chiantini and Ottaviani, 2012)

If $\min(I_1, I_2) \geq 3$ and $F \leq I_3$ then, the rank of $\underline{\mathbf{\Phi}}$ is *F* and the decomposition is unique, almost surely, if and only if $F \leq (I_1 - 1)(I_2 - 1)$

- In other words: the parameters of the CPD model are identifiable under certain rank conditions

Density Estimation: Classical methods

Two main genres:

- **Parametric models**: make strong assumptions about the structure of the data, fragile to model mismatch
  - **GMMs** (Pearson 1894, McLachlan, Basford 1988)
    - ➢ Computational and estimation challenges in the high dimensional case

Density Estimation: Classical methods

Two main genres:

- **Parametric models**: make strong assumptions about the structure of the data, fragile to model mismatch
  - **GMMs** (Pearson 1894, McLachlan, Basford 1988)
    - ➢ Computational and estimation challenges in the high dimensional case

- **Non-parametric models:** make only mild, “universal” prior assumptions about the data, such as smoothness
  - **KDE** (Rosenblatt 1956, Parzen 1962): estimates the PDF by means of a sum of kernel functions centered at the given observations
    - ➢ Computationally intractable for large M,N
  - **OSDE** (Girolami 2002; Efromovich 2010): approximates a PDF using a truncated sum of orthonormal basis functions
    - ➢ Curse of Dimensionality

Density Estimation: Modern methods

Several flavors (for Neural DE models):

- **Explicit density estimation**: explicitly define and solve for $f_X(x)$
  1. Auto-regressive neural models for DE

     e.g. **RNADE** (Uria, Murray, Larochelle 2013) -- Generally suffer from slow sampling time
  2. Flow-based neural models for DE

     e.g. **NICE** (Dinh, Krueger, Bengio 2014), **Real-NVP** (Dinh, Sohl-Dickstein, Bengio 2016) – Constrained architectures possibly not sufficiently expressive to capture all distributions

  - Point-wise density evaluation
  - Cannot impute more than very few missing elements in the input
  - No identifiability guarantees

Density Estimation: Modern methods

Several flavors (for Neural DE models):

- **Explicit density estimation**: explicitly define and solve for $f_X(x)$
  1. Auto-regressive neural models for DE
     e.g. **RNADE** (Uria, Murray, Larochelle 2013) -- Generally suffer from slow sampling time
  2. Flow-based neural models for DE
     e.g. **NICE** (Dinh, Krueger, Bengio 2014), **Real-NVP** (Dinh, Sohl-Dickstein, Bengio 2016) – Constrained architectures possibly not sufficiently expressive to capture all distributions

  - Point-wise density evaluation
  - Cannot impute more than very few missing elements in the input
  - No identifiability guarantees

- **Implicit density estimation**
  - Approximate density
    e.g. **VAEs** (Kingma and Welling 2014)
  - Frameworks that learn a model that can sample from $f_X(x)$ w/o explicitly defining it
    e.g. **GANs** (Goodfellow et al. 2014)

  - Mainly used for only one very specific task: generating samples similar to training data
  - Hard to train

A Characteristic Function approach for DE

Goal: Obtain a PDF estimate that is

- **Expressive**: flexible enough to represent a wide class of distributions
- **Tractable** and **scalable** (computationally and memory-wise)
- **Principled**
- **Accurate**

A Characteristic Function approach for DE

Goal: Obtain a PDF estimate that is

- **Expressive**: flexible enough to represent a wide class of distributions
- **Tractable** and **scalable** (computationally and memory-wise)
- **Principled**
- **Accurate**

- Given a real-valued random variable X

Fourier transform pair:
$$\left\{ \begin{aligned} \Phi_X(\nu) &:= \int_{S_X} f_X(x) e^{j\nu x} dx = E[e^{j\nu X}] \\ f_X(x) &:= \frac{1}{2\pi} \int_{-\infty}^{\infty} \Phi_X(\nu) e^{-j\nu x} d\nu \end{aligned} \right\}$$

- Expectation interpretation → estimation via sample averages

A Characteristic Function approach - 1D case

- Every PDF supported in $[0,1]$ can be uniquely represented over its support by an infinite Fourier series,

$$f_X(x) = \sum_{k=-\infty}^{\infty} \Phi_X[k] e^{-j2\pi kx}, \quad \Phi_X[k] = \Phi_X(\nu)\big|_{\nu=2\pi k}, \quad k \in \mathbf{Z}$$

A Characteristic Function approach - 1D case

- Every PDF supported in $[0,1]$ can be uniquely represented over its support by an infinite Fourier series,

$$f_X(x) = \sum_{k=-\infty}^{\infty} \Phi_X[k] e^{-j2\pi kx}, \quad \Phi_X[k] = \Phi_X(\nu)\big|_{\nu=2\pi k}, \quad k \in \mathbf{Z}$$

