Tensor Modeling of High-dimensional Distributions and its Applications in Machine Learning

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF VIRGINIA
BY

Magda Amiridi

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Prof. Nicholas D. Sidiropoulos, Advisor

December, 2022

Acknowledgments

First, I would like to thank my advisor, Professor Nikolaos D. Sidiropoulos for his valuable guidance during my Ph.D. program. Pursuing this endeavor would not have been possible without his consistent support. I am grateful to him for his impact on my research and my career development. I'm also thankful to Professors Scott Acton, Cong Shen, Jundong Li, Anil Vullikanti and Zongli Lin for serving on my Ph.D. committee, providing valuable feedback on my thesis, and influencing me greatly through their research work and lectures. Additionally, I would also like to thank my undergraduate advisor Professor George Karystinos and other Professors from Technical University of Crete for encouraging me to explore my potential and pursue a Ph.D. degree in the US.

I want to thank all members from the Apple Health-AI team, including my internship mentor Gregory Darnell, with whom I had the pleasure to have fruitful interactions and a unique learning experience. I am also grateful to my second internship mentor Cheng Qian, who gave me the opportunity to apply methods developed during my Ph.D. to solve an important practical problem in clinical trial optimization, using unique data from IQVIA.

I want to thank all my fellow labmates (past and present) in the Signal and Tensor Analytics Research (STAR) group, including Nikos Kargas, Aritra Konar, Charilaos Kanatsoulis, Paris Karakasis, Cheng Qian, Faisal Almutairi, Mikael Sorensen, Mohamed Salah, Bo Yang, Ahmed Zamzam, Kejun Huang, and Xiao Fu for the insightful discussions during our meetings.

My deepest gratitude goes to all the people I met in Virginia for being supportive during the toughest time of my life; I would not have accomplished my goals without them. I am beyond thankful to Giannis and his family for making Charlottesville a home away from home. Furthermore, I feel very lucky to have met my close friends Nikos, Charilaos, and Aritra, that became an essential part of my support system.

I would also like to thank my family, and especially my parents, for their unconditional love

and valuable advice. Their resilience has made me believe I can overcome any challenge. I thank them for their endless support. Last but not least, I would like to thank my source of happiness and joy, my husband John. John has been an incredible partner and friend throughout my journey. This work could not have been accomplished without his and his family's love and support.

Magda Amiridi, December 2022.

Dedication

To my family and friends

Abstract

Effective non-parametric modeling of probability distributions is a central problem in statistics and machine learning. The ubiquity of large amounts of data generated in real-world systems has created unprecedented opportunities to apply such models to critical machine learning tasks for making quick and informed decisions. However, inherent difficulties associated with modern data such as high-dimensionality, incomplete data (vector realizations with missing entries), and complex high-order interactions between features, pose a key challenge and necessitate expressive methods that are efficient both computationally and in memory cost. The main contribution of this dissertation is to introduce principled methods for non-parametric estimation of the data-generating distribution function using low-rank tensor models and show their potential in various real-world applications.

The first part of the dissertation focuses on non-parametric density estimation through the lens of complex Fourier series approximation and low-rank tensor modeling (a low-rank characteristic function approach). We show that any smooth compactly supported multivariate probability density function (PDF) can be approximated by a finite tensor model and under relatively mild assumptions, the proposed model can approximate any high dimensional PDF with approximation guarantees. We also show that, by virtue of uniqueness of low-rank tensor decomposition, assuming low-rank in the Fourier domain, the underlying multivariate density is identifiable. A promising extension of this work, suitable for even higher-dimensional data such as images, considers a joint dimensionality reduction and density estimation framework. The dimensionality reduction component is carried out via deep autoencoders and the latent density is modeled using a non-parametric low-rank tensor model in the Fourier domain.

However, for some applications such as problems involving hybrid random variables (continuous and discrete) or tasks requiring multivariate integration of the high-dimensional PDF (e.g., corresponding to finite or semi-infinite “box” events), modeling the joint cumulative distribution function (CDF) seems more appropriate. The second part of the dissertation introduces a novel parametrization of grid-sampled multivariate CDFs using tensor factorization in the data or in the copula domain. Furthermore, the abundance of time-series data in machine learning systems such as sensor signals collected from wrist-worn devices, brings new demands in density estimation, as it requires estimating time-varying and high-dimensional distributions, capable of accurately modeling both inter- and intra-series dependencies. Our proposed model combines the temporal modeling power of recurrent neural networks (RNNs) with the parsimony of principled low-rank density models, to propose a versatile time series high-dimensional distribution model. As a practical demonstration of the abilities of our models, we applied them to various machine learning tasks, showcasing a highly competitive performance with respect to the state-of-the-art. In particular, we apply the proposed low-rank CDF method to predict enrollment rate (ER) in clinical trials, and we demonstrate significant improvements over previous methods used in the health informatics industry. Accurate ER prediction is key for successful and timely clinical trials, and has potential for strong societal impact in human health.

Keywords: probabilistic modeling, low-rank tensor models, identifiable models, CPD decomposition, probabilistic inference, density estimation, generative models.

0.1 Intellectual Merits

This dissertation reports basic research on a fundamental problem in statistical learning: effective non-parametric modeling and estimation of high-dimensional joint distributions. By providing new theoretical contributions, algorithms and applications, our work aspires to theoretically and experimentally showcase how valuable insights from classical statistical learning and tensor factorization tools can lead to solutions that remarkably advance the state-of-the-art performance for fundamental problems in machine learning. The proposed probabilistic frameworks offer a wide range of desired benefits such as model uniqueness, easy model marginalization, prediction of any variable from (a subset of) the other variables, and sampling, which can potentially benefit many research problems in applied statistics, data mining, and machine learning where density estimation can be used as a building block.

0.2 Broader Impact

The proposed research is expected to yield exciting new theoretical and methodological insights that can be used in a broad variety of applications, including those that can have strong societal impact – e.g., healthcare applications to better diagnose, forecast, and otherwise characterise the health of individuals. Skills, tools, and knowledge gained during my Ph.D. studies have been leveraged throughout my first internship with Apple Health AI, where we developed a novel method for modeling complex time-series such as vital signs and illustrated promising forecasting performance on the Apple Heart and Movement Study. During my second internship with IQVIA, we applied our CDF method developed in [1] to solve an important practical problem in clinical trial optimization. Our experimental results demonstrated an improved performance of the proposed method in the enrollment rate prediction task over the best baselines by up to 12.2% in mean squared error on a large-scale real-world clinical trial dataset from IQVIA, while also offering direct means of quantifying uncertainty in the predictions based on the fitted model.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Goal, Motivation, and Challenges.

Probabilistic modeling is a cornerstone of artificial intelligence and machine learning as massive amounts of information are generated in data-intensive fields such as healthcare, social networks, finance, sales, marketing and many more, increasing the demand for tools that can store, detect summarize the multivariate structure of high-dimensional data, while also allowing the inherent uncertainty in these systems to be represented. Accurate modeling of probability distributions based on observed data samples lies at the core of multivariate data analysis and unsupervised learning. The joint probability distribution function encodes a complete statistical model of the data generative process and thus it can be used as a basic building block for modern tasks in intelligent systems that involve knowledge of the underlying structure of data.

Given an accurate and tractable estimate of the joint distribution function, one can generate new samples, calculate the conditional and marginal probability distributions of arbitrary subsets of variables, their corresponding expectations, or other quantities of interest on never-before-seen data. One can naturally solve a variety of types of inference problem, such as classification, regression, missing value imputation, and many other predictive tasks. This can facilitate a wide range of applications ranging from sampling realistic "synthetic" images (generative models) to anomaly detection (predicting acts of terrorism, detecting fraud, etc.), where we can use an accurate joint distribution model to label query points as anomalies if they reside in low density areas. Another potential area of applications is healthcare, where such models could be used to predict a patient's response to specific therapies based on cancer gene expression data or generate hypothetical gene expression profiles under various types of molecular and genetic perturbation. To list a few recent applications, such models have demonstrated success in generating high-fidelity images [2], [3], realistic speech synthesis [4], [5]; semi-supervised learning [6]; reinforcement learning [7]; and detecting adversarial data [8].

Methods for estimating statistical distributions can be broadly classified as either parametric or non-parametric. Parametric methods such as the Gaussian Mixture Model (GMM) [9], use a specific functional form with a fixed number of tunable parameters but they become inaccurate when the data generating process differs significantly from the assumed parametric model. Non-parametric models such as the kernel density estimator (KDE) [10, 11], histogram density estimation (HDE), and Orthogonal Series Density Estimation (OSDE) [12], [13], [14], are more unassuming and in that sense more flexible, but the flip-side is that they do not scale beyond a small number of variables (dimensions) due to the *curse of dimensionality* (CoD): the number of training data needed to obtain reasonable estimates blows up exponentially. However, real-world data (such as images, audio, and text) reside in a high dimensional and complex feature spaces while only a limited amount of observed data are directly available.

Recently, several neural network-based approaches for (both direct and indirect) distribution estimation have been proposed and demonstrated promising results in high-dimensional problems. Direct models include the so-called autoregressive models (AR) models [5,15], which decompose the distribution into a product of conditionals, and Normalizing flows (NFs) [16], which represent a density value though an invertible transformation of latent variables with known density. These methods do not construct a distribution model but rather serve for point-wise *evaluation* – they cannot impute more than very few missing elements in the input as grid search becomes combinatorial. Incomplete observations due to causes such as faulty sensors and corrupted data, present further distinct challenges to the training of these models. Such requirements call for statistical models that are flexible enough to represent a wide class of distributions from "imperfect" samples, and tractable and scalable at the same time. Indirect models, such as generative adversarial networks [17] (GANs) and do not allow for likelihood evaluation on held-out data. Furthermore, most of them do not address model identifiability, and can not guarantee the recovery of the true latent factors that generated the observed samples.

This thesis combines probability theory and tensor factorization tools to develop highly expressive and efficient statistical models, which are promising for high-impact real-world applications. In particular, the central goal of this project is to propose solutions for (non-parametric) statistical modeling of complex data – spanning high-dimensional continuous (*joint PDF*), discrete or categorical distributions (*PMF*), and distributions of (potentially) hybrid random variables (*CDF*). The proposed models are expressive enough to represent never-before-seen data, and tractable and (computationally and memory-wise) scalable at the same time (expressivity-tractability trade-off). The proposed frameworks offer a wide range of desired benefits such as model uniqueness, parsimony, easy model marginalization, prediction of any variable from (a subset of) the other variables, and fast sampling.

1.2 Thesis Outline

Our main building blocks are tensors and tensor decompositions. In Chapter 2, we review necessary background on tensors, we introduce the Canonical Polyadic Decomposition (CPD), a powerful tensor model, and discuss its identifiability properties. We also describe how CPD has been proposed for modeling finite-alphabet random vectors, a topic that was first pointed out in [18].

Although the first part of this dissertation also includes joint PMF modeling (e.g., a multivariate PMF model based on a coarse-fine hierarchical tensor partition [19]), we will mainly focus on the key novelty, which is tensors as universal PDF approximators. Before that, however, in Chapter 3 we introduce a novel information-theoretic feature selection method using tensor based joint PMF modeling [20]. In Chapters 4 and 5, we revisit the classic problem of non-parametric density estimation from a fresh perspective – through the lens of complex Fourier series approximation and tensor modeling, leading to a low-rank characteristic function approach. In Chapter 4, we show that any compactly supported density can be well-approximated by a finite *characteristic tensor* of leading complex Fourier coefficients as long as the coefficients decay sufficiently fast [21]. Chapter 5 builds on this foundation to develop a joint compression (nonlinear dimensionality reduction) and compressed density estimation framework [22]. Each approach has its own advantages. Although it does not provide a density estimate in the original space as the “baseline” method in this first part does, it offers additional flexibility and scalability.

The second part of this dissertation comprises the following two thrusts. In Chapter 6, we introduce a low-rank parametrization of grid-sampled multivariate CDFs using tensor factorization either in the data or in the copula domain, with applications in classification and missing data imputation [1]. In Chapter 7, we combine the temporal modeling power of recurrent neural networks (RNNs) with the parsimony of principled low-rank density models, to propose a versatile time series high-dimensional distribution model capable of capturing complex nonlinear dependencies between variables and time.

In Chapter 8 we present conclusions and discussion on future research directions.

1.3 Contributions

1.3.1 Chapter 3 — Information-theoretic Feature Selection via Tensor Decomposition and Submodularity

Feature selection by maximizing high-order mutual information between the selected feature vector and a target variable is the gold standard in terms of selecting the best subset of relevant features that maximizes the performance of prediction models. However, such an approach typically requires knowledge of the multivariate probability distribution of all features and the target, and involves a challenging combinatorial optimization problem. Recent work has shown that any joint Probability Mass Function (PMF) can be represented as a naive Bayes model, via Canonical Polyadic (tensor rank) Decomposition. In this paper, we introduce a low-rank tensor model of the joint PMF of all variables and indirect targeting as a way of mitigating complexity and maximizing the classification performance for a given number of features. Through low-rank modeling of the joint PMF, it is possible to circumvent the curse of dimensionality by learning 'principal components' of the joint distribution.

- By indirectly aiming to predict the latent variable of the naive Bayes model instead of the original target variable, it is possible to formulate the feature selection problem as maximization of a monotone submodular function subject to a cardinality constraint – which can be tackled using a greedy algorithm that comes with performance guarantees.
- Numerical experiments with several standard datasets suggest that the proposed approach compares favorably to the state-of-art for this important problem. The results of this chapter are reported in [20].

In chapter 4, we propose a novel approach for non-parametric *joint Probability Density Function* (PDF) estimation that builds upon tensor factorization tools [21].

1.3.2 Chapter 4 — Low-rank Characteristic Tensor Density Estimation Part I: Foundations

We show that any compactly supported continuous density can be approximated, within controllable error, by a finite *characteristic tensor* of leading complex Fourier coefficients, whose size depends on the smoothness of the density. This characteristic tensor can be naturally estimated via sample averaging from realizations of the random vector of interest. In order to circumvent this "curse of dimensionality" (CoD) and further denoise the naive sample averaging estimates, we introduce a low-rank model of the characteristic tensor, whose degrees of freedom (for fixed rank) grow linearly with the random vector dimension. Low-rank modeling significantly improves the density estimate especially for high-dimensional data and/or in the sample-starved regime. By virtue of uniqueness of low-rank tensor decomposition, under certain conditions, our method enables learning the true data-generating distribution.

- We show that a finite mixture model (approximately) follows from compactness of support and continuous differentiability (smoothness).
- Furthermore, assuming low-rank in the Fourier domain, a controllable approximation of the multivariate density is identifiable.
- We also show that a high dimensional joint PDF can be recovered by observing subsets (e.g., triples or quadruples) of variables, under certain conditions.
- The proposed model allows efficient and low-complexity inference, sampling, and density evaluation. We provide convincing experimental results on sampling, likelihood evaluation, and regression on toy, image, and many real datasets that are often used as benchmarks in our context. Our results corroborate the effectiveness of the proposed method even for datasets that have hundreds of variables. The results of this chapter are reported [21].

In chapter 5, we show that scaling to higher dimensions is possible under a joint dimensionality reduction and non-parametric density estimation framework, using a non-parametric low-rank tensor model in the Fourier domain to capture the underlying distribution of appropriate reduced-dimension representations.

1.3.3 Chapter 5 — Low-rank Characteristic Tensor Density Estimation Part II: Compression and Latent Density Estimation

Learning generative probabilistic models of complex data such as images is a core problem in machine learning, which presents significant challenges due to the *curse of dimensionality*. This paper, proposes a joint dimensionality reduction and non-parametric density estimation framework, using a novel estimator that can explicitly capture the underlying distribution of appropriate reduced-dimension representations of the input data. The idea is to jointly design a nonlinear dimensionality reducing auto-encoder to model the training data in terms of a parsimonious set of latent random variables, and learn a *canonical* low-rank tensor model of the joint distribution of the latent variables in the Fourier domain. The proposed latent density model is non-parametric and "universal", as opposed to the predefined prior that is *assumed* in variational auto-encoders. Joint optimization of the auto-encoder and the latent density estimator is pursued via a formulation which learns both by minimizing a combination of the negative log-likelihood in the latent domain and the auto-encoder reconstruction loss.

- Although it does not provide a density estimate in the original space as the "baseline" method in this first part does, this joint approach boosts the flexibility, scalability, and statistical performance in terms of prediction (regression/detection) accuracy and sampling fidelity.

- Instead of the coupled tensor factorization approach adopted in Part I, Part II tackles joint density estimation as a *hidden tensor factorization* problem using maximum likelihood learning of the latent distribution's parameters. The proposed model achieves very promising results on toy, tabular, and image datasets on regression tasks, sampling, and anomaly detection. The results of this chapter are reported in [22].

For some applications, such as tasks requiring estimates of (possibly semi-infinite) "box" probabilities, e.g. of type $Pr\{35 < \text{Age} \leq 45, \cdots, 55 < \text{Income} \leq 70\}$, or tasks involving mixed random variables, modeling the joint CDF is more appropriate, as it avoids the (potentially intractable) multi-dimensional integration of the joint PDF. In contrast to a PDF/PMF, a joint CDF (and its grid-sampled sketch) always exists, and allows modeling both discrete, continuous or hybrid (continuous and discrete) random variables.

1.3.4 Chapter 6 — Learning Multivariate CDFs and Copulas using Tensor Factorization

In Chapter 6, we aim to learn multivariate cumulative distribution functions (CDFs), as they can handle mixed random variables, allow efficient 'box' probability evaluation, and have the potential to overcome local sample scarcity owing to their cumulative nature. We show that *any* grid-sampled version of a joint CDF of mixed random variables admits a *universal* representation as a naive Bayes model via the Canonical Polyadic (tensor-rank) decomposition. By introducing a low-rank model, either directly in the raw data domain, or indirectly in a transformed (Copula) domain, the resulting model affords efficient sampling, closed form inference and uncertainty quantification, and comes with uniqueness guarantees under relatively mild conditions. We demonstrate the superior performance of the proposed model in several synthetic and real datasets and applications including regression, sampling and data imputation. Interestingly, our experiments with real data show that it is possible to obtain better density/mass estimates indirectly via a low-rank CDF model, than a low-rank PDF/PMF model. The results of this chapter are reported in [1]. This work is presented in chapter 4, where we show that:

- *Every* multivariate CDF evaluated on a predefined grid admits a compact representation via a latent variable naive Bayes model with bounded number of hidden states equal to the rank of the grid-sampled CDF tensor.

- This affords easy sampling, marginalization (by discarding the subset of factor matrices corresponding to the variables we are not interesting in), derivation of conditional distributions and expectations, and uncertainty quantification – bypassing the curse of dimensionality.

- The proposed model also affords direct and efficient estimates of (possibly semi-infinite) "box" probabilities, which is important for classification tasks. Multivariate PDF estimators, on the other hand, require multidimensional integration (analytical, numerical, or sampling-based Monte-Carlo) to estimate box probabilities, which is cumbersome and often intractable.

- At the same time, the proposed rank-constrained estimator is unassuming of the structure of the data (thus offering greater expressive power) and identifiable under relatively mild rank conditions – see [23].

- On the experimental side, our results indicate that, perhaps surprisingly, *estimating the grid-sampled CDF and then deriving a PDF estimate from it yields improved performance relative to direct PDF estimation in several machine learning applications of interest*. In addition, the performance of *the proposed non-parametric model in the Copula domain outperforms state-of-the-art Copula-based baselines*.

Building upon the framework and the methods developed in [1], we solve an important practical problem of high societal impact, using unique data and know-how of IQVIA R&D, which has a long track record in optimizing clinical trials. We leverage the results presented in [1], for reliable enrollment rate estimation, delivering probability estimates of a specific trial meeting an expected enrollment rate or probability estimates of being within a certain interval, as well as country recommendation.

- Experimental results demonstrate an improved performance of the proposed method in the enrollment rate prediction task over the best baselines by up to 12.2% in mean squared error on a real country-level trial dataset, while also offering direct means of quantifying uncertainty in the predictions based on the fitted model.

1.3.5 Chapter 7 — Modeling time series using latent history summaries and low-rank tensor densities

Probabilistic modeling of multivariate time-series is a challenging problem but fundamental for many applications in machine learning such as sensor signal modeling [24] and stock market forecasting. Although accurately modeling both inter- and intra-series dependencies is a key factor, the computational difficulties of estimating high-dimensional densities across time often limit existing methods by imposing strong assumptions on the dependencies between the variables across time. The proposed method described in this Chapter deals with these demands.

- We combine deep learning models for learning sequential dynamics with the parsimony of principled distribution models. This novel end-to-end framework ties together two threads of research: low-rank tensor modeling for flexible and efficient distribution estimation and the temporal modeling power of recurrent neural networks.

- The proposed model generates probabilistic forecasts that allow for uncertainty quantification and can handle missing inputs in some or all of the components in a multivariate time series.

- At the same time, experiments on six real-world datasets show that our method achieves significant improvements in forecasting compared to the most state-of-the-art baseline methods, as well as an average 8.07% reduction on RMSE metric for time-series imputation.

Chapter 2

Notation and Tensor Preliminaries

2.1 Notation and Definitions

Unless stated otherwise, for every Chapter of this dissertation, we use the symbols $\mathbf{x}$, $\mathbf{X}$, $\mathcal{X}$ for vectors, matrices and tensors respectively. We use the notation $\mathbf{x}(n)$, $\mathbf{X}(:,n)$, $\mathcal{X}(:,:,n)$ to refer to a particular element of a vector, a column of a matrix and a slab of a tensor. Symbols $\circ$, $\otimes$, $\circledast$, $\odot$ denote the outer, Kronecker, Hadamard and Khatri-Rao (column-wise Kronecker) product respectively. The vectorization operator is denoted as $\mathrm{vec}(\mathbf{X})$, $\mathrm{vec}(\mathcal{X})$ for a matrix and tensor respectively [23]. Additionally, $\mathrm{diag}(\mathbf{x}) \in \mathbb{R}^{I\times I}$ denotes the diagonal matrix with the elements of vector $\mathbf{x} \in \mathbb{R}^{I}$ on its diagonal. Symbols $\|\mathbf{x}\|_1$, $\|\mathbf{x}\|_2$, $\|\mathbf{X}\|_F$, d_{TV}, and $\|\mathbf{x}\|_\infty$, correspond to L_1 norm, L_2 norm, Frobenius norm, total variation distance, and infinity norm. The total variation distance between distributions p and q is defined as $d_{TV}(p,q) = \frac{1}{2}\|p-q\|_1$.

Given an N-dimensional random vector $\mathbf{X} := [X_1,\ldots,X_N]^T$, $\mathbf{X} \sim F_{\mathbf{X}}$ will denote that the random vector $\mathbf{X}$ follows distribution $F_{\mathbf{X}}$. $\mathbf{1}(A)$ is the indicator function of event A, i.e., it is 1 if and only if A is true. The set of integers $\{1,\ldots,N\}$ is denoted as $[N]$. Given M data samples, $\mathcal{D} = \{\mathbf{x}_m\}_{m=1}^{M}$ denotes the given dataset.

2.1.1 Tensor Preliminaries:

Tensors provide a natural representation for massive multidimensional data. An N-way tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ is a multidimensional array whose elements are a function of N indices $\mathcal{X}(i_1,i_2,\cdots,i_N)$, with i_n ranging from $1,\cdots,I_n$. The number of elements in the tensor

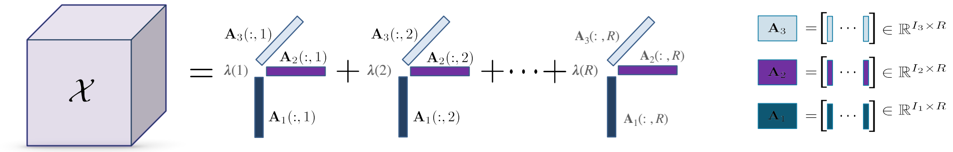


Figure 2.1: CPD model of a 3-way tensor.

$\mathcal{X}$, $\prod_{n=1}^{N} I_n$, grows exponentially with N, a problem known as the *curse of dimensionality* (CoD). Similar to the matrix case, a tensor can be represented in succinct form with tensor decompositions. It can always be decomposed as a finite sum of R rank-1 tensors for high-enough but finite R, i.e.,

$$\mathcal{X} = \sum_{h=1}^{R} \boldsymbol{\lambda}(h)\mathbf{A}_1(:,h) \circ \mathbf{A}_2(:,h), \cdots, \mathbf{A}_N(:,h), \tag{2.1}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^R$, each $\mathbf{A}_n \in \mathbb{R}^{I_n \times R}$, $\mathbf{A}_n(:,h)$ denotes the h-th column of matrix $\mathbf{A}_n$. A visualization is shown in Figure 2.1 for the case of $N = 3$. The decomposition can be compactly denoted as a collection of latent factor matrices and the weight vector $\boldsymbol{\lambda}$,

$$\mathcal{X} = [\![\boldsymbol{\lambda}, \mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_N]\!] \tag{2.2}$$

or elementwise as

$$\mathcal{X}(i_1, i_2, \cdots, i_N) = \sum_{h=1}^{R} \boldsymbol{\lambda}(h) \prod_{n=1}^{N} \mathbf{A}_n(i_n, h). \tag{2.3}$$

The vectorized form of the tensor can be expressed as

$$\text{vec}(\mathcal{X}) = \left(\odot_{n=1}^{N} \mathbf{A}_n\right) \boldsymbol{\lambda}. \tag{2.4}$$

We can express the mode-n matrix unfolding which is a concatenation of all mode-n ‘fibers’ of the tensor as

$$\mathcal{X}^{(n)} = (\odot_{k\neq n}\mathbf{A}_k)\text{diag}(\boldsymbol{\lambda}){\mathbf{A}_n}^T, \tag{2.5}$$

where $(\odot_{k\neq n}\mathbf{A}_k) = \mathbf{A}_N \odot \cdots \odot \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \odot \cdots \odot \mathbf{A}_1$.

When R is minimal, it is called the rank of $\mathcal{X}$, and the decomposition is called Canonical Polyadic Decomposition (CPD) [25, 26]. For many practical applications, low-rank decompositions can be used for extracting latent factors from tensorial datasets. If the tensor can be well approximated by a low-rank CPD model, the number of free parameters of the model to be estimated drops to $O\big(R\left(\sum_{n=1}^{N} I_n\right)\big)$.

2.2 Uniqueness of Canonical Polyadic Decomposition

A key property of the CPD is that the rank-1 components are unique under mild conditions. For a rank$-R$ tensor, the goal is to recover all the underlying factors $\mathbf{A}_n$ and $\boldsymbol{\lambda}$. The uniqueness properties of tensor rank decomposition [27] implies probability (and generative) model identifiability in our context. A model is identifiable, if and only iff there is a unique set of parameters that is consistent with what we have observed.

Theorem 2.1. *[27]: Let $k_{\mathbf{A}}$ be the Kruskal rank of $\mathbf{A}$, defined as the largest integer k such that every k columns of $\mathbf{A}$ are linearly independent. Given $\mathcal{X} = [\![\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N]\!]$, if $\sum_{n=1}^{N} k_{\mathbf{A}_n} \geq 2R + N - 1$, then the rank of $\mathcal{X}$ is R and the decomposition of $\mathcal{X}$ in rank-one terms is unique.*

Better results allowing for higher tensor rank are available for generic tensors of given rank.

Theorem 2.2. *[28]: Given $\mathcal{X} = [\![\boldsymbol{\lambda}, \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3]\!]$, assume, without loss of generality, that $I_1 \leq I_2 \leq I_3$. Let α, β be the largest integers such that $2^\alpha \leq I_1$ and $2^\beta \leq I_2$. If $R \leq 2^{\alpha+\beta-2}$ the decomposition of $\mathcal{X}$ in rank-one terms is unique almost surely.*

There are many more different uniqueness conditions for CPD. The take-home point here is that the CPD model is essentially generically unique even if R is much larger than I_1, I_2, I_3 – so long it is less than maximal possible rank.

2.3 Representing Statistical Distributions via Tensors

Learning multivariate distributions is one of the core research problems in machine learning and statistics. Modeling a data-generating distribution generally refers to the approach of inferring a function that describes the hidden structure from unlabeled data and can “explain” given samples. Given a set of independently and identically generated realizations (or data-points) of a vector

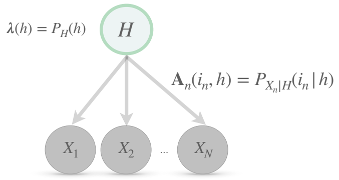

$\boldsymbol{\lambda}(h) = P_H(h)$
H
$\mathbf{A}_n(i_n, h) = P_{X_n|H}(i_n \,|\, h)$
X_1
X_2
...
X_N

Figure 2.2: Latent variable-naive Bayes model.

$\boldsymbol{X} \in \mathbf{R}^N, \boldsymbol{X} := [X_1, \ldots, X_N]^T$, the goal is to find an accurate estimate of the true underlying data distribution.

If X_n takes discrete integer values in $\{1, \cdots, I_n\}, n = 1, \ldots, N$, $\boldsymbol{X}$ is governed by a joint Probability Mass Function (PMF). We can think of the joint PMF as an N-way array $\mathcal{X} \in \mathbf{R}^{I_1 \times I_2 \times \cdots \times I_N}$, which we call a probability tensor. The size of each dimension is equal to the alphabet size $I_1, \ldots, I_N$ of the corresponding variable and the indexed elements represent the probability of the particular realization, i.e.,

$$\mathcal{X}(i_1, i_2, \ldots, i_N) = Pr(X_1 = i_1, X_2 = i_2, \ldots, X_N = i_N).$$

Every tensor $\mathcal{X}$ admits a non-negative CPD of finite rank [18], and thus we can always express the joint PMF using a non-negative CPD model

$$\mathcal{X}(i_1, \ldots, i_N) = \sum_{h=1}^{R} \boldsymbol{\lambda}(h) \prod_{n=1}^{N} \mathbf{A}_n(i_n, h), \tag{2.6}$$

for high enough R [18].

Equation (2.6) shows that every joint PMF admits a naive Bayes model *interpretation* with bounded R. The naive Bayes model assumes that there is a hidden random variable Z taking R values, such that given $H = h$ the random variables X_V are conditionally independent i.e.,

$$P_{X_V}(i_1, \ldots, i_N) = \sum_{h=1}^{R} P_H(h) \prod_{n=1}^{N} P_{X_n|H}(i_n|h). \tag{2.7}$$

By variable matching between Equations (2.6) and (2.3) and upon defining

$$\mathbf{A}_n(i_n, h) := P_{X_n|H}(i_n|h), \text{ for } n = 1, \ldots, N, \text{ and } \boldsymbol{\lambda}(h) := P_H(h),$$

we can see that the naive Bayes model can be represented by a non-negative CPD model $\mathcal{X} = [\![\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N]\!]$ with the constraints that matrices $\mathbf{A}_n$ are column-stochastic, and $\mathbf{1}^T \boldsymbol{\lambda} = 1$ [18], [19]. The observed data (so-called 'manifest' variables) are generated through an unknown mapping expressed by the conditional distributions $P_{X_n|H}$ and the prior distribution of the hidden variable P_H. One of the very appealing properties this generative model is that it affords easy sampling. According to Equation , a sample of the multivariate PMF can be generated by first drawing H according to P_H and then independently drawing samples for each variable X_n from the conditional PDF $P_{X_n|H}$. The resulting generative model can be visualized in Figure 6.1. This model is also known as mixture of unigrams or latent class model and has been applied in many applications such as topic modeling [29], clustering [30] and crowdsourcing [31]. The rank of the probability tensor is a nonlinear measure of statistical dependence of the associated random variables. If $R = 1$, the random variables are independent while full rank indicates complete statistical dependence. Partial dependence is typical, and can be modeled using a low-rank multivariate probability tensors.

In conclusion, *any* joint PMF can be represented by a latent variable model with just one hidden variable having R possible states and therefore admits a non-negative CPD of bounded rank. Employing this model, we can alleviate the CoD by focusing on 'principal components' of the joint distribution.

If $\boldsymbol{X}$ is a real valued vector with its support (the set of values that it can take) contained in $S_{\boldsymbol{X}}$, $\boldsymbol{X}$ is governed by a joint PDF. In case of continuous random variables, however, the joint PDF can no longer be directly represented by a tensor. One possible solution could be discretization, but this can lead to discretization error and possibly loss of identifiability. In Chapter 4, we show is that we can *approximately* represent any smooth joint PDF (and evaluate it at any point) by a finite *characteristic tensor* of leading complex Fourier coefficients, thereby avoiding discretization loss altogether. For mixed random variables, in Chapter 6, we target modeling the joint CDF (and its grid-sampled sketch).

Chapter 3

Information-theoretic Feature Selection via Tensor Decomposition and Submodularity

3.1 Introduction

Real-world data often exhibit complicated manifold structure in very high dimensional spaces, making conventional machine learning tools insufficient for data analysis. Knowledge discovery in a high-dimensional space with limited training examples is a difficult task that entails high computational cost in both the training and the run-time stage, large variance of the predictions due to overfitting the training samples, and poor generalization. Although adding more input variables may provide additional information, an assumption supported by the data processing inequality [32], after a certain point the performance of classification will typically degrade as the number of features continues to increase. In practice, not all features are equally important and discriminative, as many of the dimensions carry little or redundant information. Analyzing high-dimensional data therefore raises the fundamental problem of reducing dimensionality by discovering compact representations that do not incur significant loss in prediction accuracy for the ultimate task at hand. Feature selection methods try to find a lower-dimensional representation of data by removing redundant or irrelevant features. Feature selection maintains the physical meaning and dependencies between the selected features, resulting in predictive models with better interpretability [33, 34]. Feature selection aids the learning task by reducing the number of parameters to learn (and hence sample complexity), and by speeding up the associated computation.

Projecting data onto a lower dimensional space facilitates, among other tasks, exploratory data analysis and visualization, clustering, and compression of high-dimensional data. Feature selection is particularly important and challenging in biomedical data mining, where the data is characterized by relatively few training instances and a high-dimensional feature space, leading to degradation of classifier performance as noisy/uninformative features prohibit us from mining potentially useful knowledge [35]. In personalized marketing, feature selection is used for sentiment analysis of customer reviews as it aims to identify indicators in the document to infer the polar category, either positive or negative sentiment, so that products are targeted to customers where the probability of positive sentiment is higher [36]. Feature selection has also been used to improve text data clustering and classification [37], and in stock market price index prediction to reduce the cost of training time and to improve prediction accuracy [38].

To evaluate any possible subset, feature selection methods require a quality measure. Most prior information-theoretic methods for feature selection use a lower-order approximation of the Mutual Information (MI). We consider using the high-order Shannon entropy-based MI as the evaluation criterion, because it can capture any kind of relationship, linear or nonlinear, between multiple random variables.

Computing MI typically entails the estimation of a high-dimensional probability distribution. Direct estimation of the joint distribution for high-dimensional data is impossible, due to the curse of dimensionality. We thus need a 'universal' model that can capture the 'principal components' of this high-dimensional joint distribution in a parsimonious way. It has recently been shown that low-rank approximation of the joint probability tensor addresses this need [18]. Rank-F approximation represents the joint distribution as a latent variable model with just one hidden variable having F possible states. For large enough but finite F the latter model is universal – it can represent *any* joint distribution of categorical variables. Low rank CPD models have been successfully applied for joint distribution modeling and downstream tasks in applications ranging from recommender systems, see Kargas et al. [18], to anomaly detection, see Amiridi et al. [22] and sampling images from the USPS dataset where the dimensionality is $N = 256$ with tensor rank equal to $F = 8$ [21]. Higher dimensional data (MNIST, FashionMNIST, $N = 784$) have also been modeled using low rank CPD models in a reduced-dimension latent space. These results clearly support the claim that relatively low-rank CPD can effectively model relatively high-dimensional distributions. In this paper, we propose a novel dimensionality reduction framework that incorporates a low-rank model of the joint distribution, which affords disciplined subset selection through maximization of a monotone submodular function. The latter optimization is amenable to greedy solution with performance guarantees.

3.2 Related work

Numerous methods have been proposed for feature selection. This work focuses on information-theoretic feature selection methods. An extensive body of work on information-theoretic feature selection techniques exists which is based on maximizing mutual information between subsets of features and class labels. Brown et al. [39], for example, posed the feature selection problem as a conditional likelihood maximization of the class labels, given features. Most techniques are greedy methods that make use of low-dimensional MI quantities due to the difficulty associated with estimating high dimensional distributions from limited samples. Simpler methods often ignore features which have strong discriminatory power as a group but are weak as individual features. There are a growing set of methods for feature selection based on variational lower bound estimates on mutual information [40] but the properties of existing variational estimators of MI are not yet well understood. To address these problems, recent techniques consider interactions among more than two variables, by estimating / approximating higher-dimensional mutual information quantities. We selected four state-of-the-art information-theoretic feature selection methods (MRMR [41], JMIM [42], RJMI [43], and GlobalFS/iSelect [44]) as baselines.

1. MRMR: The Maximum Relevance Minimum Redundancy approach is an information-theoretic based method that uses mutual information to select features that have high relevance to the class while having low mutual information with the other selected features, thus guarding against pairwise redundancy. MRMR is a greedy algorithm like our method, but with a coarser criterion [41].

2. JMIM: The Joint Mutual Information Maximization approach starts with a feature of a maximal mutual information with the target variable Y. Then, it greedily adds the feature X that maximizes $J(X) = I(X; Y|S)$, where S is the set of already selected features [42]. Similar to the proposed method, this framework also requires the input to be discrete and

uses empirical estimators of distribution, and, consequently, information gain or entropy.

3. Rényi-based JMI (Rényi's α-order based joint MI maximization): Instead of using Shannon's definition of the mutual information, [43] defines a multivariate extension of the matrix-based Rényi's α-order joint entropy, which allows estimating the multivariate MI with respect to a desired variable Y, without evaluating the underlying PMF.

4. GlobalFS/iSelect: By denoting a new candidate feature as X^{cand}, this method [44] aims to find an optimal set of features that jointly maximize the mutual information with the class variable while penalizing low values of the increment of $I(S;Y)$ after adding X^{cand}, i.e., $I(X^{\text{cand}};Y|S')$.

Throughout our experiments, we used the respective author-provided codes downloaded from `https://github.com/danielhomola/mifs` for MRMR and JMI, `https://sites.google.com/site/vinhnguyenx` for GlobalFS/iSelect, and shared[1] by the authors of [43].

3.3 Preliminaries

3.3.1 Mutual Information and Submodularity

The Shannon entropy of a random variable X is defined as

$$H(X) = -\sum_{x} P_X(x) \log P_X(x)$$

and measures the amount of uncertainty in X. Given a second variable Y, we can quantify the uncertainty in X after Y has been observed using the conditional entropy

$$H(X|Y) = -\sum_{x,y} P_{X,Y}(x,y) \log P_{X|Y}(x|y).$$

The Mutual Information (MI) between two random variables X and Y is defined as

$$I(X;Y) = \sum_{x,y} P_{X,Y}(x,y) \log \frac{P_{X,Y}(x,y)}{P_X(x)P_Y(y)},$$

[1] Private communication – we gratefully acknowledge the authors of [43] for sharing their code with us.

and measures how far the variables X, Y are from being independent. Alternatively, we can view MI as

$$I(X;Y) = H(Y) - H(Y|X),$$

which allows us to interpret MI as the reduction of the uncertainty about Y when we are provided with knowledge of X. MI is symmetric in its arguments:

$$I(X;Y) = I(Y;X) = H(X) - H(X|Y).$$

Given the joint PMF $P_{X_V,Y}$ of $X_V = \{X_1, X_2, \ldots, X_N\}$ and Y, we consider the *high-order mutual information* between a subset of features X_S, $S \subseteq V$ and the variable Y

$$f(S) = I(X_S;Y) = H(X_S) - H(X_S|Y),$$

which quantifies the expected reduction of uncertainty about Y upon revelation of X_S. According to the non-decreasing property of MI, adding extra variables increases joint entropy, decreases conditional entropy, and increases information:

$$I(X_S;Y) \leq I(X_V;Y).$$

MI has been successfully employed in many feature selection methods due to the fact that it can capture complex relationships between the features and the target variable. However, selecting the optimal subset of features of cardinality K that maximizes the high-order mutual information is known to be NP-hard [45]. Additionally, the calculation of high-order mutual information requires a reliable estimate of the joint probability distribution.

