UNIVERSITÉ DU LUXEMBOURG

PhD-FSTC-2019-64
The Faculty of Sciences, Technology and Communication

DISSERTATION

Defence held on 22/10/2019 in Luxembourg
to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Jeremy Henri J CHARLIER
Born on 23 September 1988 in Liege (Belgium)

From Persistent Homology to Reinforcement Learning With Applications for Retail Banking

Dissertation defence committee

Dr. Radu State, Dissertation Supervisor
Professor, SnT, University of Luxembourg

Dr. Björn Ottersten, Chairman
Professor, SnT, University of Luxembourg

Dr. Jean Hilger, Vice Chairman
Head of Information Technology Department, BCEE

Dr. Djamila Aouada
Professor, SnT, University of Luxembourg

Dr. Vijay Gurbani
Professor (Adjunct), Illinois Institute of Technology
Chief Data Scientist, Vail Systems, Inc.

To Marie-Jeanne

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 60,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Jeremy Henri J Charlier
October 2019

Acknowledgements

I would like to thank particularly Dr. Habil. Radu State and Dr. Jean Hilger for having given me the opportunity to pursue a PhD at the university of Luxembourg in collaboration with the Banque et Caisse d'Épargne de l'État (BCEE). I am grateful for the time and the dedication they put to ensure the successfullness of the research project. I would like also to thank Dr. Djamila Aouada for her precise advice and challenging questions during the PhD supervision. Additionally, I am truly grateful to Patric De Waha for taking the time to supervise the various projects for which I was involved at BCEE as well as his outstanding understanding of the difficulties of a PhD research project. Furthermore, I would like to thank Dr. Henning Schulzrinne and Dr. Gaston Ormazabal for having given me the opportunity to do a research stay within the IRT lab at Columbia university. I deeply appreciate the time they invested for the supervision of the PhD research exchange, leaving me unforgettable memories. I am genuinely appreciative to Dr. Björn Ottersten for his valuable inputs during the review of this work. Moreover, I am grateful to Dr. Vijay Gurbani for his dedication and his critiques throughout the assessment of this PhD thesis. Finally, I would like to thank my fiancée, Jennifer, for her patience during this long journey and my family for their continuous support. I would like to thank all my fellow colleagues of the university of Luxembourg, the Banque et Caisse d'Epargne de l'Etat and the Columbia university as well as all the others I forgot to mention who helped to contribute to this work.

Abstract

The retail banking services are one of the pillars of the modern economic growth. However, the evolution of the client's habits in modern societies and the recent European regulations promoting more competition mean the retail banks will encounter serious challenges for the next few years, endangering their activities. They now face an impossible compromise: maximizing the satisfaction of their hyper-connected clients while avoiding any risk of default and being regulatory compliant. Therefore, advanced and novel research concepts are a serious game-changer to gain a competitive advantage.

In this context, we investigate in this thesis different concepts bridging the gap between persistent homology, neural networks, recommender engines and reinforcement learning with the aim of improving the quality of the retail banking services. Our contribution is threefold. First, we highlight how to overcome insufficient financial data by generating artificial data using generative models and persistent homology. Then, we present how to perform accurate financial recommendations in multi-dimensions. Finally, we underline a reinforcement learning model-free approach to determine the optimal policy of money management based on the aggregated financial transactions of the clients.

Our experimental data sets, extracted from well-known institutions where the privacy and the confidentiality of the clients were not put at risk, support our contributions. In this work, we provide the motivations of our retail banking research project, describe the theory employed to improve the financial services quality and evaluate quantitatively and qualitatively our methodologies for each of the proposed research scenarios.

Table of contents

List of Publications

Jeremy Charlier, Sofiane Lagraa, Jerôme François, and Radu State. Profiling Smart Contracts Interactions with Tensor Decomposition and Graph Mining. In *Workshop on MIning DAta for financial applicationS (MIDAS) in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECML-PKDD)*, 2017.

Jeremy Charlier, Radu State, and Jean Hilger. Modeling Smart Contracts Activities: A Tensor Based Approach. In *2017 IEEE Future Technologies Conference (FTC)*, pages 49–55. IEEE, 2018.

Jeremy Charlier, Radu State, and Jean Hilger. Non-Negative PARATUCK2 Tensor Decomposition Combined to LSTM Network for Smart Contracts Profiling. In *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 74–81. IEEE, 2018.

Jeremy Charlier, Radu State, and Jean Hilger. Non-Negative PARATUCK2-LSTM for Smart Contracts Monitoring. *International Journal of Computer and Software Engineering (IJCSE)*, 3, 2018.

Jeremy Charlier, Eric Falk, Radu State, and Jean Hilger. User-Device Authentication in Mobile Banking using ApHeN for PARATUCK2 Tensor Decomposition. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 886–894. IEEE, 2018.

Jeremy Charlier, Radu State, and Jean Hilger. Predicting Sparse Clients' Actions with CPOPT-Net in the Banking Environment. In *The 32nd Canadian Conference on Artificial Intelligence (Canadian AI)*, 2019.

Jeremy Charlier, Radu State, and Jean Hilger. PHom-GeM: Persistent Homology for Generative Models. In *The 6th Swiss Conference on Data Science (SDS)*, 2019.

Jeremy Charlier, Francois Petit, Gaston Ormazabal, and Radu State. Visualization of AE's Training on Credit Card Transactions with Persistent Homology. In *Workshop*

on Applications of Topological Data Analysis (ATDA) in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECML-PKDD), 2019.

Jeremy Charlier, Gaston Ormazabal, Radu State, and Jean Hilger. MQLV: Optimal Policy of Money Management in Retail Banking with Q-learning. In *Workshop on MIning DAta for financial applicationS (MIDAS) in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECML-PKDD)*, 2019.

Jeremy Charlier, Radu State, and Jean Hilger. VecHGrad for Solving Accurately Complex Tensor Decomposition. In *Submission to the thirty-fourth annual conference of the Association for the Advancement of Artificial Intelligence (AAAI)*, 2020.

Jeremy Charlier, Aman Singh, Gaston Ormazabal, Radu State, and Henning Schulzrinne. SynGAN: Towards Generating Synthetic Network Attacks using GANs. In *Submission to the thirty-fourth annual conference of the Association for the Advancement of Artificial Intelligence (AAAI)*, 2020.

List of Figures

List of Tables

Nomenclature

AE	Auto-Encoder
AI	Artificial Intelligence
ALS	Alternating Least Square
ApHeN	Approximate Hessian Newton
BFGS	Broyden–Fletcher–Goldfarb–Shanno
BSM	Black-Scholes-Merton
CG	Conjugate Gradient
CNN	Convolutional Neural Network
CP	Candecomp/Parafac
DT	Decision Tree
GAN	Generative Adversarial Network
GP-WGAN	Gradient Penalty Wasserstein Generative Adversarial Network
LSTM	Long Short Term Memory
ML	Machine Learning
MLP	Multi-Layer Perceptron
MQLV	Modified Q-Learning for Vasicek
NAG	Nesterov Accelerated Descent
NCG	Non-linear Conjugate Gradient

NN	Neural Networks
PHom	Persistent Homology
RL	Reinforcement Learning
RMSE	Root Mean Square Error
RMSProp	Root Mean Square Propagation
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TD	Tensor Decomposition
VAE	Variational Auto-Encoder
VecHGrad	Vector Hessian Gradient
WAE	Wasserstein Auto-Encoder
WGAN	Wasserstein Generative Adversarial Network

Chapter 1

Introduction

Eleven years after the biggest financial crisis of the twenty-first century, the financial industry is facing a continuously increasing pressure from the European regulators. Either new or revised European regulations are taking effect impacting all services of the banking industry. For instance, the investment banks have to comply with the revised Markets in Financial Instruments Directive [1], MIFID 2, which tries to promote more transparency, more safety and more competition in financial markets. Furthermore, to increase the safety of the financial sector, the International Financial Reporting Standard 9, IFRS9, as adopted by the European Union [2] from the International Accounting Standards Board, IASB, and the third Basel accord [3], Basel III, are now mandatory requirements. The IFRS9 accounting standard forces the banks to make provisions for future expected losses. The Basel III framework is an international framework that promotes the evaluation of risk-weighted assets triggering the amount of equity the banks should have to ensure a reliable capital strength. Directly impacting the retail banking, the revised Payment Service Directive [4], PSD2, aims at promoting more competition in the financial sector of retail banking services in the European Union. Under this directive, the retail banks lost the exclusive privilege of the distribution of payments solution such as credit cards or the confidential knowledge of the clients' financial situation. Financial companies can now challenge traditional retail banks by proposing more attractive financial packages, regarding interest rate loans or insurance policies, to gain new customers. Additionally, any bank can now consult using an application program interface the financial information of any clients from any other banks without limitation. Therefore, a bank having an aggressive marketing strategy can consult the information of the clients of the other banks before prototyping financial packages targeting the clients of the competitors. The banks will

be able to further compete between each other. Straightforwardly, this directive will have a major impact on the business opportunities and the benefits of retail banks.

Meanwhile, the society is going under a significant evolution with a new era of digitalization, contrasting the pace of evolution of the retail banking industry pictured in Figure 1.1. For the past 150 years, the retail banking remained almost identical. The survey [5] highlights that ninety-four percent of the people between eighteen and seventy-five years old have access to a computer while eighty-eight percent have access to a smartphone in thirty-one countries, either developed countries or countries in development. Furthermore, in the US in 2018, Americans spent an average time of more than eleven hours per day in front of a multimedia screen, either a smartphone, a tablet, a computer or a television [6]. It represents a significant rise of thirty minutes per day in comparison to 2017. These eleven hours are spent to watch, read, listen or write multimedia content. These drastic evolution additionally are highly dependent of

Figure 1.1 The retail banking offices and services have remained almost identical for the past 150 years. From left to right: the bank of England in the 19th century, the bank of Montreal at the beginning of the 20th century and the bank of Montreal nowadays.

the generation. For instance, millennials, people born between 1980 and 2000, are avid consumers of mobile technologies. Thirty percent of their multimedia time is spent on a smartphone. This internet generation, which will shape the future of our society, have opinions and spending habits that vary significantly from older generations [7]. Strongly impacted by the 2008 financial crisis, the millenials underline that the retail banking is at its highest risk of disruption, forcing a very conservative sector to reinvent itself. As reported by the Millennial Disruption Index (MDI) in [8], one in three millennials are effectively opened to switch to another bank in the next ninety days. Besides, the four leading worldwide banks, JP Morgan Chase, Bank of America, Citigroup and Wells Fargo, are among the ten least loved brands among millennials. Seventy-three percent would be more excited by a new offering of financial services by Google, Amazon, Apple, Paypal or Square than their national bank, illustrating their sympathy for technology companies. A shocking number of thirty-three percent of millennials even think they do not need a bank at all. These statistics are only the emerged part of the iceberg for the retail banking services. When taking into consideration all adults being more than eighteen years old, the interactions between the clients and the banks have changed significantly for the past few years [9]. Between 2012 and 2018, visiting a retail banking agency went down from few times per month to few times per year while the mobile banking, relying on smartphones and tablets, skyrocketed from few times per year to few times a month. We recall that mobile banking denotes the use of a mobile device, such as a tablet or a smartphone, to connect online to the financial accounts while the online banking uses a computer. The instantaneous access to banking services offered by the digital channels, using online banking and mobile banking, is now more privileged by the clients than the traditional visit to the retail banking agency. This trend is particularly observable in China, especially when considering the mobile payments and the transactions market. For instance, Ant Financial, an affiliate company of the Chinese group Alibaba, and Tencent, a Chinese internet-based company, performed more than 22 trillion US dollars worth of mobile payments in 2018, outpacing the entire world's debit and credit card payments volume [10]. Relying on the latest digital innovations, WeBank, China's first digital bank and affiliated to Tencent, uses a real-time blockchain core allowing unique offers and solutions for transaction payments and marketing campaigns [10]. WeBank has attracted more than 100 million clients since 2014 without the requirement of paper-based applications [10]. The retail banking changes happening currently in China illustrates strongly the future changes of retail banking activities in the next years to come for the rest of

the world. Consequently, we can easily grasp the challenges facing the retail banks due to the evolution of the habits of the society and the further digitalization of services.

In this regulatory and digitalization context, retail banks have to enrich and propose new financial services as well as to reinvent their way of thinking. They can now rely on the new source of information coming from the digital media and digital devices. Furthermore, they can use the latest techniques inherited from computer science, such as machine learning and artificial intelligence, for their survival. More precisely, the retail banks can now benefit and use the full potential of large data sets combining exclusive information of the clients, financial information and economic predictions to create value and gain a competitive advantage. These large data sets are the fuel of the machine learning and artificial intelligence algorithms. To understand better the research problem, we have to explain in more details these generic terms and the techniques related. The expression artificial intelligence (AI) is widely used to describe advanced regressions, machine learning techniques, neural networks, deep learning and reinforcement learning. In an artificial intelligence configuration, a large data set is used with an algorithm that is capable to find rules, patterns within this data set. Although the frontier between each notion is so vague that every computer scientist has his own opinion, we can demystify and shed light among the different techniques included in this novel field.
First, the term machine learning is very often used to describe clustering algorithms such as K-means [11] or K-nearest-neighbor [12], linear algebra factorization such as the Singular Value Decomposition (SVD) [13] or the Tensor Decomposition (TD) [14] used in recommender engines, decision trees [15] or Support Vector Machine (SVM) [16]. What is characteristic of these techniques is that the model fine tunes its parameters based on the data set provided as input instead of relying on hard coded and predefined rules. In most of the applications, the machine learning algorithms are defined as supervised or semi-supervised learning. In supervised learning, all of the features of the data sets are known and each example is associated with a label. In semi-supervised learning, only a fraction of the features is labeled.
Then, a second term widely used related to artificial intelligence is deep learning. Deep learning relies on neural networks [17]. A neural network, or artificial neuron, is an elementary unit receiving a weighted input. A non linear function, or activation function, such as the sigmoid function or the Rectified Linear Unit [18], is used to transform the input signal to an output signal. Neural networks are highly effective to

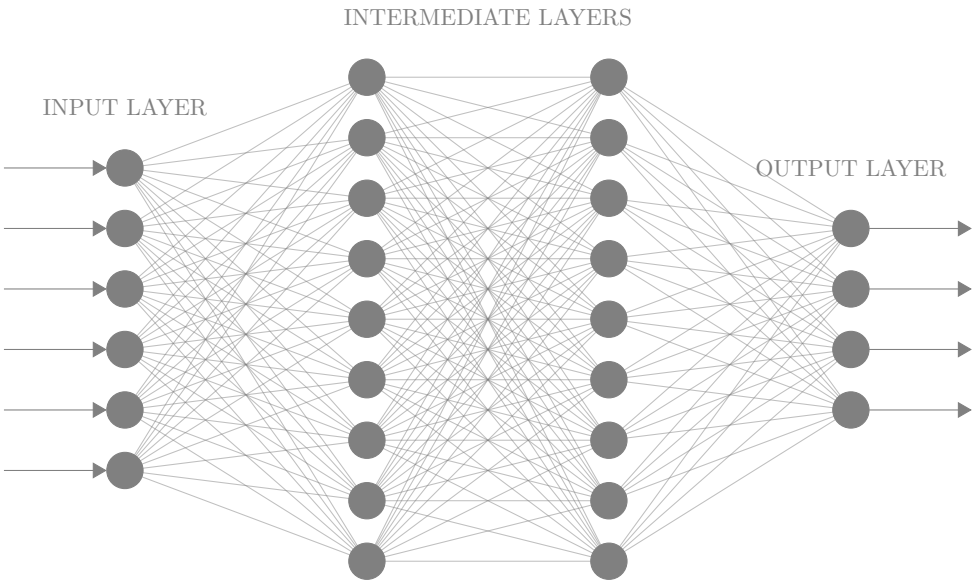

Figure 1.2 In deep learning, a large number of neurons are stacked through different layers to process the data.

solve complex tasks because of the way they can be architected, as illustrated in Figure 1.2. Neural networks can be stacked through different layers where the first layer of neurons receives the input data and the last layer of neurons outputs the processed information contained within the data. The intermediate layer, or the hidden layer, made of at least one layer of neurons, takes care of the signal processing. The collection of neurons and layers allows to increase the problem solving capabilities and often lead to a better accuracy in numerical experiments. Furthermore, the deep learning term denotes a use of a significant number of neurons and layers. Contrary to the technique aforementioned, most neural networks and deep learning applications are classified as unsupervised learning. In an unsupervised learning framework, the algorithms try to infer properties of the data set without knowing any labels.

Another technique that is gaining in popularity among computer scientists is reinforcement learning. In reinforcement learning, the objective is to learn an optimal policy that maximizes a reward based on a value function learned across a set of actions, or decisions. Each of the actions lead to a particular state and, consequently, to a particular reward. For instance, in a game emulator scenario, the optimal policy is learned through a replay experience under which the algorithm learns to play the optimal set of actions to maximize the overall score, as illustrated in Figure 1.3 with the Pacman Atari game. Relying on the Pacman game example, the goal is to eat all the bullets while avoiding the little monsters. With a reinforcement learning approach,

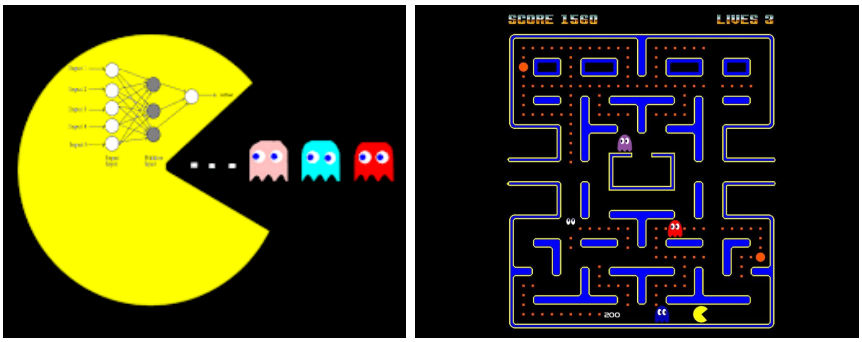

Figure 1.3 Reinforcement learning applied to the Pacman Atari game.

the algorithm learns by itself how to play based on the maximization of the reward score, reaching human level performance. Consequently, the retail banks have now at their disposal a new field of science that they can use to face their upcoming challenges.

Throughout this chapter, we have established various challenges facing the banking industry, and especially, the retail banking industry. Furthermore, we shed some light on artificial intelligence and machine learning with explanation of basic concepts in order to demystify these concepts. The objective of this work is to propose innovative and novel solutions to overcome the technical advances that caused a shift in the demographics of the population impacting the banking industry. Although the problem is large and complex, we concentrate our effort to bring research innovation and open research to retail banking. We rely on the extensive information contained within the bank's data sets in the context of evolving regulation and changing society. More precisely, we explore how to bring enriched clients' experience given limited resources, new habits and new regulations in a time of massive digitalization. Therefore, we chose to structure our approach around the following research problems:

- Accurate and efficient extension of the population size under a strong European banking regulatory framework. Currently, the innovation of the retail banking services and offers is impacted by limited information for some real-world scenarios. For instance, the range of amount for car loan applications is very often incomplete and sparse, influencing the expected risk evaluation. Therefore, there is a need to extend the historical observation samples. By relying on adversarial models and neural networks, it is possible to artificially increase the number of historical samples while keeping the crucial patterns related to existing car loan applications.

- Personalized recommendation through dimension reduction to better categorize clients' needs and clients' wills. The data sets related to the clients' personal and financial information are large and sparse. Additionally, the banks are losing their relation with their clients because of the decrease of the visits in agencies. Consequently, the banks have less information about the center of interest of their clients. Therefore, the aim here is to propose financial products that fit the needs and the interest of a client's group while trying to compress the original information to an interpretable size. Typically, the families having young children might be interested to open a savings account for the children's university studies while a young graduate might not be interested. The approach should highlight the difference of interests by categorizing the population while being humanly understandable.

- Learning through aggregated customer behavior with the determination of the optimal policy for money management with reinforcement learning. There are no institutions that can capitalize as much as the banks on the habits of the spendings of the different groups of population depending on the age, the personal situation or the revenues. Using a model-free approach based on reinforcement learning, it is possible to determine the optimal policy of money management for a category of population. It proposes to answer innovatively to traditional questions such as the limit of a loan amount or the monthly limit on the credit card, while offering complete transparency and being free of any human-bias.

In Chapter 2, we present PHom-GeM, Persistent Homology for Generative Models [19, 20], in the context of financial retail banking transactions. Different generative adversarial models, including Generative Adversarial Networks (GAN), Wasserstein Auto-Encoders (WAE) or Variational Auto-Encoders (VAE), are used with the persistent homology. The latter is a field of mathematics capable to reflect the density of the data points contained in a manifold and to evaluate the shape of the data manifold based on the number of holes. Our proposed method is used to generate synthetic transactions of high quality to increase the number of samples of existing transactions. Such approach is particularly interesting to boost the accuracy of the risk evaluation or of the recommendation predictions. Following the generation of financial transactions, we then propose, in Chapter 3, to perform recommendations of financial products depending on the clients' categorizations. We reduce the dimension of large data sets through an accurate tensor resolution scheme, VecHGrad [21], allowing to categorize

a population through variables that cannot be observed, known as latent variables. Predictions are performed using neural networks on the latent categories of population for two scenarios. We first evaluate the next financial actions of the clients to help the banks to adopt a dynamic approach for the proposal of new products fitting the clients' needs [22]. We then determine the future mobile banking authentication. It enables the bank to estimate which clients are more finance-oriented [23]. In Chapter 4, we discuss a retail banking methodology to establish an open, transparent and explainable decision making process. It targets the decisions made during financial product subscriptions such as the limit on a new credit card or the amount of money granted for a loan. This process should also be highly optimized to reduce the risk of the bank and to maximize the client's satisfaction. Using a model-free reinforcement learning approach in the context of financial transactions [24], we are able to evaluate the optimal policy for money management. It enables to answer the typical question of how and where to fix the amount limit based on a complete transparent approach, free of any-model bias, depending exclusively and solely on the clients' aggregated information. In Chapter 5, we resume the accomplishments of this thesis and we address the main contributions.

Chapter 2

PHom-GeM: Persistent Homology and Generative Models for Retail Banking Transactions

Generative models, including Generative Adversarial Networks (GANs) [25] and Auto-Encoders (AE) [17], have become a cornerstone technique to generate synthetic samples. Generative models learn the data points distribution using unsupervised learning to generate new data points with slight variations. The GANs are among the most popular neural network models to generate adversarial data. A GAN is composed of two parts: a generator that produces synthetic data and a discriminator that discriminates between the generator's output and the true data. The AE are a very popular neural network architecture for adversarial samples generation and for features extraction through order reduction. They are composed of two parts: the encoder which maps the model distribution to a latent manifold and the decoder which maps the latent manifold to a reconstructed distribution. However, AE are known to provoke chaotically scattered data distribution in the latent manifold resulting in an incomplete reconstructed distribution. Furthermore, GANs also tend to provoke chaotically scattered reconstructed distribution during their training. Consequently, generative models can originate incomplete reconstructed distributions and incomplete generated adversarial distributions. Current distance measures fail to address this problem because they are not able to acknowledge the shape of the reconstructed data manifold, i.e. its topological features, and the scale at which the manifold should be analyzed. We propose Persistent Homology [26] for Generative Models, PHom-GeM, a

new methodology to assess and measure the data distribution of a generative model. PHom-GeM minimizes an objective minimization function between the true and the reconstructed distributions and uses persistent homology, the study of the topological features of a space at different spatial resolutions, to compare the nature of the true and the generated distributions. The potential of persistent homology for generative models is highlighted in two related experiments. First, we highlight the AE partial reconstructed distribution provoked by the chaotically scattered data distribution of the latent space. Then, PHom-GEM is applied on adversarial samples synthetically generated by AE and GAN generative models. Both experiments are conducted on a real-world data set particularly challenging for traditional distance measures and generative neural network models. PHom-GeM is the first methodology to propose a topological distance measure, the bottleneck distance, for generative models used to compare adversarial samples of high quality in the context of credit card transactions.

2.1 Motivation

The field of generative models has evolved significantly for the past few years thanks to unsupervised learning and adversarial networks publications. A generative model is capable to learn and approximate any type of data distribution, thanks to neural networks, with the aim to generate new data points with small variations [17]. These models are particularly useful to increase the size of the original data set or to overcome missing data points. In [25], Goodfelow et al. introduced one of the most significant generative neural networks called Generative Adversarial Network (GAN). It is a class of generative models that plays a competitive game between two networks in which the generator network must compete against an adversary according to a game theoretic scenario [17]. The generator network produces samples from a noise distribution and its adversary, the discriminator network, tries to distinguish real samples from generated samples, respectively samples inherited from the training data and samples produced by the generator.

Meanwhile, feature extraction through dimension reduction techniques were initially driven by linear algebra with second order matrix decompositions such as the Singular Value Decomposition (SVD) [27] and, ultimately, with tensor decompositions [28, 14], a higher order analogue of the matrix decompositions. However, due to linear algebra

limitations [29, 30], several architectures for dimension reductions based on neural networks have been proposed. Dense Auto-Encoders (AE) [31, 32] are one of the most well established approaches. An AE is a neural network trained to copy its input manifold to its output manifold through a hidden layer. The encoder function sends the input space to the hidden space and the decoder function brings back the hidden space to the input space. More recently, the Variational Auto-Encoders (VAE) presented by Kingma et al. in [33] constitute a well-known approach but they might generate poor target distribution because of the KL divergence.

Nonetheless, as explained in [34], the points of the hidden space are chaotically scattered for most of the encoders, including the popular VAE. Even after proper training, groups of points of various sizes gather and cluster by themselves randomly in the hidden layer. Some features are therefore missing in the reconstructed distribution $G(Z), Z \in \mathcal{Z}$. Moreover, GANs are subject to a mode collapse during their training for which the competitive game between the discriminator and the generator networks is unfair, leading to convergence issue. Therefore, their training have been known to be very complex and, consequently, limiting their usage especially on large real world data sets. By applying some of the Optimal Transport (OT) concepts gathered in [35] and noticeably, the Wasserstein distance, Arjovsky et al. introduced the Wasserstein GAN (WGAN) in [36] to avoid the mode collapse of GANs and hazardous reconstructed distribution. Gulrajani et al. further optimized the concept in [37] by proposing a Gradient Penalty to Wasserstein GAN (GP-WGAN) capable to generate adversarial samples of higher quality. Similarly, Tolstikhin et al. in [38] applied the same OT concepts to AE and, therefore, introduced Wasserstein AE (WAE), a new type of AE generative model, that avoids the use of the KL divergence.

Nonetheless, the description of the distribution P_G of the generative models, which involves the description of the generated scattered data points [34] based on the distribution P_X of the original manifold $\mathcal{X}$, is very difficult using traditional distance measures, such as the Euclidean distance or the Fréchet Inception Distance [38]. We highlight the distribution and the manifold notations in Figure 2.1 for GAN and in Figure 2.2 for AE. Effectively, traditional distance measures are not able to acknowledge the shapes of the data manifolds and the scale at which the manifolds should be analyzed. However, persistent homology [39, 40] is specifically designed to highlight the topological features of the data [26]. Therefore, building upon the persistent homology,

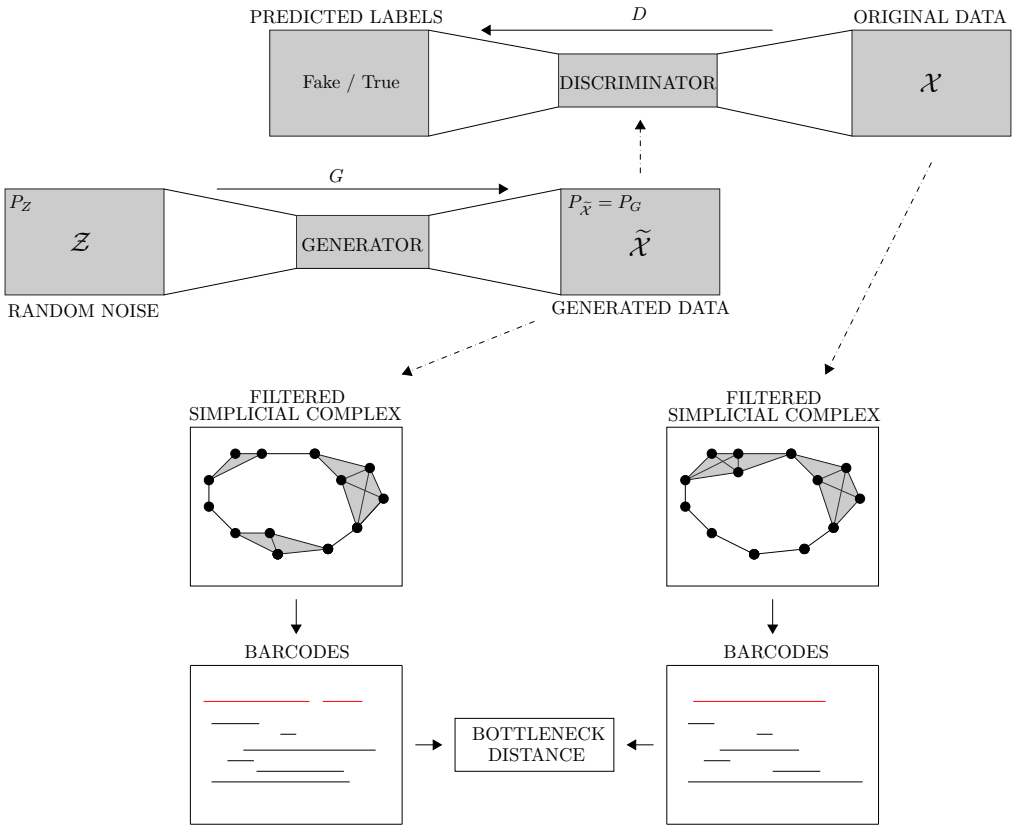

Figure 2.1 In PHom-GeM applied to GAN, the generative model G generates fake samples $\widetilde{X} \in \widetilde{\mathcal{X}}$ based on the samples $Z \in \mathcal{Z}$ from the prior random distribution P_Z. Then, the discriminator model D tries to differentiate the fake samples $\widetilde{X}$ from the true samples $X \in \mathcal{X}$. The original manifold $\mathcal{X}$ and the generated manifold $\widetilde{\mathcal{X}}$ are transformed independently into metric space sets to obtain filtered simplicial complex. It leads to the description of topological features, summarized by the barcodes, to compare the respective topological representation of the true data distribution P_X and the generative model distribution P_G.

the Wasserstein distance [41] and the generative models [38], our main contribution is to propose qualitative and quantitative ways to evaluate the scattered generated distributions and the performance of the generative models.

In this work, we describe the persistent homology features of the generated model G while minimizing the OT function $W_c(P_X, P_G)$ for a squared cost $c(x, y) = ||x - y||_2^2$ where P_X is the model distribution of the data contained in the manifold $\mathcal{X}$, and P_G the distribution of the generative model capable of generating adversarial samples. Our contributions are summarized below:

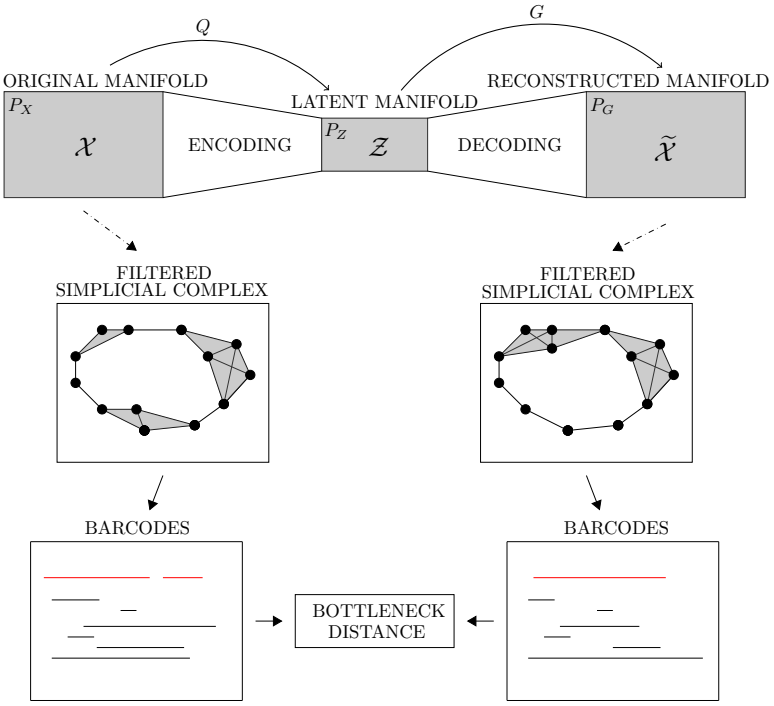

Figure 2.2 In PHom-GeM for AE, the generative model G, the decoder, generates fake samples $\widetilde{X} \in \widetilde{\mathcal{X}}$ based on the samples $Z \in \mathcal{Z}$ from a prior random distribution P_Z. Afterward, the original manifold $\mathcal{X}$ and the generated manifold $\widetilde{\mathcal{X}}$ are both transformed independently into metric space sets to obtain filtered simplicial complex. As for PHom-GeM applied to GAN, it leads to the description of the topological features, summarized by the barcodes, to compare the respective topological representation of the true data distribution P_X and the generative model distribution P_G.

- A persistent homology procedure for generative models, including GP-WGAN, WGAN, WAE and VAE, which we call PHom-GeM to highlight the topological properties of the generated distributions of the data for different spatial resolutions. The objective is a persistent homology description of the generated data distribution P_G following the generative model G.
- The use of PHom-GeM to underline the scattered latent distribution provoked by VAE in comparison to WAE on a real-word data set. We highlight the VAE's hidden layer scattered distribution using the persistent homology, confirming the original statement of VAE's scattered distribution introduced in [17].
- The application of PHom-GeM for the description of the topological properties of the encoded and decoded distribution of the data for different spatial resolutions. The objective is twofold: a persistent homology description of the encoded latent

space $Q_z := \mathbb{E}_{P_X}[Q(Z|X)]$, and a persistent homology description of the decoded latent space following the generative model $P_G(X|Z)$. The latter allows to assess the information loss of the encoding-decoding process.