- If $f_X \in C^p$, then $|\Phi_X[k]| = \mathcal{O}(\frac{1}{1+|k|^p})$ → truncated series approximation

$$\tilde{f}_X(x) = \sum_{k=-K}^{K} \widehat{\Phi}_X[k] e^{-j2\pi kx}, \quad \widehat{\Phi}_X[k] = \frac{1}{M} \sum_{m=1}^{M} e^{j2\pi k x_m}, \quad k \in \mathbf{Z}$$

A Characteristic Function approach - 1D case

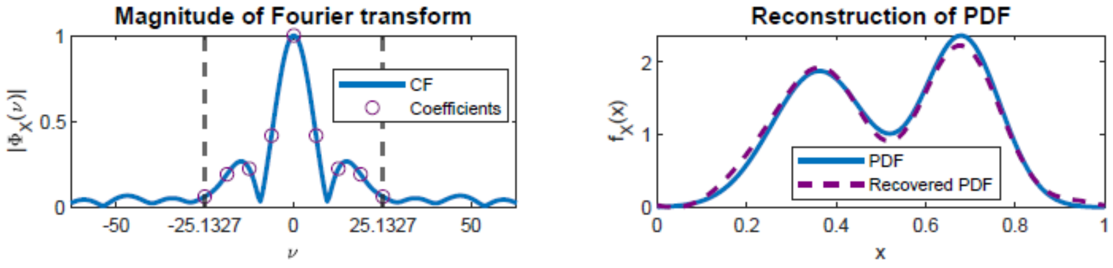
- Every PDF supported in $[0,1]$ can be uniquely represented over its support by an infinite Fourier series,

$$f_X(x) = \sum_{k=-\infty}^{\infty} \Phi_X[k] e^{-j2\pi kx}, \quad \Phi_X[k] = \Phi_X(\nu)\big|_{\nu=2\pi k}, \quad k \in \mathbf{Z}$$

- If $f_X \in C^p$, then $|\Phi_X[k]| = \mathcal{O}(\frac{1}{1+|k|^p})$ → truncated series approximation

$$\tilde{f}_X(x) = \sum_{k=-K}^{K} \widehat{\Phi}_X[k] e^{-j2\pi kx}, \quad \widehat{\Phi}_X[k] = \frac{1}{M}\sum_{m=1}^{M} e^{j2\pi k x_m}, \quad k \in \mathbf{Z}$$

- By Parseval's Theorem → $\|f - \tilde{f}\|_2^2 = \sum_{|k|>K} |\Phi_X[k]|^2$
  - Error is controllable by the smoothing parameter *K*



A Characteristic Function approach – The multivariate case

- Given a random vector $\boldsymbol{X} := [X_1, \ldots, X_N]^T$, the joint or multivariate characteristic function of $\boldsymbol{X}$ is a function $\Phi_{\boldsymbol{X}} : \mathbf{R}^N \to \mathbf{C}$ defined as

$$\boldsymbol{\Phi}_{\boldsymbol{X}}(\boldsymbol{\nu}) = E\left[e^{j\boldsymbol{\nu}^T\boldsymbol{X}}\right], \ \boldsymbol{\nu} := [\nu_1, \ldots, \nu_N]^T$$

- For any given *v*, given a set of realizations $\{\mathbf{x}_m\}_{m=1}^M$, we can estimate $\Phi_{\boldsymbol{X}}$, using a sample average

$$\widehat{\Phi}_{\boldsymbol{X}}(\boldsymbol{\nu}) = \frac{1}{M}\sum_{m=1}^{M} e^{j\boldsymbol{\nu}^T\mathbf{x}_m},$$

- The corresponding PDF can be uniquely recovered via the multidimensional inverse Fourier transform

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{(2\pi)^N}\int_{\mathbf{R}^N} \Phi_{\boldsymbol{X}}(\boldsymbol{\nu})e^{-j\boldsymbol{\nu}^T\boldsymbol{x}}d\boldsymbol{\nu}.$$

A Characteristic Function approach – The multivariate case

- Every PDF supported in $S_{\boldsymbol{X}} = [0,1]^N$ can be represented by a multivariate Fourier series,

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \sum_{k_1=-\infty}^{\infty} \cdots \sum_{k_N=-\infty}^{\infty} \Phi_{\boldsymbol{X}}[\boldsymbol{k}] e^{-j2\pi \mathbf{k}^T \mathbf{x}},$$

$$\text{where } \Phi_{\boldsymbol{X}}[\mathbf{k}] = \Phi_{\boldsymbol{X}}(\boldsymbol{\nu})\big|_{\boldsymbol{\nu}=2\pi\mathbf{k}}, \mathbf{k} = [k_1, \ldots, k_N]^T$$

A Characteristic Function approach – The multivariate case

- Every PDF supported in $S_{\boldsymbol{X}} = [0,1]^N$ can be represented by a multivariate Fourier series,