It has been shown that in the special case where the features are independent given the target variable Y (which is a very restrictive and unrealistic assumption in practice), f is monotone submodular [46, 47]. Submodular functions comprise a class of set functions $f : 2^V \to \mathbb{R}$ that satisfy the diminishing returns property

$$f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)), \forall A \subseteq B \subseteq V \text{ and } x \in V \setminus B.$$

This property states that adding an element to a smaller set results to larger increase in f

than adding it to a larger set. Moreover, if

$$f(A \cup \{x\}) \geq f(A), \forall A \subseteq V$$

holds, the function is monotone submodular. [48] showed that the problem of maximizing a monotone submodular function f subject to a cardinality constraint can be approximated with a constant multiplicative factor of

$$1 - \frac{1}{e},$$

performance guarantee to the optimal solution of the NP-hard optimization problem using a simple greedy algorithm. Here, e is the base of the natural logarithm. See more in Equation 3.4. Submodularity can be further exploited to accelerate the greedy implementation, leading to an algorithm called lazy greedy with almost linear time complexity [49].

3.4 Problem Formulation

Given a dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{M}$ of M realizations of the random variables $X_V = \{X_1, X_2, \ldots, X_N\}$ (features) and the target variable Y (label), we wish to infer a subset $S \subseteq V$ comprising $|S| \leq K$ features that best predicts Y for the given K. Ideally, K is the *intrinsic* dimension of the dataset – the minimum number of variables that carry sufficient information for accurately predicting the target variable Y. Intrinsic dimension can be alternatively viewed as the size of the smallest feature subset after which the MI between this subset and the target variable stops increasing. We begin by formulating feature selection as an optimization problem seeking to maximize the MI between the features X_S and the target variable Y,

$$\underset{S \subseteq V, |S| \leq K}{\operatorname{argmax}} \; f(S). \tag{3.1}$$

Instead of ranking each variable X_n independently of the rest, this multivariate approach, which utilizes the high-order mutual information, evaluates features according to their joint information power, enabling us to detect redundant features. Feature interaction is significant in view of the fact that groups of several features acting simultaneously may be relevant, but not the individual features alone. For example, consider the binary case, where the conditional joint PMF of two binary random variables X_1 and X_2 given $Y = 0$ and $Y = 1$, respectively is given in matrix form:

$$P_{X_1,X_2|Y=0} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}, P_{X_1,X_2|Y=1} = \begin{pmatrix} 0 & 0.5 \\ 0.5 & 0 \end{pmatrix}.$$

Then clearly X_1 is independent of Y, same for X_2, but X_1, X_2 together are not independent of Y, i.e., $X_1 \perp\!\!\!\perp Y, X_2 \perp\!\!\!\perp Y, \{X_1, X_2\} \not\perp\!\!\!\perp Y$, and we say that features X_1, X_2 constitute an interaction group – a set of features that appear to be irrelevant or weakly relevant with the class Y individually, but if considered jointly, they correlate to the class. However, the number of candidate subsets is $\binom{N}{K}$, thus an exhaustive search is too costly and practically prohibitive even for a medium feature set size K. As mentioned earlier, the problem is NP-hard [50].

Instead of solving optimization problem (3.1), we propose an intuitive and more efficient alternative approach. Since *every* joint distribution admits a naive Bayes representation, via the CPD, we take an indirect path for determining the most informative features, through the latent variable Z. We propose using the mutual information as a metric to identify the subset of the 'manifest' variables that can best identify the operational component of the distribution, or, in other words, to best predict the latent variable Z in the CPD model. The graphical model implies a dependence of the label Y on the observed variable X_n through the latent variable Z. Given Z, the features and the label Y become conditionally independent, hence if we predict Z from the features, predicting Y from Z is a simple task.

In lieu of the initial $I(X_S; Y)$ maximization approach, we therefore propose solving the surrogate problem of selecting features by maximizing the MI between the selected features and the latent variable Z, i.e.,

$$\underset{S \subseteq V, |S| \leq K}{\operatorname{argmax}} \; g(S), \text{ where } g(S) = I(X_S; Z) \quad = \sum_{x_S, z} P_{X_S,Z}(x_S, z) \log \frac{P_{X_S,Z}(x_S, z)}{P_{X_S}(x_S) P_Z(z)}.$$

Employing a CPD model for the joint PMF $P_{X_V,Y}$, feature selection can be equivalently described as dropping out all but an optimal subset S of K edges tied to the N features (Fig. 6.1). In terms of the CPD model, this means choosing a representative subset of factor matrices $\{\mathbf{A}_S\} \subset \{\mathbf{A}_V\}$ to form the reduced CPD model

$$\mathcal{X}' = [\![\boldsymbol{\lambda}, \{\mathbf{A}_S\}, \mathbf{A}_{N+1}]\!]. \tag{3.2}$$

Claim 1.

$$g(S) - \text{const} \leq f(S) \leq g(S),$$

where $\text{const} = I(X_V; Z|Y) = I(X_V; \{Z, Y\}) - I(X_V; Y)$.

Proof: Given $X_S, Y,$ and Z, the conditional mutual information is defined as $I(X_S; Z|Y) = I(X_S; \{Z, Y\}) - I(X_S; Y)$. From the definition of the conditional MI and the latent variable model, for which it holds that $I(X_S; Y|Z) = 0$, we get the following:

$$\begin{aligned} I(X_S; Y) &= I(X_S; \{Y, Z\}) - I(X_S; Z|Y) \\ &= I(X_S; Z) - I(X_S; Z|Y). \end{aligned}$$

Since MI is always non-negative, we get that

$$I(X_S; Z|Y) \geq 0$$

and thus

$$I(X_S; Y) \leq I(X_S; Z),$$

which means that our surrogate objective is an upper bound on the original objective function. Furthermore, by the non-decreasing property of MI, we get that

$$I(X_S; Z|Y) \ \leq I(X_V; Z|Y),$$

and it holds that

$$\begin{aligned} I(X_S; Z) - I(X_V; Z|Y) \leq I(X_S; Y) \leq I(X_S; Z) \Leftrightarrow \\ I(X_S; Z) - \text{const} \leq I(X_S; Y) \leq I(X_S; Z) \Leftrightarrow \\ g(S) - \text{const} \leq f(S) \leq g(S), \end{aligned}$$

where

$$\text{const} = I(X_V; Z|Y) = I(X_V; \{Z, Y\}) - I(X_V; Y).$$

The double inequality shows that we are maximizing a surrogate function that is a constant band-gap away from the desired function. Note that if we estimate the principal components of the joint PMF, we can easily estimate the band-gap. The band-gap can be further bounded as follows:

$$\text{const} = I(X_V; Z|Y) = H(Z|Y) - H(Z|X_V, Y) \leq H(Z|Y).$$

Using $H(Y) - H(Y|Z) = I(Y; Z) = H(Z) - H(Z|Y)$,

$$\text{const} \leq H(Z|Y) = H(Z) - H(Y) + H(Y|Z),$$

and thus

$$\text{const} \leq H(Z) + H(Y|Z) \leq \log(F) + H(Y|Z),$$

since the entropy of a finite-alphabet random variable is upper bounded by the logarithm of its alphabet size (achieved by the uniform distribution over its alphabet), and Z has alphabet size equal to the tensor rank, F ($F \leq 30$ in our experiments). This confirms our intuition that when Z is predictive of Y ($H(Y|Z)$ is small) the band-gap should be small. In practice we expect much smaller band-gaps, as we used multiple upper bounding steps in this derivation.

Like the original problem, the proposed alternative is NP-hard. The reason we propose it, however, is two-fold: first, given Z, all the X's become irrelevant as far as Y is concerned; and the above surrogate optimization problem where we aim to predict Z involves the maximization of a monotonic submodular reward function subject to a cardinality constraint [46,47], which is not the case when our aim is to predict Y directly from the regressors. Monotone submodular maximization subject to a cardinality constraint enjoys $1 - \frac{1}{e}$ approximation guarantee to the optimum solution, while simultaneously enabling extremely fast optimization [49].

3.5 Algorithm Description

The proposed feature selection process, called Greedy Submodular Monotone optimization using CPD (GSM-CPD), consists of four steps, namely, PMF estimation of all variables, subset generation, MI evaluation and subset selection.

3.5.1 PMF Estimation

In the first step, our algorithm utilizes a rank-F approximation of the empirical joint PMF tensor $\widehat{\mathcal{X}}$, computed using Kullback-Leibler (KL) divergence as the fitting criterion. The empirical probability tensor, which is typically sparse, is formed by computing how often an event (a

Algorithm 3.1 PMF Estimation

```
Input: Empirical PMF: $\widehat{\mathcal{X}}$, ‘Signal Rank’ $F$
Output: $\mathcal{X} = [\![\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N]\!]$
Initial guess $\mathcal{X} = [\![\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N]\!]$
$\widehat{\mathcal{Y}} \leftarrow \widehat{\mathcal{X}} / [\![\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N]\!]$
while termination condition not met do
    for all $f$ do
        $\boldsymbol{\lambda}(f) \leftarrow \boldsymbol{\lambda}(f) \widehat{\mathcal{Y}} \times_1 \mathbf{A}_1(:, f) \cdots \times_N \mathbf{A}_N(:, f)$
    end for
    for all $n$ update in parallel
        $\mathbf{A}_n \leftarrow \mathbf{A}_n * \mathrm{MTTKRP}(\widehat{\mathcal{Y}}, \{\mathbf{A}_n\}_{n=1}^N, n)$
    end for
    $\widehat{\mathcal{Y}} \leftarrow \widehat{\mathcal{X}} / [\![\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N]\!]$
end while
$\mathcal{X} \leftarrow [\![\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N]\!]$
```

realization of the feature vector) occurred in the training set. The rank-F approximation of the joint PMF captures the F principal components of the distribution and is essential for the MI calculation process, which serves to evaluate the quality of the selected feature set S. Defining KL divergence between two probability tensors $\mathcal{X}$ and $\mathcal{Y}$ as

$$\mathrm{D}_{\mathrm{KL}}(\mathcal{X} \| \mathcal{Y}) := \sum_{i_1, \ldots, i_N} \mathcal{X}(i_1, \ldots, i_N) \log \frac{\mathcal{X}(i_1, \ldots, i_N)}{\mathcal{Y}(i_1, \ldots, i_N)},$$

we propose solving the following optimization problem

$$\begin{aligned} \min_{\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N} \quad & \mathrm{D}_{\mathrm{KL}}\Big(\widehat{\mathcal{X}} \| [\![\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N]\!]\Big) \\ \text{subject to} \quad & \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{1}^T \boldsymbol{\lambda} = 1, \\ & \mathbf{A}_n \geq \mathbf{0},\ n = 1 \ldots N, \\ & \mathbf{1}^T \mathbf{A}_n = \mathbf{1}^T,\ n = 1, \ldots, N \end{aligned} \tag{3.3}$$

by employing the Expectation Maximization (EM) algorithm as described in [51] and [52]. At each iteration, EM updates the factors simultaneously, making the algorithm easily parallelizable. The exact updates for $\boldsymbol{\lambda}$, $\mathbf{A}_n$ are shown in Algorithm 3.1. Notation $\times_n$ stands for the n-mode product of a tensor with a matrix, MTTKRP denotes the n-mode matricized tensor times Khatri-Rao product, and $/$ stands for element-wise division. Note that the complexity of this operation is $\mathcal{O}(M)$. Here, M denotes the number of samples which is approximately equal to the non-zero elements of tensor $\widehat{\mathcal{X}}$.

Determining the rank F of tensor $\mathcal{X}$ is an NP-hard problem [53]. Essentially, instead of detecting the exact rank, we are interested in fitting a model that has 'meaningful' number of components – the useful 'signal rank', which is determined by cross-validation techniques. The per iteration complexity of the algorithm is dominated by the $\boldsymbol{\lambda}$-update, which is of $\mathcal{O}(MF)$ complexity, and by each $\mathbf{A}_n$-update, which is also $\mathcal{O}(MF)$ complexity.

3.5.2 Incremental Greedy Feature Selection

After fitting a low-rank CPD model to the empirical PMF, we employ a forward greedy algorithm (Alg. 3.2) for the problem of subset selection, i.e, $\max_{S \subseteq V, |S| \leq K} g(S)$. During the subset generation procedure, candidate feature subsets are generated for evaluation based on the MI. Starting with an empty set $S = \emptyset$, the algorithm incrementally builds a solution. At iteration i, it selects the feature s_i that improves the current solution the most, according to information gain

$$s_i = \operatorname*{argmax}_{s \in V \setminus S} I(X_{S \cup \{s\}}; Z) - I(X_S; Z),$$

and adds it to the current set $S \leftarrow S \cup \{s_i\}$. The process of subset generation and evaluation is repeated until $|S| = K$. The output of the greedy algorithm is always a set S such that

$$I(X_S; Z) \geq \left(1 - \frac{1}{e}\right) I(X_{S^\star}; Z), \tag{3.4}$$

where $S^\star$ is the optimal solution i.e., the set maximizing $g(S)$ among all size-K sets.

Without loss of generality, assume that at iteration $k \leq K$, $X_S = \{X_1, X_2, \ldots, X_k\}$. We can evaluate the MI between X_S and the latent variable Z as follows.

$$\begin{aligned} I(X_S; Z) &= \sum_{x_S, z} P_{X_S, Z}(x_S, z) \log \frac{P_{X_S,Z}(x_S, z)}{P_{X_S}(x_S) P_Z(z)} \\ &= \mathbf{1}^T \left[(\odot_{n=1}^k \mathbf{A}_n) \mathrm{diag}(\boldsymbol{\lambda}) * \log \frac{(\odot_{n=1}^k \mathbf{A}_n) \mathrm{diag}(\boldsymbol{\lambda})}{[(\odot_{n=1}^k \mathbf{A}_n) \boldsymbol{\lambda}] \circ \boldsymbol{\lambda}} \right] \mathbf{1}, \end{aligned}$$

Algorithm 3.2 Incremental Greedy Feature Selection

```
Input: $K$: Number of features; $\mathcal{X}$: Joint PMF tensor
Output: $S$: Estimated subset of features
$V = \{1, 2, \ldots, N\}$
$S = \emptyset$
while $|S| < K$ do
    for all $s \in V \setminus S$ do
        $MI(s) = I(X_{S \cup s}; Z)$
    end for
    $s_i \leftarrow$ feature with maximum $MI$
    $S \leftarrow \{S \cup s_i\}$
end while
```

where $*$ indicates the matrix Hadamard product and the logarithm is computed element-wise. In case of large k the above computation is prohibitive. It can be simplified using the fact that under the naive Bayes model, MI is given by

$$I(X_S; Z) = H(X_S) - H(X_S|Z) = H(X_S) - \sum_{n=1}^{K} H(X_n|Z).$$

In terms of the CPD factors, the joint entropy of X_S is given by

$$\begin{aligned} H(X_S) &= -\sum_{x_S} P_{X_S}(x_S) \log P_{X_S}(x_S) \\ &= -\sum_{i_1,\ldots,i_K} \mathcal{X}(i_1, \ldots, i_k) \log \mathcal{X}(i_1, \ldots, i_k) \\ &= -\mathbf{1}^T \left[(\odot_{n=1}^{K} \mathbf{A}_n) \boldsymbol{\lambda} * \log(\odot_{n=1}^{K} \mathbf{A}_n) \boldsymbol{\lambda} \right] \mathbf{1}, \end{aligned}$$

and the conditional entropy for each variable X_n is given by

$$\begin{aligned} H(X_n|Z) &= -\sum_{x_n, z} P_{X_n, Z}(x_n, z) \log P_{X_n|Z}(x_n|z) \\ &= -\sum_{i_n, f} \mathbf{A}_n(i_n, f) \boldsymbol{\lambda}(f) \log \mathbf{A}_n(i_n, f) \\ &= -\mathbf{1}^T \left[(\mathbf{A}_n \text{diag}(\boldsymbol{\lambda})) * \log \mathbf{A}_n \right] \mathbf{1}. \end{aligned}$$

Calculating the MI function can be expensive due to the computational bottleneck of the joint entropy, which requires I^k evaluations. To overcome this issue, we can take advantage of the fact that

$$H(X_S) = -\mathbb{E}[\log P_{X_S}(X_S)]$$

to calculate an approximation for this term, by drawing samples from the joint distribution. We randomly sample T values of the latent variable Z according to its distribution $\boldsymbol{\lambda}$ and for each value f we similarly sample from the f-th column of each factor matrix $\mathbf{A}_n(:, f), \forall n \in S$. We calculate the probability of this particular realization via the probability tensor $\mathcal{X}' = [\![\boldsymbol{\lambda}, \{\mathbf{A}_S\}]\!]$. After transforming to logarithmic scale, we sum up the results and normalize by T to get the sample mean. Note that, in our experiments drawing $1,000 - 5,000$ samples is sufficient for a well-approximated joint entropy. The per iteration complexity of the algorithm is determined by the calculation of the joint entropy which is of $\mathcal{O}(NFT)$ complexity and the conditional entropy computation which is of $\mathcal{O}(NFI)$ complexity. In total, the complexity of the algorithm is $\mathcal{O}(KNF(T + I))$.

3.6 Experiments

Datasets: We conducted experiments on multiple real-world datasets to assess the performance of the proposed GSM-CPD sequential feature selection framework against various supervised information-theoretic feature selection algorithms that are representative of the state-of-art described in Section 3.2. Note that the parameters were separately tuned to optimize the performance of each method considered. For RJMI, α is tuned on the set $\{0.1, 0.5, 1, 2\}$ and the kernel size σ is tuned on the set $\{0.1, 0.5, 1, 5, 10, 50, 100\}$. For GlobalFS/iSelect, the significance parameter α was tuned on the set $\{0.99, 0.95\}$. We have noticed that the results are not sensitive with respect to the choice of α, as also noted in [44]. All datasets are from the UCI machine learning repository [54]. A summary of the selected datasets is presented in Table 3.1. For each dataset, continuous features are uniformly discretized into 5 bins. Features that were already discrete (or categorical in nature) are left as-is. Note that we only discretize the data for the methods that assume discrete features. No discretization is used for the methods that work with continuous data, such as RJMI or GlobalFS, which are fed with the raw continuous features.

Data set	W/O FS	GSM-CPD	MRMR	JMIM	RJMI	GlobalFS/iSelect
Lung-cancer	79.58 ± 20.52	**86.43 ± 15.59**	**83.45 ± 17.35**	81.56 ± 18.50	**85.07 ± 16.59**	82.48 ± 17.37
Promoters	86.83 ± 7.64	**87.93 ± 5.45**	**87.13 ± 6.65**	**87.89 ± 5.07**	84.95 ± 7.39	86.39 ± 8.52
Splice	**94.14 ± 1.43**	**93.83 ± 1.62**	89.94 ± 1.84	89.25 ± 1.93	92.92 ± 1.49	**93.79 ± 2.41**
USCensus	**97.49 ± 0.21**	**97.41 ± 0.17**	96.95 ± 0.74	97.01 ± 0.36	**97.41 ± 0.38**	96.22 ± 0.32
CoIL2000	93.97 ± 0.28	**94.22 ± 0.11**	92.02 ± 0.13	**94.03 ± 0.14**	**94.05 ± 0.21**	92.64 ± 0.83
Chemical	94.65 ± 1.62	**96.47 ± 2.18**	**95.51 ± 2.53**	93.48 ± 1.91	**95.72 ± 1.93**	93.47 ± 1.63
Musk2	**96.79 ± 0.72**	**95.52 ± 0.70**	93.48 ± 0.76	94.65 ± 0.94	**95.35 ± 0.71**	91.33 ± 0.51
Arrhythmia	67.24 ± 2.51	**72.80 ± 5.73**	**68.59 ± 6.74**	65.92 ± 8.37	65.36 ± 5.36	**67.44 ± 6.73**
Isolet	79.10 ± 3.36	**79.58 ± 6.14**	58.49 ± 5.27	**80.75 ± 6.47**	**79.47 ± 5.64**	69.27 ± 5.52
Multi-features	**94.31 ± 1.25**	**95.08 ± 1.37**	80.41 ± 2.35	**94.98 ± 1.03**	90.80 ± 1.65	91.49 ± 1.75

Table 3.2: Test accuracy of the AdaBoost with classification stumps on a fixed number of selected features for each feature selection algorithm.

Dataset	N	M	C
Lung-cancer	57	32	3
Promoters	59	106	2
Splice	62	3190	3
USCensus	90	689338	3
CoIL2000	86	5822	2
Chemical	151	936	3
Musk2	169	6598	2
Arrhythmia	280	452	16
Isolet	618	1560	26
Multi-features	650	2000	10

Table 3.1: Summary of bench-mark datasets. N is the total number of features, M is the number of feature vectors (samples), and C is the number of classes – the size of the alphabet of the target random variable Y.

Data set	W/O FS	GSM-CPD	MRMR	JMIM	RJMI	GlobalFS/iSelect
Lung-cancer	80.00 ± 22.92	**90.05 ± 16.32**	**84.12 ± 16.37**	80.83 ± 22.92	**90.00 ± 17.88**	81.57 ± 13.93
Promoters	**90.91 ± 6.45**	**94.64 ± 5.47**	87.73 ± 6.55	**92.64 ± 6.55**	84.00 ± 6.15	82.17 ± 6.87
Splice	**95.14 ± 1.57**	**94.48 ± 1.53**	93.48 ± 2.20	89.25 ± 1.94	88.40 ± 2.20	**93.60 ± 1.87**
USCensus	93.38 ± 0.19	**97.95 ± 0.22**	96.95 ± 0.15	**97.93 ± 0.20**	**97.87 ± 0.30**	92.74 ± 0.38
CoIL2000	79.03 ± 0.21	**93.94 ± 0.07**	92.02 ± 0.04	**93.58 ± 0.07**	**93.99 ± 0.20**	92.94 ± 0.06
Chemical	62.79 ± 2.03	**78.20 ± 2.15**	**72.11 ± 2.31**	67.20 ± 1.79	**71.72 ± 2.09**	70.68 ± 1.57
Musk2	**84.69 ± 2.04**	**84.59 ± 2.12**	64.83 ± 2.17	**84.59 ± 1.14**	64.85 ± 1.93	81.94 ± 1.82
Arrhythmia	60.62 ± 3.36	**68.80 ± 4.25**	**67.80 ± 4.28**	66.81 ± 3.93	66.60 ± 5.94	**68.60 ± 5.73**
Isolet	**83.72 ± 2.38**	**83.06 ± 2.74**	71.67 ± 3.25	**83.37 ± 2.53**	82.28 ± 2.52	80.06 ± 3.42
Multi-features	**93.95 ± 2.42**	**96.15 ± 2.94**	76.05 ± 2.24	**95.95 ± 1.96**	93.75 ± 1.95	92.25 ± 2.26

Table 3.3: Test accuracy of RBF-kernel SVM on a fixed number of selected features for each feature selection algorithm.

Experimental Settings: The proposed approach, GSM-CPD, is implemented in MATLAB using the Tensor Toolbox [55] for tensor operations. For each experiment we split the dataset such that 80% of the data samples is used for training and 20% for testing, and run 10 Monte-Carlo simulations (each Monte-Carlo simulation is a random $80\% - 20\%$ split of the data).

An appropriate rank F for our model is found using $10-$fold cross-validation on the training data. For each dataset, we fit CPD models of different ranks, $F \in \{5, 10, 15, 20, 30\}$, and choose the one which on average minimizes the misclassification error on the validation set. The quality of a subset is always determined by a certain criterion – an optimal subset selected using one criterion may not be optimal according to another criterion. After extracting the optimal subset S of features using each method, each subset of features is evaluated on the testing set, in terms of the classification performance of two classifiers: AdaBoost with classification stumps and RBF-kernel SVM. Simpler classifiers, such as a linear SVM classifier, were also tested to ensure that the performance gain is driven by the selected features rather than the chosen classifier. Qualitatively similar results were obtained using simpler classifiers. We report the mean (and standard deviation) classification accuracy of each feature selection method evaluated on a certain range of percentage of selected features, namely $[1\% - 10\%]$. For each method, we keep the smallest percentage which results in largest validation accuracy. The reason for this setting is to answer the following question: Can we use a much smaller feature subset without sacrificing classification accuracy, or even better, can we enhance classifiers to improve their overall performance?

If the dataset dimensionality exceeds $N = 100$, the feature selection process is realized with the following modification. At iteration $k + 1 \leq K$, where the current subset of selected features is $X_S = \{X_1, X_2, \ldots, X_k\}$, we remodel the PDF of the subsets $X_{S \cup \{s\}}, s \in V \setminus S$ of variables and select the one that maximizes the MI. In order to limit the extra complexity incurred, we warm-start the extended model using $[\![\boldsymbol{\lambda}, \mathbf{A}_1, \cdots, \mathbf{A}_k]\!]$ from the previous iteration, which drastically reduces the number of iterations.

3.6.1 Experiments on real-world data sets

Tables 3.2,3.3 depict the predictive performance using the above feature selection methods followed by two different classifiers. In each of these tables, we include the accuracy of the classifier without any prior feature elimination (W/O FS).

Then, for each dataset and classifier we highlight the best three performing methods in bold, while underlining the best performing method. The results demonstrate the superior performance of GSM-CPD as a feature selection strategy. From individual average accuracy values, we observe that for most of the data sets, GSM-CPD can identify redundant and noisy features and thus maintain or even increase the accuracy by using a much smaller subset for the classification task. Specifically, we observe that utilizing feature subsets generated by GSM-CPD for classification can increase the classification accuracy of AdaBoost with stump functions learner by 8.61% and 8.24% on the Lung-cancer and Arrhythmia dataset respectively. Additionally, an accuracy improvement of order 1.92% and 1.27% is observed on the Chemical and Promoters dataset respectively. Better classification results are achieved using the proposed method for Multi-features, Isolet, and CoIL200 datasets. In almost all of the datasets, and especially for Lung-cancer and Arrhythmia, GSM-CPD has a clear lead compared to the baselines which demonstrates the capability of our algorithm to select the most informative features. For the rest of the datasets (Splice, USCensus, and Musk2), the best performance is achieved when using all features. Even then, there is always another reduced-dimension subset (which is at most 10% of the original dataset dimensionality) that is very close in performance, namely only $0.33\%, 0.08, 1.27\%$ accuracy reduction for Splice, USCensus, and Musk2 respectively.

We also tested using another classifier, namely a radial basis function kernel support vector machine (RBF-kernel SVM), to confirm the consistency of our comparative analysis. The results are qualitative similar to those with the AdaBoost classifier. GSM-CPD can boost the accuracy of both classifiers better than the baselines can, with the Isolet dataset being the only exception, where JMIM has a slightly better classification performance than GSM-CPD. Nonetheless, in most cases, our GSM-CPD feature selection method appears to be very effective in feature selection, and often close to optimal in terms of classification accuracy. Therefore, we can design a classifier based on reduced dataset according to GSM-CPD to improve the classification accuracy or at least without sacrificing significant accuracy.

It is also useful to compare which method achieves the highest level of dimensionality reduction on average for each dataset. The results are presented on Table 3.6. We conclude that GSM-CPD can efficiently achieve high degree of dimensionality reduction and simultaneously enhance classification accuracy with predominant features. Specifically, GSM-CPD outperforms the baselines in context of dimensionality reduction on the Promoters, Splice, USCensus, CoIL2000, Musk2, and Arrhythmia datasets. For the remaining datasets, either JMIM or RJMI can achieve a better dimensionality reduction level, although the proposed method is very close

Data set	$F = 5$	$F = 10$	$F = 15$	$F = 20$	$F = 30$
Splice	0.916	0.911	0.902	0.895	0.855
Isolet	0.629	0.703	0.784	0.764	0.729

Table 3.4: Results for test-set accuracy for different choices of rank F.

Data set	#bins $= 5$	#bins$= 10$	#bins $= 20$	#bins $= 30$
Splice	0.916	0.912	0.899	0.809
Isolet	0.784	0.798	0.802	0.801

Table 3.5: Results for test-set accuracy for different number of bins.

to that. It is also notable that by tuning the value of K it is possible identify the intrinsic dimension of the dataset by identifying the minimum number of features needed before a significant degradation in prediction performance. Next, we investigate how hyper-parameters (the number of bins (for discretization) and the rank F) influence the performance of the proposed method. The results of a linear SVM classifier for the Splice dataset for different ranks F (and fixed number of bins = 5) are shown in Table 3.4. A rank of $5 - 10$ seems to be the best for getting the subset that leads to optimal accuracy. The decreased accuracy for higher ranks and/or more bins is expected due to overfitting. For Isolet, similar results can be observed for different ranks F (and fixed number of bins = 5). We then fix rank to $F = 5$ for Splice and $F = 15$ for Isolet and vary the number of bins (see Table 3.5). One can observe that the method is not very sensitive to the choice of rank and number of bins, but a careful parameter search can lead to optimal performance.

Datasets	**GSM** **CPD**	**MRMR**	**JMIM**	**RJMI**	**GlobalFS** **iSelect**
Lung-cancer	5	7	5	5	8
Promoters	4	4	5	5	4
Splice	6	7	12	11	10
USCensus	2	12	2	13	16
CoIL2000	9	10	12	17	20
Chemical	8	7	7	11	7
Musk2	4	10	6	11	11
Arrhythmia	17	25	25	26	28
Isolet	23	62	25	19	38
Multi-features	14	36	14	12	23

Table 3.6: Average number of selected features for each algorithm and dataset.

The double inequality in Claim 1 shows that we are maximizing a surrogate function $g(S)$ that is a constant band-gap away from the desired function $f(S)$. Intuitively, when the conditional entropy $H(Y|Z)$ is small, the amount of information needed to describe the outcome of the random variable Y given the value of the latent random variable Z is small, and thus the band-gap is small. If we estimate the principal components of the joint PMF, we can easily estimate the band-gap. In Table 3.7, we present the estimated entropy $H(Y)$ and conditional entropy $H(Y|Z)$ for the various datasets considered in our experiments. We can see that given Z, the entropy of the label reduces significantly, which further supports the maximization of the surrogate function.

Dataset	$H(Y)$	$H(Y\|Z)$
Lung-cancer	1.284	0.238
Promoters	0.835	0.147
Splice	0.750	0.250
USCensus	0.715	0.149
CoIL2000	0.974	0.260
Chemical	1.094	0.369
Musk2	0.879	0.267
Arrhythmia	0.902	0.159
Isolet	1.502	0.348
Multi-features	1.221	0.354

Table 3.7: Quantifying the marginal entropy of target random variable Y and conditional entropy between Y and latent random variable Z, $H(Y|Z)$.

3.6.2 Quantifying mutual information

Since both our method and the baselines attempt to maximize some measure of mutual information between the selected subset and the target variable / class label, it is useful to compare how effective the different methods are in doing just that – maximizing mutual information.

For our next set of experiments, we proceed by fixing the reduced feature space dimensionality K to 10% of the original feature space, extracting the corresponding subsets proposed by each of the baselines, and evaluating the MI of the extracted set. To quantify the MI, we use tree methods: our GSM-CPD model, the Mutual Information Neural Estimator (MINE) [56], and the Ensemble Dependency Graph Estimator (EDGE) [57]. Our GSM-CPD model evaluates the subset quality by approximating the high-order MI of the reduced subsets using the joint PMF estimate obtained by our CPD-based approach. MINE is a non-parametric estimator for computing MI by exploiting a lower-bound based on the Donsker-Varadhan representation of the KL divergence. The lower bound on the mutual information is estimated with a neural-net-parameterized function. We fix the network architecture to 5 fully connected layers with FC$(K, 200, 200, 100, 1)$ neurons, and set (batch size, learning rate) to $(100, 5 \times 10^{-4})$ for 1000 epochs. EDGE is a dependence graph-based approach. We report the mean MI estimate over 5 runs using our CPD-MI model (see Table 3.8), MINE (see Table 3.9), and EDGE (see Table 3.10). The results show that the subset produced by our framework is able to get consistently higher mutual information compared to widely-used information theoretic feature selection techniques

Data set	**GSM** **CPD**	**MRMR**	**JMIM**	**RJMI**	**GlobalFS** **iSelect**
Lung-cancer	**0.775**	**0.752**	0.708	**0.752**	0.723
Promoters	**0.896**	**0.877**	**0.877**	0.847	0.848
Splice	**0.952**	0.894	0.932	0.934	**0.947**
USCensus	**0.987**	0.979	**0.982**	0.979	0.937
CoIL2000	**0.846**	0.822	**0.840**	0.839	0.829
Chemical	**0.894**	**0.815**	0.857	**0.834**	0.836
Musk2	0.644	0.615	**0.697**	**0.657**	0.636
Arrhythmia	**0.689**	**0.678**	0.668	0.666	0.673
Isolet	**0.590**	0.5368	**0.543**	0.529	0.504
Multi-features	**0.761**	0.603	**0.759**	0.7375	0.722

Table 3.8: MI evaluation using our CPD-MI modeling on reduced feature vectors. The higher, the better.

on the majority of the considered datasets. The results presented in Tables 3.9 and 3.8 verify the superior quality of the subset proposed by our approach. The results are reassuring, in the sense that the MINE and GSM-CPD estimates of MI are close. The results in Table 3.10 show that even though EDGE tends to overestimate MI values relative to our GSM-CPD estimator and MINE, it assigns higher MI values to subsets generated by our proposed feature selection method.

Data set	**GSM** **CPD**	**MRMR**	**JMIM**	**RJMI**	**GlobalFS** **iSelect**
Lung-cancer	**0.778**	0.733	0.711	**0.751**	0.728
Promoters	**0.899**	0.891	**0.898**	0.859	0.863
Splice	**0.938**	0.909	0.929	0.911	**0.931**
USCensus	**0.982**	0.959	0.975	**0.979**	0.952
CoIL2000	**0.858**	0.820	**0.847**	0.840	0.846
Chemical	**0.895**	0.854	**0.877**	0.854	0.848
Musk2	0.675	0.634	**0.696**	**0.683**	0.671
Arrhythmia	**0.695**	**0.687**	0.669	0.653	0.684
Isolet	**0.595**	0.524	**0.598**	0.584	0.529
Multi-features	**0.750**	0.604	**0.749**	0.708	0.714

Table 3.9: MI evaluation by MINE on reduced feature vectors.

Data set	GSM CPD	MRMR	JMIM	RJMI	GlobalFS iSelect
Lung-cancer	$\underline{\mathbf{0.864}}$	0.830	0.832	**0.852**	0.792
Promoters	$\underline{\mathbf{0.995}}$	0.975	**0.978**	0.934	0.932
Splice	$\underline{\mathbf{1.384}}$	1.070	**1.135**	1.052	1.129
USCensus	$\underline{\mathbf{1.240}}$	1.092	1.126	**1.131**	1.065
CoIL2000	$\underline{\mathbf{0.999}}$	0.907	0.965	**0.990**	0.950
Chemical	$\underline{\mathbf{0.970}}$	0.920	**0.928**	0.905	0.910
Musk2	0.736	0.732	$\underline{\mathbf{0.783}}$	**0.766**	0.734
Arrhythmia	$\underline{\mathbf{0.782}}$	**0.774**	0.733	0.731	0.734
Isolet	**0.652**	0.604	$\underline{\mathbf{0.665}}$	0.630	0.626
Multi-features	$\underline{\mathbf{0.782}}$	0.625	**0.765**	0.706	0.726

Table 3.10: MI evaluation by EDGE on reduced feature vectors.

3.7 Chapter Summary

In this work, we presented a novel low-complexity approach for identifying the most predictive and informative subset of variables for classification and data analysis without compromising classification accuracy. In the first step, we model the joint PMF of the complete set of variables using a latent variable model following the "naive" Bayes hypothesis. In earlier work, it has been shown that such a model is *universal* – it can generate any joint distribution. In our present context it naturally suggests a monotone submodular surrogate optimization problem that is amenable to greedy optimization with performance guarantees. This gives rise to the proposed GSM-CPD feature selection approach. Experiments on real-world data show that GSM-CPD can outperform well-appreciated baseline methods by a significant margin. Our contributions can be summarized as follows.

- Using the CPD model to approximate the underlying distribution of the high-dimensional data, the CoD is alleviated.

- Employing high-order mutual information, complex multi-way interactions and inter-dependencies between the features are taken into account in the feature selection process.

- PMF modeling via CPD gives rise to an efficient greedy algorithm that comes with optimality guarantees for the surrogate problem of predicting the latent variable.

○ Experiments on real-world data show that GSM-CPD can outperform well-appreciated baseline methods by a significant margin.

Chapter 4

Low-rank Characteristic Tensor Density Estimation Part I: Foundations

4.1 Introduction

Density estimation is a fundamental yet challenging problem in statistical signal processing and machine learning. Density estimation is the task of learning the joint Probability Density Function (PDF) from a set of observed data points, sampled from an unknown underlying data-generating distribution. A model of the density function of a continuous random vector provides a complete description of the joint statistical properties of the data and can be used to perform tasks such as computing the most likely value of a subset of elements ("features") conditioned on others, computing any marginal or conditional distribution, and deriving optimal estimators, such as the minimum mean squared error (conditional mean) estimator. Density estimation has a wide range of applications including classification [19, 20, 58, 59], clustering [60], data synthesis [61], data completion [62] and reconstruction related applications [63], as well as learning statistical regularities such as skewness, tail behavior, multi-modality or other structures present in the data [64].

Existing work on density estimation can be mainly categorized into parametric approaches such as Gaussian Mixture Models (GMM) [65], and non-parametric approaches such as Histogram [10] and Kernel Density Estimation (KDE) [11]. A density model must be expressive – flexible enough to represent a wide class of distributions, and tractable and scalable (computationally and memory-wise) at the same time (expressivity-tractability trade-off). Over the last several years, explicit feed-forward neural network based density estimation methods [66–68] have gained increasing attention as they provide a tractable way to evaluate high-dimensional densities *point-wise*. On the other hand implicit *generative* models such as generative adversarial networks [17] and variational autoencoders [69] can be used to obtain models which allow effective and efficient sampling. Although neural network based solutions show promise in some high-dimensional applications such as image sampling, they are not currently well-suited for many other real-world applications. They lack the ability to compute expectations, marginalize over arbitrary subsets of variables, and evaluate conditionals, as they rather serve for point-wise density *evaluation*, or *sampling*. Additionally, model identifiability (i.e., recovery of the true data-generating distribution) is a fundamental goal of PDF estimation, which most deep generative models have not yet addressed. Incomplete observations (due to causes such as faulty sensors, corrupt data, cost of acquisition, or privacy concerns) present distinct challenges to the training of these models. The majority of such models are trained on complete data only and are unable to handle missing elements in the input vector, during both training and testing ("showtime").

In this paper, we develop a novel non-parametric method for multivariate PDF estimation based on tensor rank decomposition – known as *Canonical Polyadic Decomposition* (CPD) [25, 70]. CPD is a powerful model that can parsimoniously represent high-order data tensors exactly or approximately, and its distinguishing feature is that under certain reasonable conditions it is unique – see [23] for a recent tutorial overview. We show that **any compactly supported continuous density can be approximated, within controllable error, by a finite *characteristic tensor* of leading complex Fourier coefficients, whose size depends on the smoothness of the density**. This characteristic tensor can be naturally estimated via sample averaging from realizations of the random vector of interest.

The main challenge, however, lies in the fact that the size of this tensor (the number of model parameters in the Fourier domain) grows exponentially with the number of random variables – the length of the random vector of interest. In order to circumvent this "curse of dimensionality" (CoD) and further denoise the naive sample averaging estimates, we introduce a low-rank model of the characteristic tensor, whose **degrees of freedom (for fixed rank) grow linearly with the random vector dimension**. Low-rank modeling significantly improves the density estimate especially for high-dimensional data and/or in the sample-starved regime. By virtue of uniqueness of low-rank tensor decomposition, under certain conditions, **our method**

enables learning the true data-generating distribution.