- A distance measure for persistence diagrams, the bottleneck distance, applied to generative models to compare quantitatively the true and the target distributions on any data set. We measure the shortest distance for which there exists a perfect matching between the points of the two persistence diagrams. A persistence diagram is a stable summary representation of the topological features of simplicial complex, a collection of vertices, associated to the data set.

- Finally, we propose the first application of algebraic topology and generative models on a public data set containing credit card transactions, particularly challenging for this type of models and traditional distance measures while being of high interest for the retail banking industry.

The chapter is structured as follows. We discuss the related work in Section 2.2. In Section 2.3, we review the four main generative model formulations, WGAN, GP-WGAN, WAE and VAE, derived by Arjovsky et al. in [36], Gulrajani et al. in [37], Tolstikhin et al. in [38] and Kingma et al. in [33], respectively. By using the persistent homology, we are able to compare the topological properties of the original distribution P_X and the generated distribution P_G. We additionally compare the latent distribution P_Z of the AE hidden space. We highlight experimental results in Section 2.4. We finally conclude in Section 2.5 by addressing promising directions for future work.

2.2 Related Work

Literature on Persistent Homology and Topology A major trend in modern data analysis is to consider that data has a shape, and more precisely, a topological structure. Topological Data Analysis (TDA) is a set of tools relying on computational algebraic topology to obtain precise information about the data structure. Two of the most important techniques are the persistent homology and the mapper algorithm [26].

Data sets are usually represented as points cloud of Euclidean spaces. The shape of a data set hence depends of the scale at which it is observed. Instead of trying to

find an optimal scale, the persistent homology, a method of TDA, studies the changes of the topological features (number of connected components, number of holes, ...) of a data set depending of the scale. The foundations of the persistent homology have been established in the early 2000s in [39] and [42]. They provide a computable algebraic invariant of filtered topological spaces (nested sequences of topological spaces which encode how the scale changes) called persistence module. This module can be decomposed into a family of intervals called persistence diagram or barcodes. This family records how the topology of the space is changing when going through the filtration [43]. The space of barcodes is measurable through the bottleneck distance. The space of persistence module is moreover endowed with a metric and under a mild assumption these two spaces are isometric [44]. Additionally, the *Mapper* algorithm first appeared in [45]. It is a visualization algorithm which aims to produce a low dimensional representation of high-dimensional data sets in form of a graph, and therefore, capturing the topological features of the points cloud.

Meanwhile, efficient and fast algorithms have emerged to compute the persistent homology [39],[42] as well as to construct filtered topological spaces using, for instance, the Vietoris-Rips complex [46]. It has consequently found numerous successful applications. For instance, Nicolau et al. in [47] detected subgroups of cancers with unique mutational profile. In [48], it has been shown that computational topology could be used in medicine, social sciences or sport analysis. Lately, Bendich et al. improved statistical analyses of brain arteries of a population [49] while Xia et al. were capable of extracting molecular topological fingerprints for proteins in biological science [50].

Literature on Optimal Transport and Generative Models The field of unsupervised learning and generative models has evolved significantly for the past few years [17]. One of the first most popular generative models is auto-encoders. Nonetheless, as outlined by Bengio et al. in [34], the points in the encoded hidden manifold $\mathcal{Z}$ for the majority of the encoders are chaotically scattered. Some features are missing in the reconstructed distribution $G(Z) \in \widetilde{\mathcal{X}}, Z \in \mathcal{Z}$. Thus, sampling data points for the reconstruction with traditional AE is difficult. The added constraint of Variational Auto-Encoder (VAE) in [33] by the mean of a KL divergence, composed of a reconstruction cost and a regularization term, allows to decrease the impact of the chaotic scattered distribution P_Z of $\mathcal{Z}$.

Concurrently to the emergence of AE, Goodfellow et al. introduced Generative Adversarial Network (GAN) in [25] in which two models are simultaneously trained. A generative model, the generator, captures the data distribution and a discriminative model, the discriminator, estimates the probability that a sample comes from the training data rather than the generator [17]. However, in this game theoretic scenario, the training is complex because of the competition between the generator and the discriminator, frequently leading to a mode collapse. The concepts of optimal transport have been brought up to light following the recent work of Villani in [35]. As a solution, optimal transport [35] was therefore applied to GAN in [36] with the Wasserstein distance introducing consequently the Wasserstein GAN (WGAN). By adding a Gradient Penalty (GP) to the Wasserstein distance, Gulrajani et al. in [37] proposed GP-WGAN, a new training for GANs capable of avoiding efficiently the mode collapse. As described in [51, 52] in the context of unbalanced optimal transport, Tolstikhin et al. also applied these concepts to AE in [38]. They proposed to add one extra divergence to the objective minimization function in the context of generative modeling leading to Wasserstein Auto-Encoders (WAE).

Meanwhile, serious efforts have been proposed to increase the efficiency and the accuracy of the weights optimization of neural networks including Stochastic Gradient Descent (SGD) in [53] applied to optimal transport for large-scale experiments [54] and to machine learning [55]. Other gradient descent updates, such as Adagrad [56] or Adam [57], have been introduced as alternatives to the SGD. It has led ultimately to the popular RMSProp [58], widely used for complex training of AE and GANs.

In this chapter, using the persistent homology and the bottleneck distance, we propose qualitative and quantitative ways to evaluate the performance (i) of the generated distribution of GP-WGAN, WGAN, WAE and VAE generative models and (ii) of the encoding decoding process of WAE and VAE. We build upon the work of unsupervised learning and unbalanced optimal transport with the persistent homology. We show that the barcodes, inherited from the persistence diagrams, are capable of representing the adversarial manifold $\widetilde{\mathcal{X}}$ generated by the generative models while the bottleneck distance allows us to compare quantitatively the topological features between the samples $G(Z) \in \widetilde{\mathcal{X}}$ of the generated data distribution P_G and the samples $X \in \mathcal{X}$ of the true data distribution P_X.

2.3 Proposed Method

Our method computes the persistent homology of both the true manifold $X \in \mathcal{X}$ and the generated manifold $\widetilde{X} \in \widetilde{\mathcal{X}}$ following the generative model G based on the minimization of the optimal transport cost $W_c(P_X, P_G)$. In the resulting topological problem, the points of the manifolds are transformed to a metric space set for which a Vietoris-Rips simplicial complex filtration is applied (see definition 2). PHom-GeM achieves simultaneously two main goals: it computes the birth-death of the pairing generators of the iterated inclusions while measuring the bottleneck distance between the persistence diagrams of the manifolds of the generative models.

2.3.1 Preliminaries and Notations

We follow the notations introduced in [37] and [38]. Sets and manifolds are denoted by calligraphic letters such as $\mathcal{X}$, random variables by uppercase letters X, and their values by lowercase letters x. Probability distributions are denoted by uppercase letters $P(X)$ and their corresponding densities by lowercase letters $p(x)$.

2.3.2 Optimal Transport and Dual Formulation

Following the description of the optimal transport problem [35], the study of the optimal allocation of resources, and relying on the Kantorovich-Rubinstein duality, the Wasserstein distance is computed as

$$W_c(P_X, P_G) = \sup_{f \in \mathcal{F}_L} \mathbb{E}_{X \sim P_X}[f(X)] - \mathbb{E}_{Y \sim P_G}[f(Y)] \quad , \tag{2.1}$$

where $(\mathcal{X}, d)$ is a metric space, $\mathcal{P}(X \sim P_X, Y \sim P_G)$ is a set of all joint distributions (X, Y) with marginals P_X and P_G respectively and $\mathcal{F}_L$ is the class of all bounded 1-Lipschitz functions on $(\mathcal{X}, d)$.

2.3.3 Wasserstein GAN (WGAN)

The objective of the Wasserstein minimization function applied to GAN is twofold: (i) achieve appropriate calibration of the generator and the discriminator networks thanks to the transporting mass cost function while (ii) avoiding the mode collapse of

standard GANs [17]. By using the notations and the concept introduced in [36], the WGAN objective loss function is defined by

$$W_c(\mathbb{P}_r, \mathbb{P}_\theta) = \max_{w \in \mathcal{W}} \mathbb{E}_{x \sim \mathbb{P}_r}[f_w(x)] - \mathbb{E}_{x \sim p(z)}[f_w(g_\theta(z))] \quad , \tag{2.2}$$

where $\mathbb{P}_r$ denotes the real data distribution, $\mathbb{P}_\theta$ the generative distribution of the function $g : \mathcal{Z} \times \mathbb{R}^d \to \mathcal{X}$, denoted $g_\theta(Z)$, with Z a random variable (e.g Gaussian) over another space $\mathcal{Z}$, $\{f_w\}_{w \in \mathcal{W}}$ the parameterized family of functions, all K-Lipschitz for some K, and $w \in \mathcal{W}$ are the weights w contained in the compact space $\mathcal{W}$.

2.3.4 Gradient Penalty Wasserstein GAN (GP-WGAN)

The Gradient Penalty Wasserstein GAN (GP-WGAN) [37] proposes to solve partial issues that have been left opened in [36] by adding a gradient penalty to the loss function of WGAN. The advantages of using a gradient penalty are twofolds: (i) reducing the WGAN's training error to improve the quality of the generated adversarial samples by backpropagating gradient information while (ii) avoiding clipping the weights during the WGAN's training. Therefore, the GP-WGAN objective loss function with gradient penalty is expressed such that

$$L = \mathop{\mathbb{E}}_{\widetilde{X} \sim P_G}[f(\widetilde{X})] - \mathop{\mathbb{E}}_{X \sim P_X}[f(X)] + \lambda \mathop{\mathbb{E}}_{\widehat{X} \sim P_{\widehat{X}}}[(||\nabla_{\widehat{X}} f(\widehat{X})||_2 - 1)^2] \quad , \tag{2.3}$$

where f is the set of 1-Lipschitz functions on $(\mathcal{X}, d)$, P_X the original data distribution, P_G the generative model distribution implicitly defined by $\widetilde{X} = G(Z), Z \sim p(Z)$. The input Z to the generator is sampled from a noise distribution such as a uniform distribution. $P_{\widehat{X}}$ defines the uniform sampling along straight lines between pairs of points sampled from the data distribution P_X and the generative distribution P_G. A penalty on the gradient norm is enforced for random samples $\widehat{X} \sim P_{\widehat{X}}$. For further details, we refer to [36] and [37].

2.3.5 Wasserstein Auto-Encoders

As described in [38], the WAE objective function is expressed such that

$$D_{\text{WAE}}(P_X, P_G) := \inf_{Q(Z|X) \in \mathcal{Q}} \mathbb{E}_{P_X} \mathbb{E}_{Q(Z|X)}[c(X, G(Z))] + \lambda \mathcal{D}_Z(Q_Z, P_Z) \quad , \tag{2.4}$$

where $c(X, G(Z)) : \mathcal{X} \times \mathcal{X} \to \mathcal{R}_+$ is any measurable cost function. In our experiments, we use a square cost function $c(x, y) = ||x - y||_2^2$ for data points $x, y \in \mathcal{X}$. $G(Z)$ denotes the sending of Z to X for a given map $G : \mathcal{Z} \to \mathcal{X}$. Q, and G, are any nonparametric set of probabilistic encoders, and decoders respectively.

We use the Maximum Mean Discrepancy (MMD) for the penalty $\mathcal{D}_Z(Q_Z, P_Z)$ for a positive-definite reproducing kernel $k : \mathcal{Z} \times \mathcal{Z} \to \mathcal{R}$

$$\begin{aligned} \mathcal{D}_Z(P_Z, Q_Z) &:= \text{MMD}_k(P_Z, Q_Z) \\ \mathcal{D}_Z(P_Z, Q_Z) &= ||\int_{\mathcal{Z}} k(z, .)dP_Z(z) - \int_{\mathcal{Z}} k(z, .)dQ_Z(z)||_{\mathcal{H}_k} \quad , \end{aligned} \tag{2.5}$$

where $\mathcal{H}_k$ is the reproducing kernel Hilbert space of real-valued functions mapping on $\mathcal{Z}$. For details on the MMD implementation, we refer to [38].

2.3.6 Variational Auto-Encoders

Variational Auto-Encoder (VAE) [33] were designed to combine approximate inference and probabilistic models for efficient learning of latent variables within a data set. In the VAE framework, the input x is mapped probabilistically to the latent variable z by the encoder q_ϕ and the latent variable z to x by the decoder p_θ such as $x \xrightarrow{q_\phi(x|z)} z \xrightarrow{p_\theta(z|x)} x$. The parameters ϕ and θ are used to parameterize $q(x|z)$ and $p(z|x)$ respectively. The model has to learn the generative process $p(x|z)$ for the optimization problem $\max_\theta \int_z p(z)p_\theta(x|z)$. Because of the assumption that the probability distribution are all Gaussian, the VAE loss function is defined such that

$$\begin{aligned} \mathcal{L}(\theta, \phi, \{x^i\}_{i=1}^M) &= \frac{N}{M} \sum_{i=1}^{M} \mathcal{L}(\theta, \phi, x^{(i)}) \\ \mathcal{L}(\theta, \phi, \{x^i\}_{i=1}^M) &= \frac{N}{M} \sum_{i=1}^{M} \Big\{ \underbrace{\frac{1}{L} \sum_{l=1}^{L} \log p_\theta(x^i, z^{(l)})}_{\text{reconstruction loss}} \\ &\qquad + \underbrace{\frac{1}{2} \sum_{d=1}^{D} \Big(1 + \log_{\phi,d}(x^i) - \mu_{\phi,d}^2(x^i) - \sigma_{\phi,d}^2(x^i)\Big)}_{\text{KL divergence}} \Big\} \end{aligned} \quad , \tag{2.6}$$

where the vector z is determined by $z = g_\phi(\epsilon, x) = \mu_\phi(x) + \epsilon \odot \sigma_\phi(x)$ such that z will have the distribution $q_\phi(z|x) \sim \mathcal{N}(z, \mu_\phi(x), \sigma_\phi(x))$ with $\odot$ the elementwise multiplication. The terms μ_ϕ and σ_ϕ denote the mapping from x to the mean and standard deviation vectors, respectively. Finally, N is the number of data points contained in the data set and M is the number of data points contained in the random sample.

2.3.7 Persistence Diagram and Vietoris-Rips Complex

We explain the construction of the persistence module associated to a sample of a fixed distribution on a space. First, two manifold distributions are sampled from the generative models' training. Then, we construct the persistence modules associated to each sample of the points manifolds. We refer to the first subsection of section 2.2 for pointed reference on the persistent homology.

We first associate to our points manifold $\mathcal{C} \subset \mathbb{R}^n$, considered as a finite metric space, a sequence of simplicial complexes. For that aim, we use Vietoris-Rips complex.

Definition 1 Let $V = \{1, \cdots, |V|\}$ be a set of vertices. A simplex σ is a subset of vertices $\sigma \subseteq V$. A simplicial complex K on V is a collection of simplices $\{\sigma\}$, $\sigma \subseteq V$, such that $\tau \subseteq \sigma \in K \Rightarrow \tau \in K$. The dimension $n = |\sigma| - 1$ of σ is its number of elements minus 1. Simplicial complexes examples are represented in Figure 2.3.

Definition 2 Let (X, d) be a metric space. The Vietoris-Rips complex $\mathrm{VR}(X, \epsilon)$ at scale ϵ associated to X is the abstract simplicial complex whose vertex set is X, and where $\{x_0, x_1, ..., x_k\}$ is a k-simplex if and only if $d(x_i, x_j) \leq \epsilon$ for all $0 \leq i, j \leq k$.

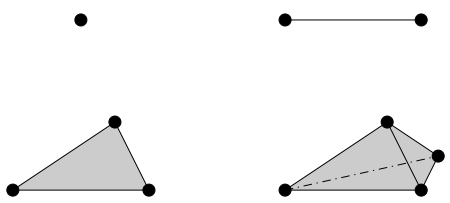

Figure 2.3 A simplical complex is a collection of numerous “simplex" or simplices, where a 0-simplex is a point, a 1-simplex is a line segment, a 2-simplex is a triangle and a 3-simplex is a tetrahedron.

We obtain an increasing sequence of Vietoris-Rips complex by considering the $\mathrm{VR}(\mathcal{C}, \epsilon)$ for an increasing sequence $(\epsilon_i)_{1 \leq i \leq N}$ of value of the scale parameter ϵ

$$\mathcal{K}_1 \overset{\iota}{\hookrightarrow} \mathcal{K}_2 \overset{\iota}{\hookrightarrow} \mathcal{K}_3 \overset{\iota}{\hookrightarrow} ... \overset{\iota}{\hookrightarrow} \mathcal{K}_{N-1} \overset{\iota}{\hookrightarrow} \mathcal{K}_N. \tag{2.7}$$

Applying the *k-th* singular homology functor $H_k(-, F)$ with coefficient in the field F [59] to the equation (2.7), we obtain a sequence of F-vector spaces, called the *k-th* persistence module of $(\mathcal{K}_i)_{1 \leq i \leq N}$

$$H_k(\mathcal{K}_1, F) \xrightarrow{t_1} H_k(\mathcal{K}_2, F) \xrightarrow{t_2} \cdots \xrightarrow{t_{N-2}} H_k(\mathcal{K}_{N-1}, F) \xrightarrow{t_{N-1}} H_k(\mathcal{K}_N, F). \tag{2.8}$$

Definition 3 $\forall\ i < j$, the *(i,j)*-persistent k-homology group with coefficient in F of $\mathcal{K} = (\mathcal{K}_i)_{1 \leq i \leq N}$ denoted $HP_k^{i \to j}(\mathcal{K}, F)$ is defined to be the image of the homomorphism $t_{j-1} \circ \ldots \circ t_i : H_k(\mathcal{K}_i, F) \to H_k(\mathcal{K}_j, F)$.

Using the interval decomposition theorem [60], we extract a finite family of intervals of $\mathbb{R}_+$ called persistence diagram. Each interval can be considered as a point in the set $D = \left\{(x, y) \in \mathbb{R}_+^2 | x \leq y\right\}$. Hence, we obtain a finite subset of the set D. This space of finite subsets is endowed with a matching distance called the bottleneck distance and defined as follow

$$d_b(A, B) = \inf_{\phi : A' \to B'} \sup_{x \in A'} \|x - \phi(x)\| \quad ,$$

where $A' = A \cup \Delta$, $B' = B \cup \Delta$, $\Delta = \{(x, y) \in \mathbb{R}_+^2 | x = y\}$ and the inf is over all the bijections from A' to B'.

2.3.8 Practical Overview of Persistence Diagrams, Barcodes and Homology Groups

Before describing our contribution PHom-GeM, we give a more practical overview of the concepts related to the filtration parameter, the barcodes, the persistence diagrams and the homology groups for the ease of understanding. Consequently, we emphasize the explanation of the key concepts without any formal formulation.

Filtration Parameter, Barcodes and Persistence Diagrams

The first key notion for the construction of barcodes and persistence diagrams, used to highlight the persistent homology features of the data points cloud, is the filtration parameter denoted by ε. First, for every data point, the size of the points is continuously and artificially increased. Therefore, the size of the points grows as they become geometrical disks. We illustrate the concept in Figure 2.4. Through the continuous process, the filtration parameter ε grows. When two disks intersect, a line is drawn between the two corresponding original data points, creating a connected component defined as a 1-simplex. As the filtration parameter keeps growing, more connected components are created as well as triangles defined as 2-simplex.

By relying on this filtration parameter ε, we can construct barcodes and persistence diagrams. We recall that a barcode diagram is a stable summary representation of a persistence diagram. We highlight the construction of the barcodes and the persistence diagram in Figure 2.5. Given a two-dimensional function, we capture the points for which there is a local extrema. The filtration starts from the bottom to reach the top. Every time a minima is observed, a barcode is created. Then, as the filtration evolves, local maxima are observed provoking the "death" of the barcode. Every "birth-death" episode can be then reported in the persistence diagram offering a slightly different visualization than the barcode diagram. Additionally, the persistence diagram can be later used to further describe the topological properties of the original data points cloud using quantitative measures, such as the bottleneck distance aforementioned.

Betti Numbers and Homology Groups

The aim of the persistent homology is to describe the shape of the data points cloud by relying on features such as connected components, loops or cavities, independent of any distance measurement. To this end, the set of features contained in the data manifold are categorized into different homological dimensions, or homology groups. We illustrate the different homology groups in Figure 2.6 for the first three homology groups. The first homology group, denoted by H_0, is used to characterize the connected components. The second homology group, H_1, defines the loops or the circles. Finally, the third homology group, H_2, designates the voids, the cavities or the spheres. We invite the reader to [59] for more details.

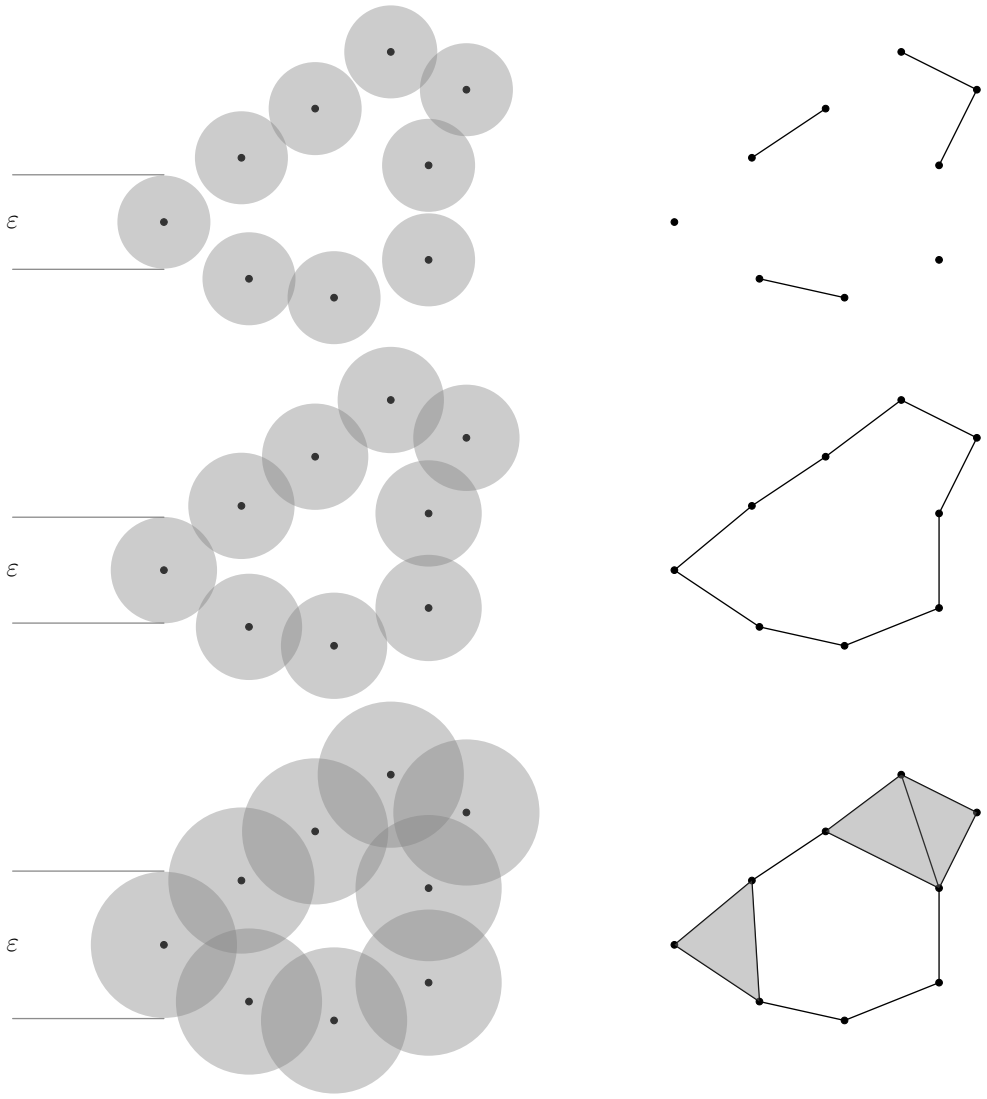

Figure 2.4 The filtration parameter ε grows around each data point continuously leading to the creation of geometrical disks. When two disks intersect, a line is drawn between the two corresponding data points resulting in a connected component defined as a 1-simplex. Triangles, defined as 2-simplex, are generated as the filtration parameter ε keeps growing.

To complete the description of the topological features of the data points cloud, the homology groups are completed with the Betti numbers. The Betti numbers are used to measure the holes of the data point cloud for each of the homology groups [40, 61]. They more precisely reflect the topological properties of a shape as being the number of $i-$dimensional holes in a simplical complex [62]. We recall the illustration of the simplicial complex in Figure 2.3. Relying on Figure 2.6, we can describe the objects with the first three Betti numbers. The connected components, highlighting

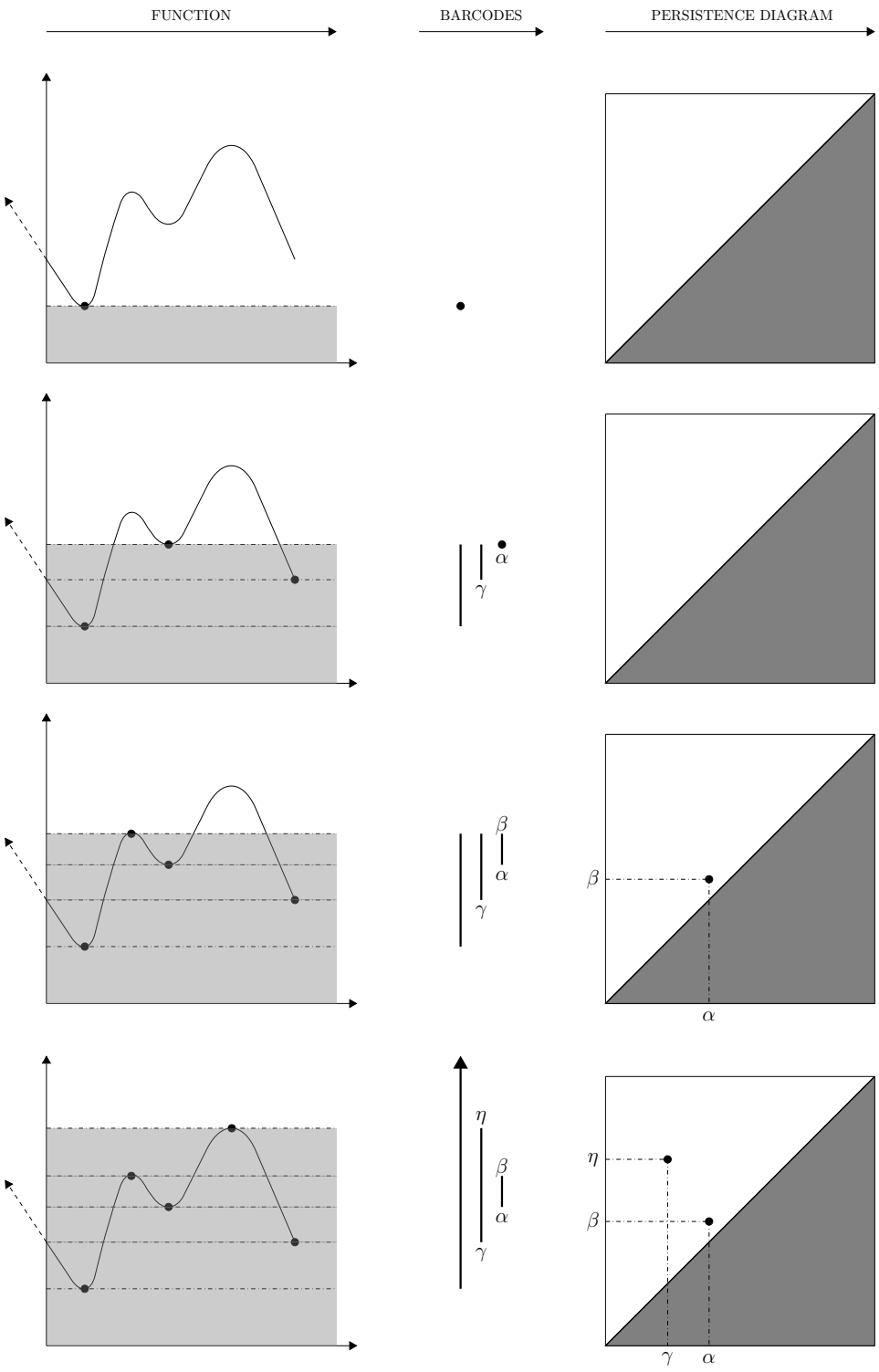

Figure 2.5 As the filtration evolves, the local minima of the function provoke the creation of a barcode while the local maxima induce the death of the barcode. Every birth-death cycle of each barcode can be represented in the persistence diagram, allowing the description of further persistent homology features.

the homology group H_0, are described with the Betti numbers $\beta_0 = 1, \beta_1 = 0, \beta_2 = 0$. Then, the circle representing the homology group H_1 has the following Betti numbers $\beta_0 = 1, \beta_1 = 1, \beta_2 = 0$. Finally, the sphere illustrating the homology group H_2 has the Betti numbers $\beta_0 = 1, \beta_1 = 0, \beta_2 = 1$.

Combining the Filtration Parameter, Homology Groups and Barcodes

Following the practical description of the key concepts of the persistent homology, we emphasize how these concepts, and noticeably the filtration parameter, the simplex, the homology groups and the barcodes, relate to each other in Figures 2.7 and 2.8. Given four original data points representing a rectangle, the continuous growth of the filtration parameter ε leads to the computation of various 1-simplex. Vertices, edges and faces are effectively continuously generated [63]. The first two homology groups, H_0 and H_1, are represented. These groups, as aforementioned, describe the connected components and the loops, respectively. The different snapshots of Figure 2.7 capture the relation between the barcodes, the homology groups and the filtration parameter with respect to the original data points and the growth of the filtration parameter ε. At the end of the filtration procedure, a persistence diagram is drawn to recapitulate the birth-death events observed with the barcodes, as shown in Figure 2.8. The persistence diagram can be later used to describe the topological properties of the original data points cloud using quantitative measures, such as the bottleneck distance.

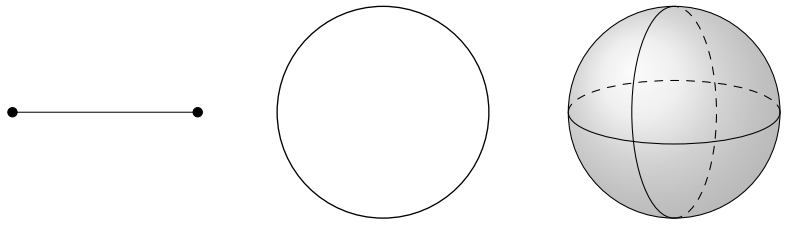

Figure 2.6 Visualization of the first three homology groups. The line, or the connected components, belongs to the first homology group H_0. The second homology group H_1 represents the circles or the loops. Finally, the homology group H_2 describes the voids, the cavities or the spheres.

2.3.9 Proposed Method: PHom-GeM, Persistent Homology for Generative Models

Bridging the gap between the persistent homology and the generative models, including GP-WGAN, WGAN, WAE and VAE, PHom-GeM uses a two-steps procedure. First, the minimization problem is solved for the generator G and the discriminator D when considering GP-WGAN and WGAN. The gradient penalty λ in equation (2.3) is fixed equal to 10 for GP-WGAN and to 0 for WGAN. For auto-encoders, the minimization problem is solved for the encoder Q and the decoder G. We use RMSProp optimizer [58] for the optimization procedure. Then, the samples of the original and generated distributions, P_X and P_G, are mapped to the persistent homology for the description of

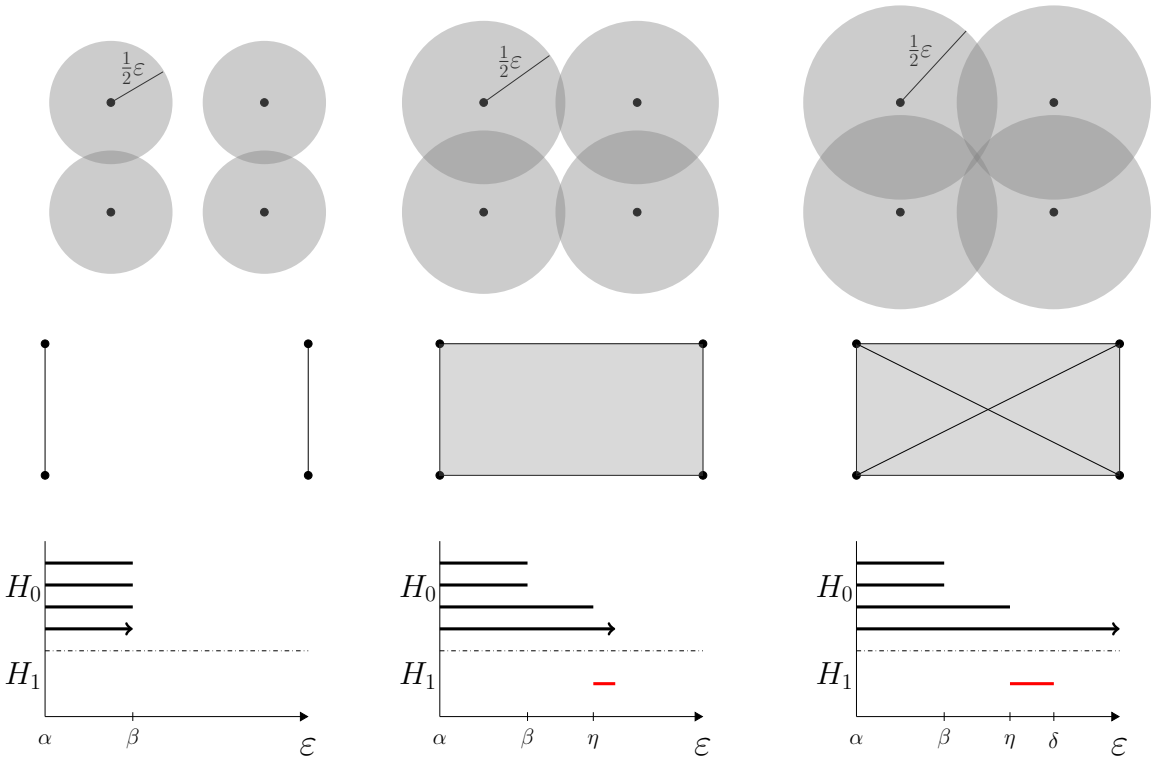

Figure 2.7 Representation of the filtration parameter ε with the homology groups H_0 and H_1 for data points inherited from a rectangle. As the filtration parameter ε progresses, the persistent homology features are highlighted with respect to each homology groups. The homology group H_0 captures the connected components and the homology group H_1 the loops. The birth-death episodes of the persistent homology features are summarized by the barcodes.