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \sum_{k_1=-\infty}^{\infty} \cdots \sum_{k_N=-\infty}^{\infty} \Phi_{\boldsymbol{X}}[\boldsymbol{k}] e^{-j2\pi \mathbf{k}^T \mathbf{x}},$$

$$\text{where } \Phi_{\boldsymbol{X}}[\mathbf{k}] = \Phi_{\boldsymbol{X}}(\boldsymbol{\nu})\big|_{\boldsymbol{\nu}=2\pi\mathbf{k}}, \mathbf{k} = [k_1, \ldots, k_N]^T$$

- If $f_X \in C^p$, then $|\Phi_{\boldsymbol{X}}[\mathbf{k}]| = \mathcal{O}\left(\frac{1}{1+\|\mathbf{k}\|_2^p}\right)$ $\longrightarrow$ truncated Fourier series approximation

$$\tilde{f}_{\boldsymbol{X}}(\mathbf{x}) = \sum_{k=-K_1}^{K_1} \cdots \sum_{k_N=-K_N}^{K_N} \Phi_{\boldsymbol{X}}[\mathbf{k}] e^{-j2\pi \mathbf{k}^T \mathbf{x}}$$

  - Known approximation error results by Mason 1980, Handscomb 2014 of the truncated series with absolute cutoffs $\{K_n\}_{n=1}^N$

A Characteristic Function approach – The multivariate case

- Every PDF supported in $S_{\boldsymbol{X}} = [0,1]^N$ can be represented by a multivariate Fourier series,

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \sum_{k_1=-\infty}^{\infty} \cdots \sum_{k_N=-\infty}^{\infty} \Phi_{\boldsymbol{X}}[\boldsymbol{k}] e^{-j2\pi \mathbf{k}^T \mathbf{x}},$$

$$\text{where } \Phi_{\boldsymbol{X}}[\mathbf{k}] = \Phi_{\boldsymbol{X}}(\boldsymbol{\nu})\big|_{\boldsymbol{\nu}=2\pi\mathbf{k}}, \mathbf{k} = [k_1, \ldots, k_N]^T$$

- If $f_X \in C^p$, then $|\Phi_{\boldsymbol{X}}[\mathbf{k}]| = \mathcal{O}\left(\frac{1}{1+\|\mathbf{k}\|_2^p}\right) \longrightarrow$ truncated Fourier series approximation

$$\tilde{f}_{\boldsymbol{X}}(\mathbf{x}) = \sum_{k=-K_1}^{K_1} \cdots \sum_{k_N=-K_N}^{K_N} \Phi_{\boldsymbol{X}}[\mathbf{k}] e^{-j2\pi \mathbf{k}^T \mathbf{x}}$$

  - Known approximation error results by Mason 1980, Handscomb 2014 of the truncated series with absolute cutoffs $\{K_n\}_{n=1}^N$

  - The smoother the underlying PDF, the faster its Fourier coefficients and the approximation error tends to zero

A low-rank Characteristic Function approach

- The truncated Fourier coefficients can be naturally represented by an *N*-way tensor

$$\underline{\mathbf{\Phi}}(k_1, \ldots, k_N) = \Phi_{\boldsymbol{X}}[\mathbf{k}]$$

- The number of parameters $(2K_1 + 1) \times \cdots \times (2K_N + 1)$, grows exponentially with *N*

A low-rank Characteristic Function approach

- The truncated Fourier coefficients can be naturally represented by an *N*-way tensor

$$\underline{\mathbf{\Phi}}(k_1, \ldots, k_N) = \Phi_{\boldsymbol{X}}[\mathbf{k}]$$

- The number of parameters $(2K_1 + 1) \times \cdots \times (2K_N + 1)$, grows exponentially with *N*

- Focus on the **principal components** of the resulting tensor -- i.e., introducing a rank-*F* parametrization of $\underline{\mathbf{\Phi}}$

$$\tilde{f}_{\boldsymbol{X}}(\mathbf{x}) = \sum_{k_1=-K}^{K} \cdots \sum_{k_N=-K}^{K} \sum_{h=1}^{F} p_H(h) \prod_{n=1}^{N} \Phi_{X_n|H=h}[k_n] e^{-j2\pi k_n x_n}$$

  - Reduction of parameters from order of $K_1 \times \cdots \times K_N$ to order of $(K_1 + \cdots + K_N)F$
  - Further denoise the naive sample average estimates

A low-rank Characteristic Function approach

- The truncated Fourier coefficients can be naturally represented by an *N*-way tensor

$$\underline{\mathbf{\Phi}}(k_1, \ldots, k_N) = \Phi_{\boldsymbol{X}}[\mathbf{k}]$$

- The number of parameters $(2K_1+1) \times \cdots \times (2K_N+1)$, grows exponentially with *N*

- Focus on the **principal components** of the resulting tensor -- i.e., introducing a rank-*F* parametrization of $\underline{\mathbf{\Phi}}$

$$\tilde{f}_{\boldsymbol{X}}(\mathbf{x}) = \sum_{k_1=-K}^{K} \cdots \sum_{k_N=-K}^{K} \sum_{h=1}^{F} p_H(h) \prod_{n=1}^{N} \Phi_{X_n|H=h}[k_n] e^{-j2\pi k_n x_n}$$