In order to handle incomplete (both training and testing) data (vector realizations with missing entries) as well as scaling up to high-dimensional vectors, we further introduce coupled low-rank decomposition of lower-order characteristic tensors corresponding to smaller subsets of variables that share 'anchor' variables, and show that this still enables recovery of the global density, under certain conditions. As an added benefit, **our approach yields a generative model of the sought density, from which it is very easy to sample from**. This is because our low-rank model of the characteristic tensor admits a latent variable naive Bayes interpretation. A corresponding result for finite-alphabet random vectors was first pointed out in [18]. In contrast to [18], our approach applies to continuous random vectors possessing a compactly supported multivariate density function. From an algorithmic standpoint, we formulate a constrained coupled tensor factorization problem and develop a Block Coordinate Descent (BCD) algorithm.

The main results and contributions of this paper can be summarized as follows:

- We show that **any smooth compactly supported multivariate PDF can be approximated by a finite tensor model, without using any prior or data-driven discretization process**. We also show that truncating the sampled multivariate characteristic function of a random vector is equivalent to using a finite separable mixture model for the underlying distribution. Note that we do not assume a latent variable mixture model; instead the **latent variable factorization falls off from compactness of support and smoothness.** Under these relatively mild assumptions, **the proposed model can approximate any high dimensional PDF with approximation guarantees**. By virtue of uniqueness of CPD, assuming low-rank in the Fourier domain, **the underlying multivariate density is identifiable**.

- We show that **high dimensional joint PDF recovery is possible under low tensor-rank conditions, even if we only observe subsets (triples) of variables**. This is a key point that enables one to handle incomplete realizations of the random vector of interest. To the best of our knowledge, no other generic density estimation approach allows this. To tackle this more challenging version of the problem, we propose an optimization framework based on coupled tensor factorization. Our approach jointly learns lower-order (3-dimensional) characteristic functions, and then assembles tensor factors to synthesize the full characteristic function model.

- The proposed model allows efficient and low-complexity inference, sampling, and density evaluation. In that sense, it a more comprehensive solution that neural density evaluation or neural generative models. **We provide convincing experimental results on sampling, likelihood evaluation, and regression on toy, image, and many real datasets that are often used as benchmarks in our context. Our results corroborate the effectiveness of the proposed method even for datasets that have hundreds of variables.**

This is the first of a two-part paper. The second part builds on this foundation to develop a joint compression (nonlinear dimensionality reduction) and compressed density estimation framework that offers additional flexibility and scalability, but does not provide a density estimate in the original space as the "baseline" method in this first part does. Each approach has its own advantages, but the second builds upon the first. It is therefore natural to present them as Part I and Part II.

4.2 Background

4.2.1 Related work

Density estimation has been the subject of extensive research in statistics and the machine learning community. Methods for density estimation can broadly be classified as either parametric or non-parametric. Parametric density estimation assumes that the data are drawn from a known parametric family of distributions, parametrized by a fixed number of tunable parameters. Parameter estimation is usually performed by maximizing the likelihood of the observed data. One of the most widely used parametric models is the Gaussian Mixture Model (GMM). GMMs can approximate any density function if the number of components is large enough [9]. However, a very large number of components may be required for good approximation of the unknown density, especially in high dimensions. Increasing the number of components introduces computational challenges and may require a large amount of data [71]. Misspecification and inconsistent estimation is less likely to occur with nonparametric density estimation.

Nonparametric density estimation is more unassuming and in that sense "universal", but the flip-side is that it does not scale beyond a small number of variables (dimensions). The most widely-used approach for nonparametric density estimation is Kernel Density Estimation (KDE) [10, 11]. The key idea of KDE is to estimate the density by means of a sum of kernel functions centered at the given observations. However, worst-case theoretical results show that its performance worsens exponentially with the dimension of the data vector [72].

Our approach falls under nonparametric methods, and is motivated by Orthogonal Series Density Estimation (OSDE) [12–14], a powerful non-parametric estimation methodology. OSDE approximates a probability density function using a truncated sum of orthonormal basis functions, which may be trigonometric, polynomial, wavelet etc. However, OSDE becomes computationally intractable even for as few as 10 dimensions, since the number of parameters grows exponentially with the number of dimensions. Unlike OSDE, our approach is able to scale to much higher dimensions.

The first work using tensor decomposition to establish identifiability of latent variable models was [73], where it was shown that, under certain conditions, a finite mixture of non-parametric product distributions is identifiable. The linear independence conditions mentioned in [73] are in fact not necessary for uniqueness; a milder condition pertaining to the sum of Kruskal-ranks of the latent factor matrices is in fact sufficient [27]. Also, [73] did not provide a companion density estimation procedure, which limits its applicability.

Later on, [74, 75] proposed using whitening-based orthogonal tensor decomposition to recover the parameters of certain latent variable (not general density) models – but this algorithm is not always feasible [76] because the whitening step cannot always find a positive semi-definite matrix via linear combination of tensor slices. This happens with positive probability [76], in which case the orthogonal decomposition algorithm fails altogether. Furthermore, if the rank of this matrix is lower than the tensor rank, then the algorithm has a "soft" failure. We also note that the approach in [74] is a kernel method that involves eigenvalue decomposition of M by M matrices, where M is the training sample size, which is prohibitive for large training sets. Our proposed algorithm is scalable (its complexity is linear in M) and it avoids earlier pitfalls. It is also worth re-iterating that, whereas there have been prior works dealing with multivariate density estimation for latent variable models such as [73, 74], our work is the first to use tensor models for high-dimensional densities, where the dimensionality is well above $3 - 10$. Part of our novelty is showing that low-rank tensor factorization can work, with remarkably low ranks, in these high dimensions (up to 256 here).

Another method also based on low-rank tensor decompositions with theoretical guarantees of identifiability (for distributions of low enough rank) has been presented in [77]. In contrast to [73, 74] and [77], our approach is "universal" for smooth, compactly supported multivariate densities, i.e., no assumptions regarding a multivariate mixture model of non-parametric product distributions are made in the present paper; we show that a latent variable factorization falls off from compactness of support and smoothness. A similar approach to [77] was considered in [78], where a tensor train model is used to approximate a discretized PDF, followed by interpolation. The authors use a conditioning chain and compute each conditional distribution given the model for the full joint distribution – this computation depends on the ordering of the variables. There are no identifiability guarantees, and the method is geared towards sampling applications. This is natural, since tensor train models do not offer easy marginalization and inference.

Recently, several density evaluation and modeling methods that rely on neural networks have been proposed. The Real-valued Neural Autoregressive Distribution Estimator (RNADE) [79] is among the best-performing neural density estimators and has shown great potential in scaling to high-dimensional distribution settings. These so-called autoregressive models (not to be confused with classical AR models for time-series) decompose the joint density as a product of one-dimensional conditionals of increasing conditioning order, and model each conditional density with a parametric model. Normalizing Flows (NF) [80] models start with a base density e.g., standard Gaussian, and stack a series of invertible transformations with tractable Jacobian to approximate the target density. Masked Autoregressive Flow (MAF) [68] is a type of NF model, where the transformation layer is built as an autoregressive neural network. These methods do not construct a joint PDF model but rather serve for point-wise density *evaluation*. That is, for any given input vector (realization), they output an estimate of the density evaluated at that particular input vector (point). For small vector dimensions, e.g., two or three, it is possible to evaluate all inputs on a dense grid, thereby obtaining a histogram-like density estimate; but the curse of dimensionality kicks in for high vector dimensions, where this is no longer an option. Additionally, these methods cannot impute more than very few missing elements in the input, for the same reason (grid search becomes combinatorial).

Another class of neural network density models are the sum-product networks (SPNs) [81,82]. SPNs are deep probabilistic models, represented by a directed acyclic graph with univariate distributions at the leaves, that decompose a joint distribution into a hierarchy of mixtures (sums) and factorizations (products). Their extension to continuous variables assumes a model for the one-dimensional marginals, e.g., Gaussian or mixture of Gaussians for each input variable, in which case the overall distribution is a mixture of separable Gaussians. In the discrete (finite-alphabet / categorical) case, our model [18] can be interpreted as a shallow SPN; in the continuous case, as considered in this paper, we do not make any assumption on the one-dimensional marginals at the leaves, so our approach can be viewed as a nonparametric shallow SPN. From the viewpoint of SPNs, we show in this paper that

1. a shallow (one-sum layer) SPN is a universal model for smooth and compactly supported multivariate densities, and the underlying 1-D densities can be recovered (versus prescribed). That is, if the true density has low rank, then it can be pinned down and its components can be 'unraveled' via the CPD.

2. the low-rank assumption works well in various practical applications.

A multi-layer SPN, on the other hand, is akin to a hierarchical Tucker model, and thus no model identifiability claims can be made for deep SPNs – they can always be replaced by a shallow SPN with sufficiently many leaves.

SPNs enjoy a tractable marginalization and inference process, but our proposed model allows for even easier marginalization (by discarding the subset of factor matrices corresponding to the variables we are not interesting in), as well as easier and closed form inference. SPNs require specific structural constraints in order to guarantee exact inference [81]. In contrast to SPNs, our model does not require architecture specification.

4.2.2 Notation

In this Chapter, we use the symbols $\mathbf{x}$, $\mathbf{X}$, $\underline{\mathbf{X}}$ for vectors, matrices and tensors respectively. We use the notation $\mathbf{x}(n)$, $\mathbf{X}(:, n)$, $\underline{\mathbf{X}}(:, :, n)$ to refer to a particular element of a vector, a column of a matrix and a slab of a tensor.

4.3 A Characteristic Function Approach

The characteristic function of a random variable X is the Fourier transform of its PDF, and it conveys all information about X. The characteristic function can be interpreted as an expectation: the Fourier transform at frequency $\nu \in \mathbb{R}$ is $E\left[e^{j\nu X}\right]$. Similarly, the multivariate characteristic function is the multidimensional Fourier transform of the density of a random vector $\boldsymbol{X}$, which can again be interpreted as the expectation $E[e^{j\boldsymbol{\nu}^T\boldsymbol{X}}]$, where $\boldsymbol{\nu}$ is a vector of frequency variables. The expectation interpretation is crucial, because ensemble averages can be estimated via sample averages; and whereas direct nonparametric density estimation at point x requires samples around x, estimating the characteristic function enables reusing all samples globally, thus enabling better sample averaging and generalization. This point is the first key to our approach. The difficulty, however, is that pinning down the characteristic function seemingly requires estimating an uncountable set of parameters. We need to reduce this to a *finite* parameterization with controllable error, and ultimately distill a parsimonious model that can learn from limited data and still generalize well. In order to construct an accurate joint distribution estimate that is scalable to high dimensions without making explicit and restrictive prior assumptions (such as a GMM model) on the nature of the density, and without requiring huge amounts of data, we encode the following key ingredients into our model.

- **Compactness of support**. In most cases, the random variables of interest are bounded, and these bounds are known or can be relatively easily estimated. The assumption of knowing the support of the sought distribution is not limiting in practice. Given that we are aiming to estimate a high-dimensional distribution we should naturally have access to much more data than is needed to estimate the support of any marginal distribution of one of the variables. Also note that we do not need to know the exact support – reasonable upper and lower bounds are enough to compress and shift the range.

- **Continuity of the underlying density and its derivatives**. The joint distribution is assumed to be sufficiently smooth in some sense, which enables the use of explicit or implicit interpolation.

- **Low-rank tensor modeling**. We show that joint characteristic functions can be represented as higher order tensors. In practice these tensor data are not unstructured. Low-rank tensor modeling provides a concise representation that captures the salient characteristics (the *principal components*) of the data distribution in the Fourier domain.

4.3.1 The Univariate Case

Before we delve into the multivariate setting, it is instructive to examine the univariate case. Given a real-valued random variable X with compact support S_X, the Probability Density Function (PDF) f_X and its corresponding Characteristic Function (CF) Φ_X form a Fourier transform pair:

$$\Phi_X(\nu) := \int_{S_X} f_X(x) e^{j\nu x} dx = E[e^{j\nu X}], \tag{4.1}$$

$$f_X(x) := \frac{1}{2\pi} \int_{-\infty}^{\infty} \Phi_X(\nu) e^{-j\nu x} d\nu. \tag{4.2}$$

Note that $\Phi_X(0) = \int_{-\infty}^{\infty} f_X(x)dx = 1$. Without loss of generality, we can apply range normalization and mean shifting so that $sX + c \in [0, 1]$ – the transformation is invertible. We may therefore assume that $S_X = [0, 1]$. Every PDF supported in $[0, 1]$ can be uniquely represented over its support by an infinite Fourier series,

$$f_X(x) = \sum_{k=-\infty}^{\infty} \Phi_X[k] e^{-j2\pi kx}, \tag{4.3}$$

where $\Phi_X[k] = \Phi_X(\nu)\big|_{\nu=2\pi k}, \quad k \in \mathbb{Z}$. This shows that *countable* parameterization through samples of the characteristic function suffices for compactly supported densities. But this is still not enough - we need a finite parametrization. Thankfully, if f_X is sufficiently differentiable in the sense that $f_X \in C^p$ i.e., all its derivatives $\frac{\partial f_X}{\partial x}, \frac{\partial^2 f_X}{\partial x^2}, \cdots, \frac{\partial^p f_X}{\partial x^p}$ exist and are continuous we have that

Lemma 4.1. *(e.g., see [83]): If $f_X \in C^p$, then*

$$|\Phi_X[k]| = \mathcal{O}\left(\frac{1}{1 + |k|^p}\right).$$

It is therefore possible to use a truncated series

$$\widehat{f}_X(x) = \sum_{k=-K}^{K} \Phi_X[k] e^{-j2\pi kx},$$

with proper choice of K that will not incur significant error. Invoking Parseval's Theorem

$$\|f_X - \widehat{f}_X\|_2^2 = \sum_{|k|>K} |\Phi_X[k]|^2,$$

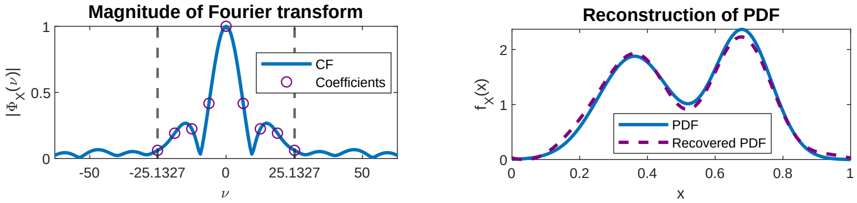


Figure 4.1: The Univariate Case: Illustration of the key idea on a univariate Gaussian mixture of two distributions with means $\mu_1 = 0.35$, $\mu_2 = 0.7$ and standard deviations $\sigma_1 = 0.1$, $\sigma_2 = 0.08$. The PDF can be (approximately) recovered from only 9 uniform samples of its Characteristic Function (CF).

which is controllable by the *smoothing parameter* K. The k-th Fourier coefficient

$$\Phi_X[k] = \int_0^1 f_X(x) e^{j2\pi kx} dx = E[e^{j2\pi kX}]$$

can be conveniently estimated via the sample mean

$$\widehat{\Phi}_X[k] = \frac{1}{M} \sum_{m=1}^{M} e^{j2\pi k x_m}$$

Here M is the number of available realizations of the random variable X.

A toy example to illustrate the idea is shown in Figure 4.1. For this example, we are given $M = 500$ realizations of a random variable X, which is a mixture of two Gaussian distributions with means $\mu_1 = 0.35$, $\mu_2 = 0.7$ and standard deviations $\sigma_1 = 0.1$, $\sigma_2 = 0.08$. The recovered PDF is very close to the true PDF using only 9 coefficients of the CF.

4.3.2 The Multivariate Case

In the multivariate case, we are interested in obtaining an estimate $\widehat{f_{\boldsymbol{X}}}$ of the true density $f_{\boldsymbol{X}}$ of a random vector $\boldsymbol{X} := [X_1, \ldots, X_N]^T$. The *joint* or *multivariate characteristic function* of $\boldsymbol{X}$ is a function $\Phi_{\boldsymbol{X}} : \mathbb{R}^N \to \mathbb{C}$ defined as

$$\Phi_{\boldsymbol{X}}(\boldsymbol{\nu}) := E\left[e^{j\boldsymbol{\nu}^T \boldsymbol{X}}\right], \tag{4.4}$$

where $\boldsymbol{\nu} := [\nu_1, \ldots, \nu_N]^T$. For any given $\boldsymbol{\nu}$, given a set of realizations $\{\mathbf{x}_m\}_{m=1}^M$, we can estimate the empirical characteristic function of the sequence as

$$\widehat{\Phi}_{\boldsymbol{X}}(\boldsymbol{\nu}) = \frac{1}{M} \sum_{m=1}^{M} e^{j\boldsymbol{\nu}^T \mathbf{x}_m}. \tag{4.5}$$

Under mixing conditions such that sample averages converge to ensemble averages, the corresponding PDF can be uniquely recovered via the multidimensional inverse Fourier transform

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{(2\pi)^N} \int_{\mathbb{R}^N} \Phi_{\boldsymbol{X}}(\boldsymbol{\nu}) e^{-j\boldsymbol{\nu}^T \boldsymbol{x}} d\boldsymbol{\nu}. \tag{4.6}$$

If the support of the joint PDF $f_{\boldsymbol{X}}(\mathbf{x})$ is contained within the hypercube $S_{\boldsymbol{X}} = [0,1]^N$, then similar to the univariate case, it can be represented by a multivariate Fourier series

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \sum_{k_1=-\infty}^{\infty} \cdots \sum_{k_N=-\infty}^{\infty} \Phi_{\boldsymbol{X}}[\boldsymbol{k}] e^{-j2\pi \mathbf{k}^T \mathbf{x}}, \text{where } \Phi_{\boldsymbol{X}}[\mathbf{k}] = \Phi_{\boldsymbol{X}}(\boldsymbol{\nu})\big|_{\boldsymbol{\nu}=2\pi\mathbf{k}}, \mathbf{k} = [k_1, \ldots, k_N]^T. \tag{4.7}$$

Lemma 4.2. *(see e.g., [83]):* For any $p \in \mathbb{N}$, if the partial derivatives $\frac{\partial^{\theta_1}}{\partial x_1^{\theta_1}} \cdots \frac{\partial^{\theta_N}}{\partial x_N^{\theta_N}} f_{\boldsymbol{X}}(\mathbf{x})$ exist and are absolutely integrable for all $\theta_1, \ldots, \theta_N$ with $\sum_{n=1}^N \theta_n \leq p$ then the rate of decay of the magnitude of the $\mathbf{k}$-th Fourier coefficient $|\Phi_{\boldsymbol{X}}[\mathbf{k}]|$ obeys $|\Phi_{\boldsymbol{X}}[\mathbf{k}]| = \mathcal{O}\left(\frac{1}{1+\|\mathbf{k}\|_2^p}\right)$.

The smoother the underlying PDF, the faster its Fourier coefficients and the approximation error tend to zero. Thus we can view the joint PDF through the lens of functions with only low frequency harmonics. Specifically, it is known [84], [85, Chapter 23] that the approximation error of the truncated series with absolute cutoffs $\{K_n\}_{n=1}^N$ is upper bounded by

$$\|f_{\boldsymbol{X}} - \widehat{f}_{\boldsymbol{X}}\|_\infty \leq C \sum_{n=1}^{N} \frac{\omega_n\left(\frac{\partial^{\theta_n}}{\partial x_n^{\theta_n}} f_{\boldsymbol{X}}, \frac{1}{1+K_n}\right)}{(1+K_n)^{\theta_n}}, \tag{4.8}$$

where

$$\omega_n(f_{\boldsymbol{X}}, \delta) := \sup_{|x_j - x_j'| \leq \delta} \left| f_{\boldsymbol{X}}(x_1, \ldots, x_j, \ldots, x_N) - f_{\boldsymbol{X}}(x_1, \ldots, x_j', \ldots, x_N) \right|,$$

and

$$C = C_2 \left(1 + C_1 \prod_{n=1}^{N} \log K_n \right).$$

C_1, C_2 are constants independent of K_n. The smoother the underlying PDF, the smaller the obtained finite parametrization error. It follows that we can approximate $f_{\boldsymbol{X}}$

$$\widehat{f}_{\boldsymbol{X}}(\mathbf{x}) = \sum_{k=-K_1}^{K_1} \cdots \sum_{k_N=-K_N}^{K_N} \Phi_{\boldsymbol{X}}[\mathbf{k}] e^{-j2\pi \mathbf{k}^T \mathbf{x}}. \tag{4.9}$$

The truncated Fourier coefficients can be naturally represented using an N-way tensor $\underline{\boldsymbol{\Phi}}$ where

$$\underline{\boldsymbol{\Phi}}(k_1, \ldots, k_N) = \Phi_{\boldsymbol{X}}[\mathbf{k}]. \tag{4.10}$$

4.4 Proposed Approach: Breaking the Curse of Dimensionality

We have obtained a finite parameterization with controllable and bounded error, but the number of parameters $(2K_1 + 1) \times \cdots \times (2K_N + 1)$ obtained by truncating $\Phi_{\boldsymbol{X}}$ as above grows exponentially with N. This curse of dimensionality can be circumvented by focusing on the *principal components* of the resulting tensor, i.e., introducing a low-rank parametrization of the *Characteristic Tensor* obtained by truncating the multidimensional Fourier series. Keeping the first F principal components, the number of parameter reduces from order of $K_1 \times \cdots \times K_N$ to order of $(K_1 + \cdots + K_N)F$. Introducing the rank-F CPD in Equation (4.9), one obtains the approximate model

$$\tilde{f}_{\boldsymbol{X}}(\mathbf{x}) = \sum_{k_1=-K}^{K} \cdots \sum_{k_N=-K}^{K} \sum_{h=1}^{F} p_H(h) \prod_{n=1}^{N} \Phi_{X_n|H=h}[k_n] e^{-j2\pi k_n x_n}, \tag{4.11}$$

where H can be interpreted as a latent (H for 'hidden') random variable, $\Phi_{X_n|H=h}[k_n]$ is the characteristic function of X_n conditioned on $H = h$

$$\begin{aligned}\Phi_{X_n|H=h}[k_n] &:= \Phi_{X_n|H=h}(\nu|h)\big|_{\nu=2\pi k_n} \\ &= \mathbb{E}_{X_n|H=h}\left[e^{j2\pi k_n X_n}\right],\end{aligned} \tag{4.12}$$

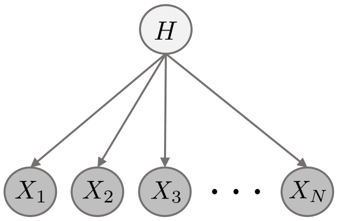


Figure 4.2: The proposed generative model $\tilde{f}_{\boldsymbol{X}}(\mathbf{x})$ admits an interpretation as a mixture of F product distributions i.e., a latent variable naive Bayes interpretation.

and we stress that for high-enough F, this representation is exact without loss of generality – see, e.g., [23]. For the rest of the paper, we consider $K = K_1 = \cdots = K_N$ for brevity.

By linearity and separability of the multidimensional Fourier transformation it follows that

$$\begin{aligned}\tilde{f}_{\boldsymbol{X}}(\mathbf{x}) &= \sum_{h=1}^{F} p_H(h) \prod_{n=1}^{N} \sum_{k_n=-K}^{K} \Phi_{X_n|H=h}[k_n]\, e^{-j2\pi k_n x_n} \\ &= \sum_{h=1}^{F} p_H(h) \prod_{n=1}^{N} f_{X_n|H}(x_n|h). \end{aligned} \tag{4.13}$$

This generative model can be interpreted as mixture of product distributions [77]. The joint PDF $f_{\boldsymbol{X}}$ is a mixture of F separable component PDFs, i.e., there exists a ‘hidden’ random variable H taking values in $\{1, \ldots, F\}$ that selects the operational component of the mixture, and given H the random variables $X_1, \ldots, X_N$ become independent (See Figure 4.2 for visualization of this model). We have thus shown the following result:

Proposition 4.1. *Truncating the multidimensional Fourier series (sampled multivariate characteristic function) of any compactly supported random vector is equivalent to approximating the corresponding multivariate density by a finite mixture of separable densities.*

Thus, by choosing appropriate K and F, it is possible to represent and approximate any compactly supported density that it is sufficiently smooth by the proposed model. See Figures 4.4, 4.3 where we showcase how each parameter affects the modeling of complex structures in 2D synthetic datasets.

Conversely, if one *assumes* that the sought multivariate density is a finite mixture of separable densities, then it is easy to show that the corresponding characteristic function is likewise a mixture of separable characteristic functions:

$$\begin{aligned}\Phi_{\boldsymbol{X}}(\boldsymbol{\nu}) &= E\left[e^{j\boldsymbol{\nu}^T\boldsymbol{X}}\right] \\ &= E_H\left[E_{\boldsymbol{X}|H}\left[e^{j\nu_1 X_1}\cdots e^{j\nu_N X_N}\right]\right] \\ &= E_H\left[\Phi_{X_1|H}(\nu_1|H)\cdots\Phi_{X_n|H}(\nu_N|H)\right] \\ &= \sum_{h=1}^{F} p_H(h)\prod_{n=1}^{N}\Phi_{X_n|H}(\nu_n|h). \end{aligned} \tag{4.14}$$

If we sample the above on any finite N-dimensional grid, we obtain an N-way tensor and its polyadic decomposition. Such decomposition is unique, under mild conditions [23]. It follows that:

Proposition 4.2. *A compactly supported multivariate ($N \geq 3$) mixture of separable densities is identifiable from (samples of) its characteristic function, under mild conditions.*

The main reasons for working in the Fourier / characteristic function domain are that

1. truncation is a “universal” approximation in the sense that it only requires smoothness of the joint PDF;
2. the Fourier transform is “global” allowing us to estimate the PDF in places where there is a scarcity of “local” samples – which is a key problem in high-dimensional cases;
3. since we limit ourselves to regular samples of the characteristic function (multivariate Fourier series), we can invert using the computationally advantageous Fast Fourier Transform; and
4. relative to the moment generating function, the characteristic function always exists.

The above analysis motivates the following course of action. Given a set of realizations $\{\mathbf{x}_m\}_{m=1}^{M}$,

1. estimate

$$\underline{\mathbf{\Phi}}[\mathbf{k}] = \frac{1}{M}\sum_{m=1}^{M} e^{j2\pi\mathbf{k}^T\mathbf{x}_m}, \tag{4.15}$$

2. fit a low-rank model

$$\underline{\mathbf{\Phi}}[\mathbf{k}] \approx \sum_{h=1}^{F} p_H(h) \prod_{n=1}^{N} \Phi_{X_n|H=h}[k_n], \tag{4.16}$$

3. and invert using

$$f_{\mathbf{X}}(\mathbf{x}) = \sum_{h=1}^{F} p_H(h) \prod_{n=1}^{N} f_{X_n|H}(x_n|h), \tag{4.17}$$

where $f_{X_n|H}(x_n|h) = \sum_{k_n=-K}^{K} \Phi_{X_n|H=h}[k_n] e^{-j2\pi k_n x_n}$. When building any statistical model, identifiability is a fundamental question. A statistical model is said to be identifiable when, given a sufficient number of observed data, it is possible to uniquely recover the data-generating distribution. When applying a non-identifiable model, different structures or interpretations may arise from distinct parametrizations that explain the data equally well. Most deep generative models do not address the question of identifiability, and thus may fail to deliver the true latent representations that generate the observations. Our approach is fundamentally different, because it builds on rigorous and controllable Fourier approximation and identifiability of the characteristic tensor.

In the Appendix (Section 4.7), we provide additional statistical insights regarding the proposed methodology, including the asymptotic behavior of the empirical characteristic function and the mean squared error reduction afforded by low-rank tensor modeling in the characteristic function domain.

Two issues remain. First, uniqueness of CPD only implies that each rank-one factor is unique, but leaves scaling/counter-scaling freedom in p_H and the conditional characteristic functions. To resolve this, we can use the fact that each conditional characteristic function must be equal to 1 at the origin (zero frequency). Likewise, p_H must be a valid probability mass function. These constraints fix the scaling indeterminacy.

We note here that, under certain rank conditions on the Fourier series coefficient tensor, the proposed method ensures that the reconstructed density is positive and integrates to one, as it should. This is due to the uniqueness properties of the Fourier series representation and the CPD: if there exists a density that generates a low-rank characteristic tensor, and that tensor can be uniquely decomposed, the sum of Fourier inverses of its components is unique, and therefore equal to the generating density. Under ideal low-rank conditions, this is true even if we ignore the constraints implied by positivity when we decompose the characteristic tensor in the Fourier domain. This is convenient because strictly enforcing those in the Fourier domain would entail cumbersome spectral factorization-type (positive semidefinite) constraints. We therefore propose the following formulation:

$$\begin{aligned} \min \quad & \|\underline{\mathbf{\Phi}} - [\![\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N]\!]\|_F^2 \\ \text{subject to} \quad & \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{1}^T\boldsymbol{\lambda} = 1, \\ & \mathbf{A}_n(K+1,:) = \mathbf{1}^T,\ n = 1 \ldots N, \end{aligned} \tag{4.18}$$

where $\mathbf{A}_n(K+1+k_n, h)$ holds $\Phi_{X_n|H=h}[k_n]$, and $\boldsymbol{\lambda}(h)$ holds $p_H(h)$.

The second issue is more important. When N is large, instantiating or even allocating memory for the truncated characteristic tensor is a challenge, because its size grows exponentially with N. Fortunately, there is a way around this problem. The main idea is that instead of estimating the characteristic tensor of all N variables, we may instead estimate the characteristic tensors of subsets of variables, such as triples, which partially share variables with other triples. The key observation that enables this approach is that the marginal characteristic function of any subset of random variables is also a constrained complex CPD model that inherits parameters from the grand characteristic tensor. Marginalizing with respect to the n'-th random variable, we have that

$$\underline{\mathbf{\Phi}}(k_1, \ldots, k_{n'} = 0, \ldots, k_N) = \sum_{h=1}^{F} \prod_{\substack{n=1 \\ n \neq n'}}^{N} \Phi_{X_n|H}[k_n] \underbrace{\Phi_{X_{n'}|H}[0]}_{=1} = \sum_{h=1}^{F} \prod_{\substack{n=1 \\ n \neq n'}}^{N} \Phi_{X_n|H}[k_n]. \tag{4.19}$$

Thus, a characteristic function of any subset of three random variables X_i, X_j, X_ℓ (triples) can be written as a third-order tensor, $\underline{\mathbf{\Phi}}_{ij\ell}$, of rank F. These sub-tensors can be jointly decomposed in a coupled fashion (see the optimization problem in (4.21)) to obtain the sought factors that allow synthesizing the big characteristic tensor. In this way, we beat the curse of dimensionality for low-enough model ranks. In addition to affording significant computational and memory reduction, unlike neural network based methods, the above approach allows us to work with fewer and even missing data during the training phase, i.e., only having access to incomplete realizations of the random vector of interest. We estimate lower-order characteristic function values from only those realizations that all three random variables in a given triple appear together. Our method can easily be adapted to work with pairs or quadruples, but reliably estimating fourth-order characteristic functions requires more sample averaging, whereas the $2-$

dimensional case requires stricter identifiability conditions. Hence working with 3-D tensors offers a good compromise between these conflicting considerations.

In earlier work, we proposed a similar approach for the categorical case where every random variable is finite-alphabet and the task is to estimate the joint probability mass function (PMF) [18]. There we showed that 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 lower-order marginals – which is reminiscent of Kolmogorov extension.

In case of continuous random variables, however, the joint PDF can no longer be directly represented by a tensor. One possible solution could be discretization, but this unavoidably leads to discretization error. In this work, what we show is that we can *approximately* represent any smooth joint PDF (and evaluate it at any point) using a low-rank tensor *in the characteristic function domain*, thereby avoiding discretization loss altogether.

Our joint PDF model enables easy computation of any marginal or conditional density of subsets of variables of $\boldsymbol{X}$. Using the conditional expectation, the response variable, taken without loss of generality to be the last variable X_N, can be estimated in the following way (see detailed derivation in Section 4.7.).

$$E\left[X_N|X_1,\ldots,X_{N-1}\right] = \frac{1}{c_1}\sum_{h=1}^{F}\boldsymbol{\lambda}(h)\prod_{n=1}^{N-1}\sum_{k_n=-K}^{K}\mathbf{A}_n(k_n,h)e^{-j2\pi k_n x_n}\sum_{k_N=-K}^{K}c_{2,k_N}\mathbf{A}_N(k_N,h) \tag{4.20}$$

where $c_1 = \sum_{h=1}^{F}\boldsymbol{\lambda}(h)\prod_{n=1}^{N-1}\sum_{k_n=-K}^{K}\mathbf{A}_n(k_n,h)e^{-j2\pi k_n x_n}$and $c_{2,k_N} = \dfrac{e^{-j2\pi k_N}}{-j2\pi k_N} + \dfrac{1-e^{-j2\pi k_N}}{[-j2\pi k_N]^2}$.

One of the very appealing properties of the proposed approach is that it is a generative model that affords easy sampling. According to Equation (4.17), a sample of the multivariate distribution can be generated by first drawing H according to p_H and then independently drawing samples for each variable X_n from the conditional PDF $f_{X_n|H}$. The resulting generative model can be visualized in Figure 4.2.

Algorithm 4.2 Low-Rank Characteristic Function based Density Estimation (LRCF-DE).

```
Input: A real-valued dataset $D \in \mathbb{R}^{N \times M}$, parameters $F, K$.
Output: The joint PDF model $f_{\boldsymbol{X}}$.
Compute $\underline{\mathbf{\Phi}}_{ij\ell} \forall i, j, \ell \in \{1, \ldots, N\},\ \ell > j > i$ from training data, using (4.15).
Initialize $\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N$ in compliance with their constraints.
repeat
    for all $n \in \{1, \ldots, N\}$ do
        Solve the optimization problem with respect to $\mathbf{A}_n$ defined in (4.22).
    end for
    Update $\boldsymbol{\lambda}$ by solving the optimization problem defined in (4.24).
until convergence criterion satisfied
Assemble the joint PDF as in equation (4.26).
```

4.4.1 Algorithm: Coupled Tensor Factorization

We formulate the problem as a coupled complex tensor factorization problem and propose a Block Coordinate Descent algorithm for recovering the latent factors of the CPD model representing the joint CF. Then, we only need to invert each conditional CF and synthesize the joint PDF. We refer to this approach as Low-Rank Characteristic Function based Density Estimation (LRCF-DE).

We begin by defining the following coupled tensor factorization problem

$$\begin{aligned} \min_{\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N} \quad & \sum_{i} \sum_{j>i} \sum_{\ell>j} \left\| \underline{\mathbf{\Phi}}_{ij\ell} - [\![\boldsymbol{\lambda}, \mathbf{A}_i, \mathbf{A}_j, \mathbf{A}_\ell]\!] \right\|_F^2 \\ \text{subject to} \quad & \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{1}^T \boldsymbol{\lambda} = 1, \\ & \mathbf{A}_n(K+1, :) = \mathbf{1}^T,\ n = 1, \ldots, N. \end{aligned} \tag{4.21}$$

Each lower-dimensional joint CF of triples, $\underline{\mathbf{\Phi}}_{ij\ell}$, can be computed directly from the observed data via sample averaging according to equation (4.15). The formulated optimization problem (4.21) is non-convex and NP-hard. However it becomes convex with respect to each variable if we fix the remaining ones and can be handled using alternating optimization. By using the mode-1 matrix unfolding of each tensor $\underline{\mathbf{\Phi}}_{ij\ell}$, the optimization problem with respect to $\mathbf{A}_i$ becomes

$$\begin{aligned} \min_{\mathbf{A}_i} \quad & \sum_{j \neq i} \sum_{\ell \neq i, \ell > j} \left\| \underline{\mathbf{\Phi}}_{ij\ell}^{(1)} - (\mathbf{A}_\ell \odot \mathbf{A}_j) \text{diag}(\boldsymbol{\lambda}) \mathbf{A}_i^T \right\|_F^2 \\ \text{subject to} \quad & \mathbf{A}_i(K+1, :) = \mathbf{1}^T. \end{aligned} \tag{4.22}$$

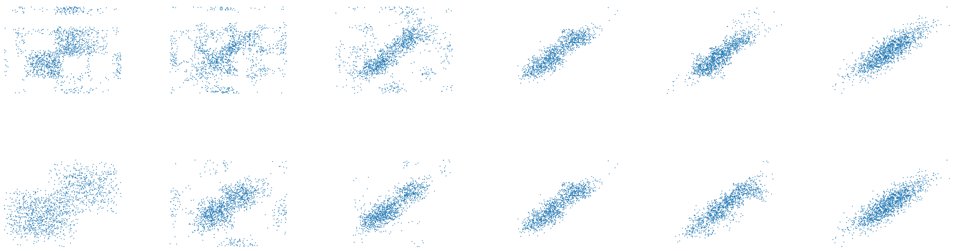


Figure 4.3: Visualization of synthetic $M' = 1500$ samples generated from the proposed model trained on the Weight-Height dataset for different F, K parameter combinations – The rightmost figure represents the ground truth. On the first row, we fixed K, $K = 4$, and varied F, $F \in [2, 4, 6, 8, 10]$ (from left to right). On the second row, we fixed F, $F = 8$, and varied K, $K \in [1, 2, 3, 4, 5]$ (from left to right).

The exact update for each factor $\mathbf{A}_i$ can be computed as

$$\mathbf{A}_i \leftarrow \mathbf{G}_i^{-1}\mathbf{V}_i, \tag{4.23}$$

where

$$\begin{aligned}
\mathbf{G}_i &= (\boldsymbol{\lambda}\boldsymbol{\lambda}^T) \circledast \sum_{j \neq i} \sum_{\ell \neq i, \ell > j} \mathbf{Q}_{\ell j}^H \mathbf{Q}_{\ell j}, \\
\mathbf{V}_i &= \operatorname{diag}(\boldsymbol{\lambda}) \sum_{j \neq i} \sum_{\ell \neq i, \ell > j} \mathbf{Q}_{\ell j}^H \underline{\boldsymbol{\Phi}}_{ij\ell}^{(1)}, \\
\mathbf{Q}_{\ell j} &= \mathbf{A}_\ell \odot \mathbf{A}_j.
\end{aligned}$$

For each update, the row of $\mathbf{A}_i$ that corresponds to zero frequency is removed and updating $\mathbf{A}_i$ becomes an unconstrained complex least squares problem. A vector of ones is appended at the same row index after each update $\mathbf{A}_i$. Due to role symmetry the same form holds for each factor $\mathbf{A}_n$.

Now, for the $\boldsymbol{\lambda}$-update we solve the following optimization problem

$$\begin{aligned}
\min_{\boldsymbol{\lambda}} \quad & \sum_i \sum_{j>i} \sum_{\ell>j} \left\| \operatorname{vec}(\underline{\boldsymbol{\Phi}}_{ij\ell}) - (\mathbf{A}_\ell \odot \mathbf{A}_j \odot \mathbf{A}_i)\boldsymbol{\lambda} \right\|_F^2 \\
\text{subject to} \quad & \boldsymbol{\lambda} \geq \mathbf{0}, \; \mathbf{1}^T\boldsymbol{\lambda} = 1.
\end{aligned} \tag{4.24}$$

The optimization problem (4.24) is a least squares problem with a probability simplex constraint.

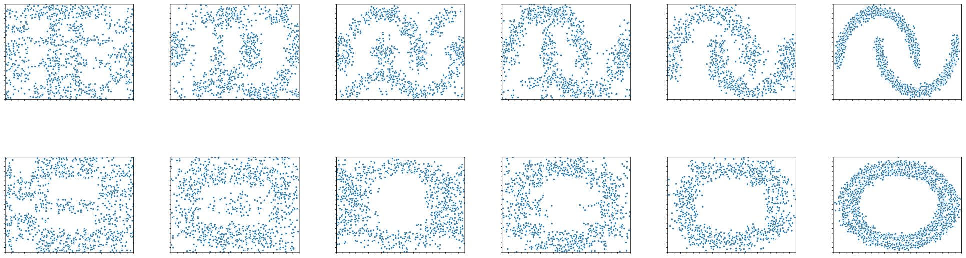


Figure 4.4: Qualitative synthetic $M' = 1500$ samples obtained from the proposed model trained on $M = 2000$ samples of a toy 2-D Moons and Circles datasets (for fixed K, $K = 11$, from left to right $F \in [1, 2, 3, 4, 6]$ – The rightmost figures represent the ground truth).