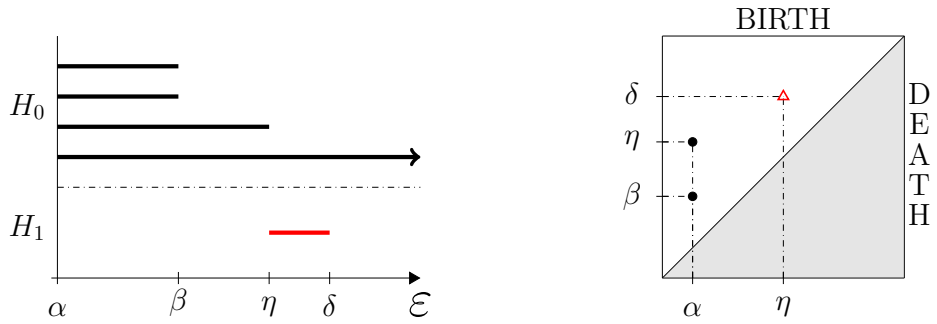

Figure 2.8 The filtration procedure leads to the construction of the barcodes of the homological groups H_0 and H_1, a stable summary representation of the persistence diagrams. The groups H_0 and H_1 are represented by black dots and red triangles, respectively, in the persistence diagram.

their respective manifolds. The points contained in the manifold $\mathcal{X}$ inherited from P_X and the points contained in the manifold $\tilde{\mathcal{X}}$ generated with P_G are randomly selected into respective batches. Two samples, Y_1 from $\mathcal{X}$ following P_X and Y_2 from $\tilde{\mathcal{X}}$ following P_G, are selected to differentiate the topological features of the original manifold $\mathcal{X}$ and the generated manifold $\tilde{\mathcal{X}}$. The samples Y_1 and Y_2 are contained in the spaces $\mathcal{Y}_1$ and $\mathcal{Y}_2$, respectively. The spaces $\mathcal{Y}_1$ and $\mathcal{Y}_2$ are then transformed into metric space sets $\widehat{\mathcal{Y}}_1$ and $\widehat{\mathcal{Y}}_2$ for computational purposes. We subsequently filter the metric space sets $\widehat{\mathcal{Y}}_1$ and $\widehat{\mathcal{Y}}_2$ using the Vietoris-Rips simplicial complex filtration. Given a line segment of length ϵ, vertices between data points are created for data points respectively separated from a smaller distance than ϵ. It leads to the construction of a collection of simplices resulting in Vietoris-Rips simplicial complex $\mathrm{VR}(\mathcal{C}, \epsilon)$ filtration. In our case, we decide to use the Vietoris-Rips simplicial complex as it offers the best compromise between the filtration accuracy and the memory requirement [26]. Subsequently, the persistence diagrams, dgm_{Y_1} and dgm_{Y_2}, are constructed. We recall a persistence diagram is a stable summary representation of the topological features of simplicial complex. The persistence diagrams allow the computation of the bottleneck distance $d_b(\mathrm{dgm}_{Y_1}, \mathrm{dgm}_{Y_2})$ for quantitative measurements of the similarities, as illustrated in Figure 2.9. Finally, the barcodes represent in a simple way the birth-death of the pairing generators of the iterated inclusions detected by the persistence diagrams. We summarized the persistent homology procedure for generative models Algorithm 1.

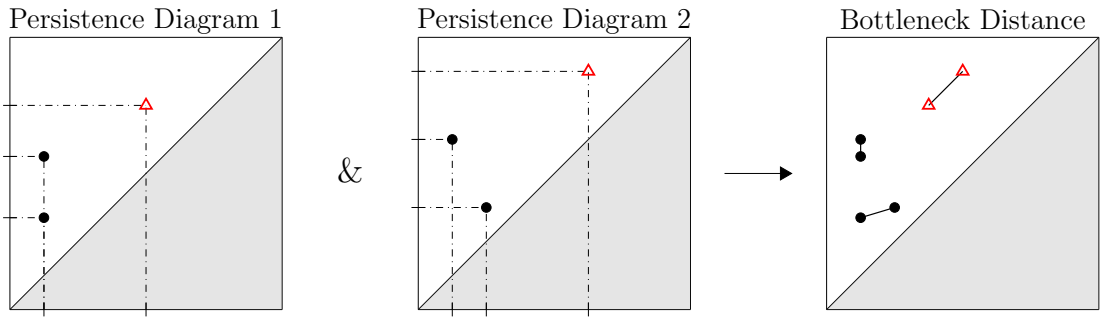

Figure 2.9 The bottleneck distance is a measure of similarity between two persistence diagrams. The points of the first and the second persistence diagrams are gathered on one persistence diagram. The distance between each of the points is measured. The bottleneck distance is computed such that it is the shortest distance for which any couple of matched points are at distance at most b.

Algorithm 1: Persistent Homology for Generative Models

Data: training and validation sets, hyperparameter λ

Result: persistent homology description of generative manifolds

begin

  */*Step 1: Generative Models Resolution*/*

  Select samples $\{x_1, ..., x_n\}$ from training set

  Select samples $\{z_1, ..., z_n\}$ from validation set

  With RMSProp gradient descent update ($\text{lr} = 0.001, \rho = 0.9, \epsilon = 10^{-6}$), optimize until convergence Q and G

    case *GP-WGAN and WGAN:* using equation 2.3

    case *WAE:* using equation 2.4

    case *VAE:* using equation 2.6

  */*Step 2: Persistence Diagram and Bottleneck Distance on manifolds of generative models*/*

  Random selection of samples $Y_1 \in \mathcal{Y}_1, Y_2 \in \mathcal{Y}_2$ from P_X and P_G

  Transform $\mathcal{Y}_1$ and $\mathcal{Y}_2$ spaces into a metric space set

  Filter the metric space set with a Vietoris-Rips simplicial complex $\text{VR}(\mathcal{C}, \epsilon)$

  Compute the persistence diagrams dgm_{Y_1} and dgm_{Y_2}

  Evaluate the bottleneck distance $d_b(\text{dgm}_{Y_1}, \text{dgm}_{Y_2})$

  Build the barcodes with respect to Y_1 and Y_2

return

2.4 Experiments

We assess on a highly challenging data set for generative models whether PHom-GeM can simultaneously achieve (i) accurate persistent homology distance measurement with the bottleneck distance, (ii) detection of the AE latent manifold scattered distribution P_Z, (iii) accurate topological reconstruction of the data points, and (iv) precise persistent homology description of the generated data points.

Data Availability and Data Description We train PHom-GeM on one real-world open data set: the credit card transactions data set from the Kaggle database[1] containing 284 807 transactions including 492 frauds. This data set is particularly interesting because it reflects the scattered points distribution of the reconstructed manifold that are obtained during generative models' training, impacting afterward the generated adversarial samples. This data set furthermore is challenging because of the strong imbalance between normal and fraudulent transactions while being of high interest for the banking industry. To preserve the transactions confidentiality, each transaction is composed of 28 components obtained with PCA without any description and two additional features *Time* and *Amount* that remained unchanged. Each transaction is labeled as fraudulent or normal in a feature called *Class* which takes a value of 1 in case of a fraudulent transaction and 0 otherwise.

Experimental Setup and Code Availability In our experiments, we use the Euclidean latent space $\mathcal{Z} = \mathcal{R}^2$ and the square cost function c previously defined as $c(x, y) = ||x - y||_2^2$ for the data points $x \in \mathcal{X}, \widetilde{x} \in \widetilde{\mathcal{X}}$. The dimensions of the true data set is $\mathcal{R}^{29}$. We kept the 28 components obtained with PCA and the amount resulting in a space of dimension 29. For the error minimization process, we used RMSProp gradient descent [58] with the parameters $\text{lr} = 0.001, \rho = 0.9, \epsilon = 10^{-6}$ and a batch size of 64. Different values of λ for the gradient penalty have been tested. We empirically obtained the lowest error reconstruction with $\lambda = 10$ for both GP-WGAN and WAE, as summarized in Table 2.1. The coefficients of the persistent homology are evaluated within the field $\mathbb{Z}/2\mathbb{Z}$. We only consider homology groups H_0 and H_1 who represent the connected components and the loops, respectively. Higher dimensional homology groups did not noticeably improve the results quality while leading to longer computational time. The simulations were performed on a computer with 16GB of

[1]The data set is available at https://www.kaggle.com/mlg-ulb/creditcardfraud.

Table 2.1 Root Mean Square Error (RMSE) after PHom-GeM gradient penalty calibration on GP-WGAN and WAE between the predicted samples and the original samples of the model (smaller is better).

Gradient Penalty	RMSE of GP-WGAN	RMSE of WAE
0.1	0.159	1.403
0.5	0.162	0.846
1.0	0.158	0.597
3.0	0.158	0.532
5.0	0.155	0.385
10.0	**0.153**	**0.376**
15.0	0.155	0.381
20.0	0.158	0.385

RAM, Intel i7 CPU and a Tesla K80 GPU accelerator. To ensure the reproducibility of the experiments, the code is available on GitHub [64].

Results and Discussions about PHom-GeM Because of the strong impact of the AE scattered distribution P_Z on the quality of the encoding-decoding process, we first assess for AE, and noticeably for VAE and WAE, if PHom-GeM, Persistent Homology for Generative Models, can measure the scattered distribution P_Z using the bottleneck distance. We then observe the consequence of the encoding process on the AE reconstructed distribution. Finally, in a third experiment, we reach the core of our contribution that is the use of the persistent homology on complex generated adversarial samples. We test PHom-GeM on the four different generative models: GP-WGAN, WGAN, WAE and VAE. For the second and third experiments, we compare the performance of PHom-GeM on two specificities: first, qualitative visualization of the persistence diagrams and barcodes and, secondly, quantitative estimation of the persistent homology closeness using the bottleneck distance between the generated manifolds $\widetilde{\mathcal{X}}$ of the generative models and the original manifold $\mathcal{X}$.

2.4.1 Detection of the Scattered Distribution of the Samples Z of the Latent Manifold $\mathcal{Z}$ for AE

PHom-GeM is capable to measure the persistent homological features, for intance the number of connected components or the number of holes, between two samples of a

distribution. Hereinafter, we use the persistent homological distance, the bottleneck distance, to measure and compare how much the WAE's and VAE's distributions P_Z of the latent manifold $\mathcal{Z}$ is scattered from a persistent homology perspective. The strength of the bottleneck distance is to measure quantitatively the topological changes in the data, either the true or the reconstructed data, while being insensitive to the scale of the analysis. Using the bootstrapping technique of [65], we successively randomly select data samples Z_i of the manifold $\mathcal{Z}$. The total number of selected samples Z_i is at least 85% of the total number of points contained in the manifold $\mathcal{Z}$ for a reliable statistical representation. Assuming the data is not scattered, the bottleneck distance between the samples Z_i is small. On the opposite, if the data is chaotically scattered in $\mathcal{Z}$, then the topological features between the samples Z_i are significantly different, and consequently, the bottleneck distance is large. We illustrate the idea in Figure 2.10.

In Table 2.2, the bottleneck distance is significantly lower for WAE than for VAE. The level of scattered chaos observed for WAE is lower than for VAE, a direct consequence of the use of optimal transport in WAE. It also means the distribution P_Z of the latent manifold $\mathcal{Z}$ is better topologically preserved for WAE than for VAE. Thus, we can expect that the reconstructed distribution $P_G(X|Z)$ of $\mathcal{X}$ is less altered for WAE than for VAE.

2.4.2 Reconstructed distribution P_Z of the AE

Following the results on the persistent homological features of the latent manifold $\mathcal{Z}$ of the WAE and the VAE, we assess in this section the impact of the quality of the distribution P_Z on the reconstructed distribution. Effectively, before generating adversarial samples, we want to evaluate how much the WAE and the VAE are capable to reproduce at their output the data injected as input.

We represent the persistence diagrams and the barcode diagrams between the original and the reconstructed distributions, respectively P_X and $P_G(X|Z)$, of the manifold $\mathcal{X}$ in Figures 2.11 and 2.12. We can notice that the original and the reconstructed distributions are more widely distributed for WAE than for VAE. Additionally, the persistence diagram and the barcode diagram of WAE are qualitatively closer to those associated with the original data manifold $\mathcal{X}$. It means the topological features are better preserved for WAE than for VAE. Such result was expected based on the observation

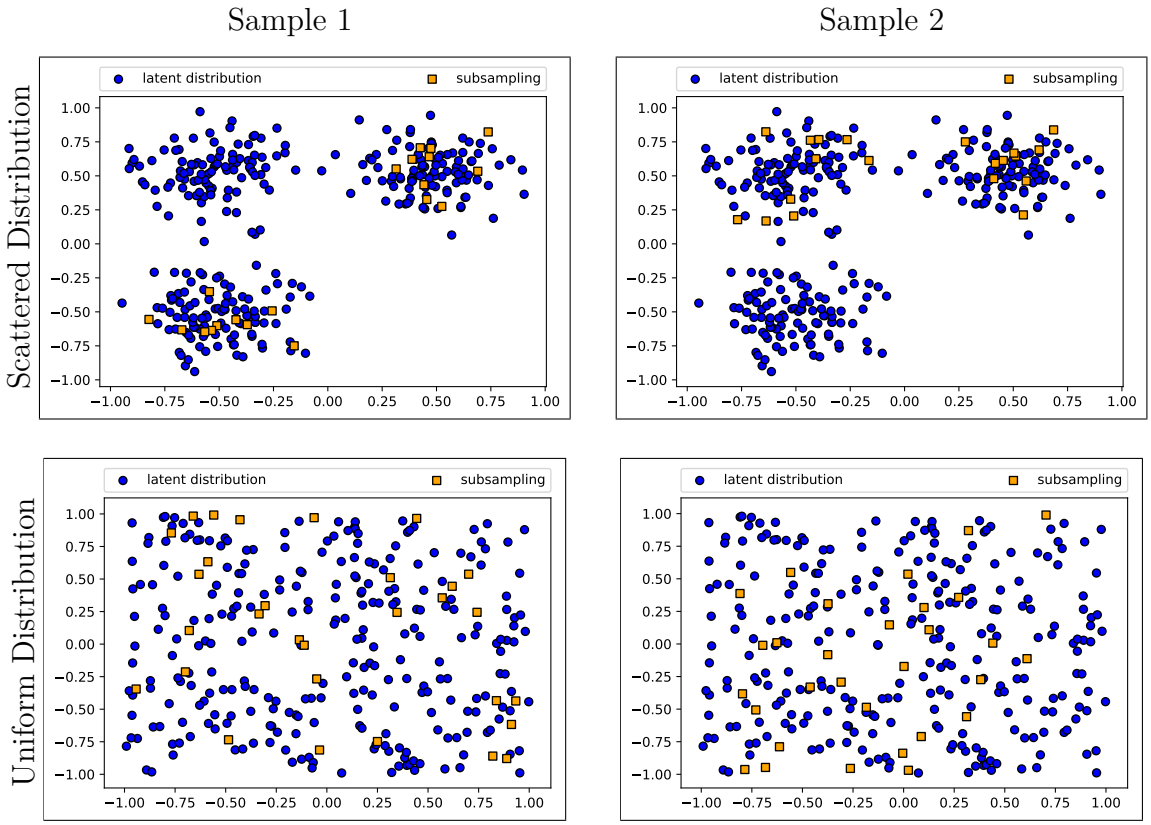

Figure 2.10 By using the bootstrapping technique of [65], we can randomly select samples of the latent manifold $\mathcal{Z}$ and then, measure and aggregate the bottleneck distance between the samples. In case of a scattered distribution, there is a strong variation between the randomly selected samples of the persistent homological properties, such as the distance between the points or the number of holes between the connected components. Therefore, a scattered distribution leads to a bigger bottleneck distance than a uniform distribution.

Table 2.2 Bottleneck distance (smaller is better) for PHom-GeM applied to WAE and PHom-GeM applied to VAE between the samples Z_i of the latent manifold $\mathcal{Z}$ following P_Z to detect the scattered distribution. The WAE better preserves the topological features during the encoding than the VAE resulting in a manifold $\mathcal{Z}$ less chaotically scattered, highlighted by the smaller bottleneck distance.

PHom-GeM on WAE	PHom-GeM on VAE	Difference (%)
0.0984	0.1372	28.28

of the scattered distribution in Subsection 2.4.1. Consequently, it highlights a better encoding-decoding process thanks to the use of an optimal transport cost function in the case of WAE. Furthermore, in Figure 2.13, a topological representation of the original and the reconstructed distributions is highlighted. We observe that the iterated inclusion chains are more similar for WAE than for VAE. For VAE, the inclusions of the reconstructed distribution are randomly scattered through the manifold without connected vertices.

In order to quantitatively assess the quality of the encoding-decoding process, we use the bottleneck distance between the persistent diagram of $\mathcal{X}$ and the persistent diagram of $G(Z)$ of the reconstructed data points. We recall the strength of the bottleneck distance is to measure quantitatively the topological changes in the data, either the true or the reconstructed data, while being insensitive to the scale of the analysis. Traditional distance measures fail to acknowledge this as they do not rely on the persistent homology and, therefore, can only reflect a measurement of the nearness relations of the data points without considering the overall shape of the data distribution. In Table 2.3, we notice the smallest bottleneck distance, and therefore, the best result, is obtained with WAE. It means WAE is capable to better preserve the topological features of the original data distribution than VAE including the nearness measurements and the overall shape.

2.4.3 Persistent Homology of Generated Adversarial Samples According to a Generative Distribution P_G

In this third experiment, we reach the core of our PHom-GeM's contribution. We evaluate on the four generative models, GP-WGAN, WGAN, WAE and VAE, with the persistent homology both the qualitative and quantitative topological properties of the generated adversarial samples with respect to the generative model distribution P_G.

Table 2.3 Bottleneck distance (smaller is better) for PHom-GeM applied to WAE and VAE between the samples X of the original manifold $\mathcal{X}$ and the reconstructed manifold $G(Z|X)$ for $Z \in \mathcal{Z}$. Because of OT, the WAE achieves better performance.

PHom-GeM on WAE	PHom-GeM on VAE	Difference (%)
0.0788	0.0878	10.25

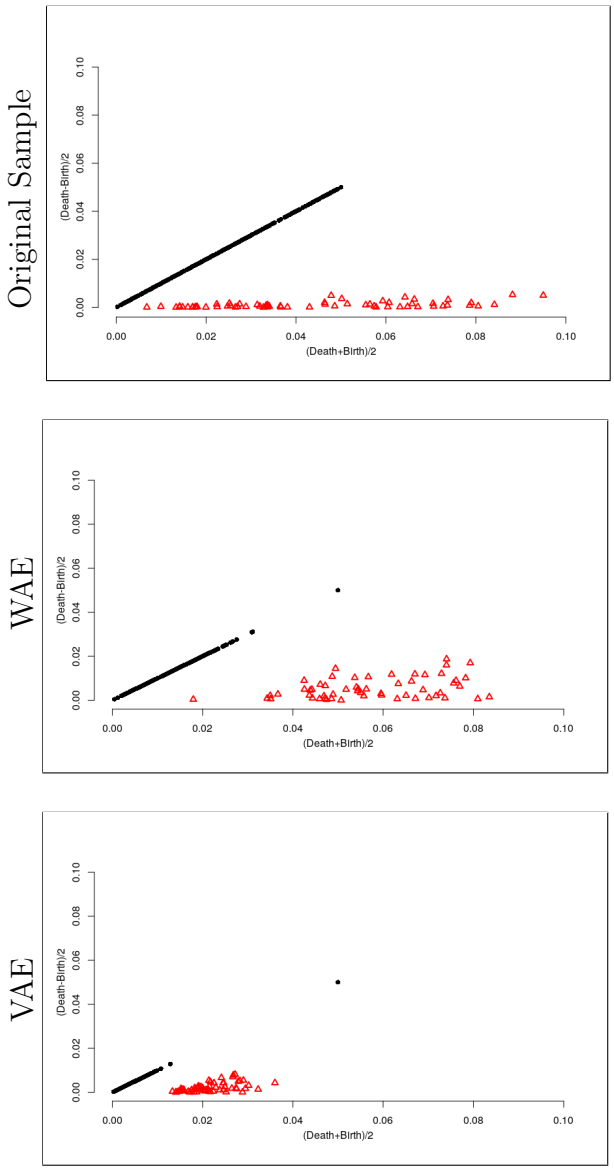

Figure 2.11 PHom-GeM applied to WAE and PHom-GeM applied to VAE's rotated persistence diagrams in comparison to the persistence diagram of the original sample showing the birth-death of the pairing generators of the iterated inclusions. Black points represent the 0-dimensional homology groups H_0 that is the connected components of the complex. Red triangles represent the 1-dimensional homology group H_1 that is the 1-dimensional features, the cycles.

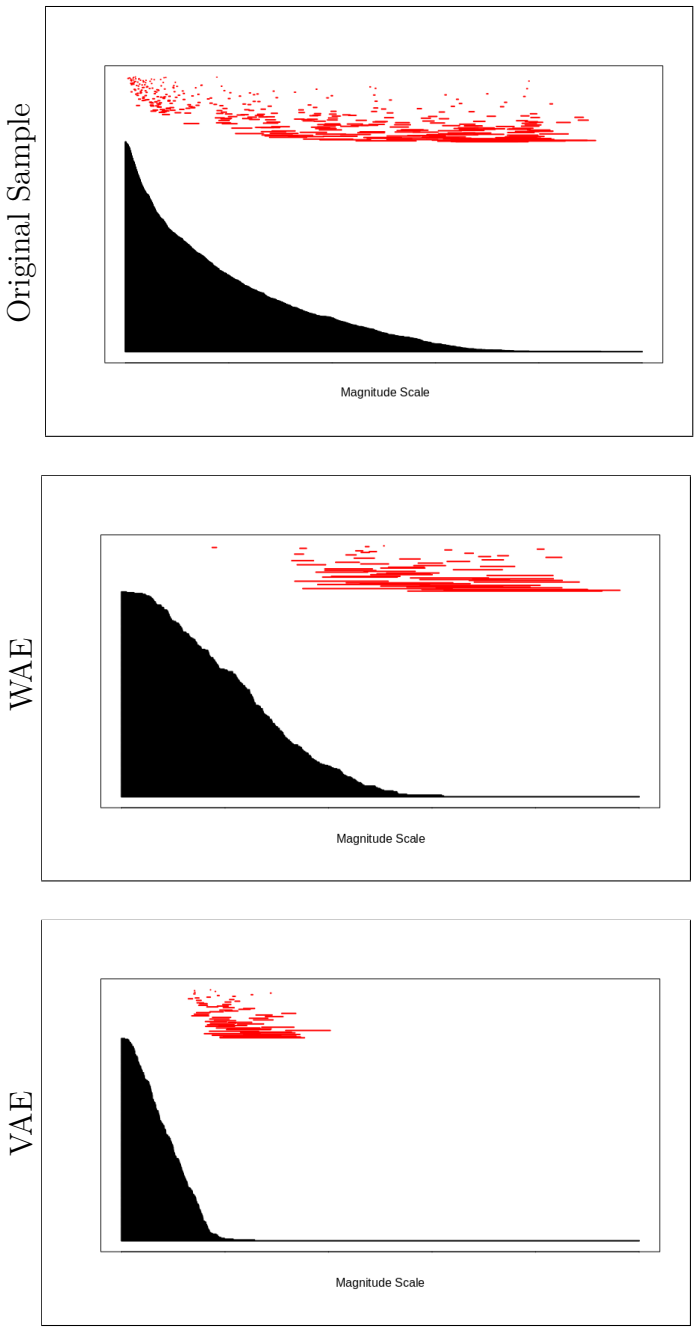

Figure 2.12 PHom-GeM applied to WAE and PHom-GeM applied to VAE's barcode diagrams in comparison to the barcode diagram of the original sample based on the persistence diagrams of Figure 2.11. The barcodes diagrams are a simple way to represent the persistence diagrams. We refer to [26] for further details on their generation.

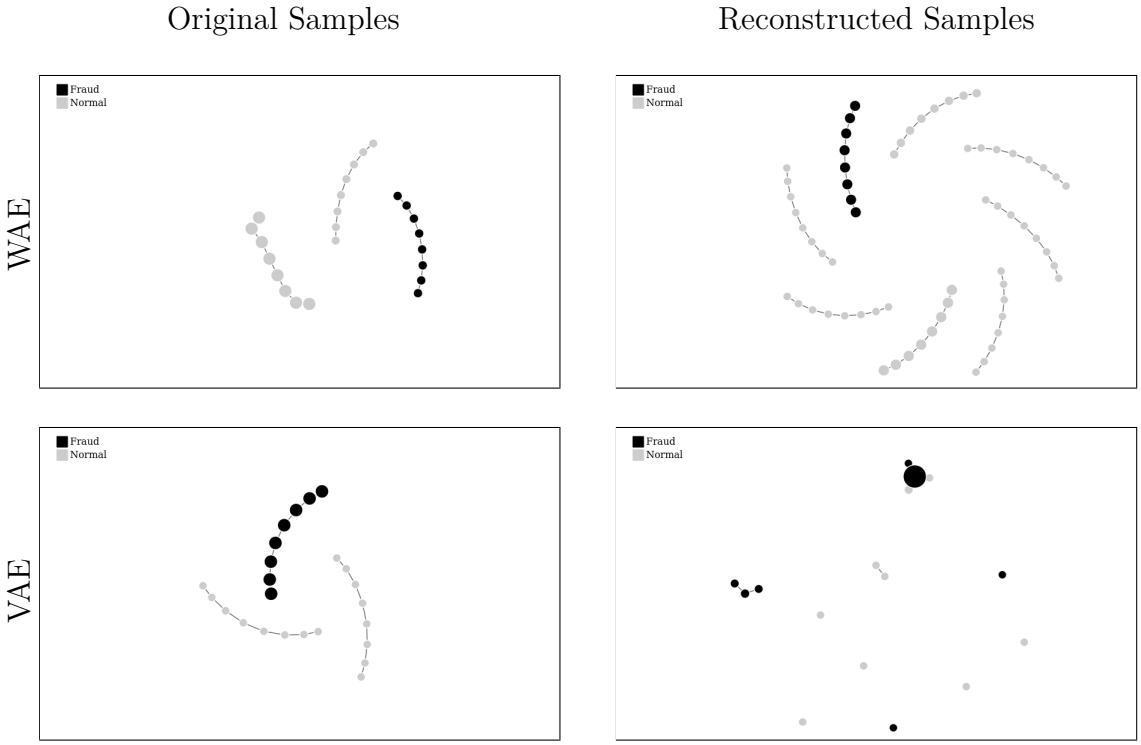

Figure 2.13 Topological representation of the reconstructed data distribution following $P_G(X|Z)$ for PHom-GeM applied to WAE and VAE. The encoding-decoding process of the WAE better preserves the topological features of the manifold $\mathcal{X}$.

The adversarial samples are compared to the original samples X of the manifold $\mathcal{X}$. On the top of Figures 2.14 and 2.15, the rotated persistence and the barcode diagrams of the original sample $\mathcal{X}$ are highlighted. In the persistence diagram, as previously mentioned, black points represent the 0-dimensional homology groups H_0, the connected components of the complex. The red triangles represent the 1-dimensional homology group H_1, the 1-dimensional features known as cycles or loops. The barcode diagram is a simple way of representing the information contained in the persistence diagram. The generated distribution P_G of GP-WGAN is the closest to the distribution P_X followed by WGAN, WAE and VAE. The spectrum of the barcodes of GP-WGAN, effectively, is very similar to the original sample's spectrum as well as denser on the right. On the opposite, the WAE and VAE's distributions P_G are not able to reproduce all of the features contained in the original distribution, underlined by the narrower and incomplete barcode spectrum.

In order to quantitatively assess the quality of the generated distributions, we use the bottleneck distance between the persistence diagram of $\mathcal{X}$ and the persistence diagram

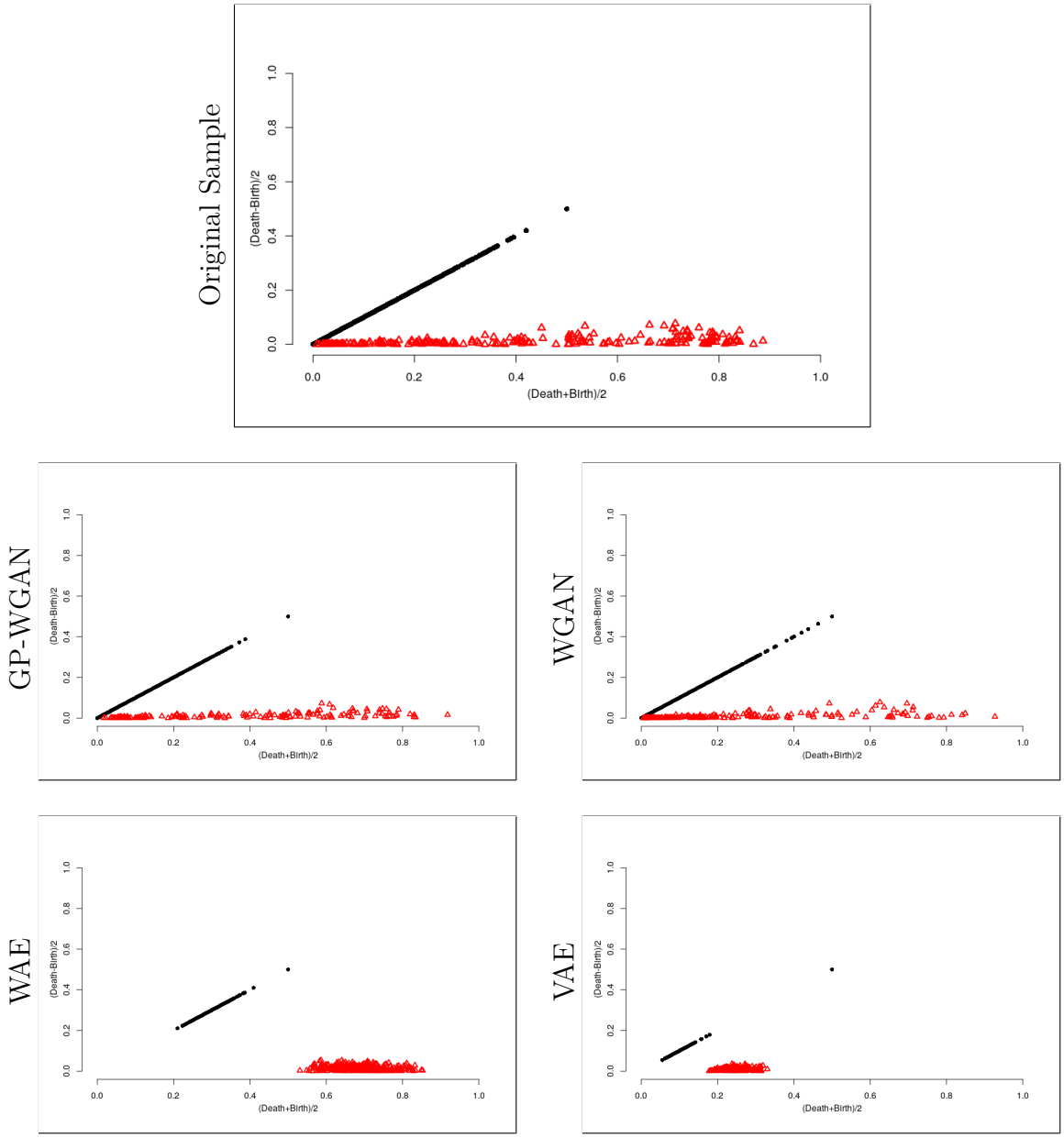

Figure 2.14 On top, the rotated persistence diagram of the original sample is represented. It illustrates the birth-death of the pairing generators of the iterated inclusions. In the persistence diagram, the black points represent the connected components of the complex and the red triangles the cycles. The GP-WGAN, WGAN, WAE and VAE's persistence diagrams allow to assess qualitatively, with the original sample persistence diagram, the persistent homology similarities between the generated and the original distribution, P_G and P_X respectively.

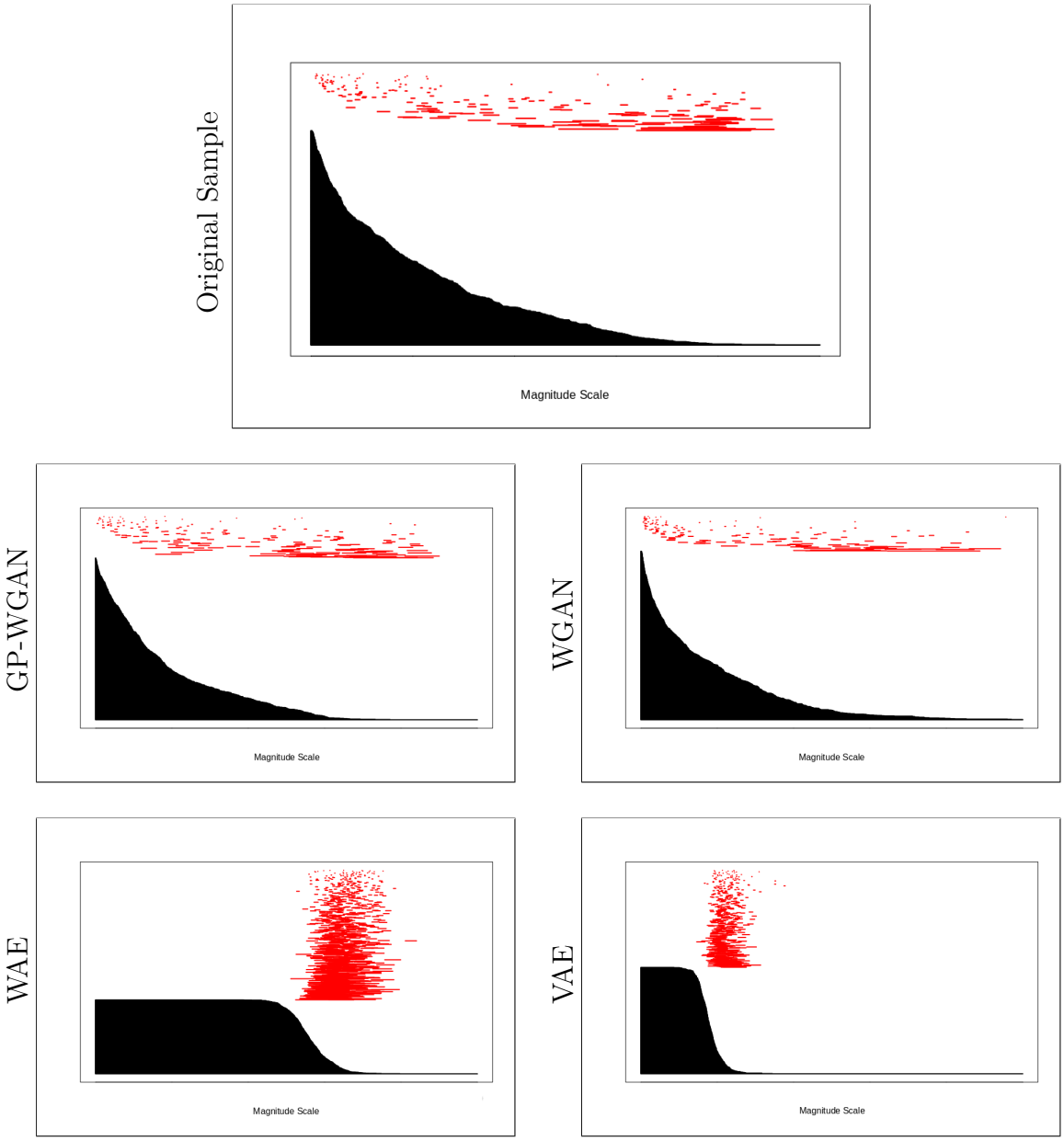

Figure 2.15 On top, the barcode diagrams of the original sample is represented. It illustrates the birth-death of the pairing generators of the iterated inclusions. In the barcode diagram, the black lines represent the connected components of the complex and the red lines the cycles. The GP-WGAN, WGAN, WAE and VAE's barcode diagrams allow to assess qualitatively, with the original sample barcode diagram, the persistent homology similarities between the generated and the original distribution, P_G and P_X respectively.