  - Reduction of parameters from order of $K_1 \times \cdots \times K_N$ to order of $(K_1 + \cdots + K_N)F$
  - Further denoise the naive sample average estimates

- Considering for brevity $K = K_1 = \cdots = K_n$, by linearity and separability of the multidimensional Fourier transformation,

$$\tilde{f}_{\boldsymbol{X}}(\mathbf{x}) = \sum_{h=1}^{F} p_H(h) \prod_{n=1}^{N} \sum_{k_n=-K}^{K} \Phi_{X_n|H=h}[k_n]\ e^{-j2\pi k_n x_n}$$

A low-rank Characteristic Function approach – Interpretation

UNIVERSITY of VIRGINIA

Mixture of product distributions - Latent variable naive Bayes interpretation

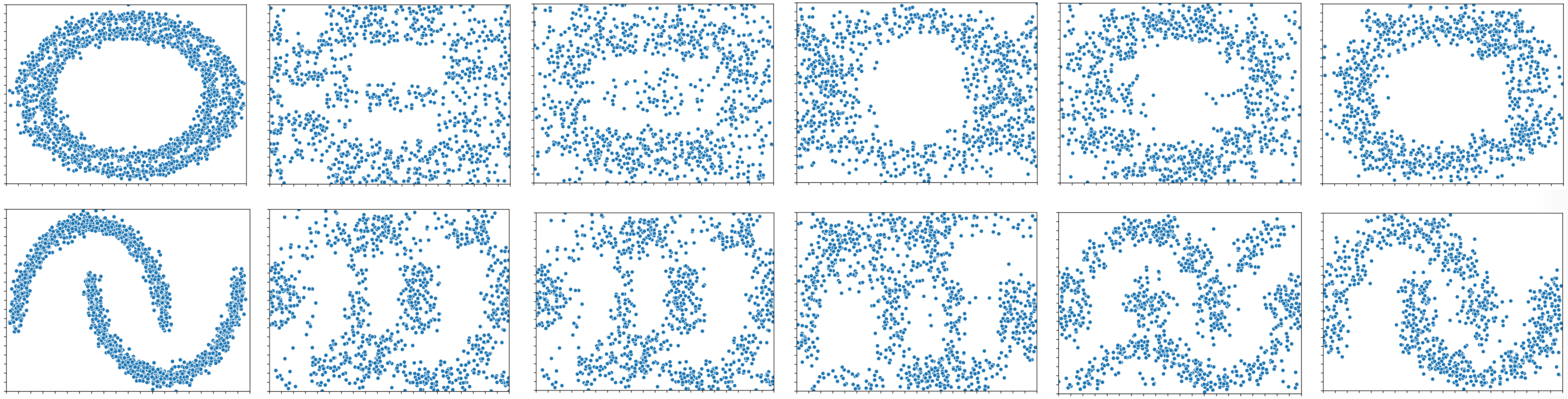
Truncating the multidimensional Fourier series of any compactly supported random vector is equivalent to approximating the corresponding multivariate density by a finite mixture of separable densities

$$\begin{aligned}\tilde{f}_{\mathbf{X}}(\mathbf{x}) &= \sum_{h=1}^{F} p_H(h) \prod_{n=1}^{N} \sum_{k_n=-K}^{K} \Phi_{X_n|H=h}[k_n]\; e^{-j2\pi k_n x_n} \\ &= \sum_{h=1}^{F} p_H(h) \prod_{n=1}^{N} f_{X_n|H}(x_n|h).\end{aligned}$$

- Yields generative model of the sought density, from which it is very easy to sample from.
- Easy marginalization.
- Easy to compute conditional densities.
- Easy to impute.

A low-rank Characteristic Function approach

- **The number of coefficients K** controls the desired smoothness of the probability model
- **The rank F** controls the expressivity of the probability model



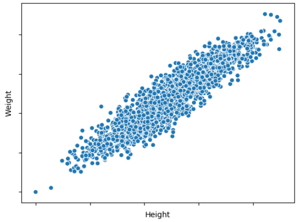
$F=2$ $F=4$ $F=6$ $F=8$ $F=10$

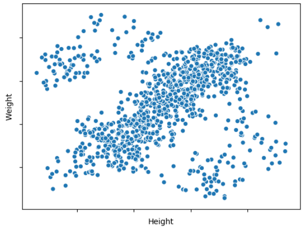
- Generating synthetic samples from our model
- For fixed K, $K=11$, given $M=2000$ samples from toy Circles and Moons 2D datasets

A low-rank Characteristic Function approach

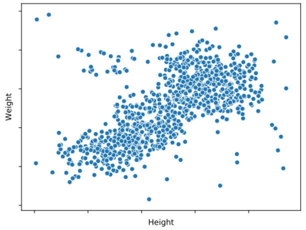
- **The number of coefficients *K*** controls the desired smoothness of the probability model
- **The rank *F*** controls the expressivity of the probability model