We use an ADMM algorithm to tackle it. Towards this end, we reformulate the optimization problem by introducing an auxiliary variable $\hat{\boldsymbol{\lambda}}$ and rewrite the problem equivalently as

$$\begin{aligned} \min_{\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}} \quad & f(\hat{\boldsymbol{\lambda}}) + r(\boldsymbol{\lambda}) \\ \text{subject to } & \hat{\boldsymbol{\lambda}}^T = \boldsymbol{\lambda}, \end{aligned}$$

where, $f(\hat{\boldsymbol{\lambda}}) = \sum_i \sum_{j>i} \sum_{\ell>j} \|\text{vec}(\underline{\boldsymbol{\Phi}}_{ij\ell}) - (\mathbf{A}_\ell \odot \mathbf{A}_j \odot \mathbf{A}_i)\hat{\boldsymbol{\lambda}}\|_F^2$ and $r(\boldsymbol{\lambda})$ is the indicator function for the probability simplex. $C = \{\boldsymbol{\lambda} | \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{1}^T \boldsymbol{\lambda} = 1\}$,

$$r(\boldsymbol{\lambda}) = \begin{cases} 0, & \boldsymbol{\lambda} \in C \\ \infty, & \boldsymbol{\lambda} \notin C. \end{cases}$$

At each iteration τ, we perform the following updates

$$\begin{aligned}
\hat{\boldsymbol{\lambda}}^{\tau+1} &\leftarrow (\mathbf{G} + \rho\mathbf{I})^{-1}(\mathbf{V} + \rho(\boldsymbol{\lambda}^{\tau} + \mathbf{u}^{\tau})) \\
\boldsymbol{\lambda}^{\tau+1} &\leftarrow \mathcal{P}_C(\boldsymbol{\lambda}^{\tau} - \hat{\boldsymbol{\lambda}}^{\tau+1} + \mathbf{u}^{\tau}) \\
\mathbf{u}^{\tau+1} &\leftarrow \mathbf{u}^{\tau} + \boldsymbol{\lambda}^{\tau+1} - \hat{\boldsymbol{\lambda}}^{\tau+1}, \\
&\text{where} \\
\mathbf{G} &= \sum_{i} \sum_{j>i} \sum_{\ell>j} \mathbf{Q}_{\ell ji}^{H} \mathbf{Q}_{\ell ji}, \\
\mathbf{V} &= \sum_{i} \sum_{j>i} \sum_{\ell>j} \mathbf{Q}_{\ell ji}^{H} \mathrm{vec}(\underline{\boldsymbol{\Phi}}), \\
\mathbf{Q}_{\ell ji} &= \mathbf{A}_{\ell} \odot \mathbf{A}_{j} \odot \mathbf{A}_{i}.
\end{aligned} \tag{4.25}$$

$\mathcal{P}_C(\mathbf{y})$ denotes the projection operator onto the convex set C – it computes the Euclidean projection of the real part of a point $\mathbf{y} = [y_1, \ldots, y_F]^T \in \mathbb{C}^F$ onto the probability simplex

$$\begin{aligned}
&\min_{\mathbf{x} \in \mathbb{R}^F} \frac{1}{2} \|\mathbf{x} - \Re(\boldsymbol{y})\|_F^2 \\
&\text{subject to} \quad \mathbf{x} \geq \mathbf{0},\ \mathbf{1}^T\mathbf{x} = 1,
\end{aligned}$$

using the method described in [86]. The overall procedure is described in Algorithm 4.2.

As the final step, the factors are assembled from the triples and the joint CF over all variables is synthesized as $\underline{\boldsymbol{\Phi}} = [\![\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N]\!]$. Given, the model of the joint CF, the corresponding joint PDF model can be recovered at any point as

$$f_{\boldsymbol{X}}(\mathbf{x}) = \sum_{h=1}^{F} \boldsymbol{\lambda}(h) \prod_{n=1}^{N} \sum_{k_n=-K}^{K} \mathbf{A}_n(k_n, h) e^{-j2\pi k_n x_n}. \tag{4.26}$$

4.5 Experiments

4.5.1 Low Dimensional Toy Data

We first show motivating results from modeling low dimensional datasets and showcase the expressivity of the proposed model as well as the significance of each parameter. Our model depends on the degree of smoothness and tensor rank (K and F). We pick the number of

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	$\mathbf{316.6 \pm 2.35}$

Table 4.1: Average test-set log-likelihood per datapoint for 5 different models on UCI datasets; **higher is better**.

Fourier series coefficients and the tensor rank through cross-validation for real data. Using these figures, one can see how the model changes when varying one of the two parameters separately. We begin by modeling $M = 2000$ samples from the Weight-Height dataset. In Figure 4.3, we present $M' = 1500$ synthetic samples obtained from the proposed model for different smoothing parameters $K \in [1, 2, 3, 4, 5]$ and ranks $F \in [2, 4, 6, 8, 10]$. By judiciously selecting the parameter search space, our approach yields an expressive generative model that can well-represent the data.

Following the same procedure, we now visualize $M = 2000$ samples from the 2-D Moons and Circles datasets. We fix the number of smoothing coefficients K, $K = 11$, and visualize synthetic $M' = 1500$ samples obtained from the proposed model for different approximation ranks $F \in [1, 2, 3, 4, 6]$ in Figure 4.4. The results show that our model is able to capture complex structures and properties of the data for an appropriate choice of rank F.

4.5.2 Real Data

We test the proposed approach on datasets (see a brief description of the datasets in Table 4.2) obtained from the UCI machine learning repository [54].

For each dataset we randomly hide 20% of data (testing set) and consider the remaining entries as observed information (training set). The parameters, which include the tensor rank F and the smoothing parameter K, are chosen using cross-validation. The smoothing parameter K is chosen from the set $\{5, 10, 20, 25, 30\}$ and the rank F from $\{5, 10, 20, 30, 50, 100\}$. We use 20% of the training data as validation data, where we seek to find the optimal parameter values maximizing the average log-likelihood of the validation samples. Once the hyperparameters are chosen, we train the model using all the training data (including the validation data) and

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

Table 4.2: Dataset information.

measure its performance on the testing set. We compare our approach against standard baselines described in section 5.2.1.

Evaluating the quality of density models is an open and difficult problem [87]. Following the approach in [68, 79], we calculate and report the average log-likelihood of unseen data samples (testing set), further averaged over 5 random data splits. The results are shown in Table 5.1. LRCF-DE has a higher average test sample log likelihood on almost all datasets. Overall, we observe that our method outperforms the baselines in 4 datasets and is comparable to the winning method in the remaining ones.

Following the derivation in Equation (4.20), we test the proposed model in several regression tasks. We evaluate and report the Mean Absolute Error (MAE) in estimating X_N for the unseen data samples in Table 4.3 and additional results for multi-output regression are presented in Table 4.4. Overall, we observe that LRCF-DE outperforms the baselines on almost all datasets, and performs comparable to the winning method in the remaining ones.

We have to stress again the fact that neural network based density estimation methods evaluate multivariate densities point-wise. These methods cannot impute more than a few missing elements in the input as grid search becomes combinatorial. Due to the interpretation of the approximation of the sought density as a finite mixture of separable densities and the coupled tensor factorization approach, our method allows us to easily work with missing data during both training and testing. Here, we showcase the results of LRCF-DE against MAF for simultaneously predicting the last two random variables of each dataset given the remaining ones.

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

Table 4.3: MAE for regression tasks.

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

Table 4.4: MAE for multi-output regression tasks.

The reduction in free parameters makes the proposed model particularly beneficial in the low-sample regime. We conducted an additional experiment on the Gas dataset to study how our model performs in terms of test-set log-likelihood when the number of samples is varied in comparison with the best performing neural network based PDF estimator from the baselines considered, namely, MAF. The results in Figure 4.5 verify that small to moderate training sample sizes result in much better LRCF-DE performance than MAF.

As our last experiment, we train LRCF-DE to learn the joint distribution of grayscale images from the USPS dataset [88], which contains 9298 images of handwritten digits of size $16 \times 16 \to N = 256$. The number of examples for each digit is shown in Table 4.5. The purpose of this experiment is to show that one can obtain reasonably accurate samples of digit images, by only modeling the distribution of triples of variables, something which has never been done before on images. We sample from the resulting 256-dimensional model, and provide visualization of the generated data. We fix the tensor rank to $F = 8$ and the smoothing parameter

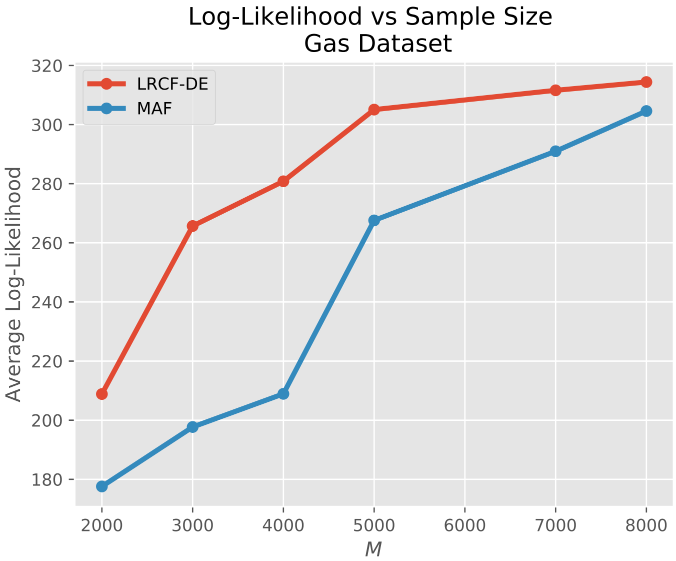


Figure 4.5: LRCF-DE is particularly beneficial in the small to moderate training sample regime. Using a limited number of training points from the Gas dataset, highlights the superior performance of LRCF-DE against the state of the art deep learning based PDF estimator (MAF).

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

Table 4.5: Images of handwritten digits - USPS dataset information.

to $K = 15$, and draw 8 random samples of each digit (class). The resulting samples are shown in Figure 4.6, and they are very pleasing – in light of the fact that our model is “agnostic”: designed for general-purpose density estimation, not specifically for realistic-looking image synthesis. It is possible to incorporate image modeling domain knowledge in the design of LRCF-DE (such as correlation between adjacent pixel values), but this is beyond the scope of this paper. This manuscript is the first part containing the foundations of a two-part paper. The second part [22] builds on this foundation to develop a joint compression (nonlinear dimensionality reduction) and compressed density estimation framework that offers additional flexibility and scalability and is used to demonstrate improved image sampling performance against well known deep learning models, including autoregressive methods and VAEs.

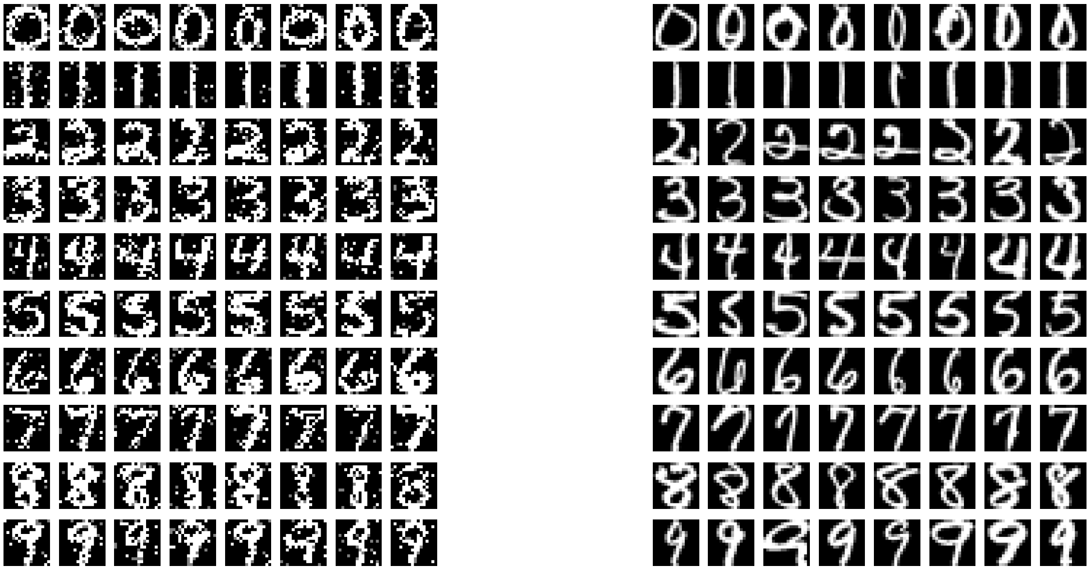


Figure 4.6: The first eight columns correspond to class-conditional synthetic samples (generated by LRCF-DE) and the rest correspond to real samples from the USPS dataset.

4.6 Chapter Summary

In this work, we have revisited the classic problem of non-parametric density estimation from a fresh perspective – through the lens of complex Fourier series approximation and tensor modeling, leading to a low-rank characteristic function approach. We showed that any compactly supported density can be well-approximated by a finite *characteristic tensor* of leading complex Fourier coefficients as long as the coefficients decay sufficiently fast. We posed density estimation as a constrained (coupled) tensor factorization problem and proposed a Block Coordinate Descent algorithm, which under certain conditions enables learning the true data-generating distribution. Results on real data have demonstrated the utility and promise of this novel approach compared to both standard and recent density estimation techniques.

4.7 Appendix

Due to the subtleties of tensor rank, the possible non-existence of best low-rank tensor approximation, and other technical issues, no perturbation theory currently exists for tensor decomposition to estimate how close low-rank approximation of a perturbed low-rank tensor is to the unperturbed low-rank tensor. In what follows, we summarize what is known for our method without imposing low-rank structure, and further explain why partially imposing low-rank structure is beneficial, using matrix results.

4.7.1 Bias, variance, consistency of the empirical characteristic function

In this appendix we summarize important properties of empirical characteristic functions as statistical estimators of the corresponding characteristic functions. We refer the reader to [89] for proofs and additional results.

By linearity of expectation, it is easy to see that the empirical characteristic function is an unbiased estimator of the corresponding characteristic function, i.e.,

$$E\left[\widehat{\Phi}_{\boldsymbol{X}}(\boldsymbol{\nu})\right] = \Phi_{\boldsymbol{X}}(\boldsymbol{\nu}),$$

for all $\boldsymbol{\nu}$ and $M \geq 1$. For the remainder of this section, we assume that $\{\mathbf{x}_m\}_{m=1}^{M}$ is i.i.d. in m. The variance of the empirical characteristic function estimate can be shown [89] to be

$$\begin{aligned}
\mathrm{Var}[\widehat{\Phi}_{\boldsymbol{X}}(\boldsymbol{\nu})] &= E\left[\left|\widehat{\Phi}_{\boldsymbol{X}}(\boldsymbol{\nu}) - E\left[\widehat{\Phi}_{\boldsymbol{X}}(\boldsymbol{\nu})\right]\right|^2\right] \\
&= E\left[\left|\widehat{\Phi}_{\boldsymbol{X}}(\boldsymbol{\nu}) - \Phi_{\boldsymbol{X}}(\boldsymbol{\nu})\right|^2\right] \\
&= \frac{1}{M}\left(1 - |\Phi_{\boldsymbol{X}}(\boldsymbol{\nu})|^2\right).
\end{aligned}$$

Note that $0 \leq \left|\Phi_{\boldsymbol{X}}(\boldsymbol{\nu})\right| \leq 1$, and therefore $\mathrm{Var}[\widehat{\Phi}_{\boldsymbol{X}}(\boldsymbol{\nu})] \leq \frac{1}{M}$. It follows that

$$\lim_{M\to\infty} E\left[\left|\widehat{\Phi}_{\boldsymbol{X}}(\boldsymbol{\nu}) - \Phi_{\boldsymbol{X}}(\boldsymbol{\nu})\right|^2\right] = 0,$$

i.e., for any fixed $\boldsymbol{\nu}$, $\widehat{\Phi}_{\boldsymbol{X}}(\boldsymbol{\nu})$ converges to $\Phi_{\boldsymbol{X}}(\boldsymbol{\nu})$ in the mean-squared sense. By the strong law of large numbers, it also converges almost surely for any fixed $\boldsymbol{\nu}$. Furthermore, for any fixed positive $T < \infty$

$$\lim_{M\to\infty} \text{sup}_{\|\boldsymbol{\nu}\|_2 \leq T}\left|\widehat{\Phi}_{\boldsymbol{X}}(\boldsymbol{\nu}) - \Phi_{\boldsymbol{X}}(\boldsymbol{\nu})\right| = 0,$$

almost surely. It can also be shown [89] that for any increasing sequence T_M such that

$\lim_{M\to\infty} \frac{\log(T_M)}{M} = 0$, it holds

$$\lim_{M\to\infty} \sup_{\|\boldsymbol{\nu}\|_2 \le T_M} \left| \widehat{\Phi}_{\boldsymbol{X}}(\boldsymbol{\nu}) - \Phi_{\boldsymbol{X}}(\boldsymbol{\nu}) \right| = 0,$$

almost surely. In our context, we only use a sampled and truncated version of the characteristic function (corresponding to a truncated multivariate Fourier series), hence T is always finite – we do not need the latter result.

It is also worth noting that the covariance of different samples of the empirical characteristic function (corresponding to different values of $\boldsymbol{\nu}$) goes to zero $\sim \frac{1}{M}$, and so does the covariance of its real and imaginary parts. As a result, for large M, the errors in the different elements of the characteristic tensor are approximately uncorrelated, with uncorrelated real and imaginary parts. This suggests that when we fit a model to the empirical characteristic function, it makes sense to use a least squares approach. Another motivation for this comes from Parseval's theorem: minimizing integrated squared error in the Fourier domain corresponds to minimizing integrated squared error between the corresponding multivariate distributions. This is true in particular when we limit the support of the distribution to a hypercube and use the samples of the characteristic function that correspond to the multivariate Fourier series, thereby replacing the multivariate integral in the Fourier domain by a multivariate sum.

4.7.2 Low-rank denoising: reduction of the mean squared error

Our use of a low-rank model in the characteristic tensor domain is primarily motivated by the need to avoid the "curse of dimensionality": using a rank-F model with $2K+1$ Fourier coefficients per mode parametrizes the whole N-dimensional multivariate density using just $FN(2K+1)$ coefficients, and avoids instantiating and storing a tensor of size $(2K+1)^N$, which is close to impossible even for moderate N. However, there is also a variance benefit that comes from this low-rank parametrization. We know from [90] that for a square $L \times L$ matrix of rank F observed in zero-mean white noise of variance σ^2, low-rank denoising attains mean squared error $cLF\sigma^2$ asymptotically in L, where c is a small constant. In practice the asymptotics kick in even for relatively small L [90]. Contrast this to the raw $L^2\sigma^2$ if one does not use the low-rank property.

For an N-way tensor of rank F, assume N is even, $K_n = K$, $\forall N$ (for simplicity of exposition), and "unfold" the characteristic tensor into a $K^{N/2} \times K^{N/2}$ matrix. In practice we use F *far* less than $K^{N/2}$, and thus the resulting matrix will be *very* low rank. Invoking [90], low-rank tensor modeling will yield a reduction in mean squared error by a factor of at least $\frac{F}{K^{N/2}}$. We say at least, because this low-rank matrix structure is implied but does not imply low-rank tensor structure, which is much stronger. Note also that mean squared error in the characteristic tensor domain translates to mean squared error between the corresponding distributions, by virtue of Parseval's theorem.

4.7.3 Derivation of (4.20)

In this appendix we present the derivation of (4.20), which is used to solve regression tasks.

$$
\begin{aligned}
&E\left[X_N|X_1,\ldots,X_{N-1}\right] \\
&= \int_0^1 x_N f_{X_N|X_1,\ldots,X_{N-1}}(x_N|x_1,\ldots,x_{N-1})dx_N \\
&= \int_0^1 x_N \frac{f_{X_1,\ldots,X_N}(x_1,\ldots,x_N)}{f_{X_1,\ldots,X_{N-1}}(x_1,\ldots,x_{N-1})}dx_N \\
&= \frac{1}{c_1}\int_0^1 x_N f_{X_1,\ldots,X_N}(x_1,\ldots,x_N)dx_N \\
&= \frac{1}{c_1}\int_0^1 x_N \sum_{h=1}^{F}\boldsymbol{\lambda}(h)\prod_{n=1}^{N}\sum_{k_n=-K}^{K}\mathbf{A}_n(k_n,h)e^{-j2\pi k_n x_n}dx_N \\
&= \frac{1}{c_1}\sum_{h=1}^{F}\boldsymbol{\lambda}(h)\prod_{n=1}^{N-1}\sum_{k_n=-K}^{K}\mathbf{A}_n(k_n,h)e^{-j2\pi k_n x_n} \\
&\qquad\qquad \sum_{k_N=-K}^{K}\mathbf{A}_N(k_N,h)\int_0^1 x_N e^{-j2\pi k_N x_N}dx_N \\
&= \frac{1}{c_1}\sum_{h=1}^{F}\boldsymbol{\lambda}(h)\prod_{n=1}^{N-1}\sum_{k_n=-K}^{K}\mathbf{A}_n(k_n,h)e^{-j2\pi k_n x_n} \\
&\qquad\qquad\qquad \sum_{k_N=-K}^{K}c_{2,k_N}\mathbf{A}_N(k_N,h),
\end{aligned}
$$

where

$$\begin{aligned} c_1 &= f_{X_1,\ldots,X_{N-1}}(x_1,\ldots,x_{N-1}) \\ &= \sum_{h=1}^{F} \boldsymbol{\lambda}(h) \prod_{n=1}^{N-1} \sum_{k_n=-K}^{K} \mathbf{A}_n(k_n,h) e^{-j2\pi k_n x_n}, \end{aligned}$$

and

$$c_{2,k_N} = \frac{e^{-j2\pi k_N}}{-j2\pi k_N} + \frac{1-e^{-j2\pi k_N}}{[-j2\pi k_N]^2}.$$

Chapter 5

Low-rank Characteristic Tensor Density Estimation Part II: Compression and Latent Density Estimation

5.1 Introduction

Accurate modeling of the multivariate structure of data based on observed data samples is one of the most fundamental topics in machine learning. A model of the joint probability density function (PDF) of a data vector encodes the complete statistical properties of the data generative process and allows one to reason about data probabilistically, uncover the (possibly low-dimensional) manifold the data live on, and ultimately generate new data. PDF estimation serves as a building block in a wide variety of applications, such as image processing [91], speech modeling [5], natural language processing [92], and anomaly detection [93]. Conventional density estimation methods, such as kernel density estimation (KDE) [94] and Gaussian mixture models (GMMs) [95] are usually designed to fit target distributions directly in the data space $\mathbb{R}^N$ and fall short in high dimensions from both computational and statistical points of view due to the *Curse of Dimensionality* – convergence slows down as the number of dimensions increases as a result of data sparsity in high-dimensional spaces. Real-world data often resides in a high-dimensional and complex feature space with only a limited amount of observed data being directly available.

Recently, the use of deep neural networks has led to substantial advances in this area. For example, generative adversarial networks (GANs) [17] can be trained to sample from very high-dimensional densities, but they do not support statistical inference or explicit density evaluation. On the other hand, variational auto-encoders (VAEs) [69] provide functionality for both (approximate) inference and sampling. VAEs assume a prior as a manually specified distribution (e.g., a simple isotropic Gaussian or mixture of Gaussians) and are trained by minimizing a reconstruction error and a divergence to force the variational posterior to fit the prior of the latent variables. However, the forced global structure in the latent space through the use of a manually specified prior may differ from the complex latent nature of the true data manifold. Thus, such simplistic assumptions may potentially harm the generalization of high dimensional data from low dimensional latent spaces. For example, it is observed that VAEs tend to generate blurry images, an effect that is usually attributed to the latent density mismatch problem [96, 97]. Finally, explicit neural models such as auto-regressive models [98] and flow-based models [16, 99] are designed to perform sampling and point-wise density evaluation. Despite their success, auto-regressive models generally suffer from slow sampling time [100] and inferior quality of samples compared to VAEs; but they are particularly useful for point-wise density evaluation. On the other hand, flow-based models, such as Real-NVP [16] and Glow [91], are efficient for sampling, but have inferior performance in evaluating the log-likelihood of the input compared to the auto-regressive ones.

The goal of this paper is to introduce a class of probabilistic latent variable models for unsupervised learning which is tailored for high dimensional datasets. The proposed class of models is non-parametric, and it learns the underlying distribution of latent representations of the input data in the Fourier domain. The proposed framework consists of two main components: an auto-encoder network, through which a lower-dimensional latent representation of the input is sought; and a nonparametric density estimation module in the latent domain. The auto-encoder compresses redundancies in the data domain while preserving the essential information, and is used as a new feature representation space where we learn the data distribution. The auto-encoder and the latent density are learned jointly via an optimization criterion that combines a data reconstruction loss and a negative log-likelihood regularization term over the latent representations of the training data.

This is the second part of a two-part paper. The first part [21] dealt with the density estimation problem in the native ("raw") input data domain, showing that any joint density that is compactly supported and continuously differentiable can be well-approximated using a low-rank tensor model in the Fourier domain. A corollary of [21] is that a finite separable mixture model (approximately) follows from compactness of support and continuous differentiability. This interpretation enables an efficient and disciplined sampling process. By introducing a low-rank tensor model in the Fourier domain via the Canonical Polyadic Decomposition (CPD) [26], a controllable approximation of the multivariate density is identifiable. The choice of tensor rank, number of Fourier coefficients, and the dimensionality of the latent space let us control the expressivity of the learned distribution. With respect to Part I [21], the key differences in this second part are the following:

- Unlike Part I, which aimed to tackle the problem directly in the original N-dimensional space, probabilistic modeling in Part II is realized in a reduced-dimension latent space and the effects are translated back into the input space through the decoder mapping. Towards this end, a *joint* nonlinear dimensionality reduction and compressed density estimation framework is proposed in Part II. The joint approach boosts the flexibility, scalability, and statistical performance in terms of prediction (regression/detection) accuracy and sampling fidelity.

- Instead of the coupled tensor factorization approach adopted in Part I, Part II tackles joint density estimation as a *hidden tensor factorization* problem using maximum likelihood learning of the latent distribution's parameters.

A high-level overview of the proposed framework is shown in Figure 5.1. A sneak peak of the expected performance of the proposed method is shown in Figure 5.2 where we use our model to learn the joint distribution of MNIST [101] images of 0s and 8s. With a suitable combination of hyper-parameters, the proposed density estimator offers considerable flexibility without sacrificing parsimony of representation. We showcase the promising results of the proposed model on benchmark image (MNIST, FMNIST [102]) and several tabular datasets on sampling, regression, and anomaly detection tasks, and on some toy but didactic examples for illustration.

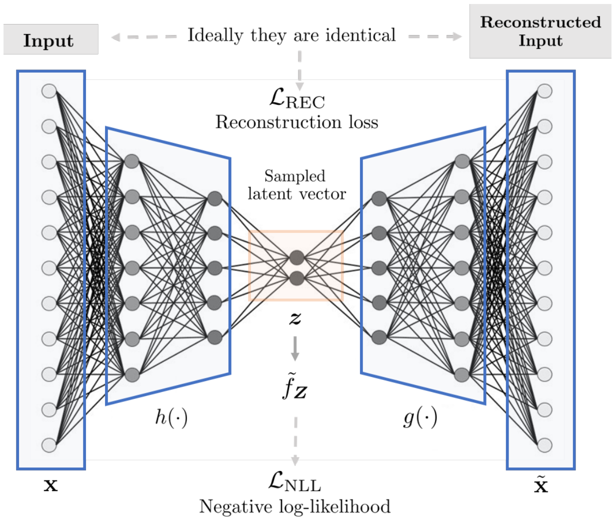


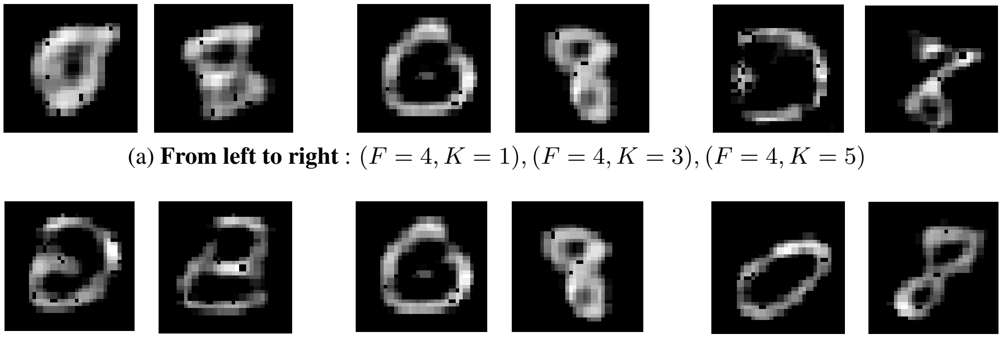
Figure 5.1: Compressed Density Estimation: An auto-encoder attempts to reproduce the input in the output layer by compressing it to fewer dimensions while retaining non-redundant information. The hidden layer becomes a bottleneck, forming a lower-dimensional representation of the data, which is used to build a non-parametric density model.

5.2 Background

5.2.1 Related work

Classic work on density estimation includes Gaussian Mixture Models, which are fragile to model mismatch due to their parametric nature, and introduce computational and estimation challenges in the high dimensional case. Conventional non-parametric models such as the Kernel Density Estimators become computationally intractable in high dimensions, since the number of parameters grows exponentially with the number of dimensions [72].

Recently, the use of deep neural networks has led to significant advances in modeling modern complex and high-dimensional data. Auto-encoders enjoy a remarkable ability to learn data representations. Auto-encoder networks such as VAEs [69] and GANs [17] learn latent representations of very high-dimensional data such as images or videos. However, GANs only support sampling, but not inference or density estimation. VAEs assume that high-dimensional data can be modeled as lying on or near a low-dimensional, nonlinear manifold which they approximate by learning nonlinear mappings while encouraging a global structure in the latent



(a) **From left to right** : $(F = 4, K = 1), (F = 4, K = 3), (F = 4, K = 5)$

(b) **From left to right** : $(F = 2, K = 3), (F = 4, K = 3), (F = 8, K = 3)$

Figure 5.2: Sneak peek: Demonstration of generated MNIST samples trained on images of 0s and 8s using the proposed CDE model. We train the model on different values of F and K to show that only a few parameters are needed to come close to the ground-truth. Increasing K generates sharper digits, while increasing F better differentiates the samples of the digits.

space through the use of a specified prior distribution. However, specifying the prior distribution may prevent them from faithfully representing the true data manifold. It was shown in [103] that choosing a simplistic prior could lead to over-regularization and, as a consequence, very poor latent representation.

In this work, we avoid the variational training by jointly training a deterministic encoder–decoder pair and an expressive density estimator in the latent space, which admits simpler optimization, and, most importantly, generates better samples than VAEs. A key advantage of our approach is that we introduce a non-parametric density model into the latent space, which by virtue of uniqueness of low rank tensor decomposition comes with identification guarantees. This approach can yield a more accurate model of the data manifold, as we will see. A conceptually similar approach was proposed in [93] and applied for unsupervised anomaly detection, the key difference being that the density of low-dimensional representations was modelled using a GMM, which is far more restrictive and does not come with identification guarantees.

Other classes of generative models include the Real-valued Neural Autoregressive Distribution Estimator (RNADE) [79] and its discrete version MADE [66], which is among the best performing neural density evaluation methods and has shown great potential in scaling to high-dimensional distribution evaluation problems. These so-called autoregressive models decompose the joint density as a product of one-dimensional conditionals of increasing conditioning order, and model each conditional density with a parametric model. Normalizing Flows (NF) [16] models, on the other hand, start with a base density e.g., standard Gaussian, and stack a series of invertible transformations with tractable Jacobian to approximate the target density. Masked Autoregressive Flow (MAF) [68] is a type of NF model where the transformation layer is built as an autoregressive neural network. Finally, Gaussianization flows (GF) [104] build upon rotation-based iterative Gaussianization. These methods do not construct an explicit joint PDF model, but rather serve for point-wise density evaluation. That is, for any given input vector (realization), they output an estimate of the density evaluated at that particular input vector.

5.2.2 Notation

In this paper, we use $\mathbf{x}$, $\mathbf{X}$, $\underline{\mathbf{X}}$ for vectors, matrices and tensors respectively. We use the notation $\mathbf{x}(k)$, $\mathbf{X}(:,k)$, $\underline{\mathbf{X}}(:,:,k)$ to refer to a particular element of a vector, a column of a matrix and a slab of a tensor. We use the notation FC (a, b, c) to describe a fully-connected layer with a input neurons and b output neurons activated by function c.

5.2.3 Canonical Polyadic Decomposition

In this section, we briefly introduce basic concepts related to tensor decomposition. A D-way tensor $\underline{\mathbf{\Phi}} \in \mathbb{C}^{K_1 \times K_2 \times \cdots \times K_D}$ is a multidimensional array whose elements are indexed by D indices. Any tensor can be decomposed as a sum of F rank-1 tensors, i.e.,

$$\underline{\mathbf{\Phi}} = [\![\boldsymbol{\lambda}, \mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_D]\!] = \sum_{f=1}^{F} \boldsymbol{\lambda}(f) \mathbf{a}_f^1 \circ \mathbf{a}_f^2 \circ \cdots \circ \mathbf{a}_f^D, \tag{5.1}$$

where $\mathbf{A}_d = [\mathbf{a}_1^d, \ldots, \mathbf{a}_F^d] \in \mathbb{C}^{K_D \times F}$ and constraining the columns $\mathbf{A}_n(:,f)$ to have unit norm, the real scalar $\boldsymbol{\lambda}(f)$ absorbs the f-th rank-one tensor's scaling. A particular element of the tensor is given by

$$\underline{\mathbf{\Phi}}(k_1, k_2, \ldots, k_D) = \sum_{f=1}^{F} \boldsymbol{\lambda}(f) \prod_{d=1}^{D} \mathbf{A}_d(k_d, f). \tag{5.2}$$

When F is minimal, it is called the rank of $\underline{\mathbf{\Phi}}$, and the decomposition is called Canonical Polyadic Decomposition (CPD) [25, 26]. CPD is a powerful model that can parsimoniously represent the high-order interactions among multi-way data exactly or approximately leading to significant reduction in the number of parameters. A key property of the CPD is that the rank-1 components are unique under mild conditions, see [23] for a tutorial overview and related identifiability results. In our present context, uniqueness of the CPD implies uniqueness of the multivariate density model that we will build using CPD in the Fourier domain, as we will see.

5.2.4 Choice of Tensor Decomposition Model

We will use CPD tensor modeling, but there are also other tensor decompositions that one might consider using, such as Tucker decompositions, particularly HOSVD/MLSVD, and tensor-train models, see [23] and references therein. Different decompositions serve different purposes and our decision to choose CPD is based on its uniqueness properties, compactness of model parametrization (scalability to higher dimensions), and ease of marginalization and sampling. Marginalization is important when we wish to predict a subset of the random variables from another subset of the random variables – one of the most important motivations for developing multivariate density models. Ease of sampling is key when we wish to efficiently draw samples from the fitted distribution, which is another common use of multivariate density models.

We briefly compare the three most commonly used tensor decompositions: CPD, Tucker (including HOSVD/MLSVD), and tensor-trains. Tucker represents data as a core tensor and D factor matrices using O($F^D + DKF$) parameters, which scales exponentially in the number of input random variables (D). CPD and tensor-train decompositions are preferred for high-order tensors since their complexity scales linearly with D. CPD represents a tensor as a sum of rank-one tensors using O(DKF) parameters and the number of free parameters in tensor-train models is O(KDF^2). We chose CPD over tensor-trains due to

1. its uniqueness properties and
2. ease of marginalization and sampling from the fitted model (these are considerably more complicated for tensor-trains).

That does not mean that tensor-trains cannot be useful in our present context; but we defer this to future work.

One of the key parameters in the CPD is the choice of rank F. Determining tensor rank is an NP-hard problem. In practice we choose rank using a validation set. Aiming to bypass the problem of rank determination, Bayesian probabilistic tensor factorization approaches such as [105, 106] have been proposed. The objective of these approaches is not multivariate PDF or PMF modeling but general tensor modeling – i.e., instead of a collection of samples of a random vector, their input is elements of a data tensor. Rank determination (subject to a user-specified upper bound on rank) is automatic in these methods, but it falls off statistical modeling assumptions that may or may not be appropriate in our context. Note that we model the complex tensor of multivariate Fourier series coefficients of the joint PDF in some latent space. We know that for differentiable PDFs the high-frequency coefficients will typically be much smaller than low-frequency ones; but the priors used on the factors in Bayesian tensor models are i.i.d. across modes.

Interestingly, reference [105] is about fitting non-negative tensors using non-negative factors. While is not useful for PDF modeling in the Fourier domain, it can potentially be used in the multivariate PMF case considered earlier in our work [18], if one can incorporate certain sum to one constraints in the framework of [105].

5.3 Compressed-domain Density Estimation

We consider the problem of general-purpose modeling of a high-dimensional continuous joint distribution $f_{\boldsymbol{X}}$ of an N-dimensional random vector $\boldsymbol{X}$ when N is large. Given a dataset $\mathcal{D}$ of M i.i.d. realizations in the N-dimensional observable space $\mathcal{D} = \{\mathbf{x}_m\}_{m=1}^{M}$, we typically wish to perform maximum likelihood learning of its parameters, i.e., to minimize the Negative Log-Likelihood (NLL)

$$\mathcal{L}_{\text{NLL}} = -\frac{1}{M}\sum_{m=1}^{M} \log\Big(f_{\boldsymbol{X}}(\mathbf{x}_m)\Big). \tag{5.3}$$

In general, $\mathcal{L}_{\text{NLL}}$ is difficult to compute or differentiate directly, since the density $f_{\boldsymbol{X}}$ can be analytically and computationally intractable. One can address this issue and evade the curse of dimensionality by using a mapping h to encode the input data samples $\mathbf{x}_m \in \mathbb{R}^N$ into much lower dimensional representations $\mathbf{z}_m \in \mathbb{R}^D$, with $D \ll N$, in the latent space.

In this work, we propose a joint dimensionality reduction (DR) and density estimation framework where the DR part is carried out through learning an auto-encoder:

$$\text{Auto-encoder: } \mathbf{x} \overset{h}{\mapsto} \mathbf{z} \overset{g}{\mapsto} \tilde{\mathbf{x}}.$$

Here, h and g denote the encoder and the decoder, respectively, and $\tilde{\mathbf{x}}$ is the reconstruction of $\mathbf{x}$.

The mapping $h : \mathbb{R}^N \to \mathbb{R}^D$ can be viewed as nonlinear dimensionality reduction, and the low-dimensional $\mathbf{z} = h(\mathbf{x})$ as the bottleneck representation of the observed vector $\mathbf{x}$. We approximate the latent domain distribution $f_{\boldsymbol{Z}}$ using the non-parametric density estimation framework in Part I of this work [21]. The density estimation framework relies on the decomposition of a D-way tensor of leading Fourier series coefficients through CPD. Part I has shown that this model is quite general, in that it can approximate any multivariate compactly supported density as long as its Fourier coefficients decay sufficiently fast, and under certain conditions it can identify the true latent model.

The choice of tensor rank, number of Fourier coefficients, and the dimensionality of the bottleneck representation are used to control the expressivity of the model. Here we propose to *jointly learn* the auto-encoder and the parameters of the density model. The combination of an auto-encoder and density estimation takes advantage of their synergistic strengths. Auto-encoders can compress input data to fewer dimensions while retaining non-redundant information, while density estimation works best in lower-dimensional spaces. The resulting training objective provides a strong underlying signal to efficiently capture the latent space distribution of the input data during training. The proposed framework can be used for missing data imputation and as a generative model.