Table 2.4 Bottleneck distance (smaller is better) with 95% of confidence interval between the samples X of the original manifold $\mathcal{X}$ and the generated samples $\widetilde{X}$ of the manifold $\widetilde{\mathcal{X}}$. Because of the Wasserstein distance and gradient penalty, GP-WGAN achieves better performance.

Gen. Model	Mean Value	Lower Bound	Upper Bound
GP-WGAN	**0.0711**	**0.0683**	**0.0738**
WGAN	0.0744	0.0716	0.0772
WAE	0.0821	0.0791	0.0852
VAE	0.0857	0.0833	0.0881

of $G(Z)$ of the generated data points. We recall the strength of the bottleneck distance is to measure quantitatively the topological changes in the data, either the true or the generated data, while being insensitive to the scale of the analysis. Traditional distance measures fail to acknowledge this as they do not rely on the persistent homology and, therefore, can only reflect a measurement of the nearness relations of the data points without considering the overall shape of the data distribution. Using the bootstrapping technique of [65], we successively randomly select data samples contained in the manifolds $\mathcal{X}$ and $\widetilde{\mathcal{X}}$ to evaluate their bottleneck distance. The total number of selected samples is at least 85% of the total number of points contained in the manifolds $\mathcal{X}$ and $\widetilde{\mathcal{X}}$ to ensure a reliable statistical representation. In Table 2.4, we highlight the mean value of the bottleneck distance for a 95% confidence interval. We also underline the lower and the upper bounds of the 95% confidence interval for each generative model. Confirming the visual observations, we notice the smallest bottleneck distance, and therefore, the best result, is obtained with GP-WGAN, followed by WGAN, WAE and VAE. It means, in our configuration, GP-WGAN is capable to generate data distribution sharing the most topological features with the original data distribution, including the nearness measurements and the overall shape. It confirms topologically on a real-world data set the claims addressed in [37] of superior performance of GP-WGAN against WGAN. Furthermore, the performance of the AE cannot match the generative performance achieved by the GANs. However, the WAE, that relies on optimal transport theory, achieves better generative distribution in comparison to the popular VAE.

2.5 Conclusion

Building upon optimal transport and unsupervised learning, we introduced PHom-GeM, Persistent Homology for Generative Models, a new characterization of the generative manifolds that uses the topology and the persistent homology to highlight the manifold features and the scattered generated distributions. We discuss the relations of GP-WGAN, WGAN, WAE and VAE in the context of unsupervised learning. Furthermore, by relying on the persistent homology, the bottleneck distance has been introduced to estimate quantitatively the alteration of the topological features occurring during the encoding-decoding process and the topological features similarities between the original distribution, the AE latent manifold distributions and the generated distributions of the generative models. It is a specificity that current traditional distance measures fail to acknowledge. We used a challenging imbalanced real-world open data set containing credit card transactions, capable of illustrating the scattered generated data distributions of the generative models, particularly suitable for the banking industry. We conducted experiments showing the performance of PHom-GeM on the four generative models GP-WGAN, WGAN, WAE and VAE. We highlighted the scattered distributions of the WAE and VAE's latent manifold, an AE limitation that will be very likely addressed successively in future research. We furthermore showed the superior persistent homology performance of GP-WGAN in comparison to the other generative models as well as the superior performance of WAE over VAE in the context of the generation of adversarial samples.

Future work will include further exploration of the topological features such as the influence of the simplicial complex. We will additionally address the use of an optimal transport persistent homological distance measure, such as the Wasserstein distance, to increase the accuracy of the topological measurements. Finally, we leave for future research how to back-propagate the persistent homology information to the objective loss function of the generative models to improve their training and the generation of adversarial samples.

Chapter 3

Accurate Tensor Resolution for Financial Recommendations

The new financial European regulations, such as the revised Payment Service Directive (PSD2) [4], are changing the retail banking services by welcoming new joiners to promote competition. Noticeably, retail banks have lost the exclusive privilege of proposing and managing financial solutions, as for instance, the personal finance management on mobile banking applications or the credit cards distribution for the transaction payment solutions. Consequently, they are now looking to optimize their resources to propose a higher quality of financial services using recommender engines. The recommendations are moving from a client-request approach to a bank-proposing approach where the banks dynamically offers new financial opportunities to their clients. By being able to estimate the financial awareness of their clients or to predict their clients' financial transactions in a context-aware environment, the banks can dynamically offer the most appropriate financial solutions, targeting their clients' future needs. In this context, we focus on the tensor decomposition [14], a collection of factorization techniques for multidimensional arrays, to build predictive financial recommender engines. Tensor decomposition are among the most general and powerful tools to perform multi-dimensional personalized recommendations. However, due to their linear algebra complexity, tensor decomposition requires accurate convex optimization schemes.
Therefore, in this chapter, we describe our novel resolution algorithm, VecHGrad for Vector Hessian Gradient, for accurate and efficient stochastic resolution over all existing tensor decomposition, specifically designed for complex decomposition. We highlight the VecHGrad's performance in comparison to state of the art machine learning optimization algorithms and popular tensor resolution scheme. The description of

VecHGrad is then followed by its direct application to the CP tensor decomposition [66, 28], a multidimensional matrix decomposition that factorizes a tensor as the sum of rank-one tensors, for the predictions of the clients' actions. By predicting the clients' actions, the retail banks can adopt an aggressive approach to propose adequate new financial products to their clients. Finally, we will address the monitoring and the prediction of user-device authentication in mobile banking to estimate the financial awareness of the clients. Because of the imbalance between the number of users and devices, we will use the VecHGrad optimization algorithm to solve the PARATUCK2 tensor decomposition [67], which expresses a tensor as a multiplication of matrices and diagonal tensors to represent and to model asymmetric relationships.

3.1 Motivation

Endorsed by the European objectives to promote the financial exchanges between the Euro members, new regulatory directives are now applicable such as the Revised Payment Service Directive (PSD2). They promote more control, more transparency and more competition while trying to reduce the contagion risk in the event of a financial crisis such as in 2008. The financial services, such as Personal Finance Management (PFM) to monitor the expenses, are no longer the exclusive privilege of the retail banks. Effectively, it allows every person having a bank account to use a PFM from a third party provider to manage its personal finance, and thus transform the banks into simple vaults. This game changing directive moreover obligates the banks to provide access, via specific Application Program Interfaces (API), to the financial data of the clients. The banks can therefore compete between each other to attract new clients by proposing them new forms of credit or new financial solutions.

Nonetheless, the retail banks now have the opportunity to use their clients' digital information for recommender engines unlocking new insights about their clients to target financial product recommendations. In this context, we concentrate on two main retail banking applications. First, we focus on the financial actions of the clients. By analyzing and predicting the financial actions of their clients, the banks gain insight knowledge of their clients' interests and clients' spendings. Therefore, they can dynamically propose products targeting the personal needs of their clients such as new credit cards or new loans. For instance, if a client likes to buy a new car every five

years and he bought the last one five years ago, the bank can approach him to propose a new car loan with interesting rates. The predictions of the financial transactions, consequently, are at the heart of the bank's marketing strategy to find or renew product subscriptions. In a second application, we then monitor the user-device authentication on mobile banking application. Through the regular authentication, the banks can create a financial profile awareness for every clients. The more frequently a client is authenticating to its mobile banking application, the more likely he will have a high interest for finance, and therefore, the more likely he will be interested by financial recommendation. This client will be first contacted to advert financial products for wealth and money optimization. The predictions of the user-device authentication consequently becomes involved in a strategy of financial recommendations.

However, the clients' actions contains a large variety of information for both applications. Therefore, the clients' transactions and the user-device authentication predictions in a financial context is multi-dimensional, sparse and complex. As a result, the proposed methodology has to extract information from a large sparse data set while being able to predict a sequence of actions. We rely on tensors, a higher order analogue of matrix decomposition, to answer these challenges. The tensors are able to scale down a large amount of data to an interpretable size using different types of decomposition, also called factorization, to model multidimensional interactions. Depending on the tensor decomposition, different latent variables can be highlighted with their respective asymmetric relationships. Fast and accurate tensor resolutions have nonetheless required specific numerical optimization methods related to preconditioning methods.

The preconditioning gradient methods use a matrix, called a preconditioner, to update the gradient before it is used. Common well-known optimization preconditioning methods include the Newton's method, which employs the exact Hessian matrix, and the quasi-Newton methods, which do not require the knowledge of the exact Hessian matrix, as described in [68]. Introduced to answer specifically some of the challenges facing Machine Learning (ML) and Deep Learning (DL), AdaGrad [56] uses the co-variance matrix of the accumulated gradients as a preconditioner. Because of the dimensions of modern optimization, specialized variants have been proposed to replace the full preconditioning methods by diagonal approximation methods such as Adam in [57], by a sketched version [69, 70] or by other schemes such as Nesterov Accelerated Gradient (NAG) [71] or SAGA [72]. It is worth mentioning that the diagonal approximation

methods are often preferred in practice because of the super-linear memory consumption of the other methods [73].

In this chapter, we take an alternative approach to preconditioning because of our aim of building accurate multidimensional financial recommendation engines. We describe how to exploit Newton's convergence using a diagonal approximation of the Hessian matrix with an adaptative line search. Our approach is motivated by the efficient and accurate resolution of tensor decomposition for which most of the ML and DL state-of-the-art optimizers fail. Our algorithm, called VecHGrad for Vector Hessian Gradient, returns the tensor structure of the gradient and uses a separate preconditioner vector. Our analysis targets non-trivial high-order tensor decomposition and relies on the extensions of vector analysis to the tensor world. We show the superior capabilities of VecHGrad over different tensor decomposition in regards to traditional resolution algorithms, such as the Alternating Least Square (ALS) or the Non-linear Conjugate Gradient (NCG) [74], and some popular ML and DL optimizers such as AdaGrad, Adam or RMSProp. The superior capabilities of VecHGrad are then used for our two main applications related to the clients' financial actions and the user-device authentication. It ensures to reach a negligible numerical error at the end of the tensor factorization process with respect to the tensor decomposition used. The compressed data set inherited from the tensor factorization is then used as input for the neurons to perform the predictions. Our main contributions are summarized below:

- We propose a new resolution algorithm, called VecHGrad, that uses the gradient and the Hessian-vector product with an adaptive line search to achieve the goal of accurate and fast optimization for complex numerical tasks. We demonstrate VecHGrad's superior accuracy at convergence and compare it with traditional resolution algorithms and popular deep learning optimization algorithms for three of the most common tensor decomposition including CP, DEDICOM [75] and PARATUCK2 on five real world data sets.
- We then apply the unique property of CP decomposition for separate modeling of each order of the clients' transactions which are the time, the client reference, the transaction label and the amount. A compressed dense data set is inherited from the resolution of the CP tensor decomposition with VecHGrad. It is used as an optimized input for the neural network removing all sparse information while

highlighting latent factors. The neurons then predict clients' financial activities to unlock financial recommendation.

- Finally, we have developed an approach with the PARATUCK2 tensor decomposition and its unique advantages of interactions modeling with imbalanced data for user-device authentication monitoring. In our application, one user can use several devices, and thus, we have considered the imbalance between the number of users and devices. The PARATUCK2 decomposition is solved with VecHGrad to ensure accurate identification of the latent components. A collection of neurons then predicts the users' authentication to estimate the future financial awareness of the clients. The banks can, therefore, better advertise their products by contacting the clients which will be the most likely to be interested.

The chapter is structured as follows. We discuss the related work in Section 3.2. In Section 3.3, we describe how VecHGrad performs a numerical optimization scheme applied to tensors without the requirement of knowing the Hessian matrix. We then build upon VecHGrad for the predictions of the clients' actions with the CP tensor decomposition and neural networks. We subsequently present our user-device authentication application for the estimation of the clients' financial awareness. We use the PARATUCK2 tensor decomposition with VecHGrad to identify the latent groups and to input a compressed data set to a collection of neurons in charge of the predictions. In Section 3.4, we highlight the experimental results of VecHGrad in comparison to ML optimizers followed by our two applications on clients' transaction predictions and user-device authentication monitoring. Finally, we conclude in Section 3.5 by addressing promising directions for future work.

3.2 Related Work

In this section, we review first the literature related to the numerical optimization algorithms from the tensor and machine learning communities. The VecHGrad resolution algorithm is effectively used in the three experiments of this chapter, and consequently, is one of the main contribution of this chapter. We then review the recommender engines and the tensor decomposition in relation with our application on predictions of sparse clients' actions. Finally, we finish by highlighting briefly some research work related to user-device authentication because of our user-device monitoring application.

Literature on numerical optimization schemes As aforementioned, VecHGrad uses a diagonal approximation of the Hessian matrix and, therefore, it is related to other diagonal approximation such as the diagonal approximation of AdaGrad [56]. The AdaGrad diagonal approximation is very popular in practice and frequently applied [73]. However, it only uses gradient information, as opposed to VecHGrad which uses both gradient and Hessian information with an adaptive line search. Other approaches extremely popular in machine learning and deep learning include Adam [57], NAG [71], SAGA [72] or RMSProp [58]. This non-exhaustive list of machine learning optimization methods is also applied to our study case since it offers a strong baseline comparison for VecHGrad.

The methods specifically designed for tensor decomposition have to be mentioned since our study case is related to tensor decomposition. Various tensor decomposition, or tensor factorization, exist for different types of applications and different algorithms are used to solve tensor decomposition [14, 76]. Each of the decomposition offers unique features for data compression and latent analysis. The most popular optimization scheme among the resolution of tensor decomposition is the Alternating Least Square (ALS). Under the ALS scheme [14], one component of the tensor decomposition is fixed, typically a factor vector or a factor matrix. The fixed component is updated using the other components. All the components are therefore successively updated at each step of the iteration process until a convergence criteria is reached, for instance a fixed number of iteration. Such resolution does not involve any derivative computation. At least one ALS resolution scheme exists for every tensor decomposition. The ALS resolution scheme was introduced in [28] and [66] for the CP/PARAFAC decomposition, in [75] for the DEDICOM decomposition and in [67] for the PARATUCK2 decomposition. Welling and Weber relied on the results of [77] applied to matrix resolution to propose a general non-negative resolution of the CP decomposition in [78]. An updated ALS scheme was presented in [79] to solve PARATUCK2. Bader et al. proposed ASALSAN in [30] to solve with non-negativity constraints the DEDICOM decomposition with the ALS scheme. While some update rules are not guaranteed to decrease the loss function, the scheme leads to overall convergence. Charlier et al. proposed recently in [80] a non-negative ALS scheme for the PARATUCK2 decomposition.

Some approaches are furthermore specifically designed for one tensor decomposition using gradient information. Most frequently, it concerns CP/PARAFAC since it has been

the most applied tensor decomposition [14]. An optimized version of the Non-linear Conjugate Gradient (NCG) for CP/PARAFAC, CP-OPT, is presented in [74, 76]. An extension of the Stochastic Gradient Descent (SGD) is described in [81] to obtain, as mentioned by the authors, an expected CP tensor decomposition. Both approaches focus specifically on CP/PARAFAC, and consequently, their performance on other tensor decomposition is not assessed. The comparison to other numerical optimizers in the experiments is additionally limited, especially when considering existing popular machine learning and deep learning optimizers. In contrast, VecHGrad is detached of any particular model structure, including the choice of tensor decomposition, and it only relies on the gradient and on the Hessian diagonal approximation, crucial for fast convergence of complex numerical optimization. VecHGrad, hence, is easy to implement and to use in practice as it does not require to be optimized for a particular model structure.

Literature on Recommender Engines and Tensor Decomposition Recommender engines have become very popular in real-world applications. The strengths of the second-order matrix factorization recommender engines have been highlighted in various publications of the last decades [82–84]. Recommender engines are generally divided into three main categories: content-based recommendations, collaborative recommendations and hybrid approaches. In content-based systems, the recommendations relies on the previous activities of a user. The collaborative recommendations engines leverages on the community. It recommends similar products to clients sharing similar interests. Finally, the hybrid recommendation combines both content-based and collaborative recommendations [85, 86]. The matrix factorization is limited to the unique modeling of the table *clients*×*products* despite the development of effective and efficient matrix factorization algorithms [87]. It cannot be enriched with additional features. As a solution, tensor recommendation engines have therefore skyrocketed in the past few years [88–90]. The tensors offer the possibility to extend the recommender engines order [91, 92] as further information can be added to the algorithm such as the time or the location.

Noticeably, tensor resolution schemes are of great interest for the end-results, building efficient and accurate multidimensional recommender engines for a wide range of application domains. Their accuracy is underlined in [93] for personalized recommendations in the context of social media. The use of enriched features such as the geospatial localization or the users' social preferences significantly improved the quality of the

recommendations. A regularized context-aware tensor factorization has been applied for location recommendation in [88]. The algorithm relied on interaction regularization and on locations information by using discrete spatial distances. What is noteworthy is that accurate recommendations are intrinsically linked to accurate predictions. The tensor factorization was even used in [94] to predict students' graduation at the university of Minnesota. Increasing the dimensionality of the data however comes at a cost as the sparsity of the information rises significantly. Different authors have therefore tried to propose various solutions to address the sparsity issues depending on the application. The tensor factorization in [95] is combined with a matrix factorization by a weight fusing method. The scheme does not nonetheless address the memory issue that comes with the sparsity of large data sets. The algorithm ParCube [96] introduces a parallelization method to speed up tensor decomposition while offering memory optimization for extremely large data sets.

Although tensor factorization achieve strong recommendations and accurate predictions, neural networks have recently been used to increase the accuracy of the predictions that allow incorporating additional features [97, 98]. The tensor factorization are combined with convolutional neural network or Long-Short-Term-Memory (LSTM) neural network. It is worth mentioning that the references to matrices are references to second order tensors. Even if the sparsity rises with the additional features, the gains in accuracy often overcome the additional computational cost. In fact, in [99], neural networks became the heart of the recommender engine and the tensors an additional feature. Hidasi et al. in [100] tried to improve the second-order tensor recommendation results by using only recurrent neural networks for short time period recommendations. However, approaches that substitute tensor recommendations have focused more on the latest deep learning progress [101–103]. The deep learning scheme is similar to the tensor factorization scheme. It extracts abstract features, or latent factors in the case of tensors, to provide the most likely events to build the recommendation.

Literature on user-device authentication The user-device authentication has significantly evolved for the past few years thanks to the new technologies. A reliable user-device authentication was proposed in [104], based on a graphical user friendly authentication. The use of Location Based Authentication (LBA) was studied in [105]. The development of recent embedded systems within smart-devices leads to new authentication processes, which were considered as a pure fiction only few years

ago. The usage of the embedded camera of smart-devices for authentication by face recognition was assessed in [106]. The face image taken by the camera of the mobile device was sent to a background server to perform the calculation which reverts then to the phone. In a similar approach, the use of iris recognition was proposed in [107]. However, the authors showed this kind of authentication was not the preferred choice of the end user. The sensors embedded into smart-devices allow, additionally, other type of biometric authentication. The different biometric authentication that could be used with smart-devices were presented in [108], such as the pulse-response, the fingerprint or even the ear shape. Although biometric or LBA solutions might offer a higher level of security for authentication, their extension toward a large scale usage is complex. The authors in [109] developed the idea that public-key infrastructure based systems, such as strong passwords in combination with physical tokens, for example, a cell phone, would be more likely to be used and largely deployed. It is nonetheless worth mentioning that the most common procedure for mobile devices authentication is still a code of four or six digits [110].

3.3 Proposed Method

In this section, we begin by describing briefly the standard form of the Newton's method, and then we introduce VecHGrad for the first order tensor optimization. We recall that a first order tensor is a vector. We then arrive at the objective of applying VecHGrad to non-trivial tensor decomposition. We next reintroduce the non-negative ALS scheme applied to the CP and the PARATUCK2 tensor decompositions, as it is the benchmark for tensor resolution algorithms. We also explain how to derive the non-negative ALS scheme for the DEDICOM decomposition based on the PARATUCK2 non-negative ALS scheme. We subsequently present the CP tensor decomposition combined with Neural Networks (NN) for the predictions of the financial activities of the clients. Finally, we reach our second application of user-device authentication in mobile banking. We explain the use of the PARATUCK2 tensor decomposition combined with NN for the predictions of authentication patterns.

The terminology hereinafter follows the one described in [14] and commonly used. Scalars are denoted by lower case letters, a. Vectors and matrices are described by boldface lowercase letters and boldface uppercase letters, respectively $\mathbf{a}$ and $\mathbf{A}$.

High order tensors are represented using Euler script notation, $\mathcal{X}$. The transpose matrix of $A \in \mathbb{R}^{I \times J}$ is denoted by A^T. The inverse of a matrix $A \in \mathbb{R}^{I \times I}$ is denoted by A^{-1}.

3.3.1 Classical Version of the Newton's Method

Hereinafter, we shortly resume the classical version of Newton's method. We invite the reader to [111, 112, 68] for a deeper review.
Let define the objective function $f : \mathbb{R}^d \to \mathbb{R}$ as a closed, convex and twice-differentiable function. For a convex set $\mathcal{C}$, we assume the constrained minimizer

$$x^* := \arg\min_{x \in \mathcal{C}} f(x) \tag{3.1}$$

is uniquely defined. We define the eigenvalues of the Hessian denoted $\gamma = \lambda_{min}(\nabla^2 f(x^*))$ evaluated at the minimum. Additionally, we assume that the Hessian map $x \mapsto \nabla^2 f(x)$ is Lipschitz continuous with modulus L at x^*, as defined below

$$\| \nabla^2 f(x) - \nabla^2 f(x^*) \| \leq L \| x - x^* \|_2 \quad . \tag{3.2}$$

Under these conditions and given an initial random point $\tilde{x}^0 \in \mathcal{C}$ such that $\| \tilde{x}^0 - x^* \|_2 \leq \frac{\gamma}{2L}$, the Newton updates are guaranteed to converge quadratically as defined below

$$\| \tilde{x}^{t+1} - x^* \|_2 \leq \frac{2L}{\gamma} \| \tilde{x}^t - x^* \|_2 \quad . \tag{3.3}$$

This result is well-known, further details can be found in [111]. In our experiments, we slightly modify Newton's method to be globally convergent by using strong Wolfe's line search, described in Subsection 3.3.2.

3.3.2 Introduction to VecHGrad for Vectors

Under the Newton's method, the current iterate $\tilde{\mathbf{x}}^t \in \mathcal{C}$ is used to generate the next iterate $\tilde{\mathbf{x}}^{t+1}$ by performing a constrained minimization of the second order Taylor expansion

$$\tilde{\mathbf{x}}^{t+1} = \arg\min_{\mathbf{x} \in \mathcal{C}} \left\{ \frac{1}{2} \left\langle \mathbf{x} - \tilde{\mathbf{x}}^t, \nabla^2 f(\tilde{\mathbf{x}}^t)(\mathbf{x} - \tilde{\mathbf{x}}^t) \right\rangle + \left\langle \nabla f(\tilde{\mathbf{x}}^t), \mathbf{x} - \tilde{\mathbf{x}}^t \right\rangle \right\} \quad . \tag{3.4}$$

We recall that ∇f and $\nabla^2 f$ denotes the gradient and the Hessian matrix, respectively, of the objective function $f : \mathbb{R}^d \to \mathbb{R}$. In the ML and DL community, the objective function is frequently called the loss function.

$$\nabla f = \underset{\mathbf{x}\in\mathcal{C}}{grad} f = [\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, ..., \frac{\partial f}{\partial x_d}] \tag{3.5}$$

$$\nabla^2 f = \mathbf{Hes} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_d} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_d \partial x_1} & \frac{\partial^2 f}{\partial x_d \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_d^2} \end{pmatrix} \tag{3.6}$$

When $\mathcal{C} \in \mathbb{R}^d$, which is the unconstrained form, the new iterate $\tilde{\mathbf{x}}^{t+1}$ is generated such that

$$\tilde{\mathbf{x}}^{t+1} = \tilde{\mathbf{x}}^t - [\nabla^2 f(\tilde{\mathbf{x}}^t)]^{-1} \nabla f(\tilde{\mathbf{x}}^t) \quad . \tag{3.7}$$

We use the strong Wolfe's line search which allows the Newton's method to be globally convergent. The line search is defined by the following three inequalities

$$\begin{aligned} &\text{i) } f(\tilde{\mathbf{x}}^t + \alpha^t \mathbf{p}^t) \leq f(\tilde{\mathbf{x}}^t) + c_1 \alpha^t (\mathbf{p}^t)^T \nabla f(\tilde{\mathbf{x}}^t) \quad , \\ &\text{ii) } -(\mathbf{p}^t)^T \nabla f(\tilde{\mathbf{x}}^t + \alpha^t \mathbf{p}^t) \leq -c_2 (\mathbf{p}^t)^T \nabla f(\tilde{\mathbf{x}}^t) \quad , \\ &\text{iii) } \mid (\mathbf{p}^t)^T \nabla f(\tilde{\mathbf{x}}^t + \alpha^t \mathbf{p}^t) \mid \leq c_2 \mid (\mathbf{p}^t)^T \nabla f(\tilde{\mathbf{x}}^t) \mid \quad , \end{aligned} \tag{3.8}$$

where $0 \leq c_1 \leq c_2 \leq 1$, $\alpha^t > 0$ is the step length and $\mathbf{p}^t = -[\nabla^2 f(\tilde{\mathbf{x}}^t)]^{-1} \nabla f(\tilde{\mathbf{x}}^t)$. Therefore, the iterate $\tilde{\mathbf{x}}^{t+1}$ becomes the following

$$\begin{cases} \tilde{\mathbf{x}}^{t+1} = & \tilde{\mathbf{x}}^t - \alpha^t [\nabla^2 f(\tilde{\mathbf{x}}^t)]^{-1} \nabla f(\tilde{\mathbf{x}}^t) \\ \tilde{\mathbf{x}}^{t+1} = & \tilde{\mathbf{x}}^t + \alpha^t \mathbf{p}^t \end{cases} . \tag{3.9}$$

Computing the inverse of the exact Hessian matrix, $[\nabla^2 f(\tilde{\mathbf{x}}^t)]^{-1}$, creates a certain number of difficulties. The inverse is therefore computed with a Conjugate Gradient (CG) loop. It has two main advantages: the calculations are considerably less expensive, especially when dealing with large dimensions [68], and the Hessian can be expressed by

a diagonal approximation. The convergence of the CG loop is defined when a maximum number of iteration is reached or when the residual $\mathbf{r}^t = \nabla^2 f(\tilde{\mathbf{x}}^t)\mathbf{p}^t + \nabla f(\tilde{\mathbf{x}}^t)$ satisfies $\| \mathbf{r}^t \| \leq \sigma \| \nabla f(\tilde{\mathbf{x}}^t) \|$ with $\sigma \in \mathbb{R}^+$. Within the CG loop, the exact Hessian matrix can be approximated by a diagonal approximation. In the CG loop, the Hessian matrix is effectively multiplied with a descent direction vector resulting in a vector. The only requirement within the main optimization loop is the descent vector. Consequently, only the results of the Hessian vector product are needed. Using the Taylor expansion, the Hessian vector product is equal to the equation below

$$\nabla^2 f(\tilde{\mathbf{x}}^t)\, \mathbf{p}^t = \frac{\nabla f(\tilde{\mathbf{x}}^t + \eta\, \mathbf{p}^t) - \nabla f(\tilde{\mathbf{x}}^t)}{\eta} \quad . \tag{3.10}$$

The term η is the perturbation and the term $\mathbf{p}^t$ the descent direction vector, fixed equal to the negative of the gradient at initialization. The extensive computation of the inverse of the exact full Hessian matrix is bypassed using only gradient diagonal approximation. Finally, we reached the objective of describing VecHGrad for first order tensor, summarized in Algorithm 2.

3.3.3 VecHGrad for Fast and Accurate Resolution of Tensor Decomposition

Hereinafter, we introduce VecHGrad in its general form, which is applicable to tensors of any dimension, of any order for any decomposition. We review further definitions and operations involving tensors before presenting the algorithm and its theoretical convergence rate.

Algorithm 2: VecHGrad, vector case

repeat
  Receive loss function $f : \mathbb{R}^d \to \mathbb{R}$
  Compute gradient $\nabla f(\tilde{\mathbf{x}}^t) \in \mathbb{R}^d$
  Fix $\mathbf{p}_0^t = -\nabla f(\tilde{\mathbf{x}}^t)$
  repeat
    Update $\mathbf{p}_k^t$ with CG loop: $\mathbf{r}_k = \nabla^2 f(\tilde{\mathbf{x}}^t) p_k^t + \nabla f(\tilde{\mathbf{x}}^t)$
  until $k = cg_{maxiter}$ OR $\| \boldsymbol{r}_k \| \leq \sigma \| \nabla f(\tilde{\boldsymbol{x}}^t)) \|$
  $\alpha^t \leftarrow$ Wolfe's line search
  Update parameters: $\tilde{\mathbf{x}}^{t+1} = \tilde{\mathbf{x}} + \alpha^t \mathbf{p}_{\text{opt}}^t$
until $t = maxiter$ OR $f(\tilde{\boldsymbol{x}}^t) \leq \epsilon_1$ OR $\| \nabla f(\tilde{\boldsymbol{x}}^t) \| \leq \epsilon_2$

Tensor Definitions and Tensor Operations

A slice is a two-dimensional section of a tensor, defined by fixing all but two indices. Alternatively, the k-th frontal slice of a third-order tensor, $\mathcal{X}_{::k}$, may be denoted more compactly as $\mathbf{X}_k$.

The vectorization operator flattens a tensor of n entries to a column vector $\mathbb{R}^n$. The ordering of the tensor elements is not important as long as it is consistent [14]. For a third order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, the vectorization of $\mathcal{X}$ is equal to

$$\text{vec}(\mathcal{X}) = \begin{bmatrix} x_{111} & x_{112} & \cdots & x_{IJK} \end{bmatrix}^T . \tag{3.11}$$

The n-mode product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ with a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ is denoted by $\mathcal{X} \times_n \mathbf{U}$. The n-mode product can be expressed either elementwise or in terms of unfolded tensors.

$$\begin{aligned} (\mathcal{X} \times_n \mathbf{U})_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} &= \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} u_{j i_n} \quad . \\ \mathcal{Y} = \mathcal{X} \times \mathbf{U} \quad &\Leftrightarrow \quad \mathbf{Y}_{(n)} = \mathbf{U}\mathbf{X}_{(n)} \end{aligned} \tag{3.12}$$

The outer product between two vectors, $\mathbf{u}, \mathbf{v} \in \mathbb{R}^I, \mathbb{R}^J$, is denoted by the symbol $\circ$

$$\mathbf{u} \circ \mathbf{v} = \begin{bmatrix} u_1 v_1 & u_1 v_2 & \cdots & u_1 v_J \\ u_2 v_1 & u_2 v_2 & \cdots & u_2 v_J \\ \vdots & \vdots & \vdots & \vdots \\ u_I v_1 & u_I v_2 & \cdots & u_I v_J \end{bmatrix} = \mathbf{u}_i \mathbf{v}_j \quad . \tag{3.13}$$

The Kronecker product between two matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{K \times L}$, denoted by $\mathbf{A} \otimes \mathbf{B}$, results in a matrix $\mathbf{C} \in \mathbb{R}^{IK \times KL}$ such that

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & \cdots & a_{IJ}\mathbf{B} \end{bmatrix} \quad . \tag{3.14}$$

The Khatri-Rao product between two matrices $\mathbf{A} \in \mathbb{R}^{I \times K}$ and $\mathbf{B} \in \mathbb{R}^{J \times K}$, denoted by $\mathbf{A} \odot \mathbf{B}$, results in a matrix $\mathbf{C}$ of size $\mathbb{R}^{IJ \times K}$. It is the column-wise Kronecker product

$$\mathbf{C} = \mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1 \quad \mathbf{a}_2 \otimes \mathbf{b}_2 \quad \cdots \quad \mathbf{a}_K \otimes \mathbf{b}_K] \quad . \tag{3.15}$$

The square root of the sum of all tensor entries squared of the tensor $\mathcal{X}$ defines its norm such that

$$\| \mathcal{X} \| = \sqrt{\sum_{j=1}^{I_1} \sum_{j=2}^{I_2} ... \sum_{j=n}^{I_n} x_{j_1, j_2, ..., j_n}^2} \quad . \tag{3.16}$$

The rank-R of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ is the number of linear components that could fit $\mathcal{X}$ exactly such that

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ ... \circ \mathbf{a}_r^{(N)} \quad . \tag{3.17}$$

The CP/PARAFAC decomposition, shown in Figure 3.1, was introduced in [66, 28]. The tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times K}$ is defined as a sum of rank-one tensors. The number of rank-one tensors is determined by the rank, denoted by R, of the tensor $\mathcal{X}$. The CP decomposition is expressed as

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \mathbf{a}_r^{(3)} \circ ... \circ \mathbf{a}_r^{(N)} \quad , \tag{3.18}$$

where $\mathbf{a}_r^{(1)}, \mathbf{a}_r^{(2)}, \mathbf{a}_r^{(3)}, ..., \mathbf{a}_r^{(N)}$ are factor vectors of size $\mathbb{R}^{I_1}, \mathbb{R}^{I_2}, \mathbb{R}^{I_3}, ..., \mathbb{R}^{I_N}$. Each factor vector $\mathbf{a}_r^{(n)}$ with $n \in \{1, 2, ..., N\}$ and $r \in \{1, ..., R\}$ refers to one order and one rank of the tensor $\mathcal{X}$.