- ***F* = 8**

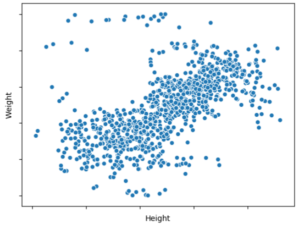




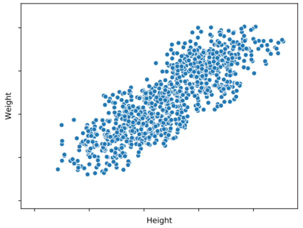
$k = 4$



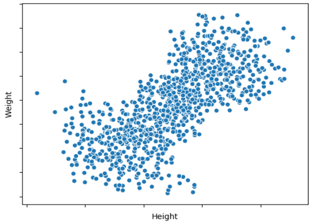
$k = 6$



$k = 8$

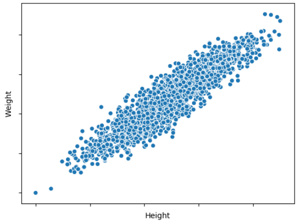


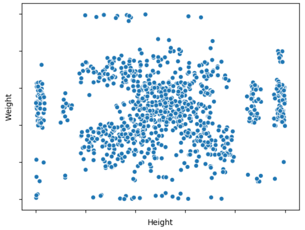
$k = 10$



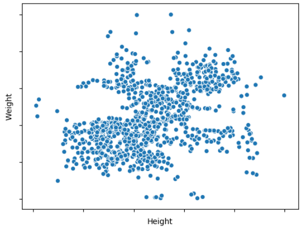
$k = 12$

- ***K* = 10**

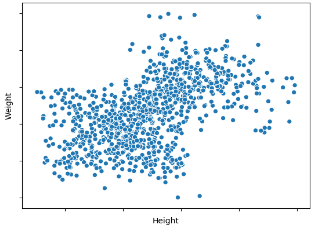




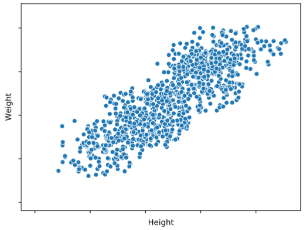
$F = 2$



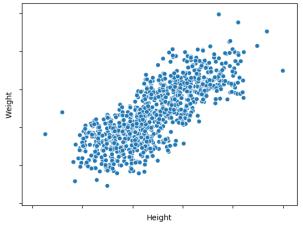
$F = 4$



$F = 6$



$F = 8$



$F = 10$

- Generating synthetic samples from our model, given samples from Weight-Height dataset

A Characteristic Function approach – Uniqueness

- Conversely, assuming that the sought joint PDF is a finite mixture of separable densities

$$
\begin{aligned}
\Phi_{\boldsymbol{X}}(\boldsymbol{\nu}) &= E\left[e^{j\boldsymbol{\nu}^T\boldsymbol{X}}\right] \\
&= E_H\left[E_{\boldsymbol{X}|H}\left[e^{j\nu_1 X_1}\cdots e^{j\nu_N X_N}\right]\right] \\
&= E_H\left[\Phi_{X_1|H}(\nu_1|H)\cdots\Phi_{X_n|H}(\nu_N|H)\right] \\
&= \sum_{h=1}^{F} p_H(h)\prod_{n=1}^{N}\Phi_{X_n|H}(\nu_n|h).
\end{aligned}
$$

Uniqueness of the Characteristic Tensor CPD

A compactly supported multivariate mixture of separable densities is identifiable from (samples of) its characteristic function, under mild conditions

Proposed approach

1. Estimate $\underline{\mathbf{\Phi}}[\mathbf{k}] = \frac{1}{M}\sum_{m=1}^{M} e^{j2\pi\mathbf{k}^T\mathbf{x}_m}$
2. Fit a low-rank model $\underline{\mathbf{\Phi}}[\mathbf{k}] \approx \sum_{h=1}^{F} p_H(h)\prod_{n=1}^{N}\Phi_{X_n|H=h}[k_n]$
3. Invert using $f_{\mathbf{X}}(\mathbf{x}) = \sum_{h=1}^{F} p_H(h)\prod_{n=1}^{N} f_{X_n|H}(x_n|h), \quad f_{X_n|H}(x_n|h) = \sum_{k_n=-K}^{K}\Phi_{X_n|H=h}[k_n]e^{-j2\pi k_n x_n}$

- **Issues**:
  1. Fix scaling/counter-scaling freedom in $p_H(\cdot)$ ⟶ constraints

$$
\begin{aligned}
\min \quad & \|\underline{\mathbf{\Phi}} - [\![\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N]\!]\|_F^2 \\
\text{subject to} \quad & \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{1}^T\boldsymbol{\lambda} = 1, \\
& \boldsymbol{A}_n(K+1,:) = \mathbf{1}^T,\ n = 1\ldots N
\end{aligned}
$$