Missing data imputation: Assume that for a given data sample $\mathbf{x}$, we observe a subset of its values denoted as $\mathbf{x}_O$ and $\mathbf{x}_M$ is the part that we do not observe. Data imputation can be performed by clamping the observed dimensions $\mathbf{x}_O$ to their values and maximizing log-likelihood with respect to the missing dimensions $\mathbf{x}_M$

$$\max_{\mathbf{x}_M} \log\left(f_{\boldsymbol{Z}}\big(h(\mathbf{x}_O, \mathbf{x}_M)\big)\right). \tag{5.4}$$

Data sampling: With g given, we can draw a realization of the random vector $\boldsymbol{Z}$ in the D-dimensional latent space from $f_{\boldsymbol{Z}}$, and back transform to a sample in the original N-dimensional space by its inverse image as

$$\mathbf{z} \sim f_{\boldsymbol{Z}},\ \tilde{\mathbf{x}} = g(\mathbf{z}). \tag{5.5}$$

Similar approaches such as VAEs pose a stochastic condition on the latent variables to comply with a fixed prior distribution $f_{\boldsymbol{Z}}$ over a low-dimensional latent space:

$$\text{VAE: } \mathbf{x} \overset{h}{\mapsto} \mathbf{z} \overset{g}{\mapsto} \tilde{\mathbf{x}},\ \mathbf{z} \sim f_{\boldsymbol{Z}}(\mathbf{z}).$$

The generative process of the VAE is carried out as

$$\mathbf{z} \sim f_{\boldsymbol{Z}}, \mathbf{x} \sim p_{\theta}(\mathbf{X}|\mathbf{Z} = \mathbf{z})$$

where a stochastic decoder

$$D_{\theta}(\mathbf{z}) = \mathbf{x} \sim p_{\theta}(\mathbf{x}|\mathbf{z}) = p(\mathbf{X}|g(\mathbf{z}))$$

links the latent space to the input space through the likelihood distribution p_{θ}. This may be limiting in case this predefined prior does not match the structure of the true data manifold, leading to a less accurate model. Our approach is fundamentally different as we avoid prior distribution matching between the variational posterior and the prior, but instead propose jointly learning a non-parametric density estimator in the latent space. Most importantly, our approach produces better samples than VAEs, as we will see. In the following sections, we give a detailed description of the two main components of our framework and the optimization procedure.

5.3.1 Compression Network

The first component of our framework seeks a non-linear mapping h to project high-dimensional input samples into a low-dimensional space. In the dimensionality reduction process, discarding some dimensions inevitably leads to information loss. We wish to preserve the available information as much as possible, and to this end we minimize the empirical approximation of the mean squared error

$$\text{MSE} := \int_{\mathbb{R}^N} \|\boldsymbol{x} - g(h(\boldsymbol{x}))\|_2^2 f_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x}. \tag{5.6}$$

Auto-encoders learn a function by fine-tuning the parameters of a feed-forward Deep Neural Network (DNN) in such a way that the reconstruction error is minimized when back projected with another feed-forward DNN. These networks need to be specified a-priori, in terms of the number of layers and neurons. In this work, we use the rectified linear unit (ReLU) activation function [107] while the rest of the parameters such as the width of each layer and the depth of the network are adjusted according to M, N.

Although we proceed with simpler networks, other types of networks (e.g., convolutional neural networks [108, 109]) can also be used. Let $h(\cdot; \boldsymbol{\theta}_h)$ and $g(\cdot; \boldsymbol{\theta}_g)$ be DNNs and $\boldsymbol{\theta}_h$, $\boldsymbol{\theta}_g$ collect the encoder and decoder network parameters, i.e., the weights and bias terms at each hidden layer. Given a finite set of samples M, the empirical reconstruction loss can be computed as

$$\mathcal{L}_{\text{REC}} := \frac{1}{M}\sum_{m=1}^{M} ||\mathbf{x}_m - g(h(\mathbf{x}_m;\boldsymbol{\theta}_h);\boldsymbol{\theta}_g)||_2^2. \tag{5.7}$$

The reconstruction loss is typically minimized using Stochastic Gradient Descent (SGD).

5.3.2 Latent Density Estimation Network

The second component of our framework is a non-parametric density estimation model [21]. The key difference is that we propose joint dimensionality reduction and density modeling in the reduced-dimension latent space so that we capture the bottleneck layer distribution, whereas [21] aimed to tackle the problem directly in the original N-dimensional space. This combination is crucial for enhanced performance and scalability. Additionally, instead of coupled tensor factorization, we consider an alternative algorithmic approach by formulating density estimation as a hidden tensor factorization problem.

Let us consider the multivariate joint PDF $f_{\boldsymbol{Z}}$ of a D-dimensional random vector $\boldsymbol{Z}$ with its support contained within the hypercube $S = [0, 1]^D$. Then, the joint PDF can be represented by a multivariate Fourier series

$$f_{\boldsymbol{Z}}(\mathbf{z}) = \sum_{k_1=-\infty}^{\infty} \cdots \sum_{k_D=-\infty}^{\infty} \Phi_{\boldsymbol{Z}}[\boldsymbol{k}] e^{-j2\pi\mathbf{k}^T\mathbf{z}}, \tag{5.8}$$

where

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

and Φ_Z is the Characteristic Function (CF). The multivariate characteristic function $\Phi_Z : \mathbb{R}^D \to \mathbb{C}$ is defined as

$$\Phi_{\boldsymbol{Z}}(\boldsymbol{\nu}) = E\left[e^{j\boldsymbol{\nu}^T\boldsymbol{Z}}\right].$$

Similar to the PDF $f_{\boldsymbol{Z}}$, its corresponding CF $\Phi_{\boldsymbol{Z}}$ contains complete information about the distribution of $\boldsymbol{Z}$, i.e., the PDF and the CF have a bijective relationship – one being the Fourier transform of the other. When the underlying PDF is sufficiently differentiable in all variables,

$f_{\boldsymbol{Z}}$ can be approximated by a truncated multivariate Fourier series with cutoffs $K_1, \ldots, K_D$ i.e.,

$$\tilde{f}_{\boldsymbol{Z}}(\boldsymbol{z}) = \sum_{k_1=-K_1}^{K_1} \ldots \sum_{k_D=-K_D}^{K_D} \Phi_{\boldsymbol{Z}}[\boldsymbol{k}] e^{-j2\pi \boldsymbol{k}^T \boldsymbol{z}}. \tag{5.9}$$

The smoother the underlying PDF the faster the convergence rate and the smaller the approximation error.

For any $p \in \mathbb{N}$, If the partial derivatives $\frac{\partial^{\theta_1}}{\partial z_1^{\theta_1}} \cdots \frac{\partial^{\theta_D}}{\partial z_D^{\theta_D}} f_{\boldsymbol{Z}}(\mathbf{z})$ of $f(\cdot)$ exist and are absolutely integrable for all $\theta_1, \ldots, \theta_D$ with $\sum_{n=1}^{D} \theta_n \leq p$ then the rate of decay of the magnitude of the $\mathbf{k}$-th Fourier coefficient $|\Phi_{\boldsymbol{Z}}[\mathbf{k}]|$ obeys [83]

$$|\Phi_{\boldsymbol{Z}}[\mathbf{k}]| = \mathcal{O}\left(\frac{1}{1 + \|\mathbf{k}\|_2^p}\right).$$

The worst-case approximation error is bounded by

$$\|f_{\boldsymbol{Z}} - \tilde{f}_{\boldsymbol{Z}}\|_\infty \leq C \sum_{d=1}^{D} \frac{\omega_d \left(\frac{\partial^{\theta_d}}{\partial z_d^{\theta_d}} f_{\boldsymbol{Z}}, \frac{1}{1+K_d} \right)}{(1 + K_d)^{\theta_d}},$$

where $C = C_2 \left(1 + C_1 \prod_{d=1}^{D} \log K_d\right)$, C_1, C_2 are constants independent of $f_{\boldsymbol{Z}}$ and the K_d's and

$$\omega_j(f_{\boldsymbol{Z}}, \delta) := \sup_{|z_j - z_j'| \leq \delta} |f_{\boldsymbol{Z}}(z_1, ..., z_j, ..., z_D) - f_{\boldsymbol{Z}}(z_1, ..., z_j', ..., z_D)| \tag{5.10}$$

measures the smoothness of $f_{\boldsymbol{Z}}$ for each component $j \in [D]$ [84], [85, Chapter 23]. Note that we can represent the truncated Fourier coefficients using a D-way tensor $\underline{\boldsymbol{\Phi}}$ where

$$\underline{\boldsymbol{\Phi}}(k_1, \ldots, k_D) = \Phi_{\boldsymbol{Z}}[\mathbf{k}]. \tag{5.11}$$

For simplicity we will assume that $K_1 = \cdots = K_D = K$. Orthogonal series PDF approximation using a truncated sum of basis functions (e.g., trigonometric, polynomial, wavelet) becomes computationally intractable in high dimensions, since the number of parameters (tensor elements) grows exponentially with the number of dimensions. To reduce the number of parameters, we introduce a low-rank parameterization of the coefficient tensor obtained by truncating the multidimensional Fourier series [21] which reduces the number of parameters from $O(K^D)$ to $O(DKF)$. Introducing the rank-F CPD we have

$$\underline{\boldsymbol{\Phi}}(k_1,\ldots,k_D) = \sum_{f=1}^{F} p_H(f) \prod_{d=1}^{D} \Phi_{Z_d|H=f}(k_d|f). \tag{5.12}$$

By linearity and separability of the multidimensional Fourier transformation, applied to rank-one components, $\tilde{f}_{\boldsymbol{Z}}(\mathbf{z})$ can be written in the following form

$$\begin{aligned}
\tilde{f}_{\boldsymbol{Z}}(\mathbf{z}) &= \sum_{k_1=-K_1}^{K_1} \cdots \sum_{k_D=-K_D}^{K_D} \Phi_{\boldsymbol{Z}}[\mathbf{k}] e^{-j2\pi \mathbf{k}^T \mathbf{z}} \\
&= \sum_{f=1}^{F} \underbrace{p_H(f)}_{\boldsymbol{\lambda}(f)} \prod_{d=1}^{D} \sum_{k_d=-K}^{K} \underbrace{\Phi_{Z_d|H=f}(k_d|f)}_{\mathbf{a}_d^f(K+1+k_d)} \underbrace{e^{-j2\pi k_d z_d}}_{\mathbf{b}_d(K+1+k_d)} \\
&= \sum_{f=1}^{F} \boldsymbol{\lambda}(f) \prod_{d=1}^{D} \mathbf{A}_d(:,f)^T \mathbf{b}_d.
\end{aligned}$$

The above joint PDF $f_{\boldsymbol{Z}}$ model can be interpreted as a mixture of F product distributions, i.e., there exists a 'hidden' random variable H taking values in $\{1,\ldots,F\}$ that selects the operational component of the mixture, and given H the random variables $Z_1,\ldots,Z_D$ become independent (Fig. 5.3). Then, given $\mathbf{z}$, we can compute the likelihood using

$$\begin{aligned}
\widehat{f}_{\boldsymbol{Z}}(\mathbf{z}) &= (\mathbf{b}_1^T \mathbf{A}_1 \circledast \cdots \circledast \mathbf{b}_D^T \mathbf{A}_D)^T \boldsymbol{\lambda} \\
&= (\circledast_{d=1}^{D} \mathbf{b}_d^T \mathbf{A}_d) \boldsymbol{\lambda}.
\end{aligned}$$

The complexity of computing the likelihood of a data point $\mathbf{z}$ is $O(DKF)$. Tensor methods are commonly used to establish that the parameters of a generative model can be identified given higher order moments. This generative model theoretically has enough flexibility to capture highly complex distributions such as image manifolds. According to this model, a sample of the multivariate latent distribution can be generated by first drawing H according to p_H and then independently drawing samples for each variable Z_d from the conditional PDF $f_{Z_d|H}$.

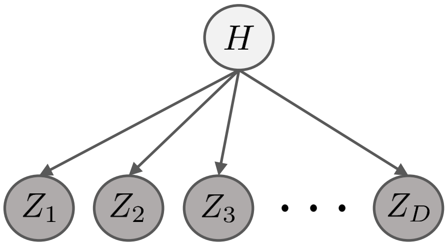


Figure 5.3: Our approach yields a generative model of the latent density, from which it is very easy to sample. This is because $f_{\boldsymbol{Z}}$ can be interpreted as a mixture of F product distributions i.e., admits a latent variable naive Bayes interpretation.

Maximum Likelihood Estimation

The above analysis suggests fitting a low-rank CPD model on the D-way Fourier series coefficient tensor $\underline{\boldsymbol{\Phi}}$. To this end, we build a generative probabilistic model that assigns high probability to the transformed observed samples. We propose fitting the Fourier tensor coefficients indirectly on the latent space representation of the training data. Let us define matrices $\mathbf{B}_d \in \mathbb{C}^{(2K+1)\times M}$ as

$$\mathbf{B}_d(K+1+k_d, m) = e^{-j2\pi k_d \mathbf{z}_m(d)}.$$

Given M samples, we define the following NLL cost term

$$\begin{aligned} \mathcal{L}_{\text{NLL}} &:= -\frac{1}{M}\sum_{m=1}^{M} \log\Big(\hat{f}_{\boldsymbol{Z}}(\boldsymbol{z}_m)\Big) \\ &= -\frac{1}{M}\sum_{m=1}^{M} \log\Big((\circledast_{d=1}^{D}(\mathbf{B}_d(:,m)^T\mathbf{A}_d))\boldsymbol{\lambda}\Big), \end{aligned} \tag{5.13}$$

where $\boldsymbol{A}_d(K+1+k_d, h)$ holds $\Phi_{Z_d|H=h}[k_d]$, $\boldsymbol{\lambda}(f)$ holds $p_H(f)$. Note that we do not instantiate the full Fourier coefficient tensor but rather recover it implicitly, by minimizing the NLL term.

We can further restrict the model and reduce its learnable parameters by 50% by noticing that each column of the factor matrix $\mathbf{A}_d$ holds a valid characteristic function which is by definition conjugate symmetric around the origin, and equal to one at the origin, i.e.,

$$\begin{aligned} \mathbf{A}_d(K+1,:) &= \mathbf{1}^T, \text{ and} \\ \mathbf{A}_d(K+1+k,:) &= \mathbf{A}_d^*(K+1-k,:), \end{aligned}$$

$$k \in [K], d \in [D].$$

Algorithm 5.1 CDE (Projected - SGD)

```
**Input:** $\mathbf{Z}, \mathbf{Z}_{\text{val}}, F, K, D, M_{\text{batch}}$
**Initialize** $\boldsymbol{\lambda}$, $\{\mathbf{A}_n\}_{n=1}^{D}$,$\boldsymbol{\theta}_g$, $\boldsymbol{\theta}_h$
**repeat**
    Sample $M_{\text{batch}}$ data points
    Update network parameters via SGD
    **for** $d = 1$ **to** $D$ **do**
        Update $\mathbf{A}_d$ via SGD
    **end for**
    Update $\boldsymbol{\lambda}$
    Project $\boldsymbol{\lambda}$ onto the probability simplex
    Compute $\mathcal{L}_{\text{NLL}} + \mathcal{L}_{\text{rec}}$ using $\mathbf{Z}_{\text{val}}$
**until** $\max_{\text{iter}}$ is reached or $\mathcal{L}_{\text{NLL}} + \mathcal{L}_{\text{rec}}$ stops diminishing
```

5.3.3 Optimization Procedure

By the above reasoning, instead of using decoupled two-stage training we suggest the following overall joint DR and density estimation optimization problem which we tackle by stochastic gradient descent

$$\begin{aligned}
&\min_{\boldsymbol{\theta}_h, \boldsymbol{\theta}_g, \{\mathbf{A}_d\}_{d=1}^{D}, \boldsymbol{\lambda}} \frac{1}{M} \sum_{m=1}^{M} \Big(\|\mathbf{x}_m - g(h(\mathbf{x}_m; \boldsymbol{\theta}_h); \boldsymbol{\theta}_g)\|^2 - \\
&- \mu \log \Big((\circledast_{d=1}^{D} (\mathbf{B}_d(:, m)^T \mathbf{A}_d)) \boldsymbol{\lambda} \Big) \Big) + \sum_{d=1}^{D} \rho \|\mathbf{A}_d\|_F^2 \\
&\text{s.t. } \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{1}^T \boldsymbol{\lambda} = 1, \\
&\quad \mathbf{A}_d(K+1, :) = \mathbf{1}^T, \\
&\quad \mathbf{A}_d(K+1+k, :) = \mathbf{A}_d^*(K+1-k, :).
\end{aligned}$$

The optimization criterion that guides CDE consists of three terms: the reconstruction loss of the auto-encoder, NLL of the density estimation component, and Frobenius norm regularization. In the above formulation, $\mu \geq 0$ is a regularization parameter which balances the reconstruction error versus the maximum likelihood estimation. The number of coefficients K controls the desired smoothness of the joint density, while the number of latent dimensions D and the rank F control the expressivity.

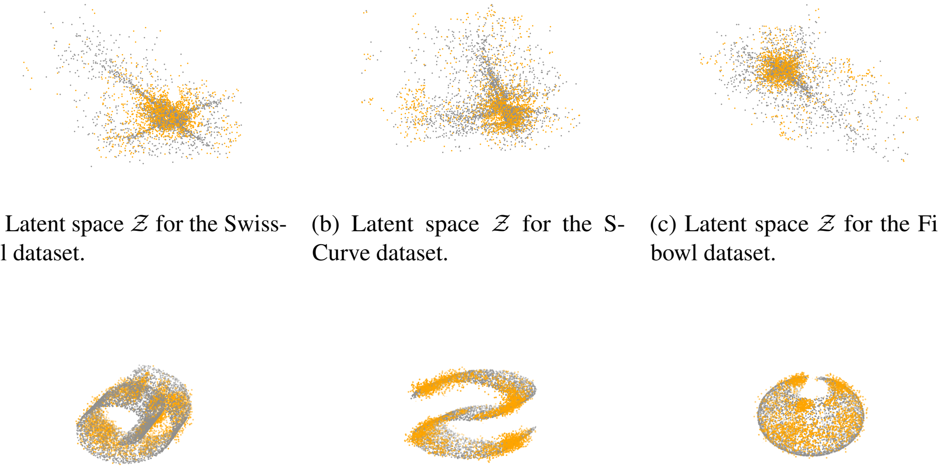
We refer to this approach as Compressed-domain Density Estimation with Hidden Tensor Factorization (CDE-HTF). Figure (5.1) presents the network structure corresponding to the final joint problem formulation. We solve the proposed optimization problem using projected Stochastic Gradient Descent (SGD). We initialize the Fourier tensor-related parameters using random initialization, while for $\boldsymbol{\theta}_g$ and $\boldsymbol{\theta}_h$, it was empirically observed that auto-encoder pre-training was most effective. At each step we update $\boldsymbol{\theta}_g$, $\boldsymbol{\theta}_h$, factors $\mathbf{A}_d$ and $\boldsymbol{\lambda}$ simultaneously by first sampling a batch of size M_{batch} and taking a gradient step. After that, we project $\boldsymbol{\lambda}$ to the probability simplex. For the termination of the algorithm we compute the cost function on a validation set and stop if a number of maximum iterations has been reached or the log-likelihood has not improved in the last T iterations. The full procedure is shown in Algorithm 5.1.

5.4 Experimental Results

In this section, we evaluate the proposed approach using various datasets and evaluation criteria, ranging from sampling of toy 3-D examples to real MNIST and Fashion-MNIST images, and regression and anomaly detection tasks using standard tabular datasets from the UCI database. We compare with density estimation and anomaly detection baselines from the deep learning literature, including standard VAEs, Real-NVP, MAF, MADE and GF for reference.

5.4.1 Toy Datasets

We begin with modeling the joint density function of a subset of MNIST images, consisting of only 0s and 8s using the proposed CDE model. For these experiments the network architecture considered is a four-layer network encoder of 784, 128, 64, 32, neurons respectively (with the decoder being a mirrored version of the encoder), and ReLU activation functions. In Figure 5.2, we visualize random samples learned by the proposed model for different values of F and K to show that only a few parameters are needed to obtain a model that is flexible enough to fit the distribution in great detail. The first row represents results for fixed $F = 4$ and different values of $K \in [1, 3, 5]$, while the second row represents results for fixed $K = 3$ and different values of $F \in [2, 4, 8]$. Increasing K generates sharper digits, while increasing F better differentiates the samples of the two digits.



(a) Latent space $\mathcal{Z}$ for the Swiss-roll dataset.

(b) Latent space $\mathcal{Z}$ for the S-Curve dataset.

(c) Latent space $\mathcal{Z}$ for the Fish bowl dataset.

(d) Data space $\mathcal{X}$ for the Swiss-roll dataset.

(e) Data space $\mathcal{X}$ for the S-Curve dataset.

(f) Data space $\mathcal{X}$ for the Fish bowl dataset.

Figure 5.4: Three toy 3-D (gray data-points) datasets and corresponding samples drawn via CDE. CDE maps raw samples to latent features through a mapping $\boldsymbol{h} : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ that is learned by an auto-encoder. The non-parametric latent density model allows efficient sample generation in the latent space (orange data-points on the images of the first row). The approximate inverse map of the decoder, back-transforms latent samples into samples in the data space (orange data-points on the images of the second row. See text for further details.

We then continue with modeling three toy 3-D datasets, namely Swiss-roll, S-curve, and Fish-bowl where we are given 3000 training data points from each dataset and we use CDE to randomly sample 5000 synthetic data points. For all the datasets considered, the auto-encoder structure we use is FC (3, 128, ReLU), FC (128, 64, ReLU), FC (64, 32, ReLU), FC (32, 2, none), FC (2, 32, ReLU), FC (32, 64, ReLU), FC (64, 128, ReLU), FC (128, 3, none), and the tensor parameters are set to $(K, F) = (5, 10)$. We provide an illustration of the 2-D latent space learned via latent synthetic samples drawn using the proposed method. Using the approximate inverse map of the decoder, we can back-transform latent samples into samples in the original data space and visualize the learned distribution in the original space. The results in Figure 5.4 showcase that the proposed framework is capable of learning the structure of the data, notably in critical regions where the curvature is very high – which is interesting.

5.4.2 Tabular Datasets

Data set	**N**	**M**	**VAE**	**Real-NVP**	**MAF**	**GF**	**CDE**
MINIBOONE	51	130065	3.69 ± 0.68	3.18 ± 0.16	3.17 ± 0.45	$\mathbf{3.15 \pm 0.52}$	$\mathbf{3.12 \pm 0.43}$
BSDS300	63	50000	0.37 ± 0.08	0.60 ± 0.02	$\mathbf{0.32 \pm 0.03}$	0.48 ± 0.03	$\mathbf{0.30 \pm 0.02}$
Gas Sensor	128	13910	1.46 ± 0.04	1.31 ± 0.31	1.23 ± 0.44	1.23 ± 0.30	$\mathbf{1.21 \pm 0.84}$
Musk	168	6598	0.40 ± 0.51	0.22 ± 0.68	$\mathbf{0.12 \pm 0.07}$	0.19 ± 0.08	$\mathbf{0.13 \pm 0.23}$
IDA2016Challenge	171	76000	0.23 ± 0.06	0.18 ± 0.09	$\mathbf{0.12 \pm 0.05}$	$\mathbf{0.10 \pm 0.09}$	$\mathbf{0.11 \pm 0.16}$
BlogFeedback	281	60021	2.43 ± 0.22	$\mathbf{2.35 \pm 0.21}$	2.37 ± 0.22	2.37 ± 0.42	$\mathbf{2.32 \pm 0.17}$
ISOLET	617	7797	1.41 ± 0.31	$\mathbf{1.09 \pm 0.04}$	1.11 ± 0.38	1.85 ± 0.21	$\mathbf{1.03 \pm 0.56}$

Table 5.1: Dataset information and test-set MAE on UCI datasets. For each test sample, we choose the response variable Y at random and estimate using Stochastic Gradient Ascent.

Next, we evaluate the proposed approach on several regression tasks using tabular datasets described in Table 5.1. We compare our approach against standard baselines. For each dataset we split the data in two subsets: 80% used for training and 20% used for testing. The parameters for each method are chosen using 5-fold cross-validation and we report the Mean Absolute Error (MAE) for the unseen data samples. The results underline the superior performance of the proposed method for inference tasks. Regarding the auto-encoder's parameters for the datasets, the number of hidden layers varies according to the dataset dimensionality – from three (MINIBOONE dataset) to six (ISOLET dataset). The most critical one is the hidden layer dimensionality $D \in \{16, 32, 48, 64\}$, and concerning tensor parameters, the important ones are the tensor rank $F \in \{5, 10, 20, 30, 50\}$ and the smoothing parameter $K \in \{5, 10, 15, 20, 30\}$. The learning, drop-out rates, and regularization parameters were sampled from a uniform distribution in the range $[0.05, 0.2]$. That is, we randomly sampled parameters from this range using a uniform distribution and used cross-validation to select the best set of parameters. The initial weight matrices were all sampled from the uniform distribution within the range $[-1, 1]$.

We used Adam optimizer [110] with a batch size of 500. Overall, we observe that CDE outperforms the baselines on almost all datasets, and performs comparable to the winning method in the remaining ones. More specifically, CDE shows significantly lower test-set MAE on MINIBOONE, Gas Sensor, and BlogFeedback dataset compared to Real-NVP and MAF, which appear to be the next best performing models. Additionally CDE has a clear lead against the VAE especially on Gas Sensor, IDA2016Challenge, and ISOLET dataset, which confirms our initial motivation of using a non-parametric latent density estimator to improve model flexibility.

5.4.3 Image Datasets

We consider grayscale images from the MNIST and Fashion-MNIST database, which both contain a set of $60,000$ training observations of 28×28 pixels ($N = 784$) from 10 classes. Regarding MNIST, the most critical parameters include the encoder architecture, which consists of four hidden layers of 784, 128, 64, 32, neurons respectively (the decoder network has a mirrored structure), ReLU activation function, tensor rank which is fixed to $F = 50$, smoothing parameter $K = 5$, and the learning rate which is fixed to $\alpha = 0.0001$. For Fashion-MNIST, the model parameters are fixed to be the same as for the MNIST dataset, with the encoder network $(784, 256, 128, 64, 32, 16)$ being the only exception. See also Figure 5.7, which shows the distribution of the components of the latent variable H after training the generative model. These bar-plots tell us that the rank of the compressed density model is essentially $F = 30$, but for exploratory modeling purposes, we set $F = 50$ (the parameter F is actually an upper bound on tensor rank) and encourage sparsity of the latent components through our optimization problem formulation. We sample from the learned lower-dimensional latent joint generative model ($D = 32$ for MNIST and $D = 16$ for Fashion-MNIST) – See Section 5.3.2 for the detailed sampling process – and provide visualization of the generated data. The resulting 100 randomly drawn samples, which are impressively more pleasing to the eye in direct comparison with other well-known models such as MADE and GFs are shown in Figures 5.5 and 5.6.

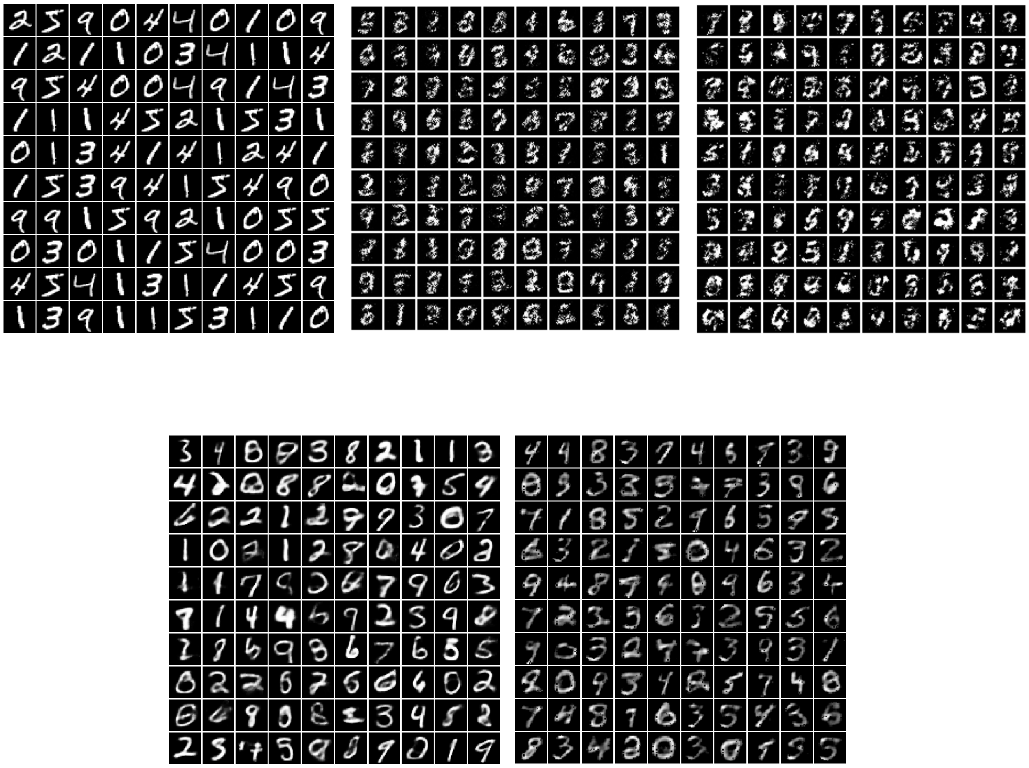


Figure 5.5: Synthetic samples drawn from the joint density of MNIST from various models. From left to right: Ground Truth, Masked auto-encoder for Distribution Estimation (MADE), Gaussianization Flows (GF), Variational auto-encoder (VAE), **Proposed: CDE**.

We note that in our case, no prior is imposed on the latent variables, so we do not have issues such as "posterior collapse" which has been observed to occur in VAEs. Degeneration ("collapse") could occur in principle for our model if the joint characteristic function is reduced to a rank-1 tensor during training. An easy way to check this would be through the mixture distribution of the latent components, i.e., the probability mass function of the latent variable H, $p_H(f)$. If after training the generative model only one element of $p_H(\cdot)$ is nonzero, i.e., only one component "survives", then the characteristic tensor is rank-1, and degeneration / collapse to a separable latent space distribution occurs. Throughout our experiments, although the datasets can be well approximated with lower ranks, such degeneration did not take place. We show this in Figure 5.7, where we demonstrate the distribution of the latent variable H after training the

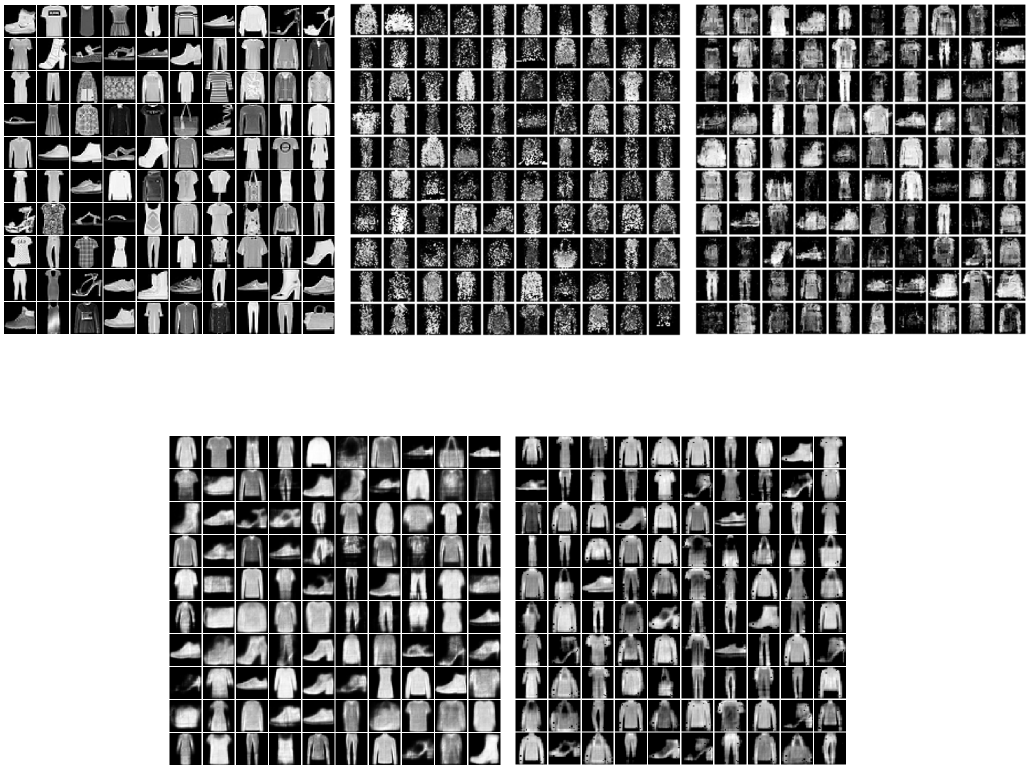


Figure 5.6: Synthetic samples drawn from the joint density of Fashion-MNIST from various models. From left to right: Ground Truth, Masked auto-encoder for Distribution Estimation (MADE), Gaussianization Flows (GF), Variational auto-encoder (VAE), **Proposed: CDE**.

generative model on MNIST (left figure) and Fashion-MNIST (right figure).

Although at first glance, the images generated by VAE have cleaner and thicker strokes, they are also more blurry and distorted than those produced by CDE. The Fashion-MNIST data help bring this out more clearly: one can see that CDE allows capturing and representing more details in the items, while the samples drawn from the VAE are much more blurry. The overall conclusion from the experiments is that while the quality of our synthetic images is competitive against the VAE and considerably better than that of the samples generated by the rest of the models considered, the proposed framework is superior for regression tasks.

Because the dimensionality of the latent space is an important factor in the architecture,

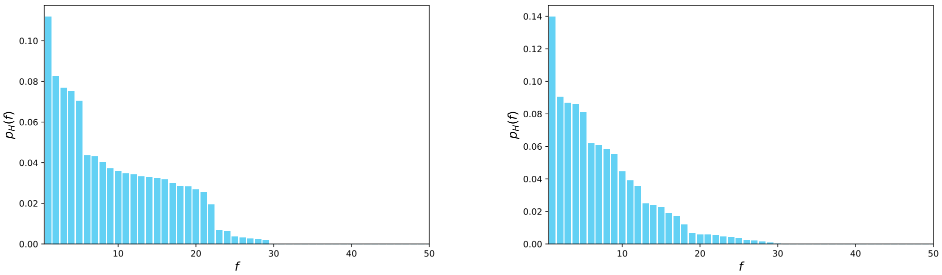


Figure 5.7: Distribution of the components of the latent variable H after training the generative model on MNIST (left figure) and Fashion-MNIST (right figure).

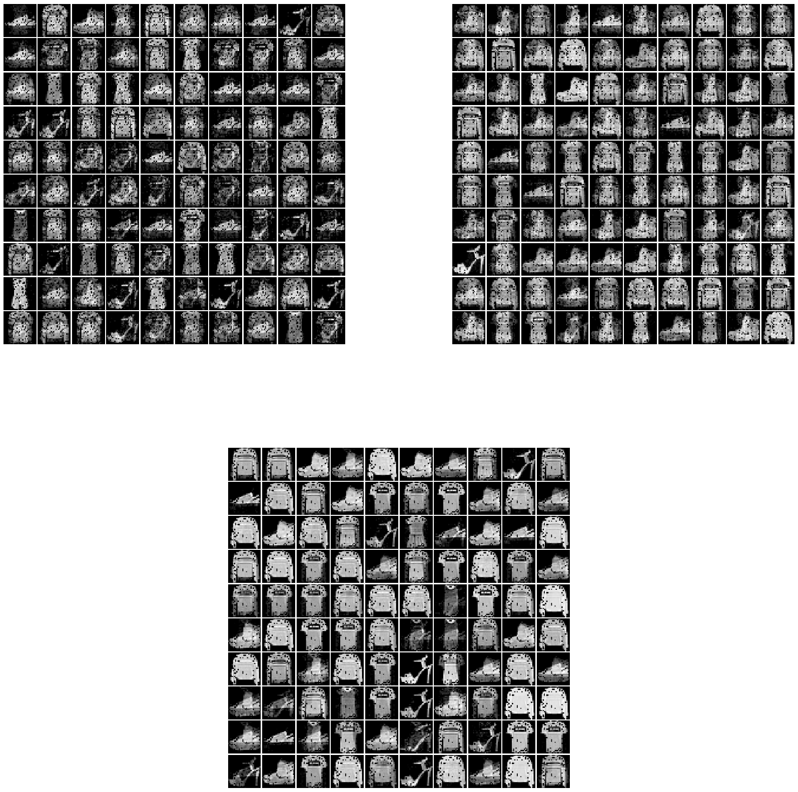


Figure 5.8: Fashion-MNIST samples for different values of latent dimensionality ($D = 6, D = 10, D = 14$).

we demonstrate sampling results for different dimensionalities on Fashion-MNIST. We repeat the experiment by setting $F = 50$ and $K = 5$. We randomly sample from the learned lower-dimensional latent joint generative model for $D = 6, 10, 14$ and provide visualization of the generated data in the input space. The resulting 100 randomly drawn samples are shown in Figure 5.8. We can see that increasing latent dimenionality from $D = 6$ to $D = 14$ significantly improves sample quality. Results for $D = 16$ can be seen on Figure 5.6 which yield high-quality image samples. Larger latent dimensionality does not further improve the samples (and eventually degrades it, as expected) so we opt for the smallest dimensionality that yields good results.

5.4.4 Anomaly Detection Using Real Data

We use four public datasets[‡]: KDDCUP99, Thyroid, Arrhythmia, and KDDCUP-Rev. The (instance number M, dimension N, anomaly ratio (%)) of each dataset is (494021, 121, 20), (3772, 6, 2.5), (452, 274, 15), and (121597, 121, 20). For categorical features, we further used one-hot representation to encode them. Regarding Thyroid, there are three classes in the original dataset. We treat the hyperfunction class as the anomaly class and the other two classes are treated as normal class. Regarding Arrhythmia, the smallest classes, including $3, 4, 5, 7, 8, 9, 14$, and 15, are combined to form the anomaly class, and the rest of the classes are combined to form the normal class. We randomly extracted 50% of the data and assigned it to the training subset and the rest to the testing subset. In our experimental setting for anomaly detection, clean training data is adopted – that is, during the training, only normal data were used. We assume that the percentage of anomalous data points is known, and our goal is to detect which data points in the testing subset are most likely to be outliers. Towards this end, at the testing stage the likelihood of each testing sample in the compressed domain was evaluated and sorted in order to detect the anomalies as the points with the smallest likelihood. By knowing the percentage of anomalies, we can indicate the exact number of outliers and output the data samples with the smallest likelihood values.

The network structures of CDE used for individual datasets are summarized as follows.

- For KDDCUP, the auto-encoder network runs with FC (120, 60, tanh), FC (60, 30, tanh), FC (30, 20, tanh), FC (20, 10, none), FC (10, 20, tanh), FC (20, 30, tanh), FC (30, 60,

[‡] Datasets can be downloaded at `https://kdd.ics.uci.edu/` and `http://odds.cs.stonybrook.edu`.

Dataset	Methods	Precision	Recall	F1
KDDCup	VAE	0.9524 (0.0047)	0.9140 (0.0052)	0.9326 (0.0052)
	DAGMM	0.9427 (0.0055)	0.9578 (0.0051)	0.9507 (0.0052)
	CDE	**0.9565 (0.0046)**	**0.9712 (0.0048)**	**0.9641 (0.0045)**
Thyroid	VAE	**0.6575 (0.0371)**	0.5743 (0.0583)	0.6357 (0.0583)
	DAGMM	0.4658 (0.0481)	0.4902 (0.0452)	0.4752 (0.0497)
	CDE	0.6560 (0.0572)	**0.6740 (0.0493)**	**0.6703 (0.0592)**
Arrythmia	VAE	0.4375 (0.0538)	0.4340 (0.0496)	0.4302 (0.0482)
	DAGMM	**0.5358 (0.0468)**	**0.5592 (0.0475)**	**0.5403 (0.0421)**
	CDE	0.5299 (0.0400)	0.5551 (0.0418)	0.5389 (0.0420)
KDDCup-rev	VAE	0.9771 (0.0058)	0.9779 (0.0004)	0.9678 (0.0018)
	DAGMM	0.9762 (0.0038)	0.9823 (0.0017)	0.9709 (0.0021)
	CDE	**0.9866 (0.0008)**	**0.9872 (0.0015)**	**0.9871 (0.0012)**

Table 5.2: Average (over 20 runs) and standard deviations (in brackets) of Precision, Recall and F1 score.

tanh), FC (60, 120, none).