The DEDICOM decomposition [75], illustrated in Figure 3.2, describes the asymmetric relationships between I objects of the tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times K}$. The decomposition groups the I objects into R latent components (or groups) and describe their pattern of interactions by computing $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{H} \in \mathbb{R}^{R \times R}$ and $\mathcal{D} \in \mathbb{R}^{R \times R \times K}$ such that

$$\mathbf{X}_k = \mathbf{A}\mathbf{D}_k\mathbf{H}\mathbf{D}_k\mathbf{A}^T \quad \text{with} \quad k = \{1, ..., K\} \quad . \tag{3.19}$$

The matrix $\mathbf{A}$ indicates the participation of object $i = 1, ..., I$ in the group $r = 1, ..., R$, the matrix $\mathbf{H}$ the interactions between the different components r and the tensor $\mathcal{D}$ represents the participation of the R latent component according to the third order K.

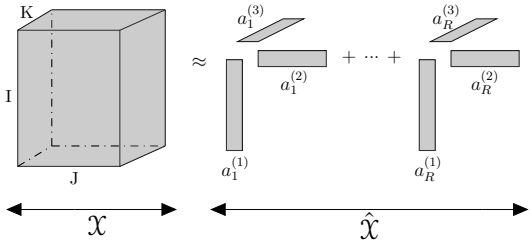

Figure 3.1 Third order CP/PARAFAC tensor decomposition

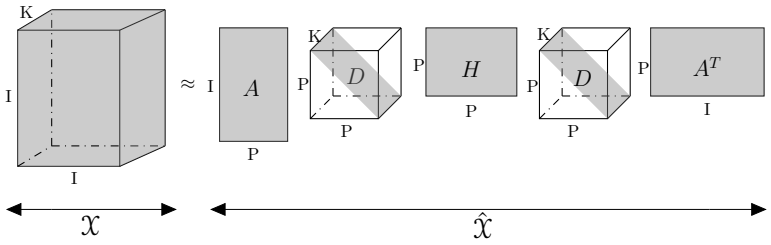

Figure 3.2 Third order DEDICOM tensor decomposition

The PARATUCK2 decomposition [67], represented in Figure 3.3, expresses the original tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ as a product of matrices and tensors

$$\mathbf{X}_k = \mathbf{A}\mathbf{D}_k^A\mathbf{H}\mathbf{D}_k^B\mathbf{B}^T \quad \text{with} \quad k = \{1, ..., K\} \quad , \tag{3.20}$$

where $\mathbf{A}$, $\mathbf{H}$ and $\mathbf{B}$ are matrices of size $\mathbb{R}^{I \times P}$, $\mathbb{R}^{P \times Q}$ and $\mathbb{R}^{J \times Q}$. The matrices $\mathbf{D}_k^A \in \mathbb{R}^{P \times P}$ and $\mathbf{D}_k^B \in \mathbb{R}^{Q \times Q}$ $\forall k \in \{1, ..., K\}$ are the slices of the tensors $\mathcal{D}^A \in \mathbb{R}^{P \times P \times K}$ and $\mathcal{D}^B \in \mathbb{R}^{Q \times Q \times K}$. The columns of the matrices $\mathbf{A}$ and $\mathbf{B}$ represent the latent factors P and Q, and therefore the rank of each object set. The matrix $\mathbf{H}$ underlines the asymmetry between the P latent components and the Q latent components. The tensors $\mathcal{D}^A$ and $\mathcal{D}^B$ measures the evolution of the latent components regarding the third order.

VecHGrad for Tensor Resolution

Here, we describe the main application of VecHGrad that is the resolution of non-trivial tensor decomposition.

The loss function, also called the objective function, is denoted by f and it is equal to

$$f(\tilde{\mathbf{x}}) = \min_{\hat{\mathcal{X}}} ||\mathcal{X} - \hat{\mathcal{X}}|| \quad . \tag{3.21}$$

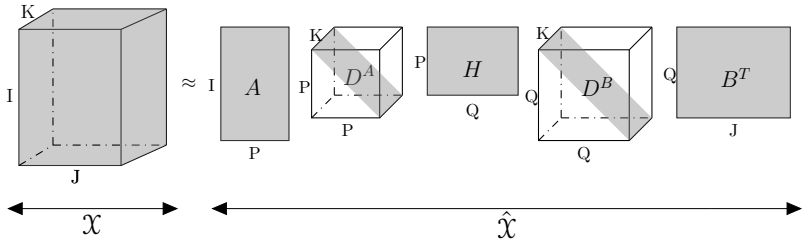

Figure 3.3 Third order PARATUCK2 tensor decomposition

The tensor $\mathcal{X}$ is the original tensor and the tensor $\hat{\mathcal{X}}$ is the approximated tensor built from the decomposition. For instance, if we consider the CP/PARAFAC tensor decomposition applied on a third order tensor, the tensor $\hat{\mathcal{X}}$ is the tensor built with the factor vectors $\mathbf{a}_r^{(1)}, \mathbf{a}_r^{(2)}, \mathbf{a}_r^{(3)}$ for $r = 1, ..., R$ initially randomized such that

$$\hat{\mathcal{X}} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \mathbf{a}_r^{(3)} \quad . \tag{3.22}$$

The vector $\tilde{\mathbf{x}}$ is a flattened vector containing all the entries of the decomposed tensor $\hat{\mathcal{X}}$. If we consider the previous example of a third order tensor $\hat{\mathcal{X}}$ of rank R factorized with the CP/PARAFAC tensor decomposition, we obtain the following vector $\tilde{\mathbf{x}} \in \mathbb{R}^{d=R(I+J+K)}$ such that

$$\tilde{\mathbf{x}} = \text{vec}(\hat{\mathcal{X}}) = \left[\mathbf{a}_1^{(1)},\ \mathbf{a}_2^{(1)}, ..., \mathbf{a}_I^{(R)}, \mathbf{a}_1^{(2)},\ \mathbf{a}_2^{(2)}, ..., \mathbf{a}_J^{(R)}, \mathbf{a}_1^{(3)},\ \mathbf{a}_2^{(3)}, ..., \mathbf{a}_K^{(R)}\right]^T \quad . \tag{3.23}$$

Since the objective is to propose a universal approach for any tensor decomposition, we rely on the finite difference method to compute the gradient of the loss function of any tensor decomposition. Thus, the method can be transposed to any decomposition just by changing the decomposition equation. The approximate gradient is based on a fourth order formula (3.24) to ensure a reliable approximation [113] of the exact gradient, defined as

$$\frac{\partial}{\partial x_i} f(\tilde{\mathbf{x}}^t) \approx \frac{1}{4!\eta}\Big(2[f(\tilde{\mathbf{x}}^t - 2\eta\mathbf{e}_i) - f(\tilde{\mathbf{x}}^t + 2\eta\mathbf{e}_i)] + 16[f(\tilde{\mathbf{x}}^t + \eta\mathbf{e}_i) - f(\tilde{\mathbf{x}}^t - \eta\mathbf{e}_i)]\Big) \quad . \tag{3.24}$$

In (3.24), the index i is the index of the variables for which the derivative is to be evaluated. The variable $\mathbf{e}_i$ is the $i-$th unit vector. The term η, the perturbation, is fixed small enough to achieve the convergence of the iterative process.

The Hessian diagonal approximation is evaluated as described in Subsection 3.3.2. During the CG optimization loop, the Hessian matrix is multiplied with a descent direction vector resulting in a vector. Therefore, only the results of the Hessian vector product is required. Using the Taylor expansion, this product is equal to the equation below

$$\nabla^2 f(\tilde{\mathbf{x}}^t)\,\mathbf{p}^t = \frac{\nabla f(\tilde{\mathbf{x}}^t + \eta\,\mathbf{p}^t) - \nabla f(\tilde{\mathbf{x}}^t)}{\eta} \quad . \tag{3.25}$$

The perturbation term is denoted by η and the descent direction vector by $\mathbf{p}^t$, fixed equal to the negative of the gradient at initialization. Consequently, the extensive computation of the inverse of the exact full Hessian matrix is bypassed using only gradient diagonal approximation. Finally, we reached the core objective of describing VecHGrad for tensors, summarized in Algorithm 3.

Theoretical Convergence Rate of VecHGrad

VecHGrad is based on the Newton's method but it relies on a diagonal approximation of the Hessian matrix instead of the full exact Hessian matrix with an adaptive line search. The reason is that although the exact Newton's method convergence is quadratic [68], the computation of the exact Hessian matrix is too time consuming for ML and DL large-scale applications, including tensor application. Therefore, VecHGrad has a superlinear convergence such that

$$\lim_{n\to\infty} \frac{\|\mathbf{B}_n - \nabla^2 f(\tilde{\mathbf{x}}_{\text{opt}})\mathbf{p}_n\|}{\|\mathbf{p}_n\|} = 0 \quad , \tag{3.26}$$

Algorithm 3: VecHGrad, tensor case

Receive tensor decomposition equation: $g : \begin{cases} \mathbb{R}^d \to \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_n} \\ \tilde{\mathbf{x}}^t \mapsto \tilde{\mathcal{X}} \end{cases}$
Receive $\tilde{\mathbf{x}}^0 = \text{vec}(\hat{\mathcal{X}})$
repeat
  Receive loss function: $f : \begin{cases} \mathbb{R}^d \to \mathbb{R} \\ \tilde{\mathbf{x}}^t \mapsto \| \mathcal{X} - g(\tilde{\mathbf{x}}^t) \| \end{cases}$
  Compute gradient $\nabla f(\tilde{\mathbf{x}}^t) \in \mathbb{R}^d$
  Fix $\mathbf{p}_0^t = -\nabla f(\tilde{\mathbf{x}}^t)$
  repeat
    Update $\mathbf{p}_k^t$ with CG loop: $\mathbf{r}_k = \nabla^2 f(\tilde{\mathbf{x}}^t)\mathbf{p}_k^t + \nabla f(\tilde{\mathbf{x}}^t)$
  until $k = cg_{maxiter}$ OR $\| \boldsymbol{r}_k \| \leq \sigma \| \nabla f(\tilde{\boldsymbol{x}})^t) \|$
  $\alpha^t \leftarrow$ Wolfe's line search
  Update parameters: $\tilde{\mathbf{x}}^{t+1} = \tilde{\mathbf{x}}^t + \alpha^t \mathbf{p}_{\text{opt}}^t$
until $t = maxiter$ OR $f(\tilde{\boldsymbol{x}}^t) \leq \epsilon_1$ OR $\| \nabla f(\tilde{\boldsymbol{x}}^t) \| \leq \epsilon_2$

with $\tilde{\mathbf{x}}_{\text{opt}}$ the point of convergence, $\mathbf{p}_n$ the search direction and $\mathbf{B}_n$ the approximation of the Hessian matrix. Practically, the convergence rate is described according to the equation below

$$q \approx \log \frac{|\tilde{\mathbf{x}}^{t+1} - \tilde{\mathbf{x}}^{t}|}{|\tilde{\mathbf{x}}^{t} - \tilde{\mathbf{x}}^{t-1}|} \left[\log \frac{|\tilde{\mathbf{x}}^{t} - \tilde{\mathbf{x}}^{t-1}|}{|\tilde{\mathbf{x}}^{t-1} - \tilde{\mathbf{x}}^{t-2}|} \right]^{-1} \quad . \tag{3.27}$$

3.3.4 Alternating Least Square for Tensor Decomposition

Because we designed VecHGrad for accurate tensor resolution, we have to compare its performance with the most popular numerical optimizer in the tensor world that is the Alternating Least Square (ALS) [14]. One component of the tensor decomposition, a factor vector or a factor matrix, is fixed and updated using the other components of the decomposition. Every component is successively updated until a convergence criteria is reached. In our experiments, we furthermore emphasize on the non-negative ALS scheme. It allows an easier interpretation of the tensor decomposition latent factors, especially in our case involving financial recommendations. We first present the standard ALS scheme [28, 66] followed by the non-negative ALS scheme based on [77, 78] for the CP tensor decomposition. We then adopt the same methodology for the PARATUCK2 tensor decomposition [79, 80]. Finally, we explain how to derive the non-negative ALS scheme for the DEDICOM tensor decomposition based on the scheme of the PARATUCK2 tensor decomposition.

Under the ALS scheme, the following minimization equation has to be solved for both the standard and the non-negative update rules

$$\min_{\hat{\mathcal{X}}} ||\mathcal{X} - \hat{\mathcal{X}}|| \quad , \tag{3.28}$$

with $\hat{\mathcal{X}}$ the approximate tensor of the decomposition and $\mathcal{X}$ the original tensor. All the factor matrices and the factor tensors are updated iteratively to solve the equation (3.28).

Standard and Non-negative ALS for CP

We recall that the CP tensor decomposition is defined by

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \mathbf{a}_r^{(3)} \circ ... \circ \mathbf{a}_r^{(N)} \tag{3.29}$$

for a N-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times N}$, a rank R, and the factor vectors $\mathbf{a}_r^{(1)}, \mathbf{a}_r^{(2)}, \mathbf{a}_r^{(3)}, ..., \mathbf{a}_r^{(N)}$ of size $\mathbb{R}^{I_1}, \mathbb{R}^{I_2}, \mathbb{R}^{I_3}, ..., \mathbb{R}^{I_N}$.

Standard ALS Method for CP

We denote the latent factor vectors by $\mathbf{A} = \sum_{r=1}^{R} a_r^{(1)}$, $\mathbf{B} = \sum_{r=1}^{R} a_r^{(2)}$ and $\mathbf{C} = \sum_{r=1}^{R} a_r^{(3)}$. For a third order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, the standard update rules are the following

$$\begin{cases} A \leftarrow & \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})(\mathbf{C}^T\mathbf{C} * \mathbf{B}^T\mathbf{B})^\dagger \\ B \leftarrow & \mathbf{X}_{(2)}(\mathbf{C} \odot \mathbf{A})(\mathbf{C}^T\mathbf{C} * \mathbf{A}^T\mathbf{A})^\dagger \\ C \leftarrow & \mathbf{X}_{(3)}(\mathbf{B} \odot \mathbf{A})(\mathbf{B}^T\mathbf{B} * \mathbf{A}^T\mathbf{A})^\dagger \end{cases} . \tag{3.30}$$

Based on the standard ALS update rules for the CP tensor decomposition, we can deduce the non-negative ALS update rules.

Non-negative ALS Method for CP

We use a tensor of size $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ with the latent factor vectors $\mathbf{A} = \sum_{r=1}^{R} a_r^{(1)}$, $\mathbf{B} = \sum_{r=1}^{R} a_r^{(2)}$ and $\mathbf{C} = \sum_{r=1}^{R} a_r^{(3)}$. Based on the non-negative update rules for matrices [77], the non-negative ALS method for CP follows the below update rules

$$a_{ir} \leftarrow a_{ir} \frac{[\mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})]_{ir}}{[\mathbf{A}(\mathbf{C} \odot \mathbf{B})^T(\mathbf{C} \odot \mathbf{B})]_{ir}} \quad , \tag{3.31}$$

$$b_{jr} \leftarrow b_{jr} \frac{[\mathbf{X}_{(2)}(\mathbf{C} \odot \mathbf{A})]_{jr}}{[\mathbf{B}(\mathbf{C} \odot \mathbf{A})^T(\mathbf{C} \odot \mathbf{A})]_{jr}} \quad , \tag{3.32}$$

$$c_{kr} \leftarrow c_{kr} \frac{[\mathbf{X}_{(3)}(\mathbf{B} \odot \mathbf{A})]_{kr}}{[\mathbf{C}(\mathbf{B} \odot \mathbf{A})^T(\mathbf{B} \odot \mathbf{A})]_{kr}} \quad . \tag{3.33}$$

We will use the non-negative ALS update rules in our experiments involving the CP tensor decomposition and the ALS algorithm.

Standard and Non-negative ALS for PARATUCK2 and DEDICOM

We recall that the equation for the PARATUCK2 tensor decomposition is equal to

$$\mathbf{X}_k = \mathbf{A}\mathbf{D}_k^A\mathbf{H}\mathbf{D}_k^B\mathbf{B}^T \quad \text{with} \quad k = \{1, ..., K\} \quad , \tag{3.34}$$

where $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$ is the original tensor, $\mathbf{A}$, $\mathbf{H}$ and $\mathbf{B}$ are matrices of size $\mathbb{R}^{I\times P}$, $\mathbb{R}^{P\times Q}$ and $\mathbb{R}^{J\times Q}$. The matrices $\mathbf{D}_k^A \in \mathbb{R}^{P\times P}$ and $\mathbf{D}_k^B \in \mathbb{R}^{Q\times Q}$ $\forall k \in \{1, ..., K\}$ are the slices of the tensors $\mathcal{D}^A \in \mathbb{R}^{P\times P\times K}$ and $\mathcal{D}^B \in \mathbb{R}^{Q\times Q\times K}$.

Standard ALS Method for PARATUCK2

We consider one level k of K, the third dimension of the tensor to ease the explanation for the resolution of the PARATUCK2 tensor decomposition. Under the standard ALS method, the equation (3.34) is rearranged to update $\mathbf{A}$ such that

$$\mathbf{X}_k = \mathbf{A}\mathbf{F}_k \quad \text{with} \quad \mathbf{F}_k = \mathbf{D}_k^A\mathbf{H}\mathbf{D}_k^B\mathbf{B}^T \quad . \tag{3.35}$$

The simultaneous least square solution for all k leads to

$$\mathbf{A} = \mathbf{X}(\mathbf{F}^\dagger)^T \quad \text{with} \quad \begin{cases} \mathbf{X} = & [\mathbf{X}_1\, \mathbf{X}_2 \cdots \mathbf{X}_k] \\ \mathbf{F} = & [\mathbf{F}_1\, \mathbf{F}_2 \cdots \mathbf{F}_k] \end{cases} \quad . \tag{3.36}$$

To update $\mathcal{D}^A$, the equation (3.34) is rearranged such that

$$\mathbf{X}_k = \mathbf{A}\mathbf{D}_k^A\mathbf{F}_k^T \quad \text{with} \quad \mathbf{F}_k = \mathbf{B}\mathbf{D}_k^B\mathbf{H}^T \quad . \tag{3.37}$$

The matrix $\mathbf{D}_k^A$ is a diagonal matrix which leads to the below update

$$\mathbf{D}_{(k,:)}^A = [(\mathbf{F}_k \odot \mathbf{A})\mathbf{x}_k]^T \quad \text{with} \quad \mathbf{x}_k = \text{vec}(\mathbf{X}_k) \quad . \tag{3.38}$$

The notation $(k, :)$ represents the *k-th* row of $\mathbf{D}_{(k,:)}^A$. To update $\mathbf{H}$, the equation (3.34) is rearranged as

$$\mathbf{x}_k = (\mathbf{B}\mathbf{D}_k^B \otimes \mathbf{A}\mathbf{D}_k^A)\mathbf{h} \quad \text{with} \quad \begin{cases} \mathbf{x}_k = & \text{vec}(\mathbf{X}_k) \\ \mathbf{h} = & \text{vec}(\mathbf{H}) \end{cases} \quad , \tag{3.39}$$

which brings the solution

$$\mathbf{h} = \mathbf{Z}^{\dagger}\mathbf{x} \quad \text{with} \quad \mathbf{Z} = \begin{pmatrix} \mathbf{B}\mathbf{D}_1^B \otimes \mathbf{A}\mathbf{D}_1^A \\ \mathbf{B}\mathbf{D}_2^B \otimes \mathbf{A}\mathbf{D}_2^A \\ \vdots \\ \mathbf{B}\mathbf{D}_k^B \otimes \mathbf{A}\mathbf{D}_k^A \end{pmatrix} \quad . \tag{3.40}$$

The methodology presented for the update of $\mathbf{A}$ and $\mathcal{D}^A$ is reproduced to update $\mathbf{B}$ and $\mathcal{D}^B$. We can now build upon the standard ALS update rules for the PARATUCK2 tensor decomposition to define the non-negative ALS update rules.

Non-negative ALS Method for PARATUCK2

In the experiments, we use the non-negative PARATUCK2 decomposition adapted from the non-negative matrix factorization presented by Lee and Seung in [77]. The matrices, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{H}$, and the tensors, $\mathcal{D}^A$ and $\mathcal{D}^B$, are computed according to the following multiplicative update rules

$$a_{ip} \leftarrow a_{ip} \frac{\left[\mathbf{X}\mathbf{F}^T\right]_{ip}}{\left[\mathbf{A}(\mathbf{F}\mathbf{F}^T)\right]_{ip}} \;,\; \mathbf{F} = \mathcal{D}^A\mathbf{H}\mathcal{D}^B\mathbf{B}^T \quad , \tag{3.41}$$

$$d^a_{pp} \leftarrow d^a_{pp} \frac{\left[\mathbf{Z}^T\mathbf{x}\right]_{pp}}{\left[\mathcal{D}^A(\mathbf{Z}\mathbf{Z}^T)\right]_{pp}} \;,\; \mathbf{Z} = (\mathbf{B}\mathcal{D}^B\mathbf{H}^T) \odot \mathbf{A} \quad , \tag{3.42}$$

$$h_{pq} \leftarrow h_{pq} \frac{\left[\mathbf{Z}^T\mathbf{x}\right]_{pq}}{\left[\mathbf{H}(\mathbf{Z}\mathbf{Z}^T)\right]_{pq}} \;,\; \mathbf{Z} = \mathbf{B}\mathcal{D}^B \otimes \mathbf{A}\mathcal{D}^A \quad , \tag{3.43}$$

$$d^b_{qq} \leftarrow d^b_{qq} \frac{[\mathbf{x}\mathbf{Z}]_{qq}}{\left[\mathcal{D}^B(\mathbf{Z}^T\mathbf{Z})\right]_{qq}} \;,\; \mathbf{Z} = \mathbf{B} \odot (\mathbf{H}^T\mathcal{D}^A\mathbf{A}^T)^T \quad , \tag{3.44}$$

$$b_{qj} \leftarrow b_{qj} \frac{\left[\mathbf{X}^T\mathbf{F}^T\right]_{qj}}{\left[\mathbf{B}(\mathbf{F}\mathbf{F}^T)\right]_{qj}} \;,\; \mathbf{F} = (\mathbf{A}\mathcal{D}^A\mathbf{H}\mathcal{D}^B)^T \quad , \tag{3.45}$$

$$\tag{3.46}$$

with

$$\begin{cases} \mathbf{X} & = [\mathbf{X}_1\, \mathbf{X}_2 \cdots \mathbf{X}_k] \\ \mathbf{x} & = \text{vec}(\mathcal{X}) \end{cases} \quad . \tag{3.47}$$

The multiplicative update rules help to an easier interpretation of the tensor decomposition since all the numbers are positive or null. We used this non-negative ALS update in our experiments involving the PARATUCK2 tensor decomposition and the ALS scheme.

Non-negative ALS method for DEDICOM

We recall that the equation for the DEDICOM tensor decomposition is equal to

$$\mathbf{X}_k = \mathbf{A}\mathbf{D}_k\mathbf{H}\mathbf{D}_k\mathbf{A}^T \quad , \quad \forall\, k = 1, ..., K \quad , \tag{3.48}$$

with the original tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times K}$, the factor matrices $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{H} \in \mathbb{R}^{R \times R}$ and the factor tensor $\mathcal{D} \in \mathbb{R}^{R \times R \times K}$.

The DEDICOM and the PARATUCK2 tensor decomposition are very similar. Therefore, we can deduce the non-negative ALS update rule for DEDICOM based on the PARATUCK2 update rules

$$a_{ip} \leftarrow a_{ip} \frac{\left[\mathbf{X}\mathbf{F}^T\right]_{ip}}{\left[\mathbf{A}(\mathbf{F}\mathbf{F}^T)\right]_{ip}} \, , \, \mathbf{F} = \mathcal{D}\mathbf{H}\mathcal{D}\mathbf{A}^T \quad , \tag{3.49}$$

$$d_{pp} \leftarrow d_{pp} \frac{\left[\mathbf{Z}^T\mathbf{x}\right]_{pp}}{\left[\mathcal{D}(\mathbf{Z}\mathbf{Z}^T)\right]_{pp}} \, , \, \mathbf{Z} = (\mathbf{A}\mathcal{D}\mathbf{H}^T) \odot \mathbf{A} \quad , \tag{3.50}$$

$$h_{pp} \leftarrow h_{pp} \frac{\left[\mathbf{Z}^T\mathbf{x}\right]_{pp}}{\left[\mathbf{H}(\mathbf{Z}\mathbf{Z}^T)\right]_{pp}} \, , \, \mathbf{Z} = \mathbf{A}\mathcal{D} \otimes \mathbf{A}\mathcal{D} \quad , \tag{3.51}$$

$$d_{pp} \leftarrow d_{pp} \frac{[\mathbf{x}\mathbf{Z}]_{pp}}{\left[\mathcal{D}^B(\mathbf{Z}^T\mathbf{Z})\right]_{pp}} \, , \, \mathbf{Z} = \mathbf{A} \odot (\mathbf{H}^T\mathcal{D}\mathbf{A}^T)^T \quad , \tag{3.52}$$

$$a_{pi} \leftarrow a_{pi} \frac{\left[\mathbf{X}^T\mathbf{F}^T\right]_{pi}}{\left[\mathbf{A}(\mathbf{F}\mathbf{F}^T)\right]_{pi}} \, , \, \mathbf{F} = (\mathbf{A}\mathcal{D}\mathbf{H}\mathcal{D})^T \quad , \tag{3.53}$$

$$\tag{3.54}$$

with

$$\begin{cases} \mathbf{X} & = [\mathbf{X}_1\, \mathbf{X}_2 \cdots \mathbf{X}_k] \\ \mathbf{x} & = \mathrm{vec}(\mathcal{X}) \end{cases} \, . \tag{3.55}$$

It is noteworthy that some factors update might not lead to a minimization of the objective function for a given iteration. However, the overall convergence through

the successive iterations is always obtained. We used this scheme in our experiments involving the ALS method and the DEDICOM tensor decomposition.

3.3.5 The CP Tensor Decomposition and Third Order Financial Predictions

Building upon VecHGrad, the CP tensor decomposition and the neural networks, we use a two-steps procedure for the predictions of the financial transactions of the clients, as illustrated in 3.4. First, the clients' transactions are stored in a third order tensor, denoted by $\mathcal{X}_{true}$. The tensor $\mathcal{X}_{true}$ is then decomposed using the CP tensor factorization to remove the sparsity contained in the transactions. The VecHGrad resolution algorithm is used with the equations (3.18) and (3.21) to ensure an accurate CP tensor decomposition. The sparse information contained in $\mathcal{X}_{true}$ is removed from the factor vectors $\mathbf{a}^{(1)}, ..., \mathbf{a}^{(N)}$ of $\mathcal{X}_{target}$. The factor vectors furthermore highlight the different latent groups of clients and transactions. In a second step, the factor vectors $\mathbf{a}^{(1)}, ..., \mathbf{a}^{(N)}$ of $\mathcal{X}_{target}$ are then mapped to the inputs of the neural networks to achieve third order financial predictions. The neural networks are able to predict the financial activities of the bank's clients through the training of the factor vectors data set to learn the function $g(.) : \mathbb{R}^3 \rightarrow \mathbb{R}^1$.

Given the shape of the factor vectors used for the predictions, it is impossible to guess at first glance which neural networks would predict the most accurately the financial transactions. We therefore use and compare different neural networks architecture and a machine learning regression for the predictions. The Decision Trees (DT) are a widely used machine learning technique [114] to predict the value of a variable by learning simple decision rules from the data [115, 116]. Their regression decision rules however have some limitations. Outpacing DT capabilities, neural networks including Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN) and Long-Short-Term-Memory (LSTM), and their applications, have therefore skyrocketed for the past few years [117]. MLP consists of at least three layers: one input layer, one output layer and one or more hidden layer [17]. Each neuron of the hidden layer transforms the values of the previous layer with a non-linear activation function. Although MLP is applied in deep learning, it lacks the possibility of modeling short term and long term events. This feature is found in LSTM [17]. The LSTM has a memory block

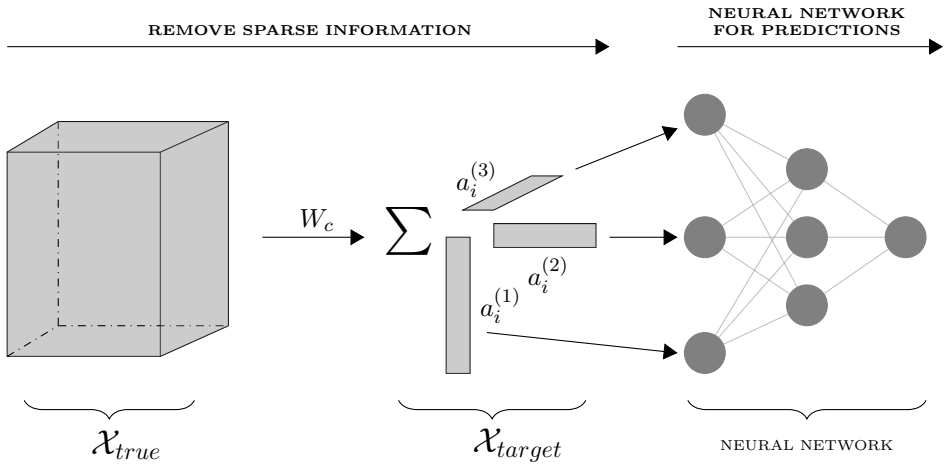

Figure 3.4 The CP tensor decomposition factorizes a third-order tensor $\mathcal{X}_{true}$ of rank R into a sum of rank-one tensors, denoted by $\mathcal{X}_{target}$, removing the sparsity of the transactions. The factor vectors are denoted by $\mathbf{a}_r^{(i)}$ with $r = 1, ..., R$ and $i = \{1, 2, 3\}$. The neural networks are used to predict the next financial transactions of the clients.

connected to the input gate and the output gate. The memory block is activated through a forget gate, resetting the memory information. Moreover, CNN is worth considering for classification and computer vision. In a CNN, the neurons are capable of extracting high order features in successive layers [118]. Through proper classification, the CNN is able to detect and predict various tasks including activities recognition [119, 120]. We summarized the methodology of the financial predictions with the CP tensor decomposition in Algorithm 4.

3.3.6 Monitoring of User-Device Authentication with the PARATUCK2 Tensor Decomposition

By adapting the methodology of the third order financial activities predictions for the user-device authentication monitoring with the PARATUCK2 tensor decomposition, we use a two-steps procedure. The procedure is capable to reduce the dimensions of the initial data set by keeping only the useful information while being able to perform accurate predictions. The proposed approach is illustrated in Figure 3.5.

The user-device authentication are first stored in a third order tensor, denoted by $\mathcal{X}_{true}$. The tensor $\mathcal{X}_{true}$ is then decomposed into three factor matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{H}$ and two factor tensors $\mathcal{D}^{\mathbf{A}}$ and $\mathcal{D}^{\mathbf{B}}$ using the PARATUCK2 tensor decomposition.

Algorithm 4: Financial predictions with the CP tensor decomposition

Receive the CP tensor decomposition equation: $g : \begin{cases} \mathbb{R}^d \rightarrow \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_n} \\ \tilde{\mathbf{x}}^t \mapsto \tilde{\mathcal{X}} \approx \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \mathbf{a}_r^{(3)} \circ \ldots \circ \mathbf{a}_r^{(N)} \end{cases}$
Receive $\tilde{\mathbf{x}}^0 = \text{vec}(\hat{\mathcal{X}})$
repeat
    VecHGrad optimization loop for t steps
        with loss function: $f : \begin{cases} \mathbb{R}^d \rightarrow \mathbb{R} \\ \tilde{\mathbf{x}}^t \mapsto \| \mathcal{X} - g(\tilde{\mathbf{x}}^t) \| \end{cases}$
    Update parameters: $\tilde{\mathbf{x}}^{t+1}$
until *convergence criteria is reached*
/* $\mathtt{A} = \mathtt{a}^{(1)}, \mathtt{B} = \mathtt{a}^{(2)}, \mathtt{C} = \mathtt{a}^{(3)}$ */
$\mathbf{A}, \mathbf{B}, \mathbf{C} \leftarrow \text{unflatten}(\tilde{\mathbf{x}}^{opt})$
Send $\mathbf{A}, \mathbf{B}, \mathbf{C}$ to the input of the NN
Training of the NN to learn the function $g(.) : \mathbb{R}^3 \rightarrow \mathbb{R}^1$
$\mathbf{y} \in \mathbb{R}^1 \leftarrow$ NN prediction of financial activities

It allows to reduce the dimensions of the data set to an interpretable size while keeping all the useful authentication patterns contained in the original tensor $\mathcal{X}_{true}$. The reason to choose the PARATUCK2 tensor decomposition additionally is that a client, a user of the mobile application, can authenticate several times per day using different devices. Our tensor decomposition has therefore to be capable of modeling the asymmetric relationships between the users and the devices. This is the case for the PARATUCK2 tensor decomposition, contrarily to the CP tensor decomposition. We rely on the VecHGrad resolution algorithm using the equations (3.20) and (3.21) to ensure an accurate PARATUCK2 tensor decomposition. Each of the factor

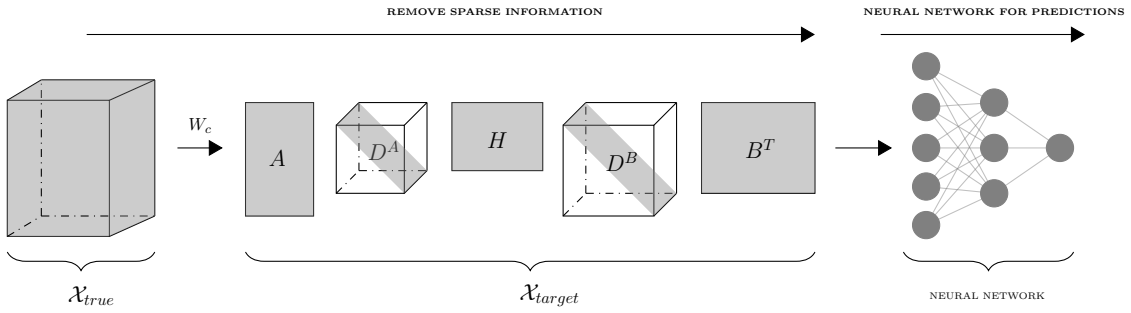

Figure 3.5 The PARATUCK2 tensor decomposition factorizes a third-order tensor $\mathcal{X}_{true}$ into a multiplication of latent matrices and tensors, denoted by $\mathcal{X}_{target}$. It removes the sparsity of the user-device authentication while preserving the data set imbalance. The factor matrices and factor tensors are denoted by $\mathbf{A}, \mathbf{H}, \mathbf{B}$ and $\mathcal{D}^{\mathbf{A}}, \mathcal{D}^{\mathbf{B}}$. The neural networks are used to predict the next user-device authentication.