  2. Allocating memory for the truncated characteristic tensor is a challenge!

Proposed approach

- Allocating memory for the truncated characteristic tensor is a challenge!
  - Model the characteristic tensors of subsets of variables (triples) $\underline{\boldsymbol{\Phi}}_{ij\ell}$
  - Key observation: lower-order marginals → also a constrained complex CPD model

$$\underline{\boldsymbol{\Phi}}(k_1, \ldots, k_{n'} = 0, \ldots, k_N) = \sum_{h=1}^{F} \prod_{\substack{n=1 \\ n \neq n'}}^{N} \Phi_{X_n|H}[k_n] \underbrace{\Phi_{X_{n'}|H}[0]}_{=1}$$

$$= \sum_{h=1}^{F} \prod_{\substack{n=1 \\ n \neq n'}}^{N} \Phi_{X_n|H}[k_n]$$

  - Jointly decompose in a coupled fashion, synthesize the full characteristic tensor
    - **Significant computational and memory reduction**
    - **Allows us to work with incomplete realizations**

Proposed approach

- Allocating memory for the truncated characteristic tensor is a challenge!
  - Model the characteristic tensors of subsets of variables (triples) $\underline{\mathbf{\Phi}}_{ij\ell}$
  - Key observation: lower-order marginals $\longrightarrow$ also a constrained complex CPD model
  - Jointly decompose in a coupled fashion, synthesize the full characteristic tensor
    - ➢ **Significant computational and memory reduction**
    - ➢ **Allows us to work with incomplete realizations**

  - We propose solving the following optimization problem:

$$\min_{\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N} \sum_{i} \sum_{j>i} \sum_{\ell>j} \left\| \underline{\mathbf{\Phi}}_{ij\ell} - [\![\boldsymbol{\lambda}, \mathbf{A}_i, \mathbf{A}_j, \mathbf{A}_\ell]\!] \right\|_F^2$$

$$\text{subject to} \quad \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{1}^T \boldsymbol{\lambda} = 1,$$

$$\mathbf{A}_n(K+1, :) = \mathbf{1}^T, \ n = 1, \ldots, N.$$

Instance of coupled tensor factorization

Algorithmic approach

- We propose solving the following optimization problem:

$$\min_{\boldsymbol{\lambda},\mathbf{A}_1,\ldots,\mathbf{A}_N} \sum_{i}\sum_{j>i}\sum_{\ell>j} \|\underline{\boldsymbol{\Phi}}_{ij\ell} - [\![\boldsymbol{\lambda},\mathbf{A}_i,\mathbf{A}_j,\mathbf{A}_\ell]\!]\|_F^2$$
$$\text{subject to} \quad \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{1}^T\boldsymbol{\lambda} = 1,$$
$$\mathbf{A}_n(K+1,:) = \mathbf{1}^T,\ n = 1,\ldots,N.$$

- Alternating optimization $\longrightarrow$ Cyclically update variables $\mathbf{A}_n, \boldsymbol{\lambda}$

- The optimization problem with respect to $\mathbf{A}_i$ becomes

$$\min_{\mathbf{A}_i} \sum_{j\neq i}\sum_{\ell\neq i,\ell>j} \|\underline{\boldsymbol{\Phi}}_{ij\ell}^{(1)} - (\mathbf{A}_\ell \odot \mathbf{A}_j)\mathrm{diag}(\boldsymbol{\lambda})\mathbf{A}_i^T\|_F^2$$
$$\text{subject to} \quad \mathbf{A}_i(K+1,:) = \mathbf{1}^T$$

Unconstrained complex least squares problem

Algorithmic approach

- We propose solving the following optimization problem:

$$\min_{\boldsymbol{\lambda},\mathbf{A}_1,\ldots,\mathbf{A}_N} \sum_i \sum_{j>i} \sum_{\ell>j} \|\underline{\boldsymbol{\Phi}}_{ij\ell} - [\![\boldsymbol{\lambda}, \mathbf{A}_i, \mathbf{A}_j, \mathbf{A}_\ell]\!]\|_F^2$$
$$\text{subject to} \quad \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{1}^T\boldsymbol{\lambda} = 1,$$
$$\mathbf{A}_n(K+1,:) = \mathbf{1}^T, \; n = 1, \ldots, N.$$

- Alternating optimization $\longrightarrow$ Cyclically update variables $\mathbf{A}_n, \boldsymbol{\lambda}$

- The optimization problem with respect to $\mathbf{A}_i$ becomes

$$\min_{\mathbf{A}_i} \sum_{j\neq i} \sum_{\ell\neq i, \ell>j} \|\underline{\boldsymbol{\Phi}}_{ij\ell}^{(1)} - (\mathbf{A}_\ell \odot \mathbf{A}_j)\text{diag}(\boldsymbol{\lambda})\mathbf{A}_i^T\|_F^2$$
$$\text{subject to} \quad \mathbf{A}_i(K+1,:) = \mathbf{1}^T$$