- The auto-encoder network for Thyroid runs with FC (6, 12, tanh), FC (12, 4, tanh), FC (4, 2, none), FC (2, 4, tanh), FC (4, 12, tanh), FC (12, 6, none).
- The auto-encoder network for Arrhythmia runs with FC (274, 64, tanh), FC (64, 32, none), FC (32, 64, tanh), FC (64, 274, none).
- The auto-encoder network for KDDCUP-Rev runs with FC (120, 60, tanh), FC (60, 30, tanh), FC (30, 20, tanh), FC (20, 10, none), FC (10, 20, tanh), FC (20, 30, tanh), FC (30, 60, tanh), FC (60, 120, none).

As metrics, precision, recall, and F1 score are calculated. We run experiments 20 times for each dataset split by 20 different random seeds. Table 5.2 reports the average scores and standard deviations (in brackets). Compared to the baselines considered, CDE achieves the highest performance – CDE is superior to both VAE and DAGMM for each evaluation criterion except for precision on the Arrhythmia dataset. This result suggests that our proposed latent nonparametric density estimation approach can provide more expressive models which can bring better performance in important detection tasks as well.

5.5 Chapter Summary

In this chapter, we introduced Compressed Density Estimation (CDE), a novel probabilistic latent density model that builds upon deep auto-encoder networks and non-parametric multivariate density modeling in the Fourier domain. We propose using an auto-encoder to embed the data into a latent code space by minimizing reconstruction error, and a regularization over the latent space which maximizes the likelihood of the hidden code vector and is modelled using a low-rank characteristic tensor approach.

We investigated whether leveraging probabilistic (non-parametric) low-rank tensor models in the Fourier domain as a latent distribution model can improve the expressivity of density models. By jointly optimizing the auto-encoder and the latent density model, we can better capture the latent distribution of data representations obtained by the auto-encoder. Experimental results demonstrated the effectiveness of the proposed joint optimization approach, which is able to learn complex high dimensional distributions using a parsimonious model with few tuning parameters.

Chapter 6

Learning Multivariate CDFs and Copulas using Tensor Factorization

6.1 Introduction

Modeling complex data distributions is a task of central interest in statistics and machine learning. Given an accurate and tractable estimate of the joint distribution function, various kinds of statistical tasks can follow naturally including fast sampling, tractable computation of expectations, and deriving conditional and marginal densities. To list a few recent applications, such models have demonstrated success in generating high-fidelity images [2], [3], realistic speech synthesis [4], [5]; semi-supervised learning [6]; reinforcement learning [7]; and detecting adversarial data [8]. The purpose of this work is to introduce a new class of universal estimators for multivariate distributions based on CDFs and the Canonical Polyadic (tensor-rank) decomposition, and to demonstrate their direct applicability and efficiency in missing data imputation, sampling, density estimation, and regression tasks.

Distribution modeling is often studied under the perspective of non-parametric PDF estimation, in which histogram, kernel [10], [11], and orthogonal series methods [12], [13], [14] are popular approaches with well-understood statistical properties. Although these estimators are data-driven and do not impose restrictive parameterisations on the form of the data distribution, they usually have poor performance on datasets of high dimensions because of the "curse of dimensionality". Currently, the most prominent methods for modeling multivariate distributions rely on neural networks [16], [68], [69], [17]. Such methods are capable of modeling higher dimensional data such as images and sound, either implicitly or explicitly. However, most of them are black-box models [111] without any identifiability guarantees, lacking the simplicity and interpretability of the classical methods. Additionally, they lack the ability to efficiently compute expectations, marginalize over subsets of variables, and evaluate conditionals, which is limiting in many critical machine learning applications.

This paper, attempts to bridge the gap between principled traditional non-parametric statistics and the scalability benefits of neural-based models by developing two variants of a rank-constrained estimator for multivariate CDFs based on tensor rank decomposition – known as Canonical Polyadic (CP) or CANDECOMP/PARAFAC Decomposition [25], [26], [23]. Our starting point is that any grid-sampled version of an N-dimensional CDF is an $N-$way cumulative probability tensor $\widehat{\mathcal{F}}$, evaluated on a predefined $I_1 \times I_2 \times \cdots \times I_N$ grid G. We will refer to $\widehat{\mathcal{F}}$ as grid-sampled CDF tensor. The evaluation grid G describes the (finite) levels/cut-offs of the CDF for every dimension and can be taken to be the cartesian product of the training samples in each dimension, or reduced via scalar or vector k-means. Each element of $\widehat{\mathcal{F}}$ can be easily estimated via sample averaging from realizations of the random vector of interest.

Any tensor can be decomposed as a sum of R rank-1 tensors, for high-enough but finite R [23]. To maintain direct control over the number of tensor parameters with growing dimensionality N, which entails $\mathcal{O}\left(\prod_{n=1}^{N} I_n\right)$ CDF tensor elements, we introduce the reconstructed approximation of $\widehat{\mathcal{F}}$, using the rank-R N-dimensional parameterization $\mathcal{F} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. Such parametrization has much fewer degrees of freedom designated by the rank and size of the the tensor. Tensor decompositions arise as a powerful tool for extracting meaningful latent structure from given data and can encode the salient characteristics of the multivariate grid-sampled CDF tensor $\widehat{\mathcal{F}}$. We seek to minimize the squared loss between $\mathcal{F}$ and the empirical CDF tensor $\widehat{\mathcal{F}}$. We formulate this task both directly, i.e., by forming and decomposing $\widehat{\mathcal{F}}$, and indirectly, as a hidden tensor factorization problem, i.e., the pertinent parts of the latent factors of the CPD model are updated from rank-1 measurements of the "hidden" discretized CDF tensor $\widehat{\mathcal{F}}$. From an algorithmic perspective, we propose alternating optimization, where each matrix factor is updated using ADMM, as well as stochastic optimization using Adam to allow scaling to larger datasets.

6.1.1 Contributions

In summary, the present paper shows that:

- *Every* multivariate CDF evaluated on a predefined grid admits a compact representation via a latent variable naive Bayes model with bounded number of hidden states equal to the rank of the grid-sampled CDF tensor.
- This affords easy sampling, marginalization (by discarding the subset of factor matrices corresponding to the variables we are not interesting in), derivation of conditional distributions and expectations, and uncertainty quantification – bypassing the curse of dimensionality.
- The proposed model also affords direct and efficient estimates of (possibly semi-infinite) "box" probabilities, which is important for classification tasks. Multivariate PDF estimators, on the other hand, require multidimensional integration (analytical, numerical, or sampling-based Monte-Carlo) to estimate box probabilities, which is cumbersome and often intractable.
- At the same time, the proposed rank-constrained estimator is unassuming of the structure of the data (thus offering greater expressive power) and identifiable under relatively mild rank conditions – see [23].
- On the experimental side, our results indicate that, perhaps surprisingly, *estimating the grid-sampled CDF and then deriving a PDF estimate from it yields improved performance relative to direct PDF estimation in several machine learning applications of interest*. In addition, the performance of *the proposed non-parametric model in the Copula domain outperforms state-of-the-art Copula-based baselines*.

6.2 Background

6.2.1 Related work

Unsupervised learning of multivariate distributions has seen tremendous progress over recent years, for the case of PDF modeling in particular. Classical methods in the literature include kernel density estimation (KDE) [10, 11], histogram density estimation (HDE), and Orthogonal Series Density Estimation (OSDE) [12–14]. All of the aforementioned methods, however are inefficient for datasets with higher dimensionalities. Neural network-based approaches for distribution estimation have recently shown promising results in high-dimensional problems. Auto-regressive (AR) models such as [5, 15] decompose the distribution into a product of conditionals, where each conditional is modeled by a parametric distribution (e.g., Gaussian or mixture of Gaussians in the continuous case). Normalizing flows (NFs) [112] represent a density value though an invertible transformation of latent variables with known density.

On the down-side, AR models are naturally sensitive to the order of the variables/features while strong network constraints of NFs can be restrictive for model expressiveness. Most importantly, AR and NF do not yield an explicit estimate of the density function; they are ‘oracles’ that can be queried to output *an estimate of the density at any given input point*, i.e., to generate samples of the sought density – the difference is important. Therefore, given a trained model, calculating expectations, marginal and conditional distributions is not straightforward with these methods. The same holds for generative adversarial networks [17] (GANs) as they do not allow for likelihood evaluation on held-out data. Furthermore, deep multivariate CDF based models such as [111], do not address model identifiability, and can not guarantee the recovery of the true latent factors that generated the observed samples.

6.2.2 Tensor modeling of distributions

Tensor models for estimating distributions have been proposed for both discrete and continuous variables. In the discrete case, the work in [18] showed that any joint PMF can be represented as an N-way probability tensor and by introducing a CPD model, every multivariate PMF can be represented by a latent variable naive Bayes model with a finite number of latent states. For continuous random vectors, the joint PDF can no longer be directly represented by a tensor. Earlier work ([74]) has dealt with latent variable models, but not general distributions. In contrast to prior work ([74], [77]), we make no assumptions regarding a multivariate mixture model of non-parametric product distributions in this paper. Another line of work (see [21], [22]) proposed a “universal” approach for smooth, compactly supported multivariate densities by representing the underlying density in terms of a finite tensor of leading Fourier coefficients. Our work requires less restrictive assumptions, as it also works with discrete or mixed random variables, of possibly unbounded support.

6.3 Methodology

6.3.1 Sketching multivariate CDFs

Let $\mathbf{X} \in \mathbf{R}^N$, $\mathbf{X} \sim F_{\mathbf{X}}$, be a random vector comprising N discrete/categorical or continuous constituent random variables (features) $\mathbf{X} := [X_1, \ldots, X_N]^T$. Given a collection $\mathcal{D} = \{\mathbf{x}_m\}_{m=1}^M$ of independently and identically generated observations $\mathbf{x}_m = [\mathbf{x}_m(1), \ldots, \mathbf{x}_m(N)]^T \in \mathbf{R}^N$ sampled from $F_{\mathbf{X}}$, the goal is to find an accurate estimate $\widehat{F}_{\mathbf{X}}$ of the true distribution function $F_{\mathbf{X}}$ with limited assumptions about the underlying model.

Let us denote the CDF cut-offs of a variable X_n as $[x_1(n), \ldots, x_{I_n}(n)]^T \in \mathbf{R}^{I_n}$. In each dimension $n \in [N]$, the cut-off points are sorted in increasing order, e.g., $x_1(n) < \cdots < x_{i_n}(n) < \cdots < x_{I_n}(n)$. Considering an evaluation (target) point $\mathbf{x}_{\boldsymbol{i}} = [x_{i_1}(1), x_{i_2}(2), \ldots, x_{i_N}(N)]^T$, a natural estimator of the sought CDF is the empirical cumulative distribution function (ECDF) $\widehat{F}_{\mathbf{X}}$,

$$\begin{aligned}\widehat{F}_{\mathbf{X}}(\mathbf{x}_{\boldsymbol{i}}) &= \frac{1}{M}\sum_{m=1}^{M} \mathbf{1}\left(\mathbf{x}_m \leq \mathbf{x}_{\boldsymbol{i}}\right) \\ &= \frac{1}{M}\sum_{m=1}^{M} \mathbf{1}\left(\mathbf{x}_m(1) \leq x_{i_1}(1), \ldots, \mathbf{x}_m(N) \leq x_{i_N}(N)\right). \qquad (6.1)\end{aligned}$$

By considering an $I_1 \times I_2 \times \cdots \times I_N$ grid G (with possibly non-uniform mesh) of target evaluation points $\mathbf{x}_{\boldsymbol{i}}$, defined as the cartesian product of the observed 1-D samples (or a reduced set of surrogates obtained via k-means), one can obtain a discretized version of the ECDF, which we will be referring to as grid-sampled (empirical) CDF. We denote the multivariate CDF evaluation grid G as $G = \{[x_{i_1}(1), x_{i_2}(2), \ldots, x_{i_N}(N)]^T \in R^N \mid i_1 \in [I_1], i_2 \in [I_2], \ldots, i_N \in [I_N]\}$.

6.3.2 Motivation

Approximating the cumulative distribution function, instead of the multivariate density [77], has significant advantages. A joint CDF (and its grid-sampled sketch) always exists in contrast to a PDF/PMF. Instead of considering one type of random variables, a CDF can simultaneously model discrete, continuous or mixed random variables. All observations “up to” a point $\mathbf{x}_{\boldsymbol{i}}$ are considered for a single point CDF estimate $F_{\mathbf{X}}(\mathbf{x}_{\boldsymbol{i}})$, allowing estimation in places where there is a scarcity of “local” samples. Additionally, it affords direct and efficient estimates of (possibly semi-infinite) “box” probabilities, e.g. of type $Pr\{35 < \text{Age} \leq 45, \cdots, 55 < \text{Income} \leq 70\}$, avoiding the (potentially intractable) multi-dimensional integration of the joint PDF. These probabilities are needed for classification and anomaly detection. For example, in the 3-D case, only 8 basic operations (additions, subtractions) of the multivariate CDF are required for estimating a single interval probability, whereas a 3-D integration is required in the joint PDF case, and the problem is compounded in higher-dimensional cases, where the only tractable option is Monte-Carlo integration, which has poor accuracy as it suffers from the curse of dimensionality – an exponential number of Monte-Carlo samples is needed for accurate estimation.

Still, sufficiently reliable estimates of $\widehat{\mathcal{F}}_{\mathbf{X}}$ in high-dimensional spaces require a number of observations that grows quickly with the tensor order N, $\mathcal{O}\left(\prod_{n=1}^{N} I_n\right)$. To reduce the number of free parameters and scale up to higher dimensions, certain assumptions must be made. In our case, the sole assumption is that the CDF tensor can be well-approximated by a low-rank model $\mathcal{F}_{\mathbf{X}}$. CPD is a powerful model that can parsimoniously represent the high-order interactions among multi-way data exactly or approximately, leading to significant reduction in the number of parameters. Even in high dimensional settings, data can be approximated to live in relatively low dimensional space by exploiting the low rank property [113], [23].

6.3.3 Modeling and Interpretation

So far we have shown that given a set of M realizations in the training set and an evaluation grid G indexed by $[i_1, i_2, \ldots, i_N]^T$, the multivariate grid-sampled empirical CDF induced by G can always be represented as a finite CDF tensor $\widehat{\mathcal{F}}_{\mathbf{X}}$, whose elements can be estimated by the following sum

$$\widehat{\mathcal{F}}_{\mathbf{X}}(i_1, i_2, \ldots, i_N) = \frac{1}{M} \sum_{m=1}^{M} \mathbf{1}\left(\mathbf{x}_m \leq \mathbf{x}_{\boldsymbol{i}}\right). \tag{6.2}$$

Assuming that $\{\mathbf{x}_m\}_{m=1}^{M}$ is i.i.d. in m, it is easy to verify that $\widehat{\mathcal{F}}$ is an unbiased estimator of the corresponding joint cumulative distribution function [114], i.e.,$\mathbb{E}\left[\widehat{\mathcal{F}}_{\mathbf{X}}(i_1, i_2, \ldots, i_N)\right] = F\left(x_{i_1}(1), \ldots, x_{i_N}(n)\right)$, for all $\left[x_{i_1}(1), \ldots, x_{i_N}(n)\right]^T \in G$ and $M \geq 1$.

We will leverage the universal approximation ability of tensors by a CPD model to build a parameterization of multivariate grid-sampled (empirical) CDFs. We focus on the first R *principal components* of the resulting tensor, i.e., we introduce a low-rank parametrization $\mathcal{F}_{\mathbf{X}}$ of the grid-sampled *CDF tensor* $\widehat{\mathcal{F}}_{\mathbf{X}}$, $\mathcal{F}_{\mathbf{X}}(i_1, i_2, \ldots, i_N) = \sum_{h=1}^{R} \boldsymbol{\lambda}(h) \prod_{n=1}^{N} \mathbf{A}_n(i_n, h)$. By

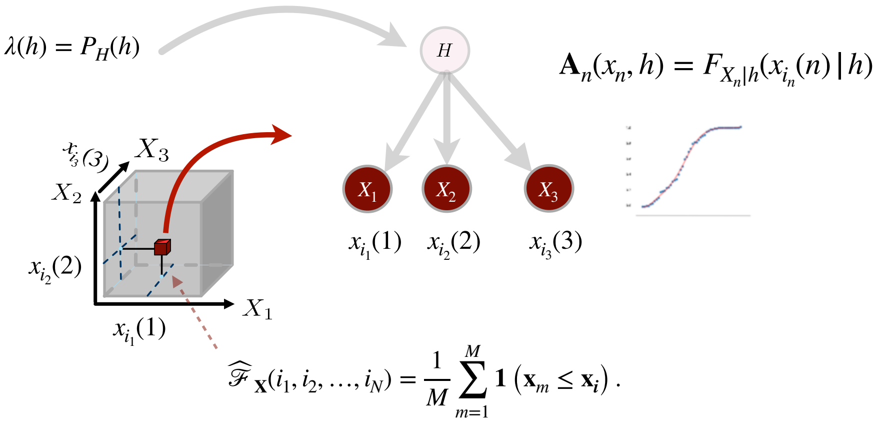

$\lambda(h) = P_H(h)$
H
$\mathbf{A}_n(x_n, h) = F_{X_n|h}(x_{i_n}(n)\,|\,h)$
$x_{i_3}(3)$
X_3
X_2
$x_{i_2}(2)$
$x_{i_1}(1)$
X_1
X_1
X_2
X_3
$x_{i_1}(1)$
$x_{i_2}(2)$
$x_{i_3}(3)$
$\widehat{\mathscr{F}}_{\mathbf{X}}(i_1, i_2, \ldots, i_N) = \frac{1}{M}\sum_{m=1}^{M} \mathbf{1}\left(\mathbf{x}_m \leq \mathbf{x}_i\right).$

Figure 6.1: Samples from the empirical grid-sampled CDF tensor can be thought as being generated via a latent variable naive Bayes model. This generative model says that observed data samples can be thought to be generated by a two-step process: First a value for H is generated according to $P_H(h)$ and then, a value $x_{i_n}(n)$ is generated according to $F_{X_n|h}$.

enforcing non-negativity and "valid CDF" constraints on each column of the CPD factors, i.e., $\mathbf{A}_n(, h)$ to be non-negative, non-decreasing, and the last element to be equal to 1, as well as simplex constraints $\mathbf{1}^T\boldsymbol{\lambda} = 1$ on $\boldsymbol{\lambda}$, it can be shown, using an argument first made for the case of multivariate PMFs by [18], that every multivariate grid-sampled CDF can be thought to be generated according to a hidden variable model,

$$\mathcal{F}_{\mathbf{X}}(i_1, \ldots, i_N) = \sum_{h=1}^{R} P_H(h) \prod_{n=1}^{N} F_{X_n|H}(x_{i_n}(n)|h). \tag{6.3}$$

This hidden variable H takes R possible values (R is the tensor rank) and follows a prior distribution $P_H(h) = Pr(H = h)$. According to the resulting model, given $H = h$, the manifest variables $X_1, \ldots, X_N$ are generated independently via an unknown mapping expressed by N univariate (discretized) conditional distributions $F_{X_n|H}(x_{i_n}(n)|h) = Pr(X_n \leq x_{i_n}(n)|H = h)$. This model is known as latent variable naive Bayes (NB) model and a visualization is presented in Figure 6.1.

Remarkably, such representation is *universal* if one allows the hidden variable to have a sufficiently rich finite alphabet. The parameters of this model $F_{X_n|H}, P_H$ will be recovered (or estimated) jointly by decomposing tensor $\widehat{\mathcal{F}}_{\mathbf{X}}$. A key property of the CPD is that the rank-1 components are unique under mild conditions. Uniqueness offers identification guarantees under ideal conditions – that is, when the true model is low rank and the correct rank is used. For model approximation, uniqueness means that there is a single model of the data *for fixed residuals* that is consistent with what we have observed. See [23] for detailed identifiability results.

The result in (6.3) states that a decomposition of multivariate grid-sampled CDFs in terms of $1-D$ conditionals can be computed via constrained CPD. Conversely, if one *assumes* that a multivariate CDF is a finite mixture of R product distributions of mixed random variables, the discretized conditional CDFs are identifiable and can be recovered by decomposing the exact CDF tensor,

$$\begin{aligned} F_{\mathbf{X}}(\mathbf{x_i}) &= \mathbb{E}\left[\mathbf{1}\left(X_1 \le x_{i_1}(1), \ldots, X_N \le x_{i_N}(N)\right)\right] \\ &= \sum_{h=1}^{R} P_H(h) \prod_{n=1}^{N} \mathbb{E}\left[\mathbf{1}\{X_n \le x_{i_n}(n)\}|h\right] \\ &= \sum_{h=1}^{R} P_H(h) \prod_{n=1}^{N} F_{X_n|h}(x_{i_n}(n)|h). \end{aligned}$$

Let us consider an evaluation grid of size I^N and refer to the element-wise grid spacing as 'resolution'. Using Lemma C.1. of [115], we have obtained the following sample complexity result for our problem. A detailed proof of the Lemma below can be found in the Appendix of this chapter, section 6.9.1. The result in Lemma 6.1 states that it is possible to learn, within error 6ϵ, any rank$-R$ grid-sampled distribution with resolution ϵ, using a number of samples that is (worst-case) log-linear in rank R and quadratic in the number of dimensions N.

Lemma 6.1. *Let F be any unknown $N-$ dimensional ("true") grid sampled CDF of $\mathbf{X}$ with rank R, alphabet size I (per variable), and resolution parameter ϵ. There is an algorithm that uses* $\mathcal{O}\left(\epsilon^{-2}(RNIlog\frac{1}{\epsilon} + NIRlogR + RN^2Ilog(2I)) \cdot log(\frac{1}{\delta})\right)$ *samples from F, and with probability $1-\delta$ outputs a distribution F' that satisfies*

$$d_{TV}(F, F') \le 6\epsilon.$$

Thanks to the naive Bayes model interpretation, we can go from the joint CDF to the joint PMF/PDF domain efficiently, by taking 1-D differences / derivatives. In the case of continuous variables (PDF), we first use interpolation (linear, bandlimited, or spline – we use linear for simplicity) followed by 1-D differentiation. For PDFs that are (approximately) band-limited with cutoff frequency ω_c, [77] showed that they can be recovered from uniform samples of the associated CDF taken $\frac{\pi}{\omega_c}$ apart. In this way, one is able to easily extract any conditional or marginal distribution at negligible complexity beyond what is needed to estimate the grid-sampled CDF tensor factorization – we simply drop factor matrices corresponding to the variables we are not interested in. We stress that multivariate interpolation/differentiation is not needed to compute any PDF, due to the separable nature of the CPD model.

Thus, given a test sample it is easy to infer a response value or label. Let us assume that the label information is stored in the $N-$th variable. Given possibly incomplete data (vector realizations with missing entries) specified by the set $X_S, S \subseteq V$, with $X_V = \{X_1, X_2, \ldots, X_{N-1}\}$, we can easily compute the posterior probability $P_{X_N|X_S}$ and derive any desired classifier or estimator and estimate its uncertainty (e.g., conditional variance, conditional tail probability) at the same time. Such uncertainty quantification is very useful in many applications.

Preprocessing step: We explore two different variants. The first method, LR-CDF, directly fits a low-rank tensor model of the grid-sampled CDF tensor, given raw data samples $\mathbf{x}_m \in \mathbb{R}^N, m \in [M]$, without preprocessing. The second approach, LR-Copula, introduces a low-rank parametrizaton of the grid-sampled version of the Copula instead. A Copula [116, 117] is a special family of CDFs with uniform marginals and it is invariant under nonlinear monotonic transformations of the individual variables. For LR-Copula, before training the models, we map each sample vector $\mathbf{x}_m \in \mathbb{R}^N$ into a vector $\mathbf{x}'_m \in [0, 1]^N$ by transforming each component separately, passing it through its (estimated) marginal CDF $x'_n = \hat{F}_{X_n}(x_n)$. By this so-called probability integral transform, we obtain a pseudo-observation vector $\mathbf{x}'$ whose multivariate CDF is $\mathbf{C}_{\mathbf{X}'}$. It is easy to see that each transformed random variable is uniformly distributed in $[0, 1]$. As we will see in our experimental results, transforming the original data (by applying non-linear marginal transformations) often yields improved low-rank modeling.

6.4 Algorithmic approach

We now turn our attention to formulating the constrained optimization problem and the algorithmic approach for low-rank fitting of the grid-sampled CDF tensor. Given a dataset $\mathcal{D}$, one can estimate and instantiate (a discretized version of) the empirical joint CDF tensor $\widehat{\mathcal{F}}$ using (6.2). We consider a rank-R approximation $\mathcal{F}$, computed using squared loss as the fitting criterion and solve the following optimization problem

$$\begin{aligned}
&\min_{\boldsymbol{\lambda},\mathbf{A}_1,\ldots,\mathbf{A}_N} \left\|\widehat{\mathcal{F}} - [\![\boldsymbol{\lambda}, \mathbf{A}_1, \ldots, \mathbf{A}_N]\!]\right\|_F^2 \\
&\text{subject to} \quad \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{1}^T\boldsymbol{\lambda} = 1, \\
&\qquad\qquad\quad \mathbf{A}_n(0,h) \geq 0, \mathbf{A}_n(I_n,h) = 1, \\
&\qquad\qquad\quad \mathbf{A}_n(i_n,h) \leq \mathbf{A}_n(i_n+1,h), \\
&\qquad\qquad\quad n \in [N], \forall h \in [R].
\end{aligned} \tag{6.4}$$

Although the problem described in (6.4) is very challenging, accurate decompositions can be efficiently computed in many practical settings. For smaller N, a straightforward way to estimate the latent factors is to directly instantiate tensor $\widehat{\mathcal{F}}$ and decompose it by employing alternating optimization (AO). Each model parameter is cyclically updated while the remaining ones are fixed at their last updated values. Then, the optimization problem with respect to each $\mathbf{A}_n$ reduces to a least-squares problem under "valid CDF" constraints on its columns. To solve these two sub-problems, we propose the Alternating Direction Method of Multipliers (ADMM) algorithm [118] because it amortizes certain expensive operations and uses a warm start to handle the constraints more easily. We refer to the overall approach as AO-ADMM.

However, when the problem is high-dimensional, computing and directly forming the full grid-sampled empirical CDF tensor could be very expensive, even intractable computationally and memory-wise. Fortunately, we can still optimize the proposed model parameters without explicitly forming the empirical CDF tensor, by minimizing

$$\left(\frac{1}{M}\sum_{m=1}^{M} \mathbf{1}\left(\mathbf{x}_m \leq \mathbf{x}_{\boldsymbol{i}}\right) - \left(\circledast_{n=1}^{N}(\mathbf{A}_n(i_n,:))\right)\boldsymbol{\lambda}\right)^2. \tag{6.5}$$

Each parameter can be estimated on-the-fly by applying stochastic approximation – i.e., sampling parts of the data at random and using the sampled piece to update the latent factors. Using this observation, we randomly sample a subset of data realizations from the training set and update the pertinent parts of the latent factors of the CPD model from rank-1 measurements of the "hidden" CDF tensor, using the sampled entries of the tensor. In the case of continuous variables, the conditional PDFs can be estimated from their corresponding CDF samples stored in $\mathbf{A}_n$ by interpolation and $1-D$ differentiation. For terminating the algorithm we compute the

MSE on a validation set and stop if the number of maximum iterations has been reached or the MSE has not improved in the last T iterations. Denoting the parameter set $\boldsymbol{\lambda}$, $\{\mathbf{A}_n\}_{n=1}^{D}$ as $\boldsymbol{\theta}$, the full procedure is shown in Algorithm 6.1. This algorithm uses the Adam optimizer with learning rate set to 0.01.

Algorithm 6.1 CDF-CPD (Projected Adam)

```
Input: Raw data or transformed data $\{\mathbf{x}_m\}_{m=1}^{M}$ in $\{\mathbf{X}, \mathbf{X}_{\text{val}}\}$, $R$, $I$, initial learning rate $\alpha$, $M_{\text{batch}}$
Output: Model parameters $\boldsymbol{\theta}^*$
Initialize model parameters $\boldsymbol{\theta}_0$
repeat
    Sample $M_{\text{batch}}$ data points
    Compute empirical estimates via Equation (6.2)
    Jointly update model parameters $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}$
    Project each $h-$th column of $\mathbf{A}_n$ using $\mathcal{P}_{\mathcal{C}}\left(\mathbf{A}_n(,h)\right)$
    Project $\boldsymbol{\lambda}$ onto the probability simplex
    Compute $\text{MSE}_{\text{val}}$ using $\mathbf{X}_{\text{val}}$
until $\text{max}_{\text{iter}}$ is reached or $\text{MSE}_{\text{val}}$ stops diminishing
```

Methods	Power	Gas	Hepmass	Miniboone
RealNVP [16]	0.17 ± 0.01	8.33 ± 0.14	-18.71 ± 0.02	-13.55 ± 0.49
Glow [91]	0.17 ± 0.01	8.15 ± 0.40	-18.92 ± 0.08	-11.35 ± 0.07
MADE MoG [66]	0.40 ± 0.01	8.47 ± 0.02	-15.15 ± 0.02	-12.27 ± 0.47
MAF-affine MoG [68]	0.24 ± 0.01	10.08 ± 0.02	-17.73 ± 0.02	-12.24 ± 0.45
MAF-affine w/o MoG [68]	0.30 ± 0.01	9.59 ± 0.02	$-17.39 \pm .02$	-11.68 ± 0.44
FFJORD [119]	0.46 ± 0.01	8.59 ± 0.12	-14.92 ± 0.08	-10.43 ± 0.04
NAF-DDSF [120]	0.62 ± 0.01	11.96 ± 0.33	-15.09 ± 0.40	-8.86 ± 0.15
LRCF - (HTF) [22]	0.68 ± 0.01	**12.15** ± 0.02	-13.78 ± 0.02	-11.08 ± 0.04
LR-CDF	0.91 ± 0.01	**12.15** ± 0.08	-11.71 ± 0.32	-7.97 ± 0.07
LR-Copula	**0.97** ± 0.01	9.73 ± 0.05	**-9.3** ± 0.18	**-7.94** ± 0.08

Table 6.1: Average log-likelihood comparison in four UCI datasets. Best performances are in bold.

In Algorithm 6.1, $\mathcal{P}_{\mathcal{C}}$ is the projection operator onto a set that defines a "valid CDF". We project each conditional by imposing the following constraints on each factor matrix: extreme point values: $\mathbf{A}_n(I_n, h) = 1$, positivity, and non decreasing $\mathbf{A}_n(:, h)$ for each h. For the latter, isotonic regression [121] finds the best least squares fit according to the following process: When a violation $\mathbf{A}_n(i_n, h) > \mathbf{A}_n(i_n + 1, h)$ is encountered, we replace this pair by their average, and back-average to the previous value as needed, to get monotonicity. We continue this process, until finally we reach $\mathbf{A}_n(I, h)$. Each factor matrix is initialized by random non-negative initialization, followed by sorting.

6.5 Experimental Study

In this section, we showcase the effectiveness of the proposed methods, namely, LR-CDF and LR-Copula (the main difference between the two methods is the use of preprocessing in the LR-Copula case), and the proposed fitting algorithms, i.e., AO-ADMM and projected Adam, against relevant baselines in a variety of machine learning tasks.

Low-Dimensional Toy Experiments: We first apply our LR-CDF (AO-ADMM) model to a range of synthetic datasets to test LR-CDF's ability to learn a known, ground truth distribution. In the 2D case, six different toy datasets have been considered: 8gaussians, circles, moons, pinwheel, swissroll, and checkerboard. We train the proposed model on $M = 5000$ samples using $I_1 = I_2 = 30$, and $R = 10$. For qualitative evaluation, we visualize (see Figure 6.5 in supplementary material Section 6.9.2) the heatmaps of the generated samples. With an appropriate choice of parameters, the proposed model is precise at capturing the modes and structure of all the datasets considered.

In the 3D case, we assume that a three dimensional PDF is a mixture of two Gaussian distributions with means $\mu_1 = [-1.7, 0.45, -2.50]$, $\mu_2 = [1.8, 2.30, -1.20]$ and co-variance matrices $\Sigma_1 = \text{diag}([0.8^2, 1.4^2, 0.7^2])$ and $\Sigma_2 = \text{diag}([1.3^2, 0.8^2, 0.8^2])$. We show (see the Figure in supplementary material Section 6.9.2) that given 5000 samples, the proposed method can reveal (samples of) each marginal CDF – in this case $\boldsymbol{\lambda}$ holds the probability of choosing component. Then, given sufficient number of CDF samples, linear interpolation and differentiation enable high-quality conditional PDF reconstruction. Using only $I = 20$ transition levels and true rank $R = 2$, the recovered PDF essentially coincides with the true PDF.

Density estimation results: In this section, we show results on density estimation for the following multivariate datasets: Power, Gas, Hepmass, and Miniboone dataset. Because Power ($N = 6$) and Gas ($N = 8$) have lower dimensionality, we employ AO-ADMM for fitting. On the other hand, for Hepmass and Miniboone, we use projected Adam. For training our model, we estimate the rank R and the number of cutoffs I, considering $I_1 = I_2 = \cdots = I_N = I$ for simplicity. We pick the pair that best fits the data by reserving 20% of the training set for validation. The set from which the tensor rank is selected is $R = [10, 20, 30, 50, 80, 100]$ and

	Banknote	*Breast*	*Concrete*	*Red*	*White*	*Yeast*
Mean	1.020 ± 0.032	1.000 ± 0.047	1.010 ± 0.035	1.000 ± 0.030	1.000 ± 0.020	1.060 ± 0.052
Copula-EM	0.604 ± 0.040	0.298 ± 0.035	0.529 ± 0.18	0.635 ± 0.072	0.772 ± 0.021	0.998 ± 0.021
MIWAE	0.446 ± 0.038	0.280 ± 0.021	0.501 ± 0.040	0.643 ± 0.026	0.735 ± 0.033	0.962 ± 0.051
MAF	0.662 ± 0.029	0.749 ± 0.084	0.974 ± 0.083	0.892 ± 0.084	0.855 ± 0.020	0.950 ± 0.041
Real-NVP	0.728 ± 0.046	0.831 ± 0.029	0.933 ± 0.075	0.972 ± 0.029	0.917 ± 0.021	0.994 ± 0.053
LR-CDF	0.574 ± 0.035	$\mathbf{0.268 \pm 0.018}$	$\mathbf{0.500 \pm 0.016}$	0.633 ± 0.090	0.743 ± 0.012	0.947 ± 0.064
LR-Copula	$\mathbf{0.439 \pm 0.038}$	0.272 ± 0.018	0.560 ± 0.042	$\mathbf{0.622 \pm 0.021}$	$\mathbf{0.711 \pm 0.032}$	$\mathbf{0.934 \pm 0.021}$

Table 6.2: Mean-squared error for single imputation for various UCI data sets (mean and standard deviations).

	Income	Credit	Heart	Car
Naive Bayes	0.206 ± 0.005	0.140 ± 0.018	0.166 ± 0.026	0.152 ± 0.017
SVM	0.179 ± 0.004	0.146 ± 0.02	0.170 ± 0.053	0.151 ± 0.016
LR-PMF	0.175 ± 0.003	0.129 ± 0.018	0.147 ± 0.023	0.089 ± 0.015
LR-CDF	$\mathbf{0.164 \pm 0.004}$	$\mathbf{0.102 \pm 0.016}$	$\mathbf{0.122 \pm 0.026}$	$\mathbf{0.069 \pm 0.008}$

Table 6.3: Misclassification error on different UCI datasets.

	Exchange Rates
Gaussian	0.927 ± 0.102
Vine	0.281 ± 0.037
Vine-TLL2	0.255 ± 0.049
LR-Copula (ours)	$\mathbf{0.172 \pm 0.108}$

Table 6.4: MSE and standard deviation for Exchange Rates (lower is better).

the set from which we pick the number of levels is $I = [10, 20, 30, 50]$. In Table 6.4, we show average log-likelihoods results over test sets, which comprise test log-likelihoods and error bars of 5 standard deviations.

We compare the results of the two models introduced in this work (LR-CDF and LR-Copula) with state-of-the-art methods for density estimation. The baselines considered here are RealNVP [16], Glow [91], MADE MoG [66], MAF-affine with and without MoG [68], FFJORD [119], NAF-DDSF [120], and LRCF -(HTF) [22]. Both of our methods significantly exceed the performance of state of the art models, suggesting an improved true density function approximation, for the Power, Hepmass, and Miniboone datasets. We stress that even though our method aims for CDF estimation, it performs better in PDF (density) estimation as well, compared to state of art density estimation methods. This is quite surprising on first sight, and it is in part due to the use of far-away samples to estimate the density in sample-starved areas, which is built-in the empirical CDF. The same holds for the characteristic function approach in [22], but still the proposed CDF approaches are better – which speaks for the appropriateness of the proposed formulation.

Imputation results: Next, we focus on the *imputation problem*: given a trained joint distribution model, and an incomplete realization $\mathbf{x} = [\mathbf{x}_{\text{OBS}}, \mathbf{x}_{\text{MISS}}]^T$, our next task is to infer the missing entries directly. In our experiments, all six datasets (we only consider continuous features) from the UCI database considered, are corrupted by removing 30% of the features uniformly at random. Our models are fitted with AO-ADMM for Banknote, Concrete, and Yeast and projected-Adam for the rest. The experiment compares the MSE of the proposed methods with other well established baselines. We chose the mean imputation, which is the simplest baseline, the Copula-EM approach [122]; the recently proposed MIWAE approach [123]; MAF, and Real-NVP. The results, averaged over 5 randomly corrupted repetitions, are presented in Table 6.1. The results suggest that both approaches provide more accurate imputations than all competitors, with CPD-Copula outperforming the rest in most cases.

PMF or CDF modeling? We also present an important set of experiments which targets to answer the following question: Which CPD model (LR-PMF [18] or LR-CDF) offers better performance on modeling mixed data? We test our LR-CDF (Adam) model using four different datasets (of mixed variables) for classification against the CPD-based joint PMF model [18] and two other standard baselines, namely, Naive Bayes classifier and linear SVMs. The results on Table 6.3 suggests that modeling the joint CDF instead of the joint PMF, as proposed in [18], may achieve a better performance in misclassification error. This further supports the merits of the proposed method, which can be useful even for discrete variables, for which we naturally turn to the PMF.

A Copula comparison: Modeling with Copulas is a popular tool in financial risk management as they are invariant to nonlinear feature scaling and useful for modeling extreme tail distributions. In the following experiment, we consider a data set of size $N = 5844$ that contains 15 years of exchange rates of Yen, Euro, Canadian dollars and the British Pound to US dollars. We estimate a parametric Gaussian copula, a Vine copula [124], a Vine copula with only TLL2 pair-copulas [**?**], and our CPD LR-Copula model (via AO-ADMM) for this dataset, and then perform data imputation, as described earlier. The proposed LR-Copula model yields significantly smaller MSE relative to the baselines, which we attribute to the unassuming non-parametric nature and strong identifiability properties of our LR-Copula model.

6.6 Application on Enrollment Rate Prediction

Clinical trials are of vital importance to global healthcare. They are a necessary step in drug development to ensure the safety and efficacy of new therapies and vaccines. One of the main reasons clinical trials are terminated in the first place is insufficient patient enrollment. Accurate enrollment rate prediction, given relevant information (trial predictors) such as phase, country, eligibility criteria, patient segment, and indication, is a key determinant of optimized clinical trial planning as it serves to minimize the overall cost and possible delays for an effective treatment from being approved. Minimizing the overall cost incurred by trials with low enrollment rates can be achieved by avoiding initiating trials that are most unlikely to meet pre-specified recruitment targets. Thus, the ability to predict the probability of proposed trials meeting enrollment goals prior to initiating the trial is highly beneficial for pharmaceutical, biotech, and medical device companies. Additionally, maximizing the expected enrollment rate of a given trial may be accomplished by careful location selection (i.e., by primarily focusing on countries likely to meet enrollment targets). Both of these aspects can be addressed if an efficient data-generating distribution of clinical trials is available.