Algorithm 5: User-device authentication monitoring with the PARATUCK2 tensor decomposition

Receive the PARATUCK2 tensor decomposition equation:

$$g : \begin{cases} \mathbb{R}^d \rightarrow \mathbb{R}^{I_1 \times I_2 \times ... \times I_n} \\ \tilde{\mathbf{x}}^t \mapsto \tilde{\mathbf{X}}_k \approx \mathbf{A}\mathbf{D}_k^A\mathbf{H}\mathbf{D}_k^B\mathbf{B}^T \quad \text{with} \quad k = \{1, ..., K\} \end{cases}$$

Receive $\tilde{\mathbf{x}}^0 = \text{vec}(\hat{\mathcal{X}})$

repeat

  VecHGrad optimization loop for t steps $= 0, 1, 2, ..., T$ to determine $\tilde{\mathbf{x}}^{opt}$

    with loss function: $f : \begin{cases} \mathbb{R}^d \rightarrow \mathbb{R} \\ \tilde{\mathbf{x}}^t \mapsto \parallel \mathcal{X} - g(\tilde{\mathbf{x}}^t) \parallel \end{cases}$

  At each step t, update vector $\tilde{\mathbf{x}}^{t+1}$

until *convergence criteria is reached*

$\mathbf{A}, \mathcal{D}^{\mathbf{A}}, \mathbf{H}, \mathcal{D}^{\mathbf{B}}, \mathbf{B} \leftarrow \text{unflatten}(\tilde{\mathbf{x}}^{opt})$

Send $\mathbf{A}$ and $\mathcal{D}^{\mathbf{A}}$ to the input of the NN

Training of the NN to learn the function $g(.) : \mathbb{R}^3 \rightarrow \mathbb{R}^1$

$\mathbf{y} \in \mathbb{R}^1 \leftarrow$ NN prediction of financial activities

matrices and of the factor tensors highlight respectively the user authentication and the device authentication. We provide more practical details in the experiments section.

In a second step, the factor matrices and the factor tensors are mapped to the inputs of the neural networks to achieve the user authentication predictions. We recall we aim at building a financial awareness score based on the client's habits on the mobile banking application. We consequently concentrate our effort on the user authentication that has been isolated from the device authentication thanks to the PARATUCK2 tensor decomposition. The factor matrix $\mathbf{A}$ and tensor $\mathcal{D}^{\mathbf{A}}$ are used to train the neural networks to learn the function $g(.) : \mathbb{R}^3 \rightarrow \mathbb{R}^1$ in order to perform predictions of the next user authentications. Similarly to the financial actions predictions of the clients, we cannot guess which neural networks will predict best the next user authentication. We therefore train independently and compare the three neural networks aforementioned including a MLP [17], a CNN[118] and a LSTM [17] as well as a DT [114]. We summarized our approach for the user-device authentication monitoring with the PARATUCK2 tensor decomposition in Algorithm 5.

3.4 Experiments

In this section, we present our three experiments in the context of accurate resolution of tensor decomposition for financial recommendation. We first present the results of the VecHGrad optimization algorithm. We then apply the superior accuracy of

the VecHGrad optimization algorithm to the task of predictions of sparse financial transactions of the bank's clients. In the last experiment, we use VecHGrad to perform efficient and precise monitoring of user-device authentication for mobile banking.

3.4.1 VecHGrad for Accurate Tensor Resolution

Hereinafter, we investigate the convergence behavior of VecHGrad in comparison to other popular numerical resolution methods inherited from both the tensor and the machine learning communities. We compare VecHGrad with ten different algorithms applied to the three main tensor decomposition with increasing linear algebra complexity, CP/PARAFAC, DEDICOM and PARATUCK2:

- ALS, Alternating Least Squares [79, 80];
- SGD, Gradient Descent [68];
- NAG, Nesterov Accelerated Gradient [71];
- Adam [57];
- RMSProp [58];
- SAGA [72];
- AdaGrad [56];
- CP-OPT and the Non-linear Conjugate Gradient (NCG) [74, 76];
- L-BFGS [121] inherited from BFGS [122–125].

Data Availability and Code Availability We highlight VecHGrad using popular data sets including CIFAR10, CIFAR100, MNIST, LFW and COCO. All the data sets are available online. Each data set has different intrinsic characteristics such as the size or the sparsity. A quick overview of the data set features is presented in Table 3.1. We chose to use different data sets as the performance of the different optimizers might vary slightly depending on the data. The overall conclusion of the experiments therefore is independent of one particular data set. The implementation and the code of the experiments are available on GitHub[1].

Experimental Setup In our experiments, we use the standard parameters for the popular ML and DL gradient optimization methods. We use $\eta = 10^{-4}$ for SGD, $\gamma = 0.9$ and $\eta = 10^{-4}$ for NAG, $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$ and $\eta = 0.001$ for

[1]The code is available at https://github.com/dagrate/vechgrad.

Table 3.1 Description of the data sets used (K: thousands).

Data Set	Labels	Size	Batch Size
CIFAR-10	image × pixels × pixels	50K × 32 × 32	64
CIFAR-100	image × pixels × pixels	50K × 32 × 32	64
MNIST	image × pixels × pixels	60K × 28 × 28	64
COCO	image × pixels × pixels	123K × 64 × 64	32
LFW	image × pixels × pixels	13K × 64 × 64	32

Adam, $\gamma = 0.9, \eta = 0.001$ and $\epsilon = 10^{-8}$ for RMSProp, $\eta = 10^{-4}$ for SAGA, $\eta = 0.01$ and $\epsilon = 10^{-8}$ for AdaGrad. We use the Hestenes-Stiefel update [126] for the NCG resolution. The convergence criteria is moreover reached when $f^{i+1} - f^i \leq 0.001$ or when the maximum number of iteration is reached. We use 100,000 iterations for the gradient-free methods, 10,000 iterations for the gradient methods and 1,000 iterations for the Hessian-based methods. We additionally fixed the number of iterations to 20 for the VecHGrad's inner CG loop, used to determine the descent direction. We invite the reader to review the code available on GitHub[2] for further knowledge about the parameters used. The simulations are conducted on a server with 50 Intel Xeon E5-4650 CPU cores and 50GB of RAM. All the resolution schemes have been implemented in Julia and are compatible for the ArrayFire GPU accelerator library.

Results and Discussions Two different experiments are performed to highlight the strengths and the weaknesses of the optimizers aforementioned. The evaluation of the optimizers' performance is first based on a visual sample and, then, we provide a detailed overview of the mean loss function at convergence and the mean calculation time over all batches for the data sets CIFAR10, CIFAR100, MNIST, LFW and COCO.

In our first experiment, we highlight visually the strengths of each of the optimization algorithms aforementioned. Figure 3.6 depicts the resulting error of the loss function of each of the methods at convergence for the PARATUCK2 tensor decomposition. We voluntarily chose latent components for which the numerical optimization would be difficult since we are interested to highlight the differences of convergence for complex optimization, and not to reproduce a good image quality. The error of the loss function, or how accurate a method is, is reflected by the blurriness of the picture. The less the image is blurry, the lower the loss function error at convergence. As it can be

[2]The code is available at https://github.com/dagrate/vechgrad.

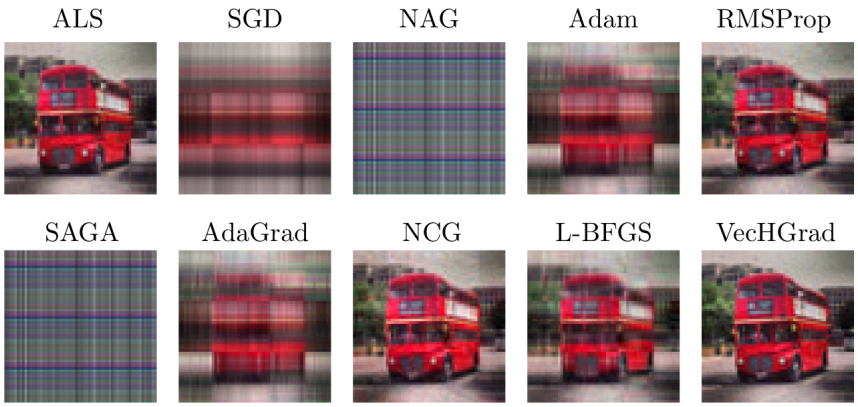

Figure 3.6 Visual simulation of the accuracy at convergence of the different optimizers for the PARATUCK2 decomposition. The accuracy at convergence is highlighted by how blurry the image is (the less blurry, the better). The popular gradient optimizers AdaGrad, NAG and SAGA failed to converge to a solution close to the original image, contrarily to VecHGrad or RMSProp.

noticed, some numerical methods, including ALS, RMSProp or VecHGrad, offer the best observable image quality at convergence, given our choice of parameters. However, other popular schemes, including NAG and SAGA, fail to converge to a solution resulting in a noisy image, far from being close to the original image.

In a second experiment, we compare in Tables 3.2 and 3.3 the loss function errors and the calculation times of the numerical optimization methods on the five ML data sets, CIFAR-10, CIFAR-100, MNIST, COCO and LFW, for the three tensor decomposition CP, DEDICOM and PARATUCK2. Both the loss function errors and the calculation times are computed based on the mean of the loss function errors and the mean of the calculation times over all batches. As it can be observed, the numerical schemes of the NAG, SAGA and AdaGrad algorithms fail to minimize the error of the loss function accurately. We have furthermore to mention that the ALS scheme offers a good compromise between the resulting errors and the required calculation times, explaining its major success across tensor decomposition applications. The gradient descent optimizers, Adam and RMSProp, and the Hessian based optimizers, VecHGrad and L-BFGS, are capable to minimize the most accurately the loss function. The NCG method achieves satisfying errors for the CP and the DEDICOM decomposition but its performance decreases significantly when trying to solve the complex PARATUCK2 decomposition. Surprisingly, the calculation times of the Adam and RMSProp gradient descents are greater than the calculation times of VecHGrad. VecHGrad is capable

Table 3.2 Mean of the loss function errors at convergence over all batches. The lower, the better (best result in bold).

Decomposition	Optimizer	CIFAR-10	CIFAR-100	MNIST	COCO	LFW
CP	ALS	318.667	428.402	897.766	485.138	4792.605
CP	SGD	2112.904	2825.710	2995.528	3407.415	7599.458
CP	NAG	4338.492	5511.272	4916.003	8187.315	18316.589
CP	Adam	1578.225	2451.217	1631.367	2223.211	6644.167
CP	RMSProp	127.961	128.137	200.002	86.792	4205.520
CP	SAGA	4332.879	5501.528	4342.708	6327.580	13242.181
CP	AdaGrad	3142.583	4072.551	2944.768	4921.861	10652.488
CP	NCG	41.990	37.086	23.320	76.478	4130.942
CP	L-BFGS	195.298	525.279	184.906	596.160	4893.815
CP	VecHGrad	$\leq$ **0.100**	$\leq$ **0.100**	$\leq$ **0.100**	$\leq$ **0.100**	$\leq$ **0.100**
DEDICOM	ALS	1350.991	1763.718	1830.830	1894.742	3193.685
DEDICOM	SGD	435.780	456.051	567.503	406.760	511.093
DEDICOM	NAG	4349.151	5722.073	4415.687	6325.638	9860.454
DEDICOM	Adam	579.723	673.316	575.341	743.977	541.515
DEDICOM	RMSProp	63.795	236.974	96.240	177.419	33.224
DEDICOM	SAGA	4285.512	5577.981	4214.771	5797.562	8128.724
DEDICOM	AdaGrad	1962.966	2544.436	1452.278	2851.649	3033.965
DEDICOM	NCG	550.554	321.332	171.181	583.430	711.549
DEDICOM	L-BFGS	423.802	561.689	339.284	435.188	511.620
DEDICOM	VecHGrad	$\leq$ **0.100**	$\leq$ **0.100**	$\leq$ **0.100**	$\leq$ **0.100**	$\leq$ **0.100**
PARATUCK2	ALS	408.724	480.312	1028.250	714.623	658.284
PARATUCK2	SGD	639.556	631.870	1306.869	648.962	495.188
PARATUCK2	NAG	4699.058	6046.024	5168.824	8205.223	14546.438
PARATUCK2	Adam	512.725	680.653	591.156	594.687	615.731
PARATUCK2	RMSProp	133.416	145.766	164.709	134.047	174.769
PARATUCK2	SAGA	4665.435	5923.178	4934.328	6350.172	8847.886
PARATUCK2	AdaGrad	1775.433	2310.402	1715.316	2752.348	2986.919
PARATUCK2	NCG	772.634	1013.032	270.288	335.532	15181.961
PARATUCK2	L-BFGS	409.666	522.158	464.259	467.139	416.761
PARATUCK2	VecHGrad	$\leq$ **0.100**	$\leq$ **0.100**	$\leq$ **0.100**	$\leq$ **0.100**	$\leq$ **0.100**

to outperform the gradient descent schemes on both accuracy and speed thanks to the use of the vector Hessian approximation, inherited from gradient information, and its adaptive strong Wolfe's line search. We can therefore conclude that VecHGrad is capable to solve accurately and efficiently complex numerical optimizations, including complex tensor decomposition, whereas, surprisingly, some popular machine learning gradient descents, such as SAGA, NAG or AdaGrad, fail or show average performance.

Table 3.3 Mean calculation times (sec.) to reach convergence over all batches. The convergence is reached when the evolution of the loss function errors between the iterations is smaller than 0.05. The results have to be analyzed with Table 3.2 as a small calculation time does not ensure a small loss function error.

Decomposition	Optimizer	CIFAR-10	CIFAR-100	MNIST	COCO	LFW
CP	ALS	5.289	4.584	2.710	5.850	4.085
CP	SGD	1060.455	1019.432	0.193	2335.060	6657.985
CP	NAG	280.432	256.196	0.400	1860.660	1.317
CP	Adam	2587.467	2771.068	2062.562	6667.673	6397.708
CP	RMSProp	2013.424	2620.088	2082.481	5588.660	4975.279
CP	SAGA	1141.374	1160.775	0.191	3504.593	3692.471
CP	AdaGrad	1768.562	2324.147	959.408	3729.306	6269.536
CP	NCG	315.132	165.983	4.910	778.279	716.355
CP	L-BFGS	2389.839	2762.555	2326.405	5936.053	5494.634
CP	VecHGrad	200.417	583.117	644.445	1128.358	1866.799
DEDICOM	ALS	21.280	70.820	14.469	55.783	158.946
DEDICOM	SGD	1826.214	1751.355	1758.625	1775.100	1145.594
DEDICOM	NAG	30.847	25.820	240.587	43.003	49.518
DEDICOM	Adam	2105.825	2128.626	1791.295	2056.036	1992.987
DEDICOM	RMSProp	1233.237	1129.172	993.429	1140.844	1027.007
DEDICOM	SAGA	27.859	30.970	64.440	28.319	32.154
DEDICOM	AdaGrad	196.208	266.057	1856.267	2020.417	2027.370
DEDICOM	NCG	2524.762	644.067	236.868	1665.704	4219.446
DEDICOM	L-BFGS	1568.677	1519.808	1209.971	1857.267	1364.027
DEDICOM	VecHGrad	592.688	918.439	412.623	607.254	854.839
PARATUCK2	ALS	225.952	209.978	230.392	589.437	625.668
PARATUCK2	SGD	1953.609	2625.722	2067.727	3002.172	2745.380
PARATUCK2	NAG	48.468	48.724	285.679	76.811	72.068
PARATUCK2	Adam	2628.211	2657.387	2081.996	2719.519	2709.638
PARATUCK2	RMSProp	1407.752	1156.370	1092.156	1352.057	1042.899
PARATUCK2	SAGA	74.248	70.952	120.861	71.398	86.682
PARATUCK2	AdaGrad	2595.478	2626.939	2073.777	292.564	2795.260
PARATUCK2	NCG	150.196	1390.013	928.071	1586.523	82.701
PARATUCK2	L-BFGS	2780.658	2656.062	2188.253	3522.249	2822.661
PARATUCK2	VecHGrad	885.246	1149.594	1241.425	1075.570	1222.827

3.4.2 Predicting Sparse Clients' Actions in the Banking Environment

New regulations have opened the competition in retail banking, especially in Europe. Retail banks lost the exclusive privilege of proposing financial solutions such as, for instance, transaction payments. They are consequently looking to propose a higher quality of financial services using recommender engines. The recommendations are

moving from a client-request approach to a bank-proposing approach where the banks dynamically offers new financial opportunities to their clients. We address this problem in our experiments with our approach. It allows the banks to predict the financial transactions of the clients in a context-aware environment and, therefore, to offer the most appropriate financial solutions to their clients for their future needs.

Clients' Transactions and Data Availability In 2016, the Santander bank released an anonymized public data set containing the financial activities of its clients[3]. The file contains activities of 2.5 millions of clients classified in 22 transactions labels for a 16 months period. The first reported transaction is on 28 January 2015 and the last one on 28 April 2016. The total number of transactions is slightly more than 17 millions. The three most common transactions are the main account transfers, direct debits and online transactions. Since the transactions are classified per month, we track monthly activities to predict the activities of the following months.

Experimental Setup and Code Availability In our simulation, we choose the 200 clients having the most frequent financial activities during the 16 months since regular activities are more interesting for the predictions modeling. The transactions are categorized into 22 different labels such as, for instance, credit card activity, payroll activity, savings or interest payments. Consequently, all the information is gathered in the tensor $\mathcal{X}_{true}$, with a size equal to $200\times22\times16$. We define the tensor rank equal to 25, considering the sparsity of the information and the number of transaction labels. For the CP decomposition with the VecHGrad resolution, we define the stopping condition as the relative change in the objective function W_c. The CP decomposition stops when $|W_{c_n} - W_{c_{n-1}}|(W_{c_{n-1}})^{-1} \leq 10^{-6}$ where W_{c_n} and $W_{c_{n-1}}$ are the values of W_c at the current and the previous iterations, respectively. We rely on the keras library for the neural network implementation. We use the Adam solver with the default parameters $\beta_1 = 0.5, \beta_2 = 0.999$ for the neural network training. The experiments were performed on a computer with 16GB of RAM, Intel i7 CPU and a Tesla K80 GPU accelerator.

Results and Discussions We first highlight the superior accuracy of the VecHGrad resolution algorithm on our financial data set. We then perform the predictions using three different type of neural networks: Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM) network. We addition-

[3]The data set is available at https://www.kaggle.com/c/santander-product-recommendation

ally cross-validate the performance of the neural network with a Decision Tree (DT). The performance of each neural network and the DT is based on four metrics: the Mean Absolute Error (MAE), the Jaccard distance, the cosine similarity and the Root Mean Square Error (RMSE).

Based on the findings of Subsection 3.4.1, we compare the performance between VecHGrad and other popular machine learning and deep learning optimizers on our clients' transactions data set. As it can be noticed in Table 3.4, the numerical errors of the objective function W_c of the CP tensor decomposition are the lowest at convergence for the VecHGrad resolution algorithm, thanks to its adaptive line search and the Hessian update. The results furthermore are very similar to the results of the subsection 3.4.1. The numerical optimizers NCG, BFGS, ALS and RMSProp lead to lowest numerical errors after VecHGrad while the updates SGD, NAG and SAGA achieve poor performance. It is worth mentioning, additionally, that the NCG optimizer allows to reach the second lowest numerical error, explaining its success within the scientific community when using the CP tensor decomposition. Our findings legitimate the use of the VecHGrad algorithm for the task of predicting the next sparse clients' actions since, in our configuration, it outperforms the popular numerical optimizers.

The predictive models, MLP, CNN, LSTM and DT, have been trained on one year period from 28 January 2015 until 28 January 2016. The activities for the next three months are then predicted with a rolling time window of one month. As shown in Figure 3.7, the LSTM models the most accurately the future personal savings activities followed by the MLP, the DT, and finally the CNN. The CNN fails visually to predict accurately the savings activity in comparison to the other three methods, while the LSTM seems to achieve the most accurate predictions. We highlight this preliminary conclusion in Table 3.5 by reporting the previously described four metrics for Figure 3.7. In Figures 3.8 and 3.9, we present the predictions of the credit card and the debit card spendings. The best results are similarly obtained with the LSTM, followed by the MLP, the DT and the CNN. In Table 3.6, we show the aggregated metrics among all transaction predictions. In all the experiments, the LSTM network predicts the activities the most accurately having the lowest errors, followed by the MLP, the DT and the CNN.

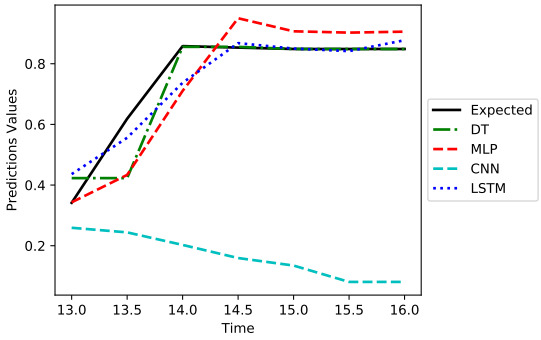

Figure 3.7 Three months predictions of the evolution of the personal savings of one latent group of clients. We can observe the prediction differences between the neural networks.

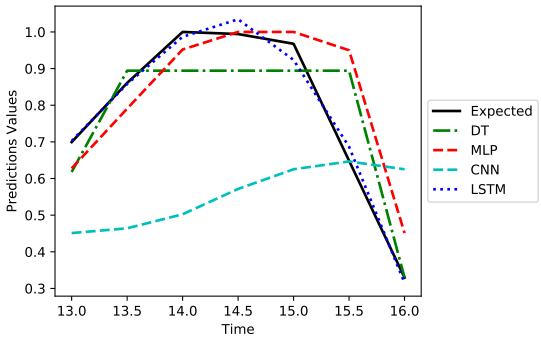

Figure 3.8 Three months predictions of the evolution of the credit card spendings of one latent group of clients. The prediction differences between the methods are highlighted.

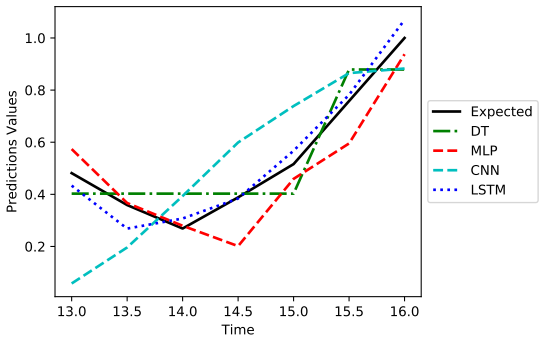

Figure 3.9 Three months predictions of the evolution of the debit card spendings of one latent group of clients. The prediction differences between the methods are highlighted.

Table 3.4 Residual errors of the objective function W_c between the different optimizers for the prediction of the clients' actions at convergence (the smaller, the better). All methods have similar computation time.

Optimizer	W_c Error
CP-VecHGrad	$\leq$ **0.100**
CP-ALS	222.280
CP-SGD	8585.511
CP-NAG	9930.186
CP-Adam	5941.293
CP-RMSProp	475.649
CP-SAGA	10646.734
CP-Adagrad	8055.968
CP-NCG	16.792
CP-BFGS	101.833

Table 3.5 Latent predictions errors on personal savings. LSTM achieves superior performance.

Error Measure	DT	MLP	CNN	LSTM
MAE	**0.040**	0.083	0.579	0.047
Jaccard dist.	0.053	0.109	0.777	**0.061**
cosine sim.	0.946	0.928	0.832	**0.966**
RMSE	0.080	0.121	0.626	**0.063**

Table 3.6 Aggregated predictions errors on all transactions. LSTM achieves superior performance.

Error Measure	DT	MLP	CNN	LSTM
MAE	0.024	0.021	0.270	**0.010**
Jaccard dist.	0.034	0.025	0.288	**0.017**
cosine sim.	0.831	0.908	0.883	**0.953**
RMSE	0.026	0.021	0.298	**0.014**

To resume, we obtain the best results when using LSTM for the predictions. Considering that most of the clients' financial actions are cyclic and occur on a monthly basis, we find legitimate that the LSTM gives the best predictions. By using our approach, the banks therefore are able to remove the sparsity of the financial transactions of the clients wile being able to predict the actions for the future weeks, a key aspect for marketing personalized financial products.

3.4.3 User-Device Authentication in Mobile Banking

Because of the increasing competition in the banking sector brought by the PSD2 directive, the banking actors are now looking to use all the digital information available from their clients. More especially, the banks can split their clients into different groups and build different financial awareness profile to target product recommendations by monitoring and predicting the user-device authentication of their mobile banking application. We consequently highlight how we can analyze the trace of the authentication through tensor decomposition. We discuss how we take into account the imbalance between the number of users and devices. We then highlight the accurate removal of sparse information using our VecHGrad resolution algorithm followed by the results of the predictions of the authentication with our methodology.

User-Computer Authentication and Data Availability For the sake of the reproducibility of the experiments, we present the approach with a public data set. In 2014, the Los Alamos National Laboratory enterprise network published the anonymized user-computer authentication logs of their laboratory [127], and available at https://csr.lanl.gov/data/auth/. Each authentication event is composed of the authentication time (in Unix time), the computer label and the user label such as, for instance, "1,U1,C1". In total, more than 11,000 users and 22,000 computers are listed representing 13 GB of data.

Construction of the user-computer authentication tensor We randomly select 150 users and 300 computers within the dataset representing more than 60 millions lines. The first two months of authentication events have been compressed into 50 time intervals, corresponding to 25 working days per month. A tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ of size of $150 \times 300 \times 50$ is built. The first dimension, denoted by I, represents the users, the second dimension, denoted by J, the computers and the last dimension, K, stands for the time intervals.

Limitations of the CP tensor decomposition The CP tensor decomposition expresses the original tensor into a sum of rank one tensors. The user-computer authentication tensor is therefore decomposed as a sum of user-computer-time rank-one tensors. However, in the case of strong imbalance, CP leads to underfitting or overfitting one of the dimension [74]. Within the dataset, we can find 2 users that connect to at

least 20 different computers. Therefore, a rank equal to 2, one per user, consequently underfits the computer connections. A rank equal to 20, one per machine, overfits the number of users. In Table 3.7, the underfitting is underlined by significant residual errors at convergence. The overfitting is detected by a good understanding of the data since the residual errors tend to be small. Hence, the PARATUCK2 decomposition is chosen to model properly each dimension of the original tensor.

PARATUCK2 Tensor Completion and Resolution PARATUCK2 decomposes the main tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ into a product of matrices and diagonal tensors as shown in Figure 3.10. The matrix $\mathbf{A}$ factorizes the users into P groups. We observe 15 different groups of users, and therefore, P equals to 15. The diagonal tensor $\mathcal{D}^A$ reflects the temporal evolution of the connections of the P users groups. The matrix $\mathbf{H}$ represents the asymmetry between the P users groups and the Q computers groups. We notice 25 different groups of machines related to different authentication profiles, and consequently, Q equals to 25. The diagonal tensor $\mathcal{D}^B$ illustrates the temporal evolution of the connections of the Q computers groups. Finally, the matrix $\mathbf{B}$ factorizes the computers into Q latent groups of computers.

Predictions for Financial Recommendation To achieve higher subscription rates during the advertising campaign of financial products, we explore the latent predictions for targeted recommendation based on the future user-computer authentication. The results of PARATUCK2 contain the users' temporal information and the computers' temporal information in the diagonal tensors $\mathcal{D}^A$ and $\mathcal{D}^B$, respectively. Predicting the users' authentication allows the banks to build a more complete financial awareness profile of their clients for optimized advertisement.

Table 3.7 In CP, for imbalanced dataset, underfitting one dimension is highlighted by significant residual errors. Overfitting is difficult to measure because of the low residual errors. A good understanding of the data is required to estimate it.

Tensor Size	Rank	Residual Errors	$\frac{\lvert f(x_n)-f(x_{n-1})\rvert}{\lvert f(x_n)\rvert}$
2×20×30	2	**50.275**	$< 10^{-6}$
2×20×30	20	**1.147**	$< 10^{-6}$

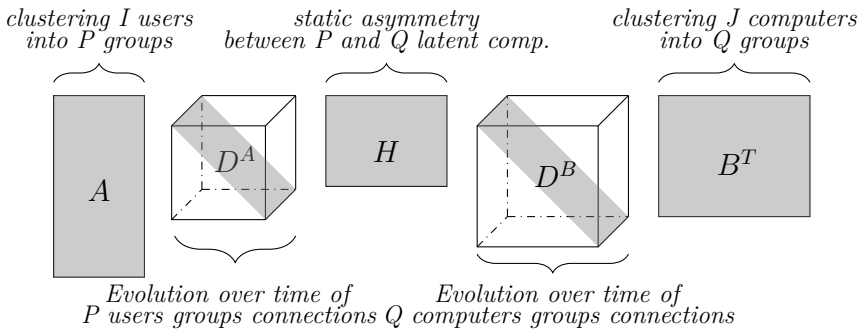

Figure 3.10 PARATUCK2 decomposition applied to user-computer authentication. The neural network predictions are performed on the tensor $\mathcal{D}^A$.

The first step of our approach is to remove the sparsity of the information and to ensure the accurate resolution of the PARATUCK2 tensor decomposition on our user-device authentication data set. We hereinafter compare the error at convergence between VecHGrad and other machine learning and deep learning optimizers. The results are presented in Table 3.8. We recall that the PARATUCK2 tensor decomposition is one of the most complex tensor decomposition to solve, and therefore, it can lead to numerical instabilities during the resolution optimization process. Similarly to the previous experiments, the VecHGrad algorithm achieves the lowest numerical errors at convergence followed by the SGD, BFGS, ALS, and RMSProp algorithms. The numerical instability of the NCG algorithm is highlighted here. Effectively, the NCG algorithm diverges when applied on the complex PARATUCK2 tensor decomposition with our user-device authentication data set. The optimizers NAG and SAGA additionally lead to the biggest numerical errors. These findings confirm the results of the previous experiments. We consequently used the VecHGrad resolution algorithm to solve the objective minimization function W_c, as illustrated in Figure 3.5, since it is the algorithm delivering the lowest numerical errors at convergence.

In Figures 3.11 and 3.12, we highlight the results of the predictions of the users' authentication for a specific group of clients, corresponding to one specific latent factor P. Four different methods have been used for the predictions, DT, MLP, CNN and LSTM. All the methods have been trained on a six weeks period. The users' authentication for the next two weeks are then predicted with a rolling time window of one day. Figures 3.11 and 3.12 highlight visually that the LSTM models the most accurately the future users' authentication. It is followed by the MLP, the DT, and finally the

Table 3.8 Residual errors of the objective function W_c between the different optimizers for the monitoring of the user-device authentication at convergence (the smaller, the better). All the methods have similar computation time.

Optimizer	W_c Error
PARATUCK2 - VecHGrad	$\leq$ **0.100**
PARATUCK2 - ALS	99.834
PARATUCK2 - SGD	8.651
PARATUCK2 - NAG	513.860
PARATUCK2 - Adam	419.812
PARATUCK2 - RMSProp	146.086
PARATUCK2 - SAGA	513.561
PARATUCK2 - Adagrad	461.413
PARATUCK2 - NCG	diverge
PARATUCK2 - BFGS	67.152

CNN. We underline this preliminary statement using six well-known error measures. The Mean Absolute Error (MAE), the Mean Directional Accuracy (MDA), the Pearson correlation, the Jaccard distance, the cosine similarity and the Root Mean Square Error (RMSE) are used to determine objectively the most accurate predictive method. Table 3.9 describes the aggregated error measures of the users' authentication. As previously seen, the LSTM is the closest to the true authentication since it has the lowest error values. Then, the MLP comes second, the DT third, and the CNN last.

We can conclude that LSTM combined with PARATUCK2 models the best the future users' authentication with the aim to better target the clients that might be interested by financial products during the bank's advertising campaigns. As the majority of the user's authentication are sequence-based, it is legitimate to find out LSTM gives the best results for the predictions. Effectively, each user has a recurrent pattern in the authentication process depending on its activities of the day. By using VecHGrad for PARATUCK2 and the LSTM for the predictions, the bank can therefore earn a significant competitive advantage for the personalized products recommendation, relying only on its clients' authentication on the mobile application.

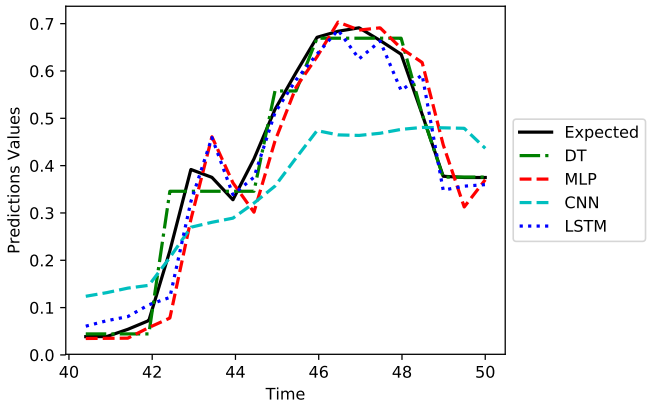

Figure 3.11 Two weeks prediction of the evolution of a latent users' authentication group according to the different models used.