Unconstrained complex least squares problem

- The optimization problem with respect to $\boldsymbol{\lambda}$ becomes

$$\min_{\boldsymbol{\lambda}} \sum_i \sum_{j>i} \sum_{\ell>j} \|\text{vec}(\underline{\boldsymbol{\Phi}}_{ij\ell}) - (\mathbf{A}_\ell \odot \mathbf{A}_j \odot \mathbf{A}_i)\boldsymbol{\lambda}\|_F^2$$
$$\text{subject to} \quad \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{1}^T\boldsymbol{\lambda} = 1.$$

Least squares problem with probability simplex constraints
ADMM

Algorithmic approach

- We propose solving the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N} \quad & \sum_i \sum_{j>i} \sum_{\ell>j} \| \underline{\boldsymbol{\Phi}}_{ij\ell} - [\![\boldsymbol{\lambda}, \mathbf{A}_i, \mathbf{A}_j, \mathbf{A}_\ell]\!] \|_F^2 \\ \text{subject to} \quad & \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{1}^T \boldsymbol{\lambda} = 1, \\ & \mathbf{A}_n(K+1, :) = \mathbf{1}^T, \ n = 1, \ldots, N. \end{aligned}$$

- Alternating optimization → Cyclically update variables $\mathbf{A}_n, \boldsymbol{\lambda}$

- The optimization problem with respect to $\mathbf{A}_i$ becomes

$$\begin{aligned} \min_{\mathbf{A}_i} \quad & \sum_{j \neq i} \sum_{\ell \neq i, \ell > j} \| \underline{\boldsymbol{\Phi}}_{ij\ell}^{(1)} - (\mathbf{A}_\ell \odot \mathbf{A}_j) \mathrm{diag}(\boldsymbol{\lambda}) \mathbf{A}_i^T \|_F^2 \\ \text{subject to} \quad & \mathbf{A}_i(K+1, :) = \mathbf{1}^T \end{aligned}$$

Unconstrained complex least squares problem

- The optimization problem with respect to $\boldsymbol{\lambda}$ becomes

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & \sum_i \sum_{j>i} \sum_{\ell>j} \| \mathrm{vec}(\underline{\boldsymbol{\Phi}}_{ij\ell}) - (\mathbf{A}_\ell \odot \mathbf{A}_j \odot \mathbf{A}_i) \boldsymbol{\lambda} \|_F^2 \\ \text{subject to} \quad & \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{1}^T \boldsymbol{\lambda} = 1. \end{aligned}$$

Least squares problem with probability simplex constraints
ADMM

- The corresponding joint PDF model can be recovered ***at any point*** as

$$f_{\mathbf{X}}(\mathbf{x}) = \sum_{h=1}^{F} \boldsymbol{\lambda}(h) \prod_{n=1}^{N} \sum_{k_n=-K}^{K} \mathbf{A}_n(k_n, h) e^{-j2\pi k_n x_n}$$

Experiments

- Performance evaluation using 7 UCI high-dimensional datasets
  - Average log-likelihood of unseen data samples
  - Regression tasks
  - Image sampling

- 10 Monte Carlo simulations
- 80% training, 20% test (5 - fold cross - validation for parameter selection)
  - Parameters: Tensor rank, smoothing parameter

- Standard baselines
  - 2 Classic literature (GMMs, KDE)
  - 2 State of the art Neural Density Estimators (RNADE, MAF)

Results

- **Average log-likelihood** of unseen data samples
- Our method achieves a higher average test sample log likelihood in almost all datasets!

Data set	MoG	KDE	RNADE	MAF	LRCF-DE
Red wine	11.9 ± 0.29	9.9 ± 0.16	14.41 ± 0.16	$\mathbf{15.2 \pm 0.09}$	$\mathbf{16.4 \pm 0.67}$
White wine	16.1 ± 1.48	14.8 ± 0.12	17.1 ± 0.26	$\mathbf{17.3 \pm 0.20}$	$\mathbf{18.4 \pm 0.17}$
F-O.TP	125.4 ± 7.79	103.05 ± 0.84	$\mathbf{152.48 \pm 5.62}$	149.6 ± 8.32	$\mathbf{154.34 \pm 8.43}$
PCB	152.9 ± 3.88	147.6 ± 1.63	171.7 ± 2.75	$\mathbf{179.6 \pm 1.62}$	$\mathbf{194.4 \pm 2.43}$
Superconductivty	134.7 ± 3.47	127.2 ± 2.82	140.2 ± 1.03	$\mathbf{143.5 \pm 1.32}$	$\mathbf{146.1 \pm 2.31}$
Corel Images	211.7 ± 1.04	201.4 ± 1.18	$\mathbf{223.6 \pm 0.88}$	218.2 ± 1.35	$\mathbf{222.6 \pm 1.25}$
Gas Sensor	310.3 ± 3.47	296.48 ± 1.62	$\mathbf{316.3 \pm 3.57}$	315.4 ± 1.458	$\mathbf{316.6 \pm 2.35}$

Average test-set log-likelihood per datapoint for 5 different models on UCI datasets; **higher is better**.

Data set	N	M
Red wine	11	1599
White wine	11	4898
First-order theorem proving (F-O.TP)	51	6118
Polish companies bankruptcy (PCB)	64	10503
Superconductivty	81	21263
Corel Images	89	68040
Gas Sensor Array Drift (Gas Sensor)	128	13910

Results

- **Regression**: our joint PDF model enables easy computation of any marginal or conditional density of subsets of variables
- Estimate the output using the **conditional expectation**
  - Report the Mean Absolute Error

Data set	MoG	KDE	RNADE	MAF	LRCF-DE
Red wine	1.28	1.13	0.66	0.63	0.56
White wine	1.79	1.31	0.80	0.75	0.59
F-O.TP	1.86	1.46	0.63	0.52	0.48
PCB	5.6	7.73	4.43	4.52	3.85
Superconductivty	18.56	19.96	16.46	16.38	16.53
Corel Images	0.53	0.93	0.27	0.27	0.28
Gas Sensor	29.7	35.3	26.8	26.2	26.7

Data set	N	M
Red wine	11	1599
White wine	11	4898
First-order theorem proving (F-O.TP)	51	6118
Polish companies bankruptcy (PCB)	64	10503
Superconductivty	81	21263
Corel Images	89	68040
Gas Sensor Array Drift (Gas Sensor)	128	13910

Data set	LRCF-DE	MAF
Red wine	0.82	0.91
White wine	0.93	0.97
First-order theorem proving (F-O.TP)	0.69	0.72
Polish companies bankruptcy (PCB)	4.97	5.46
Superconductivty	20.84	20.72
Corel Images	1.36	1.59
Gas Sensor Array Drift (Gas Sensor)	25.7	26.1

Multi-output regression: Predicting the last two random variables

- Our method outperforms the baselines in almost all datasets and performs comparable to the winning method in the remaining ones

Results

- Image synthesis: Our generative model affords easy sampling
- USPS dataset $N = 256$: Fix the tensor rank to $F = 8$, $K = 15$ and draw 8 random samples of each digit (class)

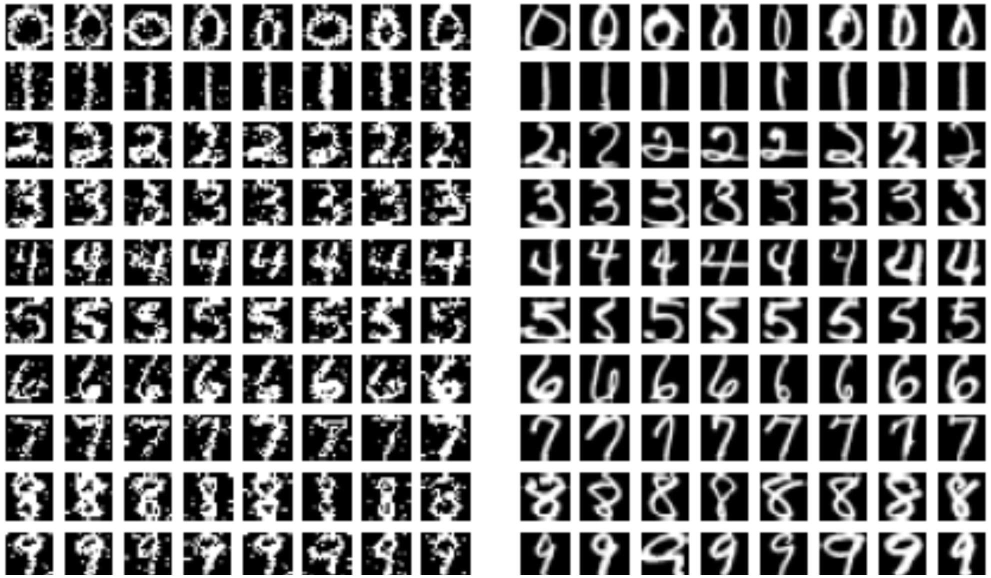


Figure: Class-conditional synthetic vs real samples from the USPS dataset.

	0	1	2	3	4	5	6	7	8	9	Total
Samples	1553	1269	929	824	852	716	834	792	708	821	9298

Nonparametric Multivariate Density Estimation: A Low-Rank Characteristic Function Approach

Recap

- We revisited the classic problem of nonparametric density estimation from a fresh perspective
  - Through the lens of complex Fourier series approximation
  - Tensor modeling

- We showed that
  - Any compactly supported density can be approximated by a finite characteristic tensor of leading complex Fourier coefficients, whose size depends on the smoothness of the density
  - We posed density estimation as a constrained (coupled) tensor factorization problem and proposed a Block Coordinate Descent algorithm
  - Under certain conditions enables learning the true data-generating distribution

UNIVERSITY of VIRGINIA

THANK YOU!

Questions?