The need for probabilistic analysis in predicting the enrollment rate of clinical trials using limited number of samples motivates the proposed method. We explore a data-driven, machine learning approach for effectively modeling the multivariate distribution of trial predictors/features (trial phase, location, primary indication, therapeutic area) and enrollment rate based on grid-sampled CDFs. This work builds upon the framework and the methods developed in [1] to solve an important practical problem of high societal impact, using unique data and know-how of IQVIA R&D, which has a long track record in optimizing clinical trials. We leverage the results presented in [1], which show that one can always sketch a multivariate CDF in terms of a multidimensional empirical cumulative probability array, i.e., a finite grid-sampled CDF tensor, and *every* multivariate grid-sampled CDF can be thought to be generated according to a latent variable Naive Bayes model with a bounded number of hidden states through CPD corresponding to the tensor rank.

To jointly model both discrete (e.g., phase, country) and continuous variables (e.g., enrollment-rate), we will sketch the trial-related multivariate distribution in terms of a grid-sampled CDF tensor $\widehat{\mathcal{F}}$, where each mode represents the (finite) levels/cut-offs of the CDF for every individual trial feature (e.g., phase I, II, III, IV for the trial phase feature and USA, Germany, etc. for the location feature) and each element of that tensor can be easily estimated via sample averaging. By introducing the reconstructed approximation of $\widehat{\mathcal{F}}$, using the rank-R parameterization $\mathcal{F}$, one can get a “universal” approach, which is unassuming of the structure of the data, striking an excellent trade-off between model generality and identifiability under mild conditions by virtue of the uniqueness of CPD. The resulting model also yields a probabilistic method which allows easy likelihood estimation, computation of conditional distributions, and closed-form inference. The above enable us to reliably predict the expected enrollment rate given trial-related information, provide the probability of a new trial achieving a target enrollment rate, as well as country or site recommendation.

We evaluated the proposed approach against state-of-the-art machine learning methods by leveraging a large-scale real-world clinical trial dataset from IQVIA. We focus on four trial features to instantiate and decompose the multivariate CDF tensor, namely, trial location, therapeutic area or primary indication, trial phase, and final enrollment rate. The proposed method demonstrates improved performance in the enrollment rate prediction task over the best baselines by up to 12.2% in MSE, while also allowing direct uncertainty quantification (e.g., through the conditional variance which can be easily calculated from the derived conditional distribution). Additionally, we demonstrate how the proposed approach can be utilized to benefit country recommendation, maximizing the likelihood of a given trial towards reaching an enrollment rate target, at negligible complexity beyond what is needed to estimate the grid-sampled CDF tensor.

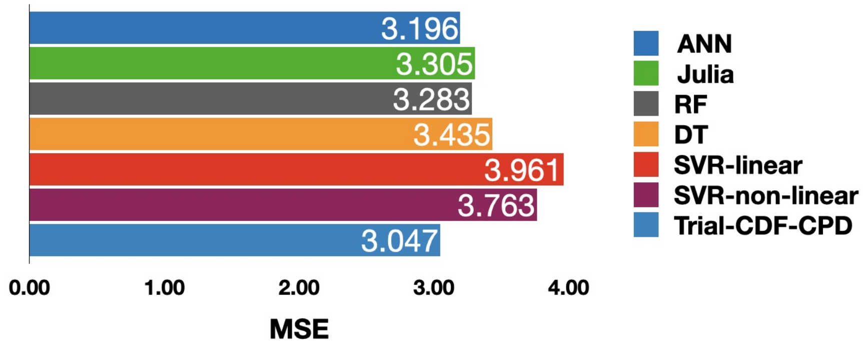


Figure 6.2: Model evaluation for the first configuration: country, phase, primary therapeutic area, enrollment rate.

6.7 Enrollment Rate Prediction from IQVIA's Real-World Clinical Trial Database

We used information available prior to the initiation of the trials as features to obtain test-set enrollment rate estimates. We utilized country-level trial data from IQVIA's real-world clinical trial database to conduct the evaluation. The dataset contains the country-level enrollment rates which were collected from 5470 clinical trials in 54 countries between 03-31-2000 and 09-12-2018. The trials are across Phase I to Phase IV and with 15 indications. After basic prepossessing, the resulting dataset contained 5337 clinical trials across 30 countries, 4 phases, and 15 indications. Overall, the available features include geographic location of the trial, phase, primary therapeutic area, primary indication and the outcome trial enrollment rate for each study at a country-level.

We split the dataset such that 70% of the data samples are used for training and 30% for testing, and we ran 10 Monte-Carlo simulations. We consider two different tensor configurations, where the first one is based on country, phase, primary therapeutic area, and enrollment rate, while the second one is based on country, phase, primary indication, and enrollment rate, which is more granular than the first one. In the results, we report the parameter combination, i.e., rank R and number of thresholds/ cut-offs for the enrollment rate I, that achieves the best possible results, namely $R = 20$ and $I = 25$.

We assess the performance of the proposed Trial-CDF-CPD model in enrollment rate regression tasks under Mean Squared Error (MSE), against various supervised learning baselines,

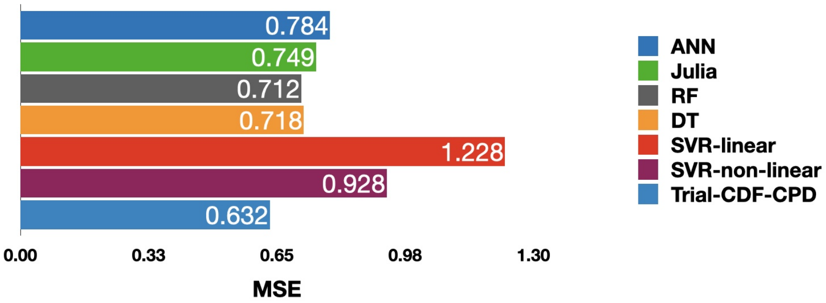


Figure 6.3: Model evaluation for the second configuration: country, phase, primary indication, enrollment rate.

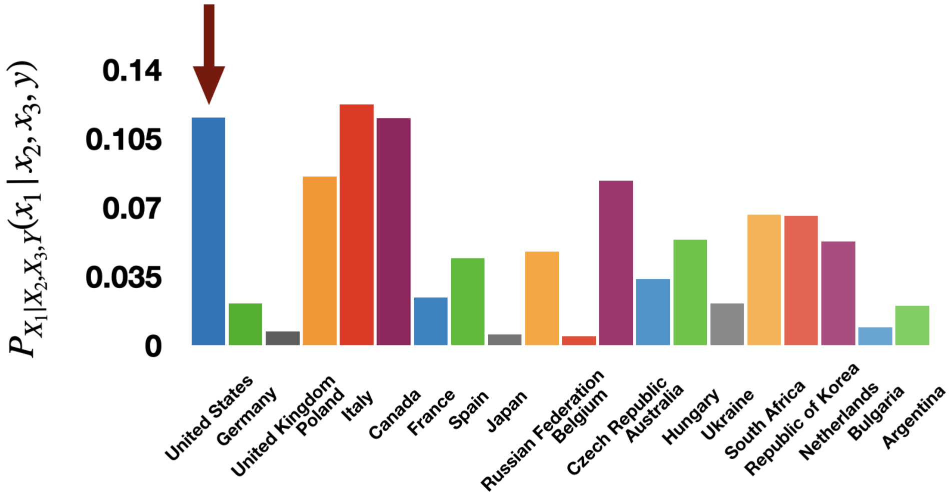


Figure 6.4: Country recommendation: Given a prespecified phase and primary indication, we use the trained model to output the conditional probability for each candidate country achieving an expected enrollment rate. In this example, Trial-CDF-CPD assigns a high likelihood to United States (ground truth) achieving the expected enrollment rate. However, our method also indicates other countries (Poland, Italy, Canada, Czech Republic) that are within top-5 countries with highest likelihood values. To encourage diversity, clinical trials require engaging a wide range of countries instead of frequently selected countries (USA, Germany etc).

that are representative of the state-of-art. In our experiments, we studied the following machine learning regression techniques: Decision Tree (DT), both linear and non-linear Support Vector Regression (SVR), Artificial Neural Networks (ANN), JULIA [125], and Random Forest (RF). We explore the space of parameters and report the results stemming from their best variation for each machine learning algorithm.

We obtained enrollment rate predictions under both conditional expectation and conditional mode (MAP), with the latter achieving the best empirical performance. Figures 6.2 and 6.3 report the enrollment rate prediction performance (MAP) for configurations (1) and (2) correspondingly. For the first set of experiments, we observe a 4.662% reduction in MSE using Trial-CDF-CPD against the next best baseline and a 12.219% reduction for the second configuration. At the same time, out of the baselines, only our method can output the confidence of its decision, which is important in the present healthcare context. Furthermore, our model is not restricted to this specific application but can also be used to answer the following question: Which countries can achieve an expected predefined enrollment rate? An example can be visualized in Figure 6.4, where Trial-CDF-CPD indicates top-5 locations (United States, Poland, Italy, Canada, Czech Republic), which includes the one selected in real world (United States), that have a high likelihood of achieving an expected enrollment rate. This knowledge could be useful during trial planning and cite selection to minimize the risk of not achieving a desired enrollment rate for a given trial.

6.8 Chapter Summary

In this paper, we introduced a new class of models for general purpose distribution estimation. Our work has shown that *every* multivariate CDF admits a compact grid-sampled approximation in terms of a latent variable naive Bayes model with a bounded number of hidden states through CPD corresponding to the tensor rank. We have considered the statistical and computational efficiency of the proposed estimators and highlighted desired properties such as easy marginalizability over subsets of variables, fast sampling, and identifiability under mild assumptions. Furthermore, our experimental results indicate that estimating a grid-sampled CDF yields better data modeling than direct PDF estimation. We have also presented results that clearly support the broad applicability of low-rank grid-sampled CDF estimators in various tasks of interest in machine learning and statistics.

6.9 Supplementary Material

In this section, we provide a proof for sample complexity analysis of grid-sampled CDF-CPD estimators and additional experimental material to better evaluate the performance of the proposed method.

6.9.1 Sample Complexity

Using Lemma C.1. of [115], we have the following sample complexity result for our problem.

Lemma 1: *Let F be any unknown $N-$ dimensional ("true") grid sampled CDF of $\mathbf{X}$ with rank R, alphabet size I (per variable), and resolution parameter ϵ'. There is an algorithm that uses* $\mathcal{O}\left(\epsilon'^{-2}(RNIlog\frac{1}{\epsilon'} + NIRlogR + RN^2Ilog(2I)) \cdot log(\frac{1}{\delta})\right)$ *samples from F, and with probability $1-\delta$ outputs a distribution F' that satisfies $d_{TV}(F, F') \leq 6\epsilon'$.*

Proof: Let us start by considering the 2D rank-1 case.

We consider a dictionary of rank-1 CDFs generated by latent factor vectors whose elements are drawn from a finite quantization. An element of the grid-sampled CDF matrix $\hat{\mathbf{F}}(i, j)$ can be expressed as

$$\hat{\mathbf{F}}(i, j) = (a_i + \tilde{\epsilon}_i)(b_j + \tilde{\epsilon}_j)$$

where a_i and b_j are the exact latent factors of $\mathbf{F}(i, j) = a_i b_j$. If the quantization error for both dimensions is no more than ϵ, i.e., $|\tilde{\epsilon}_i| \leq \epsilon$, $|\tilde{\epsilon}_j| \leq \epsilon$, and using that the factors are 1-D CDFs (follows from marginalization) and thus $a_i \leq 1, b_j \leq 1$, there exists a rank-1 CDF in the dictionary such that

$$|\mathbf{F}(i, j) - \hat{\mathbf{F}}(i, j)| \leq 2\epsilon + \epsilon^2,$$

and thus, the overall error becomes

$$d_{TV}(\mathbf{F}, \hat{\mathbf{F}}) \leq IJ(2\epsilon + \epsilon^2)$$

Assuming that $I_1 = I_2 = \cdots = I_N = I$ for simplicity, the bound in the N-dimensional rank-1 case is

$$d_{TV}(\mathbf{F}, \hat{\mathbf{F}}_{\text{rank-1}}) \leq (2I)^N \epsilon.$$

This is because there are I^N grid-sampled CDF elements overall, and the expression for the per-element error involves a total of $2^N - 1$ terms (product of N pairwise sums, with one term corresponding to the exact value of the CDF), each of which is bounded above by $\epsilon < 1$. In the N-dimensional case of rank-R, the overall error is bounded by

$$d_{TV}(\mathbf{F}, \hat{\mathbf{F}}_{\text{rank-}R}) \leq R(2I)^N \epsilon.$$

We will denote this bound as ϵ', $\epsilon' = R(2I)^N \epsilon$.

By choosing b bits to quantize each element of a conditional CDF, the resolution that we get is $\epsilon = \frac{1}{2^b}$. The number of distributions in our quantized dictionary is then $M = 2^{RNIb}$.
By focusing on M,

$$M = 2^{RNIb} = 2^{-RNIlog\epsilon}$$

and replacing

$$\epsilon = \frac{\epsilon'}{R(2I)^N},$$

we get that

$$\begin{aligned} M &= 2^{RNIb} \\ &= 2^{-RNI(log\epsilon' - log(R(2I)^N))} \\ &= 2^{RNIlog\frac{1}{\epsilon'} + RNI(logR + log(2I)^N)} \\ &= 2^{RNIlog\frac{1}{\epsilon'} + NIRlogR + RN^2Ilog(2I)}. \end{aligned}$$

By plugging in M in Lemma C.1 considered in [115] and replacing the worst case number of samples $\mathcal{O}(\epsilon'^{-2} logM \cdot log(\frac{1}{\delta}))$, we get the final result,

$$\mathcal{O}\left(\epsilon'^{-2}(RNI\mathrm{log}\frac{1}{\epsilon'} + NIR\mathrm{log}R + RN^2I\mathrm{log}(2I)) \cdot \mathrm{log}(\frac{1}{\delta})\right).$$

6.9.2 2D, 3D-Synthetic Experiments

In this section, we employ synthetic experiments to showcase the effectiveness of the proposed algorithm. We first showcase 2D sampling results from six different toy datasets: 8gaussians, circles, moons, pinwheel, swissroll, and checkerboard (see Figure 6.5). Please refer to the discussion of implementation details in the main paper.

Next, we assume that a $3-$dimensional PDF of a random vector $\boldsymbol{X} := [X_1, X_2, X_3]^T$ is a

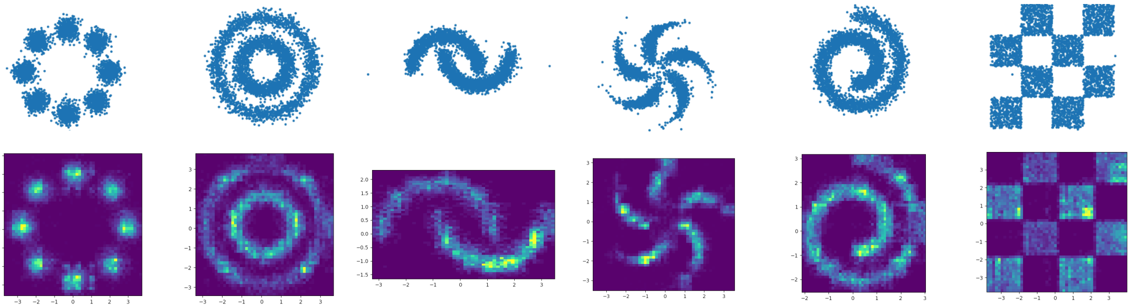


Figure 6.5: Visualization of training samples (upper row) and synthetic samples generated with a learned CPD-CDF (lower row).

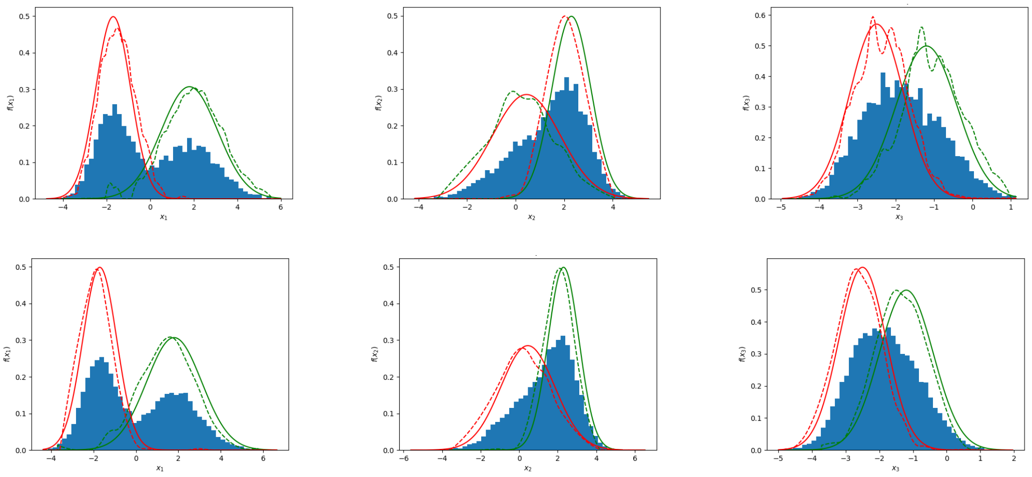


Figure 6.6: Illustration of the key idea on a 3-D two-component Gaussian mixture. Each row shows the estimated component PDFs of X_1, X_2, X_3 in dashed line, and the true component PDFs in solid line. We also visualize the compound histogram per dimension, given M training samples. The first row corresponds to $M = 2000$, $R = 2$, $I = 20$ and the second to $M = 4000$, $R = 2$, $I = 20$.

mixture of two Gaussian distributions with means $\mu_1 = [-1.7, 0.45, -2.50]$, $\mu_2 = [1.8, 2.30, -1.20]$ and co-variance matrices

$$\Sigma_1 = \begin{bmatrix} 0.8^2 & 0 & 0 \\ 0 & 1.4^2 & 0 \\ 0 & 0 & 0.7^2 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 1.3^2 & 0 & 0 \\ 0 & 0.8^2 & 0 \\ 0 & 0 & 0.8^2 \end{bmatrix}.$$

We show (see Figure 6.6) that given only 2000 (top row) or 4000 (bottom row) samples, the proposed method can effectively estimate samples of each marginal CDF, which after interpolation and differentiation yield high-quality conditional PDF reconstruction. For these plots, $I = 20$, $R = 2$, and cubic interpolation of the marginal CDFs was used.

Chapter 7

Modeling time series using latent history summaries and low-rank tensor densities

7.1 Introduction

The ubiquity of multivariate time series data in real-world systems, has created unprecedented opportunities for machine learning tools to capture system patterns across time, provide informative insights about the evolution of complex systems, and predict abnormal events at an early stage. Accurate machine learning models for multivariate time series analysis have been applied for financial risk management [126], demand forecasting [127], for monitoring an individual's health trajectory [128], or identifying unexpected system events, such as sensor faults [129].

For real-world applications, time-series often exhibit strong nonlinear inter-dependencies (e.g., statistical dependencies between opening price, closing price, and daily highs for stock price prediction [130]; cross-series correlations/effects between sensor signals from different brain regions collected from EEG apparatus; occurrence of anomalies in a sensor system can often be detected by measurements of multiple nearby sensors). Many approaches in the literature, [131], [132], [133] avoid multivariate modeling overall, fit a separate model per series, or assume linear relationships among variables. Although multivariate time-series analysis poses a challenge due to modeling and computational difficulties in high-dimensional settings, leveraging information provided from complex interactions is crucial for accuracy. At the same time, accounting for uncertainties in time series predictions is of vital importance for risk-averse decision-making, especially for critical and high-stakes application domains – such as medicine and finance. An accurate joint probabilistic time-series model addresses this need as it allows assessing confidence in the predictions for downstream tasks. Thus, in modeling multivariate (continuous) sequential data $\mathbf{x}_1, \ldots, \mathbf{x}_t$, we ideally aim to accurately capture the joint density $f(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t)$, from which we can derive the conditional density of the transitions $f(\mathbf{x}_t | \mathbf{x}_1, \ldots, \mathbf{x}_{t-1})$. Most previous work aims to estimate $f(\mathbf{x}_t | \mathbf{x}_1, \ldots, \mathbf{x}_{t-1})$ directly and makes restrictive assumptions such as tractable density classes [134] or structural assumptions [135], that may affect the flexibility and expressivity of these approaches.

The goal of this paper is to accurately model the joint multivariate density of $\mathbf{x}_t$ and a latent history summary of $\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}$. Building upon previous work [135] on deep learning for modeling time series data, we propose an end-to-end approach for multivariate time series modeling consisting of a (multi-output) recurrent neural network (RNN) to capture necessary information from historical data $\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}$, and an expressive joint density model that learns the joint distribution of the updated hidden state $\mathbf{h}_{t-1}$ and $\mathbf{x}_t$, i.e., the joint density of $\left[\mathbf{h}_{t-1}^T, \mathbf{x}_t^T\right]^T$. We capitalize on the "universality" of tensor-based density models to accurately learn the distribution of complex high-dimensional data and the ability of RNNs and their variants (e.g., LSTM [136], GRU [137], etc.) for effectively modeling sequential dynamics. We follow a recent line of work (see [21], [22]), which proposed a "universal" approach for smooth, compactly supported multivariate densities by representing the underlying density in terms of a low-rank tensor of leading Fourier coefficients. By virtue of uniqueness of low-rank tensor decomposition, this model enables learning the true data-generating distribution, under certain conditions [23]. Additionally, it allows scaling to hundreds of dimensions without sacrificing model expressivity, enables efficient and low-complexity inference from data with missing entries, and likelihood evaluation.

The RNN learns relations through time and thanks to its memory state, it outputs a parsimonious latent representation of previous data. The low-rank distribution model learns the series progressions from the latent representations and current observations, while remaining computationally tractable and without compromising expressivity. The key to success is that these two components are trained jointly in an end-to-end fashion using backpropagation and maximum likelihood as the optimization criterion. In this way, model parameters are learned simultaneously to capture only the necessary information from the past that will enable accurate learning and future state prediction.

7.1.1 Key Contributions

Our key contributions can be summarized as follows:

- *Expressivity benefits:* We introduce a parametrization of the joint density $f(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t)$ without imposing idealized assumptions on the underlying data generating process, which could hinder accuracy (linear inter-dependencies, independent components etc). Instead, the proposed model allows for multivariate time-series analysis while respecting the cross-series relationships between variables.

- The overall model allows for uncertainty quantification (e.g., via the conditional variance), which is crucial for monitoring unexpected fluctuations and supports better-informed decision making.

- *Applicability for both time-series forecasting and imputation:* We demonstrate that our model achieves superior forecasting performance of complex dynamics and inter-dependencies while scaling to high dimensions. Thanks to the ease of marginalization of the resulting probabilistic model, the proposed approach also allows direct imputation of incomplete time-series. We are able to deal with missing values by evaluating, for every missing data point, its conditional expectation given the non-missing values.

- *Efficient training:* We introduce an end-to-end model based on optimizing an LSTM-RNN network in combination with the tensor parameters. Both components encourage parsimony in the resulting model, leading to a significant reduction in generative model parameters, which enables fast and efficient training.

The paper first covers related work in Section 7.2 and provides background context in Section 7.3. Section 7.4 describes the proposed model in detail, including the proposed joint end-to-end training of its two constituent components. Experiments are detailed in Section 7.5. We conclude with discussion in Section 7.6.

7.2 Related Work

The literature on probabilistic time-series modeling has a long history. Fundamental time-series methods, such as vector auto-regressive (VAR) models [138], a multivariate extension of auto-regressive (AR) models [139], assume linear relationships and become over-parameterized as the number of component series increases. Classic time-series models, such as auto-regressive moving average (ARMA) and auto-regressive integrated moving average model (ARIMA) [140] and exponential smoothing (ES) [141] are often employed in univariate mode to separately learn one model per scalar time-series, for simplicity and to avoid over-parameterization. There is rich literature on multivariate state-space models (SSMs) exemplified by the Kalman filter [142–144].

An important limitation of univariate and linear multivariate approaches is that they fail when there is strong and nonlinear coupling, respectively. For multivariate probabilistic time-series analysis, ideally, a full joint distribution at each time step has to be modeled. Many of the methods that estimate a joint global model can only handle a limited number of time series [145], [146] due to the number of parameters that need to be estimated, which can be restrictive for many modern application domains.

Multivariate approaches such as Gaussian Processes (GPs) [147], have been shown to require prior knowledge in order to attain competitive performance against classical forecasting methods such as ARIMA and ES [148]. Furthermore, the kernels employed in GPs can have a strong impact on model performance. Recently, SSMs were revisited through the lens of deep learning, with DeepAR [149] being among the earliest methods in this space. DeepAR is a multi-task univariate forecasting method based on LSTMs, which outputs a probabilistic forecast via the parameters of a parametric distribution (of the next time step) (e.g., μ and σ of a Gaussian distribution). Following this line of work, current methods typically aim to estimate the multivariate conditional density $f(\mathbf{x}_t|\mathbf{x}_1, \ldots, \mathbf{x}_{t-1})$ by introducing a combination of RNNs and tractable distribution classes. Specifically, they employ recurrent neural networks to compress the history $\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}$ into a state vector $\mathbf{h}_{t-1}$ with probabilistic forecasts parameterized by pre-selected tractable distribution classes such as Gaussian Copulas [135], Normalizing Flows [150], or Diffusion models [151]. For example, multivariate methods such as Time-GAN [152] utilize GANs to generate time series. Another related work combines variational methods with sequence models such as variational RNN (VRNN) [134]. A VRNN applies a VAE to each hidden state of an RNN over the input series. VRNN, however, does not provide a probabilistic model in the input space, so uncertainties in forecasts can not be directly estimated. The same holds for the Temporal Latent Autoencoder (TLAE), where the temporal model is applied across a low dimensional space [153]. In [135], the authors used a Gaussian copula to convert non-Gaussian observations to follow a standard Gaussian distribution, a multitask univariate LSTM [136] to model the latent states, and the conditional distribution is given by a low-rank plus diagonal covariance Gaussian copula. Apart from the sensitivity to the choice of rank, imposing the assumptions above can limit the distributional expressiveness of the resulting model.

In light of this limitation, more flexible models such as temporal conditioned Normalizing Flows [150] have been proposed. This method uses a multivariate RNN to output a latent representation $\mathbf{h}_{t-1}$ of process history up to time $t-1$, and Normalizing Flows [16] to model the conditional distribution $f(\mathbf{x}_t|\mathbf{h}_{t-1})$. Although Normalizing Flows (NFs) can potentially represent any complex density though a flow of successive (invertible) transformations, in practice, strong network constraints of NFs can be restrictive for model expressiveness.

In this paper, we advocate using Low-Rank Characteristic Function (tensor) based Density Estimators (LRCF-DE) [21] as an appealing alternative. The motivation for doing so comes from [21], [22], which have demonstrated that non- parametric low-rank tensor modeling in the Fourier domain achieves superior performance in multivariate density estimation, while also allowing easy marginalization, easy likelihood evaluation, and model identifiability. Some background on tensors, low-rank tensor models, and their use in density estimation is necessary at this point, before we proceed.

7.3 Background

7.3.1 Notation

We use $\mathbf{x}$, $\mathbf{X}$, $\underline{\mathbf{X}}$ for vectors, matrices and tensors respectively. We use $\mathbf{x}(k)$, $\mathbf{X}(:,k)$, $\underline{\mathbf{X}}(:,:,k)$ to denote a particular element of a vector, a column of a matrix, and a slab of a tensor, respectively. We denote the multivariate observation vector at time t as $\mathbf{x}_t \in \mathbb{R}^N$, and its $n-$th element at time t as $\mathbf{x}_t(n)$. We use $\bar{\mathbf{x}}$ to denote the expected value of the ground truth signal $\mathbf{x}$, and $\hat{\mathbf{x}}$ to denote an estimate of $\mathbf{x}$. Symbols $\circ$, $\circledast$, $\odot$ denote the outer, Hadamard, and Khatri-Rao product, respectively. The set of integers $\{1,\ldots,N\}$ is denoted as $[N]$.

We employ six different metrics to evaluate the performance of our model, i.e., mean absolute error (MAE) at moment t, $MAE = \frac{1}{N}\sum_{n=1}^{N}|\mathbf{x}_t(n) - \hat{\mathbf{x}}_t(n)|$, mean absolute percentage error

(MAPE), $MAPE = \frac{1}{N}\sum_{n=1}^{N}\frac{|\mathbf{x}_t(n)-\hat{\mathbf{x}}_t(n)|}{|\mathbf{x}_t(n)|}$, mean square error (MSE) and root mean square error (RMSE), $RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_t(n) - \hat{\mathbf{x}_t}(n))^2}$, the Root Relative Squared Error (RSE), $RSE = \frac{\sqrt{\sum_{n=1}^{N}(\mathbf{x}_t(n)-\hat{\mathbf{x}}_t(n))^2}}{\sqrt{\sum_{n=1}^{N}(\mathbf{x}_t(n)-\bar{\mathbf{x}}))^2}}$, and Maximum Error (Max Error).

7.4 Proposed Model: Joint probabilistic modeling of latent history and next state

7.4.1 Technical Approach

Prominent models for probabilistic analysis of high-dimensional time-series estimate the the conditional one-step-ahead distribution given the current historical summary, using a combination of RNN-based models with parametric densities (e.g., Gaussian Copulas [135]), which are often further constrained (e.g., diagonal or low-rank covariance matrices) or are sensitive to hyper-parameters (e.g., rank). Ideally, a joint distribution of $[\mathbf{h}_{t-1}^T, \mathbf{x}_t^T]^T$ has to be modeled, with a controllable number of parameters. The goal of the proposed approach is to introduce a scalable probabilistic model of N dependent time-series $\mathbf{x}_t$ with a parameterization that provides the flexibility to fit an arbitrary data distribution and represent higher-order statistical dependencies.

We build upon the ideas of [150], [135] and [21], to introduce a tensor-based method for joint densities with tractable likelihood in combination with an LSTM-RNN architecture to model the multivariate temporal dynamics of the time series. At a high level, the proposed model uses an LSTM network to capture the transition of latent states and a non-parametric density estimate based on tensor decompositions to jointly model the latent state and the next snapshot of time series observations. We will show that, in addition to enhanced modeling accuracy, the proposed approach yields a more general method that comes with a number of key advantages (e.g., a marginalizable probabilistic model), making it very competitive with respect to the state-of-the-art.

We consider contiguous sequences of size T, sampled from the complete time series history of the training data. Each sequence consists of past observations $\mathbf{x}_1, \ldots, \mathbf{x}_{t_0-1}$ specified by a context window, and future observations $\mathbf{x}_{t_0}, \ldots, \mathbf{x}_T$. We begin by focusing on the problem of learning the conditional density of temporal transitions $f(\mathbf{x}_{t_0}, \ldots, \mathbf{x}_T|\mathbf{x}_1, \ldots, \mathbf{x}_{t_0-1})$. Under the chain rule decomposition, the conditional distribution of future values given the observed sequence, can be equivalently written as

$$f(\mathbf{x}_{t_0} \dots \mathbf{x}_T | \mathbf{x}_1, \dots, \mathbf{x}_{t_0-1}) = \prod_{t=t_0}^{T} f(\mathbf{x}_t | \mathbf{x}_{1:t-1}). \tag{7.1}$$

The first assumption of this paper is that a latent state vector $\mathbf{h}_{t-1} \in \mathbb{R}^M$ can summarize the history of the time series up to its previous time step $t-2$ via the hidden state of an LSTM-RNN. Following [150], an LSTM-RNN is assumed to encode the time series sequence up to time point $t-1$ via the updated hidden state $\mathbf{h}_{t-1}$:

$$\mathbf{h}_{t-1} = \mathrm{RNN}_\theta(\mathbf{x}_{t-1}, \mathbf{h}_{t-2}), \tag{7.2}$$

where RNN_θ is a multi-layer recurrent neural network with LSTM cells parameterized by shared weights θ and $\mathbf{h}_0 = \mathbf{0}$.

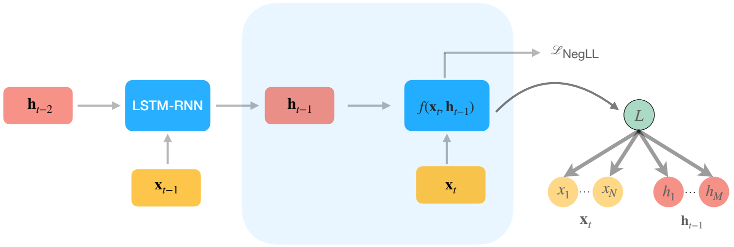


Figure 7.1: The LSTM network learns relations through time and outputs a parsimonious latent representation $\mathbf{h}_{t-1}$ of previous data. The low-rank distribution model learns the series progressions from the latent representations $\mathbf{h}_{t-1}$ and current observation $\mathbf{x}_t$. Our approach yields a generative model of the joint density $f_{\boldsymbol{X},\boldsymbol{H}}$, which is very easy to marginalize. This is because $f_{\boldsymbol{X},\boldsymbol{H}}$ can be interpreted as a mixture of F product distributions i.e., admits a latent variable naive Bayes interpretation.

At each time point, the LSTM-RNN gets input data $\mathbf{x}_{t-1}$ and the previous state $\mathbf{h}_{t-2}$, and outputs a state for the next timestep $\mathbf{h}_{t-1}$, which is utilized along with the current observation $\mathbf{x}_t$ for estimating $f(\mathbf{x}_t|\mathbf{h}_{t-1})$. Provided that this assumption holds, the following equation approximates the problem of probabilistically modeling several steps ahead to rolling prediction: the prediction at time $t-1$ is input to the model to predict the value at time t. Under this model, we can factorize the temporal distribution of the observations as follows:

$$f(\mathbf{x}_{t_0} \ldots \mathbf{x}_T | \mathbf{x}_1, \ldots, \mathbf{x}_{t_0-1}) \approx \prod_{t=t_0}^{T} f(\mathbf{x}_t | \mathbf{h}_{t-1}) = \prod_{t=t_0}^{T} \frac{f(\mathbf{x}_t, \mathbf{h}_{t-1})}{f(\mathbf{h}_{t-1})}. \tag{7.3}$$

Using the mere definition of conditional densities, according to Eq.7.3, an expressive model for the joint density $f_{\boldsymbol{X},\boldsymbol{H}}(\mathbf{x}, \mathbf{h})$ contains all the necessary information for the task. The joint density $f_{\boldsymbol{X},\boldsymbol{H}}(\mathbf{x}, \mathbf{h})$ should ideally allow several properties: (i) efficient evaluation of the likelihood, (ii) efficient evaluation of conditional or marginal distributions during the inference time, (iii) expressivity even if the number of time series components N is large. Assuming that the joint density $f_{\boldsymbol{X},\boldsymbol{H}}(\mathbf{x}, \mathbf{h})$ is time-invariant, given a set of realizations $\{\mathbf{x}_t, \mathbf{h}_t\}$ we can use them to estimate $f_{\boldsymbol{X},\boldsymbol{H}}(\mathbf{x}, \mathbf{h})$. An advantage of modeling the joint instead of the conditional distribution lies in that the joint approach takes into account the statistics of the history vector and learn collaboratively, which could encourage better history representation learning and generalize to unseen history vectors better. The overall model, as illustrated in Fig.7.1, uses the observation of the last time step $\mathbf{x}_{t-1}$ as well as the recurrent state $\mathbf{h}_{t-2}$ to produce the state $\mathbf{h}_{t-1}$ which is concatenated to the current observation $\mathbf{x}_t$ and passed through the probabilistic model. We introduce a parameterization of $f_{\boldsymbol{X},\boldsymbol{H}}(\mathbf{x}, \mathbf{h})$ using a CPD decomposition in the Fourier series domain, i.e., fitting a low-rank CPD model on the $(N + M)$-way Fourier series coefficient tensor $\underline{\boldsymbol{\Phi}}$. Instead of learning conditional densities as in [150], [135], the model is optimized by learning $f_{\boldsymbol{X},\boldsymbol{H}}(\mathbf{x}, \mathbf{h}; \theta)$ and updating $\mathbf{h}_t$ jointly using Eq.7.2 via the maximum likelihood principle. The LSTM network reduces the parameter space of the probabilistic model, by encoding the time-series history into an $M-$dimensional latent vector $\mathbf{h}_t \in \mathbb{R}^M$, while the LRCF model pushes $\mathbf{h}_t$ to only keep the information from the past that will lead towards better future modeling. We call the proposed method LRCF-LSTM.

7.4.2 Training the Overall Model

Training is performed by randomly sampling context and prediction length sized windows from the training time series data and optimizing the set of all parameters θ of both the low-rank tensor and the LSTM-RNN. In practice, the time series $\mathbf{x}_1, \ldots, \mathbf{x}_T$ in a batch $\mathcal{B}$ are selected from a random time window of size T with different starting points from the training data, so that the proposed model takes into account most of the information from different sub-sequences. This is important when the given time-series dataset has small history. We train the components jointly under the maximum likelihood principle, i.e. by minimizing the loss function

$$\mathcal{L}_{\text{NegLL}} := -\sum_{\mathbf{x}_{t_0:T} \in \mathcal{B}} \sum_{t=t_0}^{T} \log f_{\boldsymbol{X},\boldsymbol{H}}(\mathbf{x}_t, \mathbf{h}_{t-1}; \theta), \tag{7.4}$$

using stochastic gradient descent-based optimization.

We introduce a truncated multivariate Fourier series over $X_1, X_2, X_3 \ldots, X_N$, and $H_1, \ldots, H_M$ with cutoffs $K_1, \ldots, K_{N+M}$. For simplicity, we assume that $K_1 = \cdots = K_{N+M} = K$, we represent the concatenation of $\boldsymbol{X} \in \mathbb{R}^N$ and $\boldsymbol{H} \in \mathbb{R}^M$ in vector $\boldsymbol{Z} \in \mathbb{R}^{N+M}$. To reduce the number of parameters, we introduce a rank-F parameterization of the coefficient tensor obtained by truncating the multidimensional Fourier series [21] which reduces the number of parameters from $O(K^{N+M})$ to $O(KF(N+M))$. Notice that M is controlled by the dimensionality of the LSTM-RNN output and serves as a tunable hyper-parameter in the proposed architecture (see details in Section 7.5). Let $\boldsymbol{A}_d(K+1+k_d, f)$ hold $\Phi_{X_d|L=f}(k_d)$, $\boldsymbol{\lambda}(f)$ hold $p_L(f)$ and let us define matrices $\mathbf{B}_d \in \mathbb{C}^{K \times (T-t_0)}$ as $\mathbf{B}_d(K+1+k_d, t) = e^{-j2\pi k_d \mathbf{z}_t(d)}$. Introducing the rank-$F$ CPD, the loss function becomes

$$\mathcal{L}_{\text{NegLL}} := -\sum_{\mathbf{x}_{t_0:T} \in \mathcal{B}} \sum_{t=t_0}^{T} \log\Big((\circledast_{d=1}^{N+M} (\mathbf{B}_d(:,t)^T \mathbf{A}_d)) \boldsymbol{\lambda} \Big) \tag{7.5}$$

The above model can be used to obtain any marginal model (e.g., a model of the marginal density $f(\mathbf{h}_{t-1})$) by dropping out all but the subset of factor matrices $\{\mathbf{A}_S\}$, where $S \in \{N+1, \cdots, M\}$,

$$f_{\boldsymbol{H}}(\mathbf{h}) = (\circledast_{d=N+1}^{N+M} (\mathbf{B}_d(:,t)^T \mathbf{A}_d)) \boldsymbol{\lambda})). \tag{7.6}$$

Since we can easily obtain $f(\mathbf{h}_{t-1})$ from $f(\mathbf{x}_t, \mathbf{h}_{t-1})$ using Eq.7.6, we can easily estimate $f(\mathbf{x}_t|\mathbf{h}_{t-1})$. Forecasting can then be performed by computing the conditional expectation. Missing data imputation can likewise be tackled efficiently. For example, if certain elements of any $\mathbf{x}_t$ are missing, they can be imputed from whatever is available using conditional expectation. It is important to note that these conditional expectation computations are closed-form.

7.5 Experimental Evaluation

We compare our proposed model to baselines in public benchmarks on forecasting and imputation tasks.