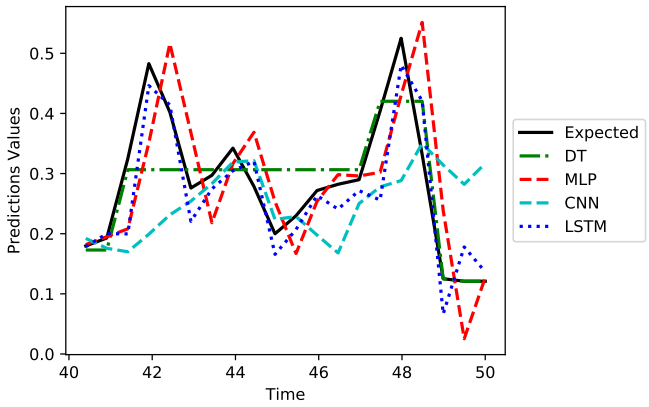

Figure 3.12 Two weeks prediction of the evolution of another latent users' authentication group according to the different models used.

Table 3.9 Aggregated latent predictions errors of the users' authentication with decision tree and neural networks

Error Measure	DT	MLP	CNN	LSTM
MAE	0.0965	0.0506	0.1106	**0.0379**
MDA	0.1579	0.7447	0.5263	**0.6842**
Pearson corr.	0.8537	0.9598	0.8885	**0.9753**
Jaccard dist.	0.2257	0.1206	0.2648	**0.0911**
cosine sim.	0.9587	0.9891	0.9745	**0.9914**
RMSE	0.1306	0.0695	0.3140	**0.0477**

3.5 Conclusion

Building upon tensor decomposition, numerical optimization, preconditioning methods and neural networks, we proposed a predictive method to target personalized recommendations for the banking industry. Because of the requirement of precise and accurate resolution of tensor decomposition, we introduced VecHGrad, a Vector Hessian Gradient optimization method. VecHGrad uses partial information of the second derivative and an adaptive strong Wolfe's line search to ensure faster convergence. We conducted experiments on five real world data sets, CIFAR10, CIFAR100, MNIST, COCO and LFW, very popular in machine learning and deep learning. We highlighted that VecHGrad is capable to outperform the accuracy of the widely used gradient based resolution methods such as Adam, RMSProp or Adagrad, and the linear algebra update ALS on three different tensor decomposition, CP, DEDICOM and PARATUCK2, offering different levels of complexity. Based on the results of this experiment, we used VecHGrad resolution algorithm for the task of predicting the next financial actions of the bank's clients in a sparse environment. We rely on a public data set proposed by the Santander bank. We proposed a predictive method in which the sparsity of the financial transactions is removed before performing the predictions on future client's transactions. The sparsity of the financial transactions is removed using the popular CP tensor decomposition, which decomposes the initial tensor containing the financial transaction into a sum of rank-one tensors. The predictions are performed using different type of neural networks including LSTM, CNN, MLP and a decision tree. Due to the recurrent activities of most of the financial transactions, we underlined the best results were found when the CP tensor decomposition was used with LSTM. We furthermore presented our methodology for a novel application in the context of mobile banking application. We introduced the use of the PARATUCK2 tensor decomposition and neural networks for the monitoring and the predictions of imbalanced user-device authentication. The PARATUCK2 tensor decomposition expresses a tensor as a multiplication of matrices and diagonal tensors and, therefore, it is highly suitable for imbalanced data sets. The user-device authentication on mobile banking allows to build a financial awareness profile to better target potential subscription by the clients on new products. We rely on a public data set proposed by the Los Alamos National Laboratory enterprise network for the experiments. The resolution of the PARATUCK2 tensor decomposition was performed using VecHGrad to ensure a decomposition of high accuracy. We performed users' authentication predictions using LSTM, CNN,

MLP and a decision tree, evaluated on different distance measures. Similarly to the predictions of the next clients' transactions the best results were obtained with LSTM because of the recurrent and cyclic patterns of the user-device authentication.

Future work will concentrate on the influence of the adaptive line search for the VecHGrad algorithm. We effectively observed that the performance of the algorithm is strongly correlated with the performance of the the adaptive line search optimization. We will simultaneously look to reduce the memory cost of the adaptive line search as it has a crucial impact for a GPU implementation as well as a limited memory resolution for a usage on very large data sets. Concerning the predictions of the financial activities for personal financial recommendation, a smaller time frame discretization, weekly or daily, will be assessed with other financial transactions. It will offer a larger choice of financial product recommendations depending on the clients' mid-term and long-term interests. Finally, the financial recommendation depending of the user-device authentication on a mobile banking application will be further extended by incorporating additional features. Noticeably, the navigation usage, the time gap between each action and the type of device used will be monitored to further improve the bank's advertising campaigns of their products to the appropriate clients.

Chapter 4

MQLV: Q-learning for Optimal Policy of Money Management in Retail Banking

Reinforcement learning [128] is one of the best approaches to train a computer game emulator capable of human level performance. In a reinforcement learning approach, an optimal value function is learned across a set of actions, or decisions, that leads to a set of states giving different rewards, with the objective to maximize the overall reward. A policy assigns to each state-action pairs an expected return. We call an optimal policy a policy for which the value function is optimal. QLBS [129], Q-Learner in the Black-Scholes(-Merton) Worlds, applies the reinforcement learning concepts, and noticeably, the popular Q-learning algorithm [130], to the financial stochastic model of Black, Scholes and Merton [131, 132]. It is, however, specifically optimized for the geometric Brownian motion and the vanilla options. Its range of application is, therefore, limited to vanilla option pricing within the financial markets. We propose MQLV, Modified Q-Learner for the Vasicek model, a new reinforcement learning approach that determines the optimal policy of money management based on the aggregated financial transactions of the clients. It unlocks new frontiers to establish personalized credit card limits or to fulfill bank loan applications, targeting the retail banking industry. MQLV extends the simulation to mean reverting stochastic diffusion processes and it uses a digital function, a Heaviside step function expressed in its discrete form, to estimate the probability of a future event such as a payment default. In our experiments, we first show the similarities between a set of historical financial transactions and Vasicek

generated transactions and, then, we underline the potential of MQLV on generated Monte Carlo simulations. MQLV is the first Q-learning Vasicek-based methodology addressing transparent decision making processes in retail banking.

4.1 Introduction

A major goal of the reinforcement learning (RL) and Machine Learning (ML) community is to build efficient representations of the current environment to solve complex tasks. In RL, an agent relies on multiple sensory inputs and past experience to derive a set of plausible actions to solve a new situation [133]. While the initial idea around reinforcement learning is far from new [134–136], significant progress has been achieved recently by combining neural networks and Deep Learning (DL) with RL, to either increase the quality of the signals from the multiple sensory inputs or the number of linear function approximators. DL [137, 138] has, for instance, allowed the development of a novel agent combining RL with a class of deep artificial neural networks [133, 139] resulting in Deep Q-Network (DQN). The Q refers to the Q-learning algorithm introduced in [130]. It is an incremental method that successively improves its evaluations of the quality of the state-action pairs. The DQN approach achieves human level performance on Atari video games using unprocessed pixels as inputs. In [140], deep RL with double Q-Learning was proposed to challenge the DQN approach while trying to reduce the overestimation of the action values, a well-known drawback of the Q-learning and DQN methodologies. Meanwhile, the extension of the DQN approach from discrete action domain to continuous action domain, directly from the raw pixels to inputs, was successfully achieved for various simulated tasks [141].

Nonetheless, most of the proposed models focused on gaming theory and computer game simulation and very few to the financial world. In QLBS [129], a RL approach is applied to the Black, Scholes and Merton (BSM) financial framework for derivatives [131, 132], a cornerstone of the modern quantitative finance. In the BSM model, the dynamic of a stock market is defined as following a Geometric Brownian Motion (GBM) to estimate the price of a vanilla option on a stock [142]. A vanilla option is an option that gives the holder the right to buy or sell the underlying asset, a stock, at maturity for a certain price, the strike price. In Figure 4.1, we describe the payoff $(S_T - K)^+$ at maturity of a vanilla call option with S_T the spot price at maturity of the stock and K the strike price. The holder of the vanilla option will execute his right to buy the stock

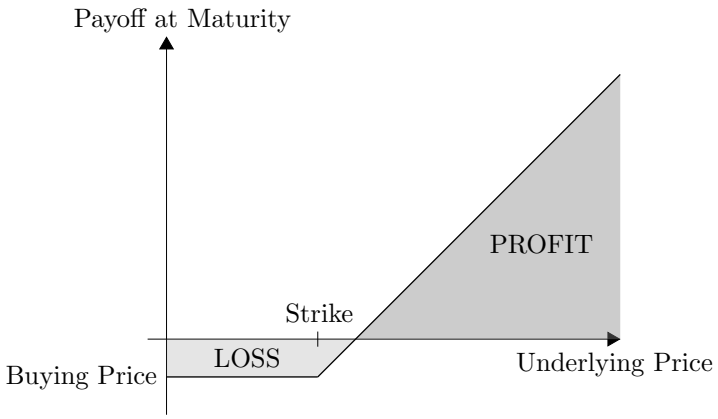

Figure 4.1 Payoff at maturity of a vanilla call option. The option allows to buy the underlying asset, a stock, at maturity for a certain price, the strike price. A profit is generated if the underlying price is higher than the strike at maturity, otherwise the buyers makes a loss.

at the strike price as soon as the payoff is positive and compensates the buying cost of the option. QLBS is one of the first approach to propose a complete RL framework for finance. As mentioned by the author, a certain number of topics are, however, not covered in the approach. For instance, it is specifically designed for vanilla options and it fails to address any other type of financial applications. Additionally, the initial generated paths rely on the popular GBM but there exist a significant number of other popular stochastic models depending on the market dynamics [143].

In this work, we describe a RL approach tailored for personal recommendation in retail banking regarding money management to be used for loan applications or credit card limits. The method is part of a banking strategy trying to reduce the customer churn in a context of a competitive retail banking market. We rely on the Q-learning algorithm and on a mean reverting diffusion process to address this topic. It leads ultimately to a fitted Q-iteration update and a model-free and off-policy setting. The diffusion process reflects the time series observed in retail banking such as transaction payments or credit card transactions. We furthermore introduce a new terminal digital function, Π, defined as a Heaviside step function in its discrete form for a discrete variable $n \in \mathbb{R}$. The digital function is at the core of our approach for retail banking since it can evaluate the future probability of an event including, for instance, the future default probability of a client based on his spendings. Our method converges to an optimal policy, and to optimal sets of actions and states, respectively the spendings

and the available money. The retail banks can, consequently, determine the optimal policy of money management based on the aggregated financial transactions of the clients. The banks are able to compare the difference between the MQLV's optimal policy and the individual policy of each client. It contributes to an unbiased decision making process while offering transparency to the client. Our main contributions are summarized below:

- A new RL framework called MQLV, Modified Q-Learning for Vasicek, extending the initial QLBS framework [129]. MQLV uses the theoretical foundation of RL learning and Q-Learning to build a financial RL framework based on a mean reverting diffusion process, the Vasicek model [144], to simulate data, in order to reach ultimately a model-free and off-policy RL setting.

- The definition of a digital function to estimate the future probability of an event. The aim is to widen the application perspectives of MQLV by using a characteristic terminal function that is usable for a decision making process in retail banking such as the estimation of the default probability of a client.

- The first application of Q-learning to determine the clients' optimal policy of money management in retail banking. MQLV leverages the clients aggregated financial transactions to define the optimal policy of money management, targeting the risk estimation of bank loan applications or credit cards.

- The introduction of an update function applied to MQLV to compensate the overestimation of the action values, characteristic of the Q-Learning algorithm. The objective is to minimize the overestimation of the event probabilities measured by the digital function.

The chapter is structured as follows. We discuss the related work in Section 4.2. We review QLBS and the Q-Learning formulations derived by Halperin in [129] in the context of the Black, Scholes and Merton model in Section 4.3. We describe MQLV according to the Q-Learning algorithm that leads to a model-free and off-policy setting in Section 4.4. We highlight experimental results in Section 4.5. We conclude in Section 4.6 by addressing promising directions for future work.

4.2 Related Work

The foundations of modern reinforcement learning [134, 136] established the theoretical framework to learn policies for sequential decision problems by proposing a formulation of the cumulative future reward signal. The Q-learning algorithm introduced in [135] is one of the cornerstone of reinforcement learning. However, the convergence of the Q-Learning algorithm was solved several years later. It was shown that the Q-Learning algorithm with non-linear function approximators [145] with off-policy learning [146] could provoke a divergence of the Q-network. The reinforcement learning community therefore focused on linear function approximators [145] to ensure convergence.

The emergence of neural networks and deep learning [17] contributed to address the use of reinforcement learning with neural networks. At an early stage, deep auto-encoders were used to extract feature spaces to solve reinforcement learning tasks [147]. Thanks to the release of the Atari 2600 emulator [148], a public data set was then available answering the needs of the RL community for larger simulation. The Atari emulator allowed a proper performance benchmark of the different reinforcement learning algorithms and offered the possibility to test various architectures. The Atari games were used to introduce the concept of deep reinforcement learning [133, 139]. The authors used a convolutional neural network trained with a variant of Q-learning to successfully learn control policies directly from high dimensional sensory inputs. They reached human-level performance on many of the Atari games. Shortly after, the deep reinforcement learning was challenged by double Q-Learning within a deep reinforcement learning framework [140]. The double Q-Learning algorithm was initially introduced in [149] in a tabular setting. The double deep Q-Learning gave more accurate estimates and lead to much higher scores than the one observed in [133, 139]. An ongoing work is consequently to further improve the results of the double deep Q-learning algorithms through different variants. The authors used a quantile regression to approximate the full quantile function for the state-action return distribution in [150], leading to a large class of risk-sensitive policies. It allowed them to further improve the scores on the Atari 2600 games simulator. Similarly, a new algorithm, C51, which applies the Bellman's equation to the learning of the approximate value distribution was designed in [151] and showed state-of-the-art performance.

Meanwhile, a certain number of publications focused on model-free policies and actor-critic framework. Stochastic policies were trained in [152] with a replay buffer to avoid divergence. It was showed in [153] that deterministic policy gradients (DPG) exist, even in a model-free environment. The DPG approach was subsequently extended in [154] using a deviator network. A deviator network backpropagates different signals to train the network. Continuous control policies were learned using backpropagation, and therefore, introducing the Stochastic Value Gradient SVG(0) and SVG(1) in [155]. Recently, Deep Deterministic Policy Gradient (DDPG) was presented in [141] to learn competitive policies using an actor-critic model-free algorithm based on a DPG algorithm that can operate over continuous action spaces.

4.3 Background

We define $A_t \in \mathcal{A}$ the action taken at time t for a given state $X_t \in \mathcal{X}$ and the immediate reward by R_{t+1}. To avoid any confusion between the stochastic diffusion process and the different states of the environment, the ongoing state is denoted by $X_t \in \mathcal{X}$ and the stochastic diffusion process by $S_t \in \mathcal{S}$ at time t. The discount factor that trades off the importance of immediate and later rewards is expressed by $\gamma \in [0; 1]$.

We recall a policy is a mapping from states to probabilities of selecting each possible action [128]. By following the notations of [129], the policy π such that

$$\pi : \{0, \dots, T-1\} \times \mathcal{X} \to \mathcal{A} \tag{4.1}$$

maps at time t the current state $X_t = x_t$ into the action $a_t \in \mathcal{A}$

$$a_t = \pi(t, x_t) \quad . \tag{4.2}$$

The value of a state x under a policy π, denoted by $v_\pi(x)$ when starting in x and following π thereafter, is called the state-value function for policy π and it is defined as

$$v_\pi = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | X_t = x \right] \quad . \tag{4.3}$$

The action-value function, $q_\pi(x, a)$ for policy π defines the value of taking action a in state x under a policy π as the expected return starting from x, taking the action a, and thereafter following policy π such that

$$q_\pi(x, a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | X_t = x, A_t = a \right] \quad . \tag{4.4}$$

The optimal policy, π_t^*, is the policy that maximizes the state-value function

$$\pi_t^*(X_t) = \arg\max_\pi V_t^\pi(X_t) \quad . \tag{4.5}$$

The optimal state-value function, V_t^*, satisfies the Bellman optimality equation such that

$$V_t^*(X_t) = \mathbb{E}_t^{\pi^*} \left[R_t(X_t, u_t = \pi_t^*(X_t), X_{t+1}) + \gamma V_{t+1}^*(X_{t+1}) \right] \quad . \tag{4.6}$$

The Bellman equation for the action-value function, the Q-function, is defined as

$$Q_t^\pi(x, a) = \mathbb{E}_t \left[R_t(X_t, a_t, X_{t+1}) | X_t = x, a_t = a \right] + \gamma \mathbb{E}_t^\pi \left[V_{t+1}^\pi(X_{t+1}) | X_t = x \right] \quad . \tag{4.7}$$

The optimal action-value function, Q_t^*, is obtained for the optimal policy with

$$\pi_t^* = \arg\max_\pi Q_t^\pi(x, a). \tag{4.8}$$

The optimal state-value and action-value functions are connected by the following system of equations

$$\begin{cases} V_t^* = \max_a Q^*(x, a) \\ Q_t^* = \mathbb{E}_t \left[R_t(X_t, a, X_{t+1}) \right] + \gamma \mathbb{E}_t \left[V_{t+1}^*(X_{t+1} | X_t = x) \right] \end{cases} . \tag{4.9}$$

Therefore, we can obtain the Bellman optimality equation

$$Q_t^*(x, a) = \mathbb{E}_t \left[R_t(X_t, a_t, X_{t+1}) + \gamma \max_{a_{t+1} \in \mathcal{A}} Q_{t+1}^*(X_{t+1}, a_{t+1}) | X_t = x, a_t = a \right] \quad . \tag{4.10}$$

Using the Robbins-Monro update [53], the update rule for the optimal Q-function with on-line Q-learning on the data point $(X_t^{(n)}, a_t^{(n)}, R_t^{(n)}, X_{t+1}^{(n)})$ is expressed by the following equation with α a constant step-size parameter

$$Q_t^{*,k+1}(X_t, a_t) = (1-\alpha^k)Q_t^{*,k}(X_t, a_t) + \alpha^k \left[R_t(X_t, a_t, X_{t+1}) + \gamma \max_{a_{t+1} \in \mathcal{A}} Q_{t+1}^{*,k}(X_{t+1}, a_{t+1}) \right] \quad . \tag{4.11}$$

4.4 Algorithm

We describe, in this section, how to derive a general recursive formulation for the optimal action. It is equivalent to an optimal hedge under a financial framework such as, for instance, portfolio or personal finance optimization. We additionally present the formulation of the action-value function, the Q-function. Both the optimal hedge and the Q-function follow the assumption of a continuous space scenario generated by the Vasicek model with Monte Carlo simulation. For the ease of understanding, we explain the meaning of the RL concepts into the retail banking world. The states are defined as the amount of money available, the actions as spending or receiving money in your bank account, the reward as the probability of avoiding a payment default, and the policy as the mapping from the perceived states (the amount of money) to actions (credit or debit transactions). Figure 4.2 gathers these RL concepts.

By relying on the financial framework established in [129], we consider a mean reverting diffusion process, also known as the Vasicek model [144],

$$dS_t = \kappa(b - S_t)dt + \sigma dB_t \quad . \tag{4.12}$$

The term κ is the speed reversion, b the long term mean level, σ the volatility and B_t the Brownian motion. The solution of the stochastic equation is equal to

$$S_t = S_0 e^{-\kappa t} + b(1 - e^{-\kappa t}) + \sigma e^{-\kappa t} \int_0^t e^{\kappa s} dB_s \quad . \tag{4.13}$$

Therefore, we define a new time-uniform state variable, i.e. without a drift, as

$$\begin{cases} S_t = X_t + S_0 e^{-\kappa t} + b(1 - e^{-\kappa t}) \\ \text{with } X_t = \sigma e^{-\kappa t} \int_0^t e^{\kappa s} dB_s - [S_0 e^{-\kappa t} + b(1 - e^{-\kappa t})] \end{cases} . \tag{4.14}$$

We illustrate the idea of the time-uniform variable in Figures 4.3, 4.4 and 4.5. In the context of finance and retail banking, a time-uniform variable is a variable that is independent of the concepts of inflation and deflation. Instead of estimating the

price of a vanilla option as proposed in [129], we are interested to estimate the future probability of an event using the Q-learning algorithm and a digital function. We recall the digital function is at the center of our approach with the Vasicek model. It effectively allows the modeling of the default probabilities, a key aspect to determine the optimal policy for money management. We first define the terminal condition reflecting that with the following equation

$$Q_T^*(X_T, a_T = 0) = -\Pi_T - \lambda Var\left[\Pi_T(X_T)\right] \quad , \tag{4.15}$$

where Π_T is the digital function at time $t = T$ defined such that

$$\Pi_T = 1_{S_T \geq K} = \begin{cases} 1 \text{ if } S_T \geq K \\ 0 \text{ otherwise} \end{cases} \quad , \tag{4.16}$$

and the second term, $\lambda Var\left[\Pi_T(X_T)\right]$, is a regularization term with $\lambda \in \mathbb{R}^+ \ll 0$. The digital function can be approximated using the BSM formula by combining vanilla options, as illustrated in Figure 4.6. We use a backward loop to determine the value of Π_t for $t = T - 1, ..., 0$

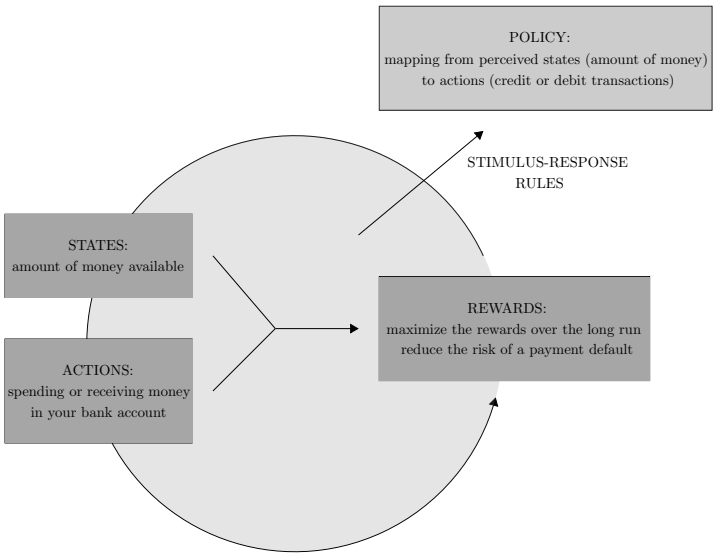

Figure 4.2 In MQLV, reinforcement learning is applied to the retail banking. The optimal policy of money management is learned by maximizing the rewards, defined as avoiding a payment default, based on state-action sequences, respectively spending or receiving money in your bank account.

$$\Pi_t = \gamma\left(\Pi_{t+1} - a_t \Delta S_t\right) \quad \text{with} \quad \Delta S_t = S_{t+1} - \frac{S_t}{\gamma} = S_{t+1} - e^{r\Delta t} S_t \quad . \tag{4.17}$$

Following the definition of the equations (4.6) and (4.17), we express the one-step time dependent random reward with respect to the cross-sectional information $\mathcal{F}_t$ as follows

$$\begin{aligned} R_t(X_t, a_t, X_{t+1}) &= \gamma a_t \Delta S_t(X_t, X_{t+1}) - \lambda Var\left[\Pi_t | \mathcal{F}_t\right] \\ &\text{with } Var\left[\Pi_t | \mathcal{F}_t\right] = \gamma^2 \mathbb{E}_t\left[\hat{\Pi}_{t+1}^2 - 2a_t \Delta\hat{S}_t \hat{\Pi}_{t+1} + a_t^2 \Delta\hat{S}_t^2\right] \end{aligned} \quad . \tag{4.18}$$

The term $\Delta\bar{S}_t$ is defined such that $\Delta\bar{S}_t = \frac{1}{N}\Delta S$, $\Delta\widehat{S} = \Delta S - \Delta\bar{S}_t$ and $\hat{\Pi}_{t+1} = \Pi_{t+1} - \bar{\Pi}_{t+1}$ with $\bar{\Pi}_{t+1} = \frac{1}{N}\Pi_{t+1}$. Because of the regularizer term, the expected reward R_t is quadratic in a_t and has a finite solution. We therefore inject the one-step time dependent random reward equation (4.18) into the Bellman optimality equation (4.10) to obtain the following Q-learning update, Q^*, and the optimal action, a^*, to be solved within a backward loop $\forall t = T-1, ..., 0$

$$\begin{aligned} Q_t^*(X_t, a_t) &= \gamma \mathbb{E}_t\left[Q_{t+1}^*(X_{t+1}, a_{t+1}^*) + a_t \Delta S_t\right] - \lambda Var\left[\Pi_t | \mathcal{F}_t\right] \\ a_t^*(X_t) &= \mathbb{E}_t\left[\Delta\hat{S}_t \hat{\Pi}_{t+1} + \frac{1}{2\lambda\gamma}\Delta S_t\right]\left[\mathbb{E}_t\left[\left(\Delta\hat{S}_t\right)^2\right]\right]^{-1} \end{aligned} \quad . \tag{4.19}$$

We refer to [129] for further details about the analytical solution, a^*, of the Q-learning update (4.19). Our approach uses the N Monte Carlo paths simultaneously to determine the optimal action a^* and the optimal action-value function Q^* to learn the policy π^*. We thus do not need an explicit conditioning of X_t at time t. We assume a set of M basis function $\{\Phi_n(x)\}$, with $n = 1, \ldots, M$, for which the optimal action $a_t^*(X_t)$ and the optimal action-value function, $Q_t^*(X_t, a_t^*)$, can be expanded

$$a_t^*(X_t) = \sum_n^M \phi_{nt}\Phi_n(X_t) \quad \text{and} \quad Q_t^*(X_t, a_t^*) = \sum_n^M \omega_{nt}\Phi_n(X_t) \quad . \tag{4.20}$$

The coefficients ϕ and ω are computed recursively backward in time $\forall t = T-1, \ldots, 0$. We subsequently define the minimization problem to evaluate ϕ_{nt} such that

$$G_t(\phi) = \sum_{k=1}^N \left[-\sum_n^M \phi_{nt}\Phi_n(X_t^k)\Delta S_t^k + \gamma\lambda\left(\Pi_{t+1}^k - \sum_n^M \phi_{nt}\Phi_n(X_t^k)\Delta\widehat{S}_t^k\right)^2\right] \quad . \tag{4.21}$$

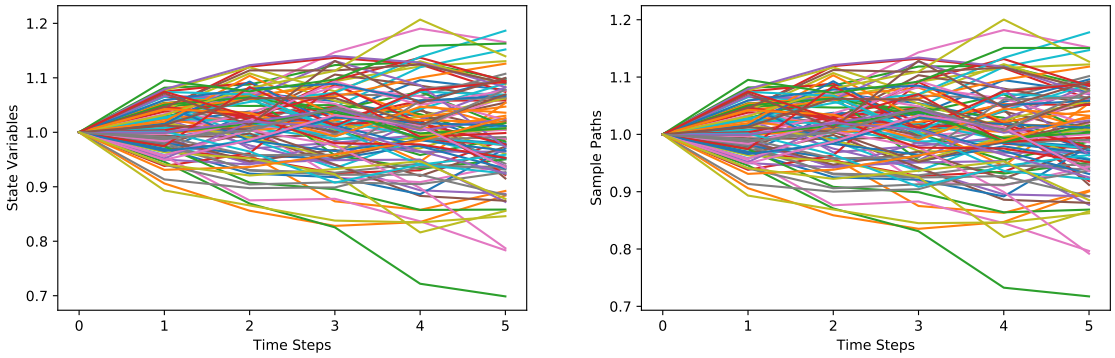

Figure 4.3 Time uniform state variables (left) in the context of a null inflation sample paths (right). Both curves are similar since there is no time-dependency modeled here in the case of a null inflation.

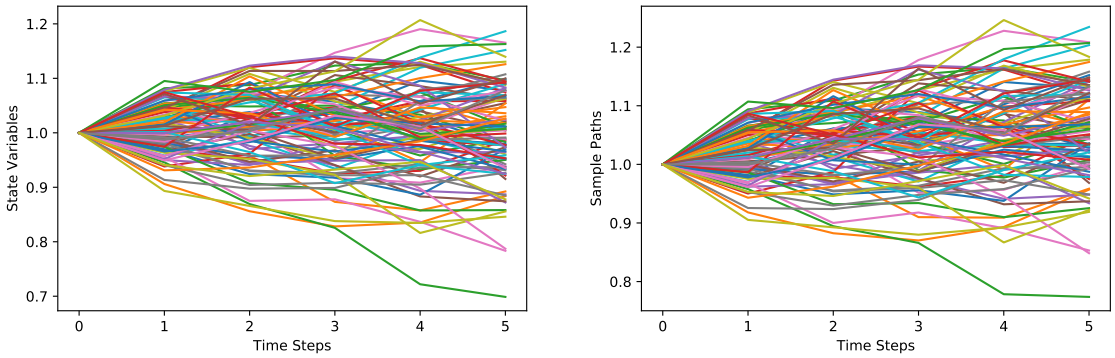

Figure 4.4 Time uniform state variables (left) in the context of a strong inflation (right). The time uniform state variables only capture the fluctuation of the transactions. Therefore, the long term impact of the inflation leading to the strong increase of the sample paths has no effect on the state variables.

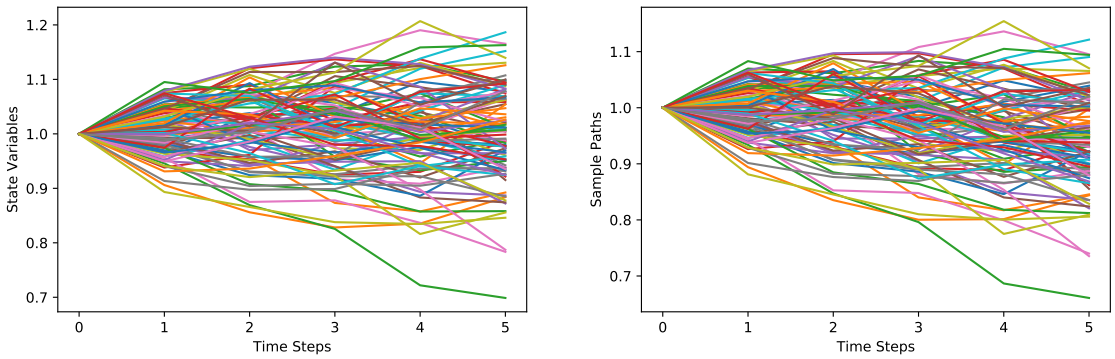

Figure 4.5 Time uniform state variables (left) in the context of a strong deflation (right). The deflation leads to the sample paths decrease. The time uniform state variables, however, are not impacted and only capture the fluctuation of the transactions.

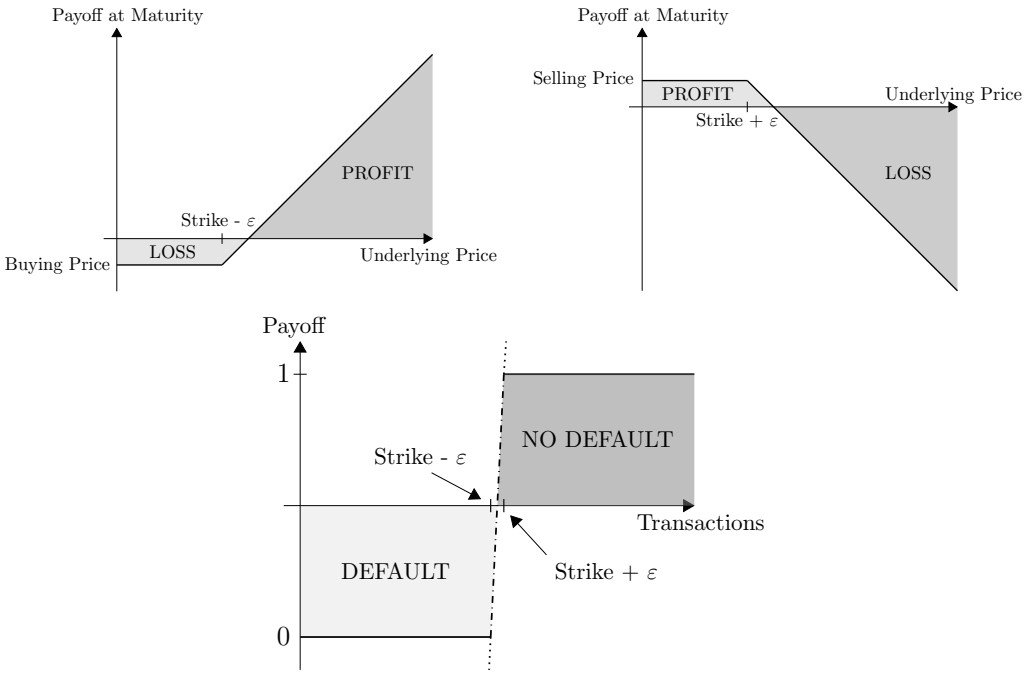

Figure 4.6 The BSM formula is commonly used to price vanilla options. The combination of vanilla options at different strikes allows the replication of the digital function. In our case, the BSM formula allows to replicate the thresholds for which we observe a payment default. It offers a benchmark comparison of the values found by our approach denoted by MQLV.

The equation (4.21) leads to the following set of linear equations $\forall n = 1, \ldots, M$

$$\begin{cases} A_{nm}^{(t)} = \sum_{k=1}^{N} \Phi_n(X_t^k)\Phi_m(X_t^k)(\Delta \widehat{S}_{t^k})^2 \\ B_n^{(t)} = \sum_{k=1}^{N} \Phi_n(X_t^k) \left[\widehat{\Pi}_{t+1}^k \Delta \widehat{S}_t^k + \frac{1}{2\gamma\lambda} \Delta S_t^k \right] \end{cases} \quad \text{with } \sum_{m}^{M} A_{nm}^{(t)} \phi_{mt} = B_n^{(t)} \quad . \tag{4.22}$$

The coefficients of the optimal action $a_t^*(X_t)$ are therefore determined by

$$\phi_t^* = A_t^{-1} B_t \quad . \tag{4.23}$$

We hereinafter use the Fitted Q Iteration (FQI) [149, 156] to evaluate the coefficients ω. The optimal action-value function, $Q^*(X_t, a_t)$, is represented in its matrix form according to the basis function expansion of the equation (4.20) such that

$$
\begin{aligned}
Q_t^*(X_t, a_t) = &\left(1, a, \frac{1}{2}a_t^2\right) \begin{pmatrix} W_{11}(t) & W_{12}(t) & \dots & W_{1M}(t) \\ W_{21}(t) & W_{22}(t) & \dots & W_{2M}(t) \\ W_{31}(t) & W_{32}(t) & \dots & W_{3M}(t) \end{pmatrix} \begin{pmatrix} \Phi_1(X_t) \\ \vdots \\ \Phi_M(X_t) \end{pmatrix} \quad . \\
= &A_t^T W_t \Phi(X_t) = A_t^T U_W(t, X_t)
\end{aligned} \tag{4.24}
$$

Based on the least-square optimization problem, the coefficients W_t are determined using backpropagation $\forall t = T-1, ..., 0$ as follows

$$
\begin{aligned}
\mathcal{L}_t(W_t) = &\sum_{k=1}^{N} \left(R_t(X_t, a_t, X_{t+1}) + \gamma \max_{a_{t+1} \in \mathcal{A}} Q_{t+1}^*(X_{t+1}, a_{t+1}) - W_t \Psi_t(X_t, a_t) \right)^2 \quad . \\
&\text{with } W_t \Psi(X_t, a_t) + \epsilon \underset{\epsilon \to 0}{\longrightarrow} R_t(X_t, a_t, X_{t+1}) + \gamma \max_{a_{t+1} \in \mathcal{A}} Q_{t+1}^*(X_{t+1}, a_{t+1})
\end{aligned} \tag{4.25}
$$

for which we derive the following set of linear equations

$$
\begin{cases}
M_n^{(t)} = \displaystyle\sum_{k=1}^{N} \Psi_n(X_t^k, a_t^k) \left[\eta \left(R_t(X_t, a_t, X_{t+1}) + \gamma \max_{a_{t+1} \in \mathcal{A}} Q_{t+1}^*(X_{t+1}, a_{t+1}) \right) \right] \quad . \\
\text{with } \eta \sim B(N, p)
\end{cases} \tag{4.26}
$$

The term $B(N, p)$ represents the binomial distribution for n samples with probability p. It plays the role of a dropout function when evaluating the matrix M_t to compensate the well-known drawback of the Q-learning algorithm that is the overestimation of the Q-function values. We finally reach the definition of the optimal weights to determine the optimal action a^* as follows

$$
W_t^* = S_t^{-1} M_t \quad . \tag{4.27}
$$

The proposed model does not require any assumption on the dynamics of the time series, neither transition probabilities nor policy or reward functions. It is an off-policy model-free approach. We highlight the overall scheme of MQLV with aggregated transactions in Figure 4.7. The computation of the optimal policy, the optimal actions and the optimal Q-function is summarized in Algorithm 6.

Algorithm 6: Modified Q-learning for the Vasicek model with the digital value function Π

Data: time series of maturity T, either from generated or true data
Result: optimal Q-function Q^*, optimal action a^*, value of digital function Π

begin
  /**Condition at T**/
  $a_T^*(X_T) = 0$
  $Q_T(X_T, a_T) = -\Pi_T = -1_{S_T \geq K}$ using equation (4.16)
  $Q_T^*(X_T, a_T^*) = Q_T(X_T, a_T)$
  /**Backward Loop**/
  for $t \leftarrow T-1$ **to** 0 **do**
    /**Evaluate the coefficients ϕ**/
    compute A_t, B_t using equation (4.22)
    $\phi_t^* \leftarrow A_t^{-1} B_t$
    /**Evaluate the coefficients ω**/
    compute S_t, M_t using equation (4.26)
    $W_t^* \leftarrow S_t^{-1} M_t$
    $a_t^*(X_t) = \sum_n^M \phi_{nt}^* \Phi_n(X_t)$
    $Q^*(X_t, a_t) = A_t^T W_t^* \Phi_(X_t)$
  /**Compute the digital function value to estimate the event probability at $t = 0$**/
  `print`$(\Pi_0 = mean(Q_0^*))$
return

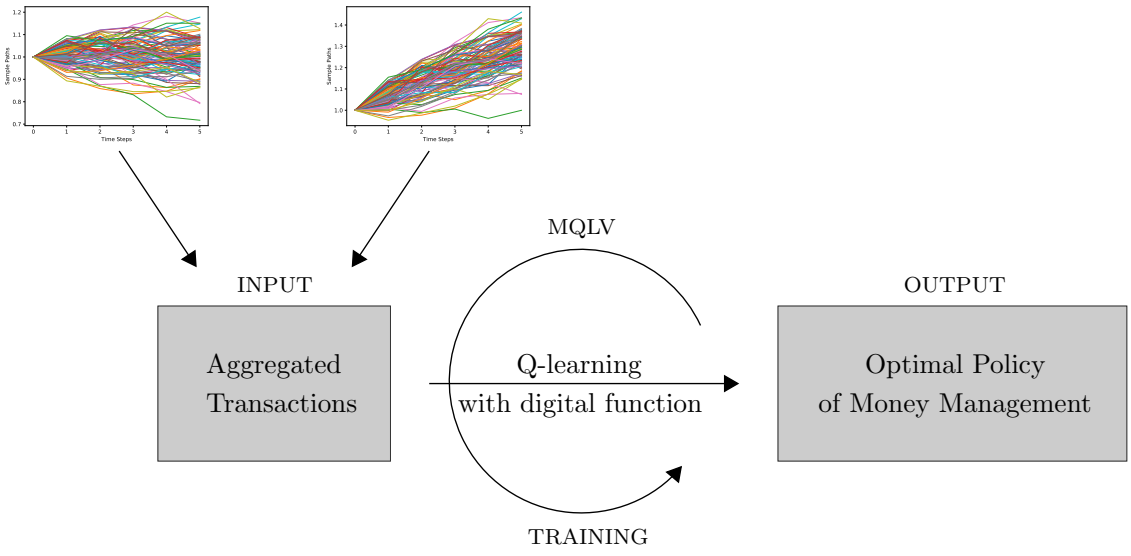

Figure 4.7 MQLV takes as input the aggregated financial transactions. The training is performed using the Bellman equation updated with the digital function. At convergence, the optimal policy of money management is obtained.

4.5 Experiments

We empirically evaluate the performance of MQLV. We initially highlight the similarities between historical payment transactions and Vasicek generated transactions. We then evaluate the Q-values overestimation with the closed formula of [131, 132], hereinafter denoted by BSM's closed formula. We finally underline the MQLV's capabilities to learn the optimal policy of money management based on the estimation of future event probabilities in comparison to the BSM's closed formula. We rely on synthetic data sets because of the privacy and the confidentiality issues of the retail banking data sets.

4.5.1 Data Availability and Data Description

One of our contributions is to bring a RL framework designed for retail banking. However, none of the data sets can be released publicly because of the highly sensitive information they contain. We therefore show the similarities between a small sample of anonymized transactions and Vasicek generated transactions [144]. We then use the Vasicek mean reverting stochastic diffusion process to generate larger synthetic data sets similar to the original retail banking data sets. The mean reverting dynamic is particularly interesting since it reflects a wide range of retail banking transactions including the credit card transactions, the savings history or the clients' spendings. Three different data sets were generated to avoid any bias that could have been introduced by using only one data set. We choose to differentiate the number of Monte Carlo paths between the data sets to assess the influence of the sampling size on the results. The first, second and third data sets contain 20,000, 30,000 and 40,000 paths. We release publicly the data sets[1] to ensure the reproducibility of the experiments.

4.5.2 Experimental Setup and Code Availability

In our experiments, we generate synthetic data sets using the Vasicek model with a parameter $S_0 = 1.0$ corresponding to the value of the time series at $t = 0$, a maturity of six months $T = 0.5$, a speed reversion $a = 0.01$, a long term mean $b = 1$ and a volatility $\sigma = 0.15$. The numbers were fixed such that any limitations of the methodology would be quickly observed because the choice of the parameters of the Vasicek model does

[1]The code and the data sets are available at https://github.com/dagrate/MQLV.

not have any influence on the results of the Q-learning approach. The number of time steps is fixed equal to 5. We additionally use different strike values for the experiments explicitly mentioned in the Results and Discussions subsection. The simulations were performed on a computer with 16GB of RAM, Intel i7 CPU and a Tesla K80 GPU accelerator. To ensure the reproducibility of the experiments, the code is available at the address[1] aforementioned.

4.5.3 Results and Discussions

We first highlight the similarities between the dynamic of a small sample of anonymized transactions and Vasicek generated transactions for one client [157] in Figure 4.8 because we cannot release publicly an anonymized transactions data set due to privacy, confidentiality and regulatory issues. The financial transactions in retail banking are periodic and often fluctuates around a long term mean, reflecting the frequency and the amounts of the spendings habits of the clients. The akin dynamic of the original and the generated transactions is highlighted by the small RMSE of 0.03. We also performed a least square calibration of the Vasicek parameters to assess the model's plausibility. We can observe in Table 4.1 that the Vasicek parameters have the same magnitude and, therefore, it supports the hypothesis that the Vasicek model could be used to generate synthetic transactions.

We then highlight the motivations of the dropout function of MQLV, capable to limit the overfit of the Q-values. We use the standard example of vanilla option to facilitate the comparison with the BSM's closed formula [131, 132], used as reference values for the cross-validation of our approach. The absolute difference between the standard Q-learning and the dropout Q-learning algorithms with respect to the BSM's closed formula values are computed in Figure 4.9 for a European vanilla call option. The absolute differences of the BSM's closed formula are all equal to zero since they are the reference values. We can observe that the standard Q-learning update leads to higher differences for all strikes, between 1% and 2% depending on the threshold values. When applying the dropout function, the values obtained are closer to the target values, therefore minimizing the absolute difference. These observations are confirmed quantitatively in Table 4.2. The valuation differences are reported for the BSM's closed formula, the standard and the dropout Q-learning updates for the three generated data

sets. The values of the dropout Q-learning approach are always closer to the BSM's target values. We can conclude that, for our simulation, the dropout update of the equation (4.26) reduces the discrepancy with the BSM's target values.

We present the core of our contribution in the following experiment. We aim at learning the optimal policy of money management. It is particularly interesting for bank loan applications where the differences between a client's spendings policy and the optimal policy can be compared. We show that MQLV is capable of evaluating accurately the probability of a default event using a digital function, which highlights the learning of the optimal policy of money management. Effectively, if the MQLV's learned policy is different than the optimal policy, then the probabilities of default events are not accurate. The estimation of future event probabilities for different strike values is represented in Figure 4.10. We rely on the BSM's closed formula for the vanilla option pricing [131, 132] to approximate the digital function values [143]. We used, therefore, the BSM's values as reference values to cross-validate the MQLV's values. MQLV achieves a close representation of the event probabilities for the different strike values in Figure 4.10. The curves of both the MQLV and the BSM's approaches are similar with a RMSE of 1.5016. This result highlights that the learned Q-learning policy of MQLV is sufficiently close to the optimal policy to compute event probabilities almost identical to the probabilities of the BSM's formula approximation.

We gathered quantitative results in Table 4.3 for a deeper analysis of the MQLV's results. The event probability values are listed for the three data sets. We chose a set of parameters for the Vasicek model such that our configuration is free of any time-dependency. We therefore expect a probability value of 50% at a threshold of 1 because the standard deviation of the generated data sets is only induced by the standard deviation of the normal distribution, used to simulate the Brownian motion. Surprisingly, the MQLV values at a strike of 1 are closer to 50% than the BSM's values for all the data sets. We can conclude, subsequently, that, for our configuration, MQLV is capable to learn the optimal policy of money management which is reflected by the accurate evaluation of the event probabilities.

We chose to generate three new data sets with new Vasicek parameters a and σ to underline the potential of MQLV and the universality of the results. In Table 4.4, we computed the event probabilities for different strikes for the newly generated data sets.

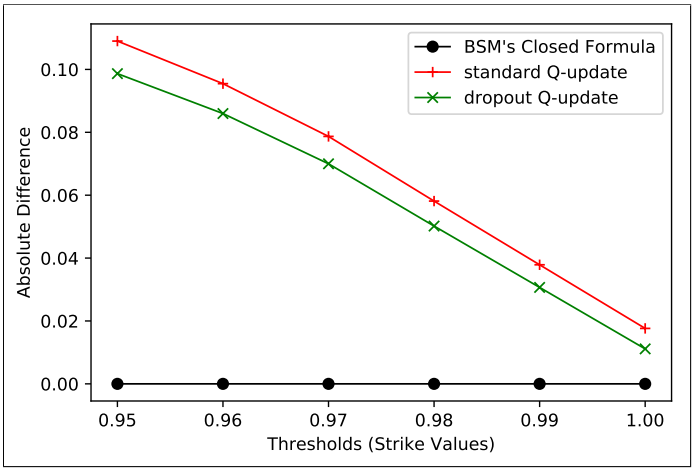

Figure 4.9 Absolute differences between the standard and the dropout Q-learning updates for the price evaluation of a European vanilla call. The benchmark of the Q-learning algorithms is the BSM's closed formula and, therefore, the differences are set to 0. The dropout Q-update limits the overestimation of the Q-values leading to smaller differences with the BSM's closed formula.

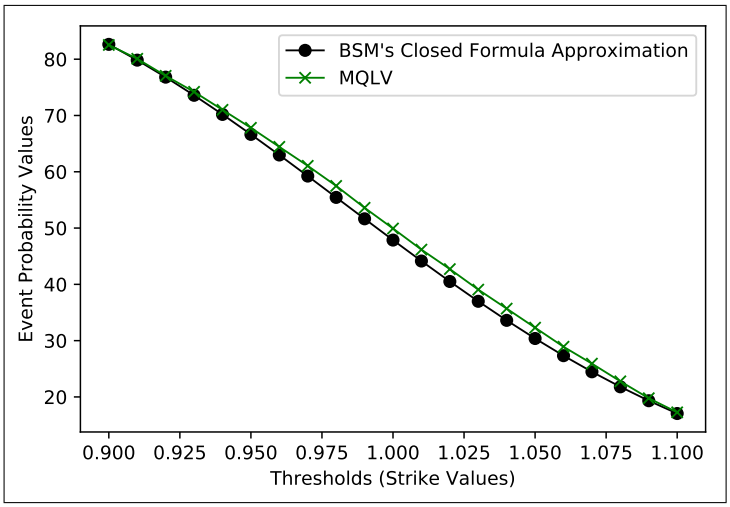

Figure 4.10 Event probability values calculated by MQLV and BSM's closed formula approximation for 40,000 Monte Carlo paths with Vasicek parameters $a = 0.01, b = 1$ and $\sigma = 0.15$. The BSM's closed formula approximation values are used as reference values. The event probabilities of MQLV are close to the BSM's values with a total RMSE of 1.502. It illustrates that MQLV is able to learn the optimal policy leading to accurate event probabilities.

The parameter b remains unchanged since we want to keep a configuration free of any time-dependency. We notice that MQLV is capable to estimate a probability of 50% for a strike of 1 which can only be obtained if MQLV is able to learn the optimal policy. We also observe that the BSM's approximation does lead to a lower accuracy. We showed in this experiment that our model-free and off-policy RL approach, MQLV, is able to learn the optimal policy reflected by the accurate probability values independently of the data sets considered and of the Vasicek parameters.

Limitations of the BSM's closed formula used for cross validation In our experiments, we observed, surprisingly, that the BSM's closed formula approximation underestimates the event probability values. The volatility is the only parameter playing a significant role in the generation of the time series and, therefore, the event probability should be equal to the mean of the distribution used to generate the random numbers. The Brownian motion is simulated with a standard normal distribution with a 0.5 mean. The BSM's closed formula did not, however, lead to a probability of 0.5 but to slightly smaller values because of the limit of their theoretical framework [131, 132]. We hence observed that MQLV was more accurate than the BSM's closed formula in our configuration.

4.6 Conclusion

We introduced Modified Q-Learning for Vasicek or MQLV, a new model-free and off-policy reinforcement learning approach capable of evaluating an optimal policy of money management based on the aggregated transactions of the clients. MQLV is part of a banking strategy that looks to minimize the customer churn by including more transparency and more customization in the decision process related to bank loan applications or credit card limits. It relies on a digital function, a Heaviside step function expressed in its discrete form, to estimate the future probability of an event such as a payment default. We discuss its relation with the Bellman optimality equation and the Q-learning update. We conducted experiments on synthetic data sets because of the privacy and confidentiality issues related to the retail banking data sets. The generated data sets followed a mean reverting stochastic diffusion process, the Vasicek model, simulating retail banking data sets such as transaction payments. Our experiments showed the performance of MQLV with respect to the BSM's closed formula for vanilla options. We also highlighted that MQLV is able to determine an

optimal policy, an optimal Q-function, the optimal actions and the optimal states reflected by accurate probabilities. We surprisingly observed that MQLV led to more accurate event probabilities than the popular BSM's formula in our configuration.

Future work will address the creation of a fully anonymized data set illustrating the retail banking daily transactions with a privacy, confidentiality and regulatory compliance. We will also evaluate the MQLV's performance for data sets that violate the Vasicek assumptions. We will furthermore assess the influence of the basis function on the results accuracy. We effectively observed that the Q-learning update could minor the real probability values for simulation involving a small temporal discretization such as $\Delta t = 200$. Preliminary results showed it is provoked by the basis function approximator error. We will address this point in future research. We will finally extend the Q-learning update to other scheme for improved accuracy and incorporate a deep learning framework in our approach.

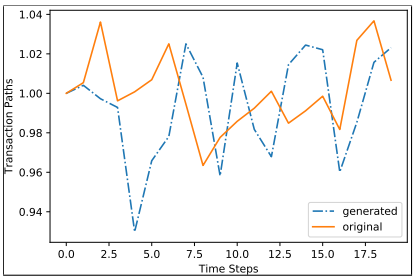

Figure 4.8 Samples of original and Vasicek generated transactions for one client. The two samples oscillate around a long term mean of 1 and have a similar pattern, highlighted by the RMSE of 0.03 in Table 4.1.

Table 4.1 RMSE error between the samples of original transactions and generated Vasicek transactions of Figure 4.8. We also calibrated the Vasicek parameters according to the original transactions to validate the model's plausibility.

Description	Value
RMSE	0.0335
Vasicek speed reversion a	0.5444
Vasicek long term mean b	0.9001
Vasicek volatility σ	0.2185

Table 4.2 Valuation differences for a vanilla call option between the BSM's closed formula, the standard Q-learning update and the dropout Q-learning update. The differences are presented for the three generated data sets for different strikes. The dropout function helps to reduce the overestimation of the Q-values. The dropout Q-values are, consequently, closer to the reference values of the BSM's closed formula.

Data Set	Number of Paths	Strike Values	BSM's Values (%)	No Dropout Q-Values (%)	Dropout Q-Values (%)
1	20,000	0.95	**7.096**	7.223	**7.214**
1	20,000	0.96	**6.448**	6.557	**6.549**
1	20,000	0.99	**4.727**	4.764	**4.759**
1	20,000	1.00	**4.230**	4.241	**4.237**
2	30,000	0.95	**7.096**	7.215	**7.208**
2	30,000	0.96	**6.448**	6.552	**6.546**
2	30,000	0.99	**4.727**	4.769	**4.764**
2	30,000	1.00	**4.230**	4.249	**4.246**
3	40,000	0.95	**7.096**	7.205	**7.195**
3	40,000	0.96	**6.448**	6.543	**6.534**
3	40,000	0.99	**4.727**	4.765	**4.758**
3	40,000	1.00	**4.230**	4.247	**4.240**

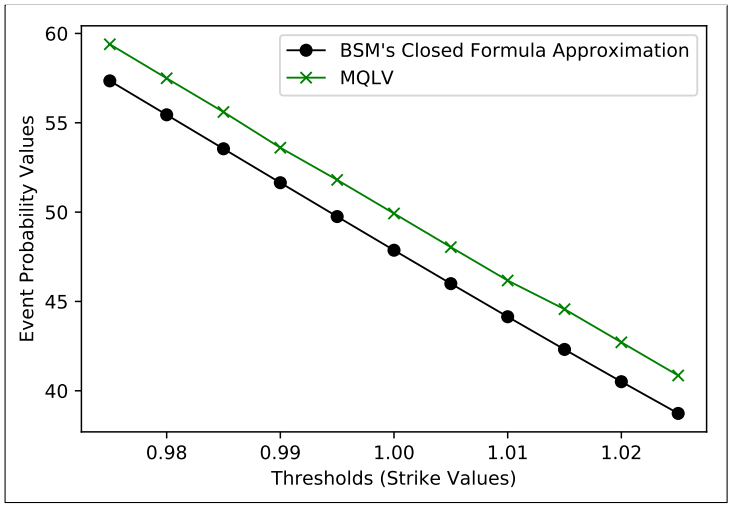

Figure 4.11 Zoom in of the results pictured in Figure 4.10. The differences between MQLV and BSM are further highlighted. Both curves have the same slope of -381 between the strike values of 0.98 and 1.02. We can already highlight that the MQLV's probabilities illustrate the MQLV's ability to learn the optimal policy.

Table 4.3 Valuation differences of the digital values for event probabilities according to different strikes between the BSM's closed formula approximation and MQLV. Given our time-uniform configuration, the event probability values should be close to 50% for a strike value of 1. The MQLV values are close to the theoretical target of 50% at a strike of 1 highlighting the MQLV's capabilities to learn the optimal policy. The BSM's closed formula approximation slightly underestimates the probability values.

Data Set	Number of Paths	Strike Values	BSM's Approx. Values (%)	MQLV Values (%)	Absolute Difference
1	20,000	0.92	76.810	**77.098**	0.288
1	20,000	0.98	55.447	**57.920**	2.473
1	20,000	1.00	47.867	**50.235**	2.368
1	20,000	1.02	40.509	**42.865**	2.356
2	30,000	0.92	76.810	**76.953**	0.143
2	30,000	0.98	55.447	**57.760**	2.313
2	30,000	1.00	47.867	**50.043**	2.176
2	30,000	1.02	40.509	**42.744**	2.235
3	40,000	0.92	76.810	**77.047**	0.237
3	40,000	0.98	55.447	**57.491**	2.044
3	40,000	1.00	47.867	**49.924**	2.057
3	40,000	1.02	40.509	**42.713**	2.204

Table 4.4 Event probabilities for data sets generated with different Vasicek parameters a and σ. The parameter b remains unchanged to keep a time-uniform configuration to facilitate the results explainability. We can deduce that MQLV is able to learn the optimal policy because the MQLV's probabilities are close to the theoretical target of 50% at a strike of 1. MQLV is also more accurate than BSM's formula in this configuration.

Parameters $a; b; \sigma$	Number of Paths	Strike Values	BSM's App. Values (%)	MQLV Values (%)	Absolute Difference
0.01; 1; 0.10	50,000	0.98	59.856	**61.223**	1.366
0.01; 1; 0.10	50,000	1.00	48.562	**50.001**	1.439
0.01; 1; 0.10	50,000	1.02	37.596	**39.044**	1.447
0.01; 1; 0.30	50,000	0.98	49.558	**53.647**	4.089
0.01; 1; 0.30	50,000	1.00	45.767	**49.997**	4.230
0.01; 1; 0.30	50,000	1.02	42.088	**46.194**	4.106
0.10; 1; 0.15	50,000	0.98	55.447	**57.540**	2.093
0.10; 1; 0.15	50,000	1.00	47.867	**50.015**	2.148
0.10; 1; 0.15	50,000	1.02	40.509	**42.638**	2.129
0.30; 1; 0.15	50,000	0.98	55.447	**57.586**	2.139
0.30; 1; 0.15	50,000	1.00	47.867	**50.022**	2.155
0.30; 1; 0.15	50,000	1.02	40.509	**42.542**	2.033

Chapter 5

Conclusion

In this concluding chapter, we resume the accomplishments of this research project and we point out the main contributions. Throughout the different chapters, we proposed advanced analytical techniques and state of the art computer science concepts to answer the specific needs facing the retail banking industry in a context of evolving regulations and new clients' behavior. We first highlighted how to overcome the lack of financial data using generative adversarial models and persistent homology. We then addressed multidimensional financial recommendations targeting new or renewed products subscriptions for improved marketing banking strategy. We finally introduced a model-free reinforcement learning approach specifically designed for personal financial advice and custom-tailored requests, trying to limit the customer churn.

5.1 Persistent Homology for Generative Models to Generate Artificial Financial Data

In our current modern age where digital traces and data are omnipresent, the retail banking industry frequently encounters situations where an insufficient amount of data is available to evaluate accurately the risks of the bank or to propose adequate solutions to the clients. In car loan applications for instance, some age categories of the applicants or loan amounts are historically under represented. Extending the available data is a crucial need. We therefore introduced Persistent Homology for Generative Models. In PHom-GeM, we used four different types of neural networks, including gradient penalty Wasserstein generative adversarial network, generative adversarial network, Wasserstein auto-encoders and variational auto-encoders, to generate

synthetic credit card transactions based on a small sample of existing credit card transactions. PHom-GeM characterizes the generative manifolds using persistence homology to highlight manifold features of existing and artificially generated data. We chose to apply PHom-GeM on credit card transactions since the latter is particularly challenging for neural networks and persistent homology. Given the large number of features per transactions, we underline in our experiments that PHom-GeM is able to build effective representation of a high dimensional data set. We moreover introduced the bottleneck distance to compare quantitatively the differences between the original and the generated transaction samples.

PHom-Gem establishes new opportunities to artificially generate and evaluate high dimensional data of retail banking. However, several questions still need to be addressed in future research. First, we used the popular Vietoris-Rips simplicial complex for the construction of the simplex tree. It is known to offer a good compromise between the computational cost and the accuracy of the results. But what is the influence of the simplicial complex? Second, we used the bottleneck distance to compare quantitatively the persistence diagram samples. Given the recent progress of the theoretical framework around optimal transport, the use of the Wasserstein distance for the quantitative comparison between the persistence diagrams of the different samples should provide interesting insights. Finally, PHom-GeM trains the neural network models and compare the original and the artificial samples in a two-steps procedure. We leave for future research how to back-propagate the persistent homology information to the objective loss function of the generative models to improve their training and the generation of adversarial samples[1].

5.2 Accurate Tensor Resolution for Multidimensional Financial Recommendation

The European financial regulation authorities have been increasing regulatory pressure on all the financial actors since the financial crisis of 2008. Under the revised Payment Service Directive aiming the retail banking industry, the banks are losing partly their historical privileges such as the distribution of the credit card payment solutions or

[1]This chapter of the thesis was published in the 6th IEEE Swiss Conference on Data Science (SDS) and in the ATDA 2019 Workshop in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD).

the management of the personal finance. Targeting the clients' needs and proposing dynamically new financial products is part of a marketing banking strategy to retain the clients. Nonetheless, the task is complex and highly dimensional. We therefore build upon tensor decomposition, a collection of factorization techniques for multidimensional arrays, and neural networks. We first designed an optimized resolution algorithm for the resolution of complex tensor decomposition. Our resolution algorithm, VecHGrad for Vector Hessian Gradient, uses partial information of the second derivative and an adaptive strong Wolfe's line search to ensure faster convergence. We showed the superior performance of VecHGrad on several well-known research data sets, such as CIFAR-10, CIFAR-100 and MNIST, against the state of the art optimizers used in deep learning and linear algebra, including alternating least square, Adam, Adagrad and RMSProp. We then arrived at our first application in the context of financial recommendations. We predicted the next financial actions of the bank's clients in a sparse environment. Such approach is of particular interest for the renewal of consumer loans or the renewal of credit cards subscriptions. We used a public transactions data set of the Santander bank. Our predictive method removes the sparsity of the financial transactions before predicting the future client's transactions. The CP tensor decomposition, which decomposes the initial tensor containing the financial transaction into a sum of rank-one tensors, removes the transactions sparsity. The next financial transactions are then predicted using different type of neural networks. We obtained the best predictions when using Long Short Term Memory neural network because of the recurrent nature of the financial transactions activities. In our second financial application, we addressed the authentication on mobile banking application. The traces of the mobile banking application are of particular interest since they can be used to build financial awareness profiles for every clients. We monitored and predicted imbalanced user-device authentication with the PARATUCK2 tensor decomposition and neural networks. The PARATUCK2 tensor decomposition is highly suitable for imbalanced data sets since it expresses a tensor as a multiplication of matrices and diagonal tensors. We relied on a public data set of user-device authentication proposed by the Los Alamos National Laboratory enterprise network for our quantitative experiments. Similarly to our first application, the best results were obtained with Long Short Term Memory neural network because of the recurrent and cyclic patterns of the user-device authentication.

Based on the recent publications in computer science conferences, we do believe that some remaining opened questions about adaptive line search will be addressed in future research. We noticeably observed that the performance of the VecHGrad algorithm is strongly correlated to the performance of the the adaptive line search optimization. In the context of big data and very large data bases, the memory cost of the adaptive line search additionally has a crucial impact. In our client's transactions predictions case study, a different time frame discretization could furthermore be assessed. It would contribute to address a larger choice of financial product recommendations depending on the clients' mid-term and long-term interests. Concerning our second financial applications, the traces of the mobile banking application could finally be further enriched by incorporating traces such as the navigation usage or the time gap between each actions. It would further improve the design of financial awareness profile and, consequently, the potential impact of the bank's advertising campaigns[2].

5.3 Reinforcement Learning for Decision Making Process in Retail Banking

Following the new habits of younger generations and the financial regulations promoting more competition, the retail banks face a significant risk of high customer churn. Everyone nowadays can easily change from one bank to another to benefit from more attractive financial opportunities. In this chapter, we described a reinforcement learning approach at the core of the banking strategy to limit the customer churn in the retail banking market. The approach is tailored for personal policy of money management to be used for bank loan applications or credit card limits. We presented MQLV, Modified Q-Learning for the Vasicek model, a model-free and off-policy reinforcement learning approach capable of evaluating the optimal policy of money management based on the aggregated transactions of the clients. We introduced a digital function, a Heaviside step function expressed in its discrete form, to estimate the future probability of an event such as a payment default. This contributed to the evaluation of the optimal policy of money management helping to determine the limits on the credit cards or

[2]This chapter of the thesis was published in the 2017 IEEE Future Technologies Conference (FTC), the 2018 IEEE Big Data and Smart Computing (BigComp) Conference, the 6th International Workshop on Data Science and Big Data Analytics (DSBDA) in conjunction with IEEE International Conference on Data Mining (ICDM 2018) and the 32nd Canadian Conference on Artificial Intelligence. This chapter was also submitted to the thirty-fourth annual conference of the Association for the Advancement of Artificial Intelligence (AAAI).

the amount of a bank loan. In our experiments, we generated artificial data sets reflecting the transaction data sets observable in retail banking because of the privacy, confidentiality and regulatory issues of the clients' historical transactions. Different data sets with different configuration were generated to ensure the legitimacy of the results. We showed the ability of MQLV to learn an optimal policy reflected by the accurate estimation of event probabilities. We compared the MQLV's probabilities with the probabilities computed with the BSM's closed formula. We successfully determined an optimal policy, an optimal Q-function, the optimal actions and the optimal states.

Reinforcement learning applied to the retail banking industry for money management is a very promising concept. We established the foundations for future research in this field. In our opinion, the first research direction should address the creation of a full anonymized data set of high quality illustrating the retail banking daily transactions with a privacy, confidentiality and regulatory compliance. The second research direction should emphasize the choice of the basis function in comparison to the event probability estimation. Effectively, we observed that some fine temporal time discretization could underestimate the real probability values. Considering the recent success of deep learning with reinforcement learning applied to video games, we believe that the approach could benefit in using deep learning for function approximators. Finally, we used the popular Q-learning algorithm for our approach. Nonetheless, a large variety of other algorithms are available, such as the double Q-learning algorithm, offering wide possibilities to extend the MQLV framework[3].

[3]This chapter of the thesis was published in the MIDAS 2019 Workshop in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD).

References

[1] European Parliament and European Council. Directive 2014/65/eu of the European parliament and of the council of 15 May 2014 on markets in financial instruments and amending directive 2002/92/ec and directive 2011/61/EU, 2014. [Online; accessed 26-April-2019].

[2] International Accounting Standards Board. International Financial Reporting Standard 9, 2014. [Online; accessed 26-April-2019].

[3] Basel Committee on Banking Supervision. Third Basel Accord, 2010. [Online; accessed 26-April-2019].

[4] European Commission. Payment Services (PSD 2) - Directive (EU), 2015. [Online; accessed 26-April-2019].

[5] Deloitte Luxembourg. Digital banking benchmark, 2017. [Online; accessed 20-April-2019].

[6] The Nielsen Company (US) LLC. Q1 2018 nielsen total audience report, 2018. [Online; accessed 20-April-2019].

[7] International Fintech UAB. Millennials: Good news for fintech, bad news for banks, 2018. [Online; accessed 20-April-2019].

[8] Viacom Media Networks. The millennial disruption index, 2015. [Online; accessed 20-April-2019].

[9] PricewaterhouseCoopers. PwC's 2018 digital banking consumer survey: Mobile users set the agenda, 2018. [Online; accessed 20-April-2019].

[10] Brett King. *Bank 4.0: Banking Everywhere, Never at a Bank*. Wiley, 2018.

[11] Stuart Lloyd. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[12] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[13] Gene Golub and William Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 2(2):205–224, 1965.

[14] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3), 2009.

[15] William A Belson. Matching and prediction on the principle of biological classification. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 8(2):65–75, 1959.

[16] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[17] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[18] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[19] Jeremy Charlier, Francois Petit, Gaston Ormazabal, and Radu State. Visualization of AE's Training on Credit Card Transactions with Persistent Homology. In *Workshop on Applications of Topological Data Analysis (ATDA) in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECML-PKDD)*, 2019.

[20] Jeremy Charlier, Radu State, and Jean Hilger. PHom-GeM: Persistent Homology for Generative Models. In *The 6th Swiss Conference on Data Science (SDS)*, 2019.

[21] Jeremy Charlier, Radu State, and Jean Hilger. VecHGrad for Solving Accurately Complex Tensor