Implementation details: All datasets have been split into training set (60%), validation set (20%) and test set (20%) in chronological order. We split all datasets by using all data prior to a fixed date for the training and validation and by using rolling windows for the test set. Parameters are learned with the Adam optimizer with the learning rate set to 0.0001, and hyper-parameter values, which include the rank F, the number of Fourier coefficients K, and the number of LSTM cells M are obtained based on a grid-search from the sets $[5, 10, 20, 30, 50, 80, 100]$, $[5, 10, 15, 20, 30]$ and $[10, 20, 40, 80, 120]$ respectively, using the validation set. The RNN consists of two LSTM layers for every experiment. We construct batches of size 64, with the context length equal to the prediction length. We mainly use the Root Mean Square Error (RMSE) and its scaled version, the Root Relative Squared Error (RSE) as evaluation metrics. Both for RMSE and RSE, lower scores indicate better performance.

Datasets: The Solar dataset consists of solar power production records sampled every 10 minutes from 137 stations in Alabama [154]. The future window is set from 30 to 120 minutes over the Solar-Energy data. The Traffic dataset consists of 48 months (2015-2016) hourly data describing the road occupancy rates measured by different sensors on San Francisco Bay area freeways [155]. The future window is set from 3 to 12 hours for the Traffic data. The Electricity dataset consists of hourly time series of the electricity consumption of 370 customers [54]. The Exchange-rates dataset includes daily exchange rates of eight foreign countries including Australia, British, Canada, Switzerland, China, Japan, New Zealand and Singapore ranging from 1990 to 2016 [154].

Baselines: Our evaluation is extensive, covering relevant baselines, including LSTM-FCNN, where an LSTM layer and a Fully Connected Neural Network are combined to perform multivariate time series forecasting, VAR [139], a multivariate linear autoregressive model, DeepAR [149], Vec-LSTM ind-scaling, which models the dynamics via an RNN and outputs the parameters of an independent Gaussian distribution [135], GP LR-Cop [135], which parameterizes a $f(\mathbf{x}_t, \mathbf{h}_{t-1})$ using a Gaussian copula with a low-rank-plus-diagonal parametrization of the covariance matrix, TCNF-MAF and Transformer-MAF [150], which uses RNN or Transformer respectively to model the temporal conditioning and Masked Autoregressive Flow for the distribution emission model. For each baseline, we use the recommended architectures and parameters reported for the given datasets in the corresponding papers. For LSTM-FCNN, we use a two layer LSTM, with the hidden state size chosen from $\{25, 50, 100, 200\}$, followed by a three-layer Fully Connected Network, with the number of hidden units chosen from $\{20, 50, 100\}$. The LSTM-FCNN is trained in an end-to-end fashion using a least-squares loss, thus directly approximating the nonlinear MMSE (conditional expectation) point estimator of $\mathbf{x}_t$ given $\mathbf{h}_{t-1}$, instead of estimating the joint distribution and computing conditional expectations indirectly from the joint distribution.

Results: Results in Table 7.1 show the corresponding RSE values averaged over 5 independent runs. The results indicate that LRCF-LSTM forecasts outperform current state-of-the-art multivariate forecasting approaches in most cases (including both probabilistic forecasts and point forecasts obtained by LSTM-FCNN). Compared with other models, the gain on Solar and Traffic datasets can be as significant as 3.7% and 4.0% in RSE. Apart from the Exchange-Rate dataset, we demonstrate improvement over [150], which further supports the claim that replacing Normalizing Flows with an LRCF-DE model is beneficial for improved forecasting accuracy. To empirically evaluate the sample complexity for all methods, we additionally test how changing the percentage of training (we retain 40%, 50%, and 70% of the training set) affects the forecasting performance on the Solar-Energy dataset. The results in Figure 7.2 suggest that in the small sample regime (40% training) the direct point estimate method (LSTM-FCNN) works better. However, the performance of our method quickly overpowers LSTM-FCNN, as more training samples become available. At the same time, our method showcases a better overall performance compared to the probabilistic baselines across different splits.

Dataset	Solar-Energy				Traffic				Electricity				Exchange-Rate			
Methods \ Horizon	3	6	12	24	3	6	12	24	3	6	12	24	3	6	12	24
LSTM-FCNN	.1927	.2505	.3363	.4755	.5205	.5522	.5605	.5695	.1008	.1144	.1193	.1298	.0185	.0264	.0408	.0626
VAR	.2435	.3790	.5911	.8699	.5991	.6218	.6252	.6293	.0995	.1035	.1050	.1054	.0228	.0279	.0353	.0445
DeepAR	.1843	.2559	.3254	.4643	.4777	.4893	.4950	.4973	.0864	.0931	.1007	.1007	.0226	.0280	.0356	.0449
Vec-LSTM ind-scaling	.1803	.2347	.3234	.4389	.4487	.4658	.4641	.4765	.0823	.0916	.0964	.1006	**.0174**	**.0241**	.0341	**.0444**
GP LR-Cop	.1898	.2580	.3472	.4441	.4212	.4586	.4679	.4743	.0762	.0917	.0966	.0994	.0175	**.0244**	**.0338**	.0447
TCNF-MAF	**.1704**	**.2257**	**.3072**	**.4050**	**.4095**	**.4470**	.4640	.4641	**.0740**	**.0898**	.0940	.0980	.0175	**.0244**	**.0339**	**.0442**
Transformer-MAF	.1778	.2348	.3109	.4270	.4162	.4754	**.4461**	**.4535**	.0745	**.0878**	**.0916**	**.0953**	.0194	.0259	.0349	.0456
LRCF-LSTM	**.1651**	**.2202**	**.2981**	**.3897**	**.4017**	**.4295**	**.4324**	**.4354**	**.0732**	.0898	**.0915**	**.0939**	**.0170**	.0245	.0340	.0450

Table 7.1: Benchmark on Solar-Energy, Traffic, Electricity and Exchange-Rate. Root Relative Squared Error (RSE) is reported for different horizons {3, 6, 12, 24}. All results have been averaged over 5 runs.

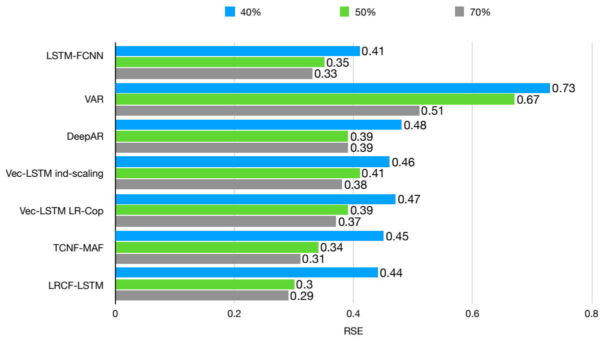


Figure 7.2: How the forecasting performance (RSE – the smaller the better) changes for different models by varying the training data percentage from 40% to 70% for the Solar-Energy dataset.

Forecasting stock values: We also study the NASDAQ dataset [156], which includes the stock prices of 81 major corporations, collected minute-by-minute. The future window over the NASDAQ data is set from 3 to 12 minutes. Results in Table 7.3 show RMSE values of different methods, averaged over 5 runs, demonstrating that our method offers better modeling of complex inter-dependencies, validating the effectiveness of the proposed framework. To further explore the modeling abilities of our model with respect to the best performing time-series probabilistic models, we employ the Apple stock dataset obtained from Yahoo Finance, which contains as features the volume, high, low, opening, and closing prices of Apple (AAPL), Meta (META), Netflix (NFLX), and Google (GOOGL). Using the data from 1st January 2017 up to 31st December 2020 to train all models, we consider multivariate forecasting of the

$5-$dimensional vector over the first 250 days in 2021. We use the Mean Absolute Errors (MAE), Mean Absolute Percentage Errors (MAPE), Mean Squared Errors (MSE), and Max Error to measure the performance. Our results are shown in Figure 7.3 and Table 7.2, where we demonstrate the forecasts of our method and of [135] and [150] for the closing price. The results indicate that the proposed model can accurately predict trends for AAPL, META, and NFLX. However, all methods fall short of reasonably predicting the evolultion of GOOGL in 2021 (our method is slightly better, but still fails). Of course, 2021 was a very unusual year due to the pandemic and other factors, so the fact that the other three stocks can be so well predicted so much in advance is already better than expected.

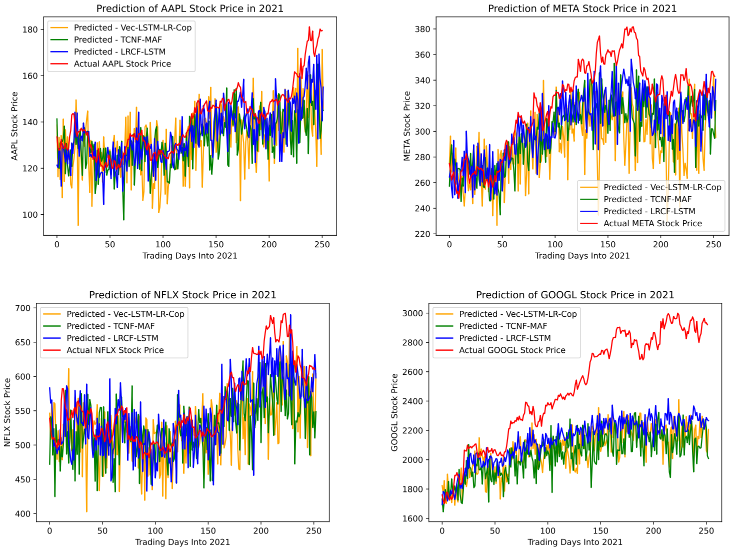


Figure 7.3: We use AAPL, META, NFLX, GOOGL (clockwise from top left) multivariate stock related data from 1st January 2017 up to 31 December 2020 to train all models and test (use the models to sequentially generate fully unsupervised predictions using previous predictions) over the first 250 days of 2021. The proposed model yields better forecasts for AAPL, META, NFLX. No model is able to forecast the meteoric rise of GOOGL that year.

Imputation: The second task of interest is time-series imputation. In the experiments, we use the localization for human activity UCI [54] dataset (Activity), consisting of the multivariate kinematic time series recording the motion state of 5 people performing 11 kinds of activities. Each person wore four sensors (tags) on her/his left/right ankle, chest, and belt to record the

Dataset	AAPL				META				NFLX				GOOGL			
Metrics Methods	MSE	MAE	MAPE	Max Error	MSE	MAE	MAPE	Max Error	MSE	MAE	MAPE	Max Error	MSE	MAE	MAPE	Max Error
GP LR-Cop	245.79	12.28	0.085	53.40	3462.92	44.65	0.077	191.89	1685.32	32.48	0.097	137.25	245493.80	424.08	0.159	1037.62
TCNF - MAF	200.47	10.77	0.073	48.93	3240.03	42.68	0.073	178.42	931.09	24.59	0.073	96.20	276872.97	451.14	0.169	1071.00
LRCF-LSTM	111.99	7.89	0.054	43.19	1735.25	32.55	0.057	178.40	646.03	20.1	0.061	86.27	179519.31	353.69	0.131	804.31

Table 7.2: AAPL, META, NFLX, GOOGL multivariate stock forecasting results.

Methods	3	6	12
VAR	2.725	3.049	3.048
DeepAR	3.522	3.903	4.595
Vec-LSTM ind-scaling	4.195	4.295	4.574
GP LR-Cop	5.795	5.891	7.685
TCNF-MAF	**0.366**	**0.522**	**0.754**
LRCF-LSTM	**0.365**	**0.516**	**0.739**

Table 7.3: Benchmark on NASDAQ. Root Mean Squared Error (RMSE) is reported for different horizons $\{3, 6, 12\}$. All results have been averaged over 5 runs.

3-dimensional coordinates. In the experiments, we utilize the RMSE to measure the imputation performance. We compare the performance of our LRCF-LSTM model using conditional expectation with both standard baselines and more recent ones. The standard baselines include mean, last value repetition, replacing with a random value from the last 10 observed values, weighted moving average, interpolation, Kalman smoothing and the more recent ones include GP-VAE [157] and TSGAN [158]. The results are presented in Table 7.4. Our model incurs 8.21256%, 8.74636%, and 7.25191% decrease in RMSE against the second best model for 10%, 30%, and 50% missing data respectively.

7.6 Conclusions

Real-world time-series data has introduced many new challenges for data modeling. In this work we aim to sufficiently capture high dimensional, complex dependencies across time by learning embeddings of time steps and modeling them jointly with current observations using flexible density models. We propose modeling an LSTM network to capture the transition dynamics of latent representations of observed data and a non-parametric density model based on tensor decompositions and the characteristic function representation, to model the latent

	10%	30%	50%
Mean	.813	.873	.933
Last	.792	.862	.936
Random-Last-10	.722	.792	.879
Weighted Moving Average	.670	.726	.796
Interpolation	.705	.785	.825
Kalman Smoothing	.677	.762	.812
GP-VAE [157]	.641	.724	.794
TSGAN [158]	.621	.686	.786
LRCF-LSTM	.570	.626	.729

Table 7.4: Performance comparison (in RMSE) of time series imputation methods under different missing rates.

representations jointly with current observations. We obtain a flexible model that allows efficient likelihood evaluation during training time and efficient evaluation of conditional expectations during inference time. We employ extensive experiments to validate the accuracy of the proposed approach with respect to the state-of-the-art on forecasting and on time-series imputation.

Chapter 8

Thesis Summary and Future Directions

In this dissertation we studied the general problem of distribution estimation under different settings and challenges. We developed tensor-based approaches which are principled, come with identifiability guarantees, while at the same time achieve a superior performance with respect to the state-of-the-art. We addressed important and challenging problems that arise in real world applications (e.g., anomaly detection, modeling complex high-dimensional data such as images), described how tensor-based probabilistic models can be utilized towards them, and validated their potential via a detailed experimental examination using complex real-world data.

8.1 Summary

In chapter 3, based an interesting link between tensors and multivariate statistics that was first poionted out in [18], we showed that by indirectly aiming to predict the latent variable of the naive Bayes model instead of the original target variable, it is possible to formulate the feature selection problem as maximization of a monotone submodular function subject to a cardinality constraint. This problem can be tackled using a greedy algorithm that comes with performance guarantees.

In chapter 4 we studied the problem of non-parametric density estimation. If the sought density is compactly supported, then its characteristic function can be approximated, within controllable error, by a finite tensor of leading Fourier coefficients, whose size depends on the smoothness of the underlying density. This tensor can be naturally estimated from observed and possibly incomplete realizations of the random vector of interest, via sample averaging. In order to circumvent the curse of dimensionality, we introduced a low-rank model of this characteristic tensor, which significantly improves the density estimate especially for high-dimensional data and/or in the sample-starved regime. By virtue of uniqueness of low-rank tensor decomposition, under certain conditions, our method enables learning the true data-generating distribution.

In chapter 5, we built on the results of chapter 4 and developed a joint dimensionality reduction and non-parametric density estimation framework, using the aforementioned density estimator that can explicitly capture the underlying distribution of appropriate reduced-dimension representations of the input data. The idea is to jointly design a nonlinear dimensionality reducing auto-encoder to model the training data in terms of a parsimonious set of latent random variables, and learn a canonical low-rank tensor model of the joint distribution of the latent variables in the Fourier domain. The proposed latent density model is non-parametric and “universal,” as opposed to the predefined prior that is assumed in variational auto-encoders.

In chapter 6, we learn multivariate cumulative distribution functions (CDFs), as they can handle mixed random variables, allow efficient ‘box’ probability evaluation, and have the potential to overcome local sample scarcity owing to their cumulative nature. We show that any grid-sampled version of a joint CDF of mixed random variables admits a universal representation as a naive Bayes model via the Canonical Polyadic (tensor-rank) decomposition. By introducing a low-rank model, either directly in the raw data domain, or indirectly in a transformed (Copula) domain, the resulting model affords efficient sampling, closed form inference and uncertainty quantification, and comes with uniqueness guarantees under relatively mild conditions.

In the last chapter 7, we pursue probabilistic modeling of multivariate time-series. We propose a novel approach which ties together two threads of research: low-rank tensor modeling for flexible and efficient density estimation and the temporal modeling power of recurrent neural networks (RNNs). This end-to-end framework leverages RNNs for learning sequential dynamics and the parsimony of principled tensor-based density models to obtain a time-series generative model with a controllable number of parameters. The proposed model generates probabilistic forecasts that allow for uncertainty quantification and can handle missing inputs in some or all of the components in a multivariate time series.

8.2 Future Directions

- Privacy-preserving density estimation: We have shown that an accurate distribution estimate can be used to perform a variety of important analysis tasks in critical fields such as the healthcare (e.g., output the likelihood estimate of a clinical trial achieving a certain enrollment rate goal). For settings involving sensitive data (e.g., healthcare data), the construction and subsequent release of a density estimate could potentially leak private information. Thus, continued attention is required to develop privacy-preserving methods for density estimation. The inputs to our density model introduced in chapter 4 are sample average estimates of $E[e^{j\boldsymbol{\nu}^T\boldsymbol{X}}]$, thus information about individual input vectors in the data is washed out, due to averaging and the modulo 2π periodicity of the complex exponential. We can in fact discard all raw data for the purposes of joint PDF estimation. Future work could explore how secure this strategy really is for healthcare.

- Variational Inference using Low-Rank Characteristic Function based Density Estimators: The choice of posterior distributions is one of the core problems in Variational autoencoders (VAE). In most cases, variational inference employs simple families of posterior approximations (Gaussian densities) in order to allow for efficient inference. VAEs are known to produce samples of poor quality when implemented in their most basic form – they require more powerful generative models in order to be competitive. Given the enhanced expressivity of Low-Rank Characteristic Function based density estimators, one could introduce such models to approximate posterior distributions. Although Normalizing Flows have also been employed to this task [80], the proposed class of densities allows much faster sampling times thanks to the latent variable naive Bayes interpretation. Potential future work could explore if the overall model can enhance sample quality with respect to [80].

References

[1] M. Amiridi and N. Sidiropoulos, “Learning multivariate CDFs and copulas using tensor factorization,” 2022. [Online]. Available: https://arxiv.org/abs/2210.07132

[2] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for Generative Adversarial Networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.

[3] A. Brock, J. Donahue, and K. Simonyan, “Large scale GAN training for high fidelity natural image synthesis,” *arXiv preprint arXiv:1809.11096*, 2018.

[4] W.-N. Hsu, Y. Zhang, and J. Glass, “Learning latent representations for speech generation and transformation,” *arXiv preprint arXiv:1704.04222*, 2017.

[5] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” *arXiv preprint arXiv:1609.03499*, 2016.

[6] A. Odena, “Semi-supervised learning with Generative Adversarial Networks,” *arXiv preprint arXiv:1606.01583*, 2016.

[7] G. Ostrovski, M. G. Bellemare, A. Oord, and R. Munos, “Count-based exploration with neural density models,” in *International Conference on Machine Learning*. PMLR, 2017, pp. 2721–2730.

[8] H. Dolatabadi, S. Erfani, and C. Leckie, “Advflow: Inconspicuous black-box adversarial attacks using Normalizing Flows,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 871–15 884, 2020.

[9] G. J. McLachlan and K. E. Basford, *Mixture models: Inference and applications to clustering*. M. Dekker New York, 1988, vol. 38.

[10] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *Ann. Math. Statist.*, vol. 27, no. 3, pp. 832–837, 09 1956.

[11] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[12] M. Girolami, "Orthogonal series density estimation and the kernel eigenvalue problem," *Neural computation*, vol. 14, no. 3, pp. 669–688, 2002.

[13] S. Efromovich, "Orthogonal series density estimation," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 467–476, 2010.

[14] A. B. Tsybakov, *Introduction to nonparametric estimation*. Springer Science & Business Media, 2008.

[15] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, "Pixelcnn++: Improving the pixelCNN with discretized logistic mixture likelihood and other modifications," *arXiv preprint arXiv:1701.05517*, 2017.

[16] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *Proc. ICLR*, 2017. [Online]. Available: https://arxiv.org/abs/1605.08803

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 2672–2680.

[18] N. Kargas, N. Sidiropoulos, and X. Fu, "Tensors, learning, and "Kolmogorov extension" for finite-alphabet random vectors," *IEEE Transactions on Signal Processing*, vol. 66, no. 18, pp. 4854–4868, 2018.

[19] M. Amiridi, N. Kargas, and N. Sidiropoulos, "Statistical learning using hierarchical modeling of probability tensors," in *2019 IEEE Data Science Workshop (DSW)*. IEEE, 2019, pp. 290–294.

[20] M. Amiridi, N. Kargas, and N. D. Sidiropoulos, “Information-theoretic feature selection via tensor decomposition and submodularity,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 6195–6205, 2021.

[21] M. Amiridi, N. Kargas, and N. Sidiropoulos, “Low-rank characteristic tensor density estimation part I: Foundations,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 2654–2668, 2022.

[22] M. Amiridi, N. Kargas, and N. Sidiropoulos, “Low-rank characteristic tensor density estimation part II: Compression and latent density estimation,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 2669–2680, 2022.

[23] N. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, “Tensor decomposition for signal processing and machine learning,” *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[24] M. Amiridi, G. Darnell, and S. Jewell, “Latent temporal flows for multivariate analysis of wearables data,” 2022. [Online]. Available: http://arxiv.org/abs/2210.07475

[25] F. L. Hitchcock, “The expression of a tensor or a polyadic as a sum of products,” *Journal of Mathematics and Physics*, vol. 6, no. 1-4, pp. 164–189, 1927.

[26] R. A. Harshman, “Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multimodal factor analysis,” *UCLA Working Papers Phonetics*, vol. 16, pp. 1–84, 1970.

[27] N. Sidiropoulos and R. Bro, “On the uniqueness of multilinear decomposition of N-way arrays,” *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 14, no. 3, pp. 229–239, 2000.

[28] L. Chiantini and G. Ottaviani, “On generic identifiability of 3-tensors of small rank,” *SIAM Journal on Matrix Analysis and Applications*, vol. 33, no. 3, pp. 1018–1037, 2012.

[29] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, “Text classification from labeled and unlabeled documents using EM,” *Machine learning*, vol. 39, no. 2-3, pp. 103–134, 2000.

[30] N. L. Zhang, “Hierarchical latent class models for cluster analysis,” *J. Mach. Learn. Res.*, vol. 5, no. 6, pp. 697–723, Jun. 2004.

[31] Y. Zhang, X. Chen, D. Zhou, and M. I. Jordan, “Spectral methods meet EM: A provably optimal algorithm for crowdsourcing,” in *Advances in Neural Information Processing Systems*, 2014, pp. 1260–1268.

[32] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 1991.

[33] J. Tang, S. Alelyani, and H. Liu, “Feature selection for classification: A review,” in *Data Classification*. CRC Press, 2014, pp. 37–64.

[34] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, “Feature selection: A data perspective,” *ACM Computing Surveys*, vol. 50, no. 6, pp. 1–45, 2018.

[35] M. Ritchie, L. Hahn, N. Roodi, L. Bailey, W. Dupont, F. Parl, and J. Moore, “Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer,” *American journal of human genetics*, vol. 69, no. 1, pp. 138–147, 2001.

[36] A. Duric and F. Song, “Feature selection for sentiment analysis based on content and syntax models,” *Decision support systems*, vol. 53, no. 4, pp. 704–711, 2012.

[37] C. C. Aggarwal and C. Zhai, *Mining text data*. Springer Science & Business Media, 2012.

[38] C.-L. Huang and C.-Y. Tsai, “A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting,” *Expert Systems with Applications*, vol. 36, no. 2, pp. 1529–1539, 2009.

[39] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján, “Conditional likelihood maximisation: a unifying framework for information theoretic feature selection,” *The journal of machine learning research*, vol. 13, pp. 27–66, 2012.

[40] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker, “On variational bounds of mutual information,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 5171–5180.

[41] Hanchuan Peng, Fuhui Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[42] M. Bennasar, Y. Hicks, and R. Setchi, "Feature selection using joint mutual information maximisation," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8520–8532, 2015.

[43] S. Yu, L. G. S. Giraldo, R. Jenssen, and J. C. Principe, "Multivariate extension of matrix-based Renyi's α-order entropy functional," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[44] N. X. Vinh, J. Chan, and J. Bailey, "Reconsidering mutual information based feature selection: A statistical significance view," in *Proceedings of the twenty-eighth AAAI conference on artificial intelligence, Québec City*, 2014, pp. 2092–2098.

[45] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[46] A. Krause and C. Guestrin, "Near-optimal nonmyopic value of information in graphical models," in *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, 2005, pp. 324–331.

[47] A. Krause and D. Golovin, *Submodular Function Maximization*. Cambridge University Press, 2014, pp. 71–104.

[48] L. A. Nemhauser, G. L. Wolsey and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[49] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," in *Optimization Techniques*, 1978, pp. 234–243.

[50] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature extraction: foundations and applications*. Springer, 2008, vol. 207.

[51] M. V. Shashanka, B. Raj, and P. Smaragdis, "Probabilistic latent variable models as nonnegative factorizations," *Computational Intelligence and Neuroscience*, vol. 2008, 2008.

[52] K. Huang and N. D. Sidiropoulos, “Kullback-Leibler principal component for tensors is not NP-hard,” in *Proceedings of the 51st Asilomar Conference on Signals, Systems, and Computers*, 2017, pp. 693–697.

[53] C. J. Hillar and L.-H. Lim, “Most tensor problems are NP-hard,” *Journal of the ACM*, vol. 60, no. 6, pp. 1–39, 2013.

[54] M. Lichman, “UCI machine learning repository,” 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[55] B. W. Bader and T. G. Kolda, “Efficient MATLAB computations with sparse and factored tensors,” *SIAM Journal on Scientific Computing*, vol. 30, no. 1, pp. 205–231, December 2007.

[56] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, “Mutual information neural estimation,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 531–540.

[57] M. Noshad, Y. Zeng, and A. O. Hero, “Scalable mutual information estimation using dependence graphs,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2962–2966.

[58] T. Schmah, G. E. Hinton, S. L. Small, S. Strother, and R. S. Zemel, “Generative versus discriminative training of RBMs for classification of fMRI images,” in *Advances in Neural Information Processing Systems*, 2009, pp. 1409–1416.

[59] E. L. Ray, K. Sakrejda, S. A. Lauer, M. A. Johansson, and N. G. Reich, “Infectious disease prediction with kernel conditional density estimation,” *Statistics in Medicine*, vol. 36, no. 30, pp. 4908–4929, 2017.

[60] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *International Conference on Knowledge Discovery and Data Mining*, vol. 96, 1996, pp. 226–231.

[61] H. Zen and A. Senior, “Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 3844–3848.

[62] D. Titterington and J. Sedransk, “Imputation of missing values using density estimation,” *Statistics & Probability Letters*, vol. 8, no. 5, pp. 411–418, 1989.

[63] J. Ballé, V. Laparra, and E. P. Simoncelli, “Density modeling of images using a generalized normalization transformation,” in *International Conference on Learning Representations*, 2016.

[64] B. W. Silverman, *Density estimation for statistics and data analysis*. Routledge, 2018.

[65] K. Pearson, “Contributions to the mathematical theory of evolution,” *Philosophical Transactions of the Royal Society of London. A*, vol. 185, pp. 71–110, 1894.

[66] M. Germain, K. Gregor, I. Murray, and H. Larochelle, “MADE: Masked autoencoder for distribution estimation,” in *International Conference on Machine Learning (ICML)*, 2015, pp. 881–889.

[67] B. Uria, M.-A. Côté, K. Gregor, I. Murray, and H. Larochelle, “Neural autoregressive distribution estimation,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 7184–7220, 2016.

[68] G. Papamakarios, T. Pavlakou, and I. Murray, “Masked autoregressive flow for density estimation,” in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 2338–2347.

[69] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *International Conference on Learning Representations*, 2014.

[70] R. A. Harshman *et al.*, “Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multimodal factor analysis,” *UCLA Working Papers Phonetics*, vol. 16, pp. 1–84, 1970.

[71] J. Chen, “Optimal rate of convergence for finite mixture models,” *The Annals of Statistics*, pp. 221–233, 1995.

[72] D. W. Scott, “Feasibility of multivariate density estimates,” *Biometrika*, vol. 78, no. 1, pp. 197–205, 1991.

[73] E. S. Allman, C. Matias, and J. A. Rhodes, “Identifiability of parameters in latent structure models with many observed variables,” *The Annals of Statistics*, vol. 37, no. 6A, pp. 3099–3132, 2009.

[74] L. Song, A. Anandkumar, B. Dai, and B. Xie, “Nonparametric estimation of multi-view latent variable models,” in *International Conference on Machine Learning*. PMLR, 2014, pp. 640–648.

[75] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, “Tensor decompositions for learning latent variable models,” *Journal of machine learning research*, vol. 15, pp. 2773–2832, 2014.

[76] T. G. Kolda, “Symmetric orthogonal tensor decomposition is trivial,” *arXiv preprint arXiv:1503.01375*, 2015.

[77] N. Kargas and N. Sidiropoulos, “Learning mixtures of smooth product distributions: Identifiability and algorithm,” in *22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019, pp. 388–396.

[78] S. Dolgov, K. Anaya-Izquierdo, C. Fox, and R. Scheichl, “Approximation and sampling of multivariate probability distributions in the tensor train decomposition,” *Statistics and Computing*, vol. 30, no. 3, pp. 603–625, 2020.

[79] B. Uria, I. Murray, and H. Larochelle, “RNADE: The real-valued neural autoregressive density-estimator,” in *Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 2175–2183.

[80] D. Rezende and S. Mohamed, “Variational inference with Normalizing Flows,” in *International Conference on Machine Learning*. PMLR, 2015, pp. 1530–1538.

[81] H. Poon and P. Domingos, “Sum-product networks: A new deep architecture,” in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 689–690.

[82] I. París, R. Sánchez-Cauce, and F. J. Díez, “Sum-product networks: A survey,” *arXiv preprint arXiv:2004.01167*, 2020.

[83] G. Plonka, D. Potts, G. Steidl, and M. Tasche, *Numerical Fourier Analysis*. Springer, 2018.

[84] J. C. Mason, "Near-best multivariate approximation by Fourier series, Chebyshev series and Chebyshev interpolation," *Journal of Approximation Theory*, vol. 28, no. 4, pp. 349–358, 1980.

[85] D. C. Handscomb, *Methods of numerical approximation: lectures delivered at a Summer School held at Oxford University, September 1965*. Elsevier, 2014.

[86] W. Wang and M. A. Carreira-Perpinán, "Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application," *arXiv preprint arXiv:1309.1541*, 2013.

[87] L. Theis, A. v. d. Oord, and M. Bethge, "A note on the evaluation of generative models," in *4th International Conference on Learning Representations (ICLR)*, 2016.

[88] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems*, 1990, pp. 396–404.

[89] N. G. Ushakov, *Selected topics in characteristic functions*. Walter de Gruyter, 2011.

[90] M. Gavish and D. L. Donoho, "Optimal shrinkage of singular values," *IEEE Transactions on Information Theory*, vol. 63, no. 4, pp. 2137–2152, 2017.

[91] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 215–10 224.

[92] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," *arXiv preprint arXiv:1511.06349*, 2015.

[93] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *International Conference on Learning Representations*, 2018, pp. 1–19.

[94] B. W. Silverman, *Density estimation for statistics and data analysis*. CRC press, 1986, vol. 26.

[95] G. J. McLachlan and D. Peel, *Finite mixture models*. John Wiley & Sons, 2004.

[96] B. Dai and D. Wipf, "Diagnosing and enhancing VAE models," *arXiv preprint arXiv:1903.05789*, 2019.

[97] M. Rosca, B. Lakshminarayanan, and S. Mohamed, "Distribution matching in variational inference," *arXiv preprint arXiv:1802.06847*, 2018.

[98] A. V. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel Recurrent Neural Networks," in *International Conference on Machine Learning*, vol. 48, 20–22 Jun 2016, pp. 1747–1756.

[99] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Non-linear independent components estimation," *arXiv preprint arXiv:1410.8516*, 2015.

[100] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel, "Flow++: Improving flow-based generative models with variational dequantization and architecture design," in *International Conference on Machine Learning*, 2019, pp. 2722–2730.

[101] Y. LeCun, "The MNIST database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[102] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[103] M. D. Hoffman and M. J. Johnson, "ELBO surgery: yet another way to carve up the variational evidence lower bound," in *Workshop in Advances in Approximate Bayesian Inference, NIPS*, vol. 1, 2016, p. 2.

[104] C. Meng, Y. Song, J. Song, and S. Ermon, "Gaussianization flows," in *International Conference on Artificial Intelligence and Statistics*, 2020, pp. 4336–4345.

[105] L. Cheng, X. Tong, S. Wang, Y.-C. Wu, and H. V. Poor, "Learning nonnegative factors from tensor data: Probabilistic modeling and inference algorithm," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1792–1806, 2020.

[106] L. Xu, L. Cheng, N. Wong, and Y.-C. Wu, "Overfitting avoidance in tensor train factorization and completion: Prior analysis and inference," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 1439–1444.

[107] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *ICML*, 2010.

[108] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[109] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep Convolutional Neural Networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[110] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *3rd International Conference on Learning Representations, ICLR*, 2015.

[111] P. Chilinski and R. Silva, “Neural likelihoods via Cumulative Distribution Functions,” in *Conference on Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 420–429.

[112] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, “Normalizing flows for probabilistic modeling and inference,” *Journal of Machine Learning Research*, vol. 22, no. 57, pp. 1–64, 2021.

[113] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[114] J. Shao, *Mathematical statistics*. Springer Science & Business Media, 2003.

[115] C. Daskalakis, I. Diakonikolas, and R. A. Servedio, “Learning k-modal distributions via testing,” in *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 2012, pp. 1371–1385.

[116] M. Sklar, “Fonctions de repartition an dimensions et leurs marges,” *Publ. inst. statist. univ. Paris*, vol. 8, pp. 229–231, 1959.

[117] R. B. Nelsen, *An introduction to Copulas*. Springer science & business media, 2007.

[118] K. Huang, N. Sidiropoulos, and A. P. Liavas, “A flexible and efficient algorithmic framework for constrained matrix and tensor factorization,” *IEEE Transactions on Signal Processing*, vol. 64, no. 19, pp. 5052–5065, 2016.

[119] W. Grathwohl, R. T. Q. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud, “FFJORD: Free-form continuous dynamics for scalable reversible generative models,” *International Conference on Learning Representations*, 2019.

[120] C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville, “Neural autoregressive flows,” in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR, 10–15 Jul 2018, pp. 2078–2087. [Online]. Available: http://proceedings.mlr.press/v80/huang18d.html

[121] R. E. Barlow and H. D. Brunk, “The isotonic regression problem and its dual,” *Journal of the American Statistical Association*, vol. 67, no. 337, pp. 140–147, 1972.

[122] Y. Zhao and M. Udell, “Missing value imputation for mixed data via Gaussian copula,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 636–646.

[123] P.-A. Mattei and J. Frellsen, “Miwae: Deep generative modelling and imputation of incomplete data sets,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 4413–4423.

[124] K. Aas, C. Czado, A. Frigessi, and H. Bakken, “Pair-Copula constructions of multiple dependence,” *Insurance: Mathematics and economics*, vol. 44, no. 2, pp. 182–198, 2009.

[125] C. Qian, K. Huang, L. Glass, R. S. Srinivasa, and J. Sun, “Julia: Joint multi-linear and nonlinear identification for tensor completion,” *arXiv preprint arXiv:2202.00071*, 2022.

[126] C.-H. Du, Y.-S. Chiang, K.-C. Tsai, L.-C. Liu, M.-F. Tsai, and C.-J. Wang, “Fridays: A financial risk information detecting and analyzing system,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 9853–9854.

[127] M. W. Seeger, D. Salinas, and V. Flunkert, “Bayesian intermittent demand forecasting for large inventories,” *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[128] B. M. Goodale, M. Shilaih, L. Falco, F. Dammeier, G. Hamvas, and B. Leeners, “Wearable sensors reveal menses-driven changes in physiology and enable prediction of the fertile

window: observational study," *Journal of medical Internet research*, vol. 21, no. 4, p. e13404, 2019.

[129] F. Giannoni, M. Mancini, and F. Marinelli, "Anomaly detection models for IoT time series data," *arXiv preprint arXiv:1812.00890*, 2018.

[130] S. Selvin, R. Vinayakumar, E. Gopalakrishnan, V. K. Menon, and K. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *2017 International Conference on Advances in Computing, Communications and Informatics*. IEEE, 2017, pp. 1643–1647.

[131] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[132] R. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder, *Forecasting with exponential smoothing: the State Space approach*. Springer Science & Business Media, 2008.

[133] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization," *CoRR, vol. abs/1509.08333*, 2015.

[134] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," *Advances in neural information processing systems*, vol. 28, pp. 2980–2988, 2015.

[135] D. Salinas, M. Bohlke-Schneider, L. Callot, R. Medico, and J. Gasthaus, "High-dimensional multivariate forecasting with low-rank Gaussian Copula processes," *Advances in neural information processing systems*, vol. 32, 2019.

[136] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[137] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[138] C. A. Sims, "Macroeconomics and reality," *Econometrica: Journal of the Econometric Society*, pp. 1–48, 1980.

[139] G. Box and G. Jenkins, *Time Series Analysis: Forecasting and Control*, ser. Holden-Day series in time series analysis and digital processing. Holden-Day, 1976. [Online]. Available: https://books.google.com/books?id=1WVHAAAAMAAJ

[140] G. E. Box, G. M. Jenkins, and J. F. MacGregor, “Some recent advances in forecasting and control,” *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 23, no. 2, pp. 158–179, 1974.

[141] E. McKenzie, “General Exponential Smoothing and the equivalent ARMA process,” *Journal of Forecasting*, vol. 3, no. 3, pp. 333–344, 1984.

[142] R. E. Kalman, “A new approach to linear filtering and prediction problems,” 1960.

[143] J. D. Hamilton, *Time series analysis*. Princeton University press, 2020.

[144] C. Liu, S. C. Hoi, P. Zhao, and J. Sun, “Online ARIMA algorithms for time series prediction,” in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[145] A. J. Patton, “A review of Copula models for economic time series,” *Journal of Multivariate Analysis*, vol. 110, pp. 4–18, 2012.

[146] Z. Yu, M. Zhu, M. Trapp, A. Skryagin, and K. Kersting, “Leveraging probabilistic circuits for nonparametric multi-output regression,” in *Uncertainty in Artificial Intelligence*. PMLR, 2021, pp. 2008–2018.

[147] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[148] G. Corani, A. Benavoli, and M. Zaffalon, “Time series forecasting with Gaussian Processes needs priors,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2021, pp. 103–117.

[149] D. Salinas, V. Flunkert, and J. Gasthaus, “DeepAR: Probabilistic forecasting with autoregressive recurrent networks,” *International Journal of Forecasting*, vol. 8, no. 2, pp. 136–153, 2019.

[150] K. Rasul, A.-S. Sheikh, I. Schuster, U. Bergmann, and R. Vollgraf, “Multivariate probabilistic time series forecasting via conditioned Normalizing Flows,” *arXiv preprint arXiv:2002.06103*, 2020.

[151] K. Rasul, C. Seward, I. Schuster, and R. Vollgraf, "Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8857–8868.

[152] J. Yoon, D. Jarrett, and M. Van der Schaar, "Time-series Generative Adversarial Networks," 2019.

[153] N. Nguyen and B. Quanz, "Temporal latent auto-encoder: A method for probabilistic multivariate time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 10, 2021, pp. 9117–9125.

[154] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *The 41st international ACM SIGIR conference on research & development in information retrieval*, 2018, pp. 95–104.

[155] M. Cuturi, "Fast global alignment kernels," in *Proceedings of the 28th International Conference on Machine Learning*, 2011, pp. 929–936.

[156] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," *arXiv preprint arXiv:1704.02971*, 2017.

[157] V. Fortuin, D. Baranchuk, G. Rätsch, and S. Mandt, "GP-VAE: Deep probabilistic time series imputation," in *International conference on artificial intelligence and statistics*. PMLR, 2020, pp. 1651–1661.

[158] Y. Luo, X. Cai, Y. Zhang, J. Xu, and X. Yuan, "Multivariate time series imputation with Generative Adversarial Networks," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 1603–1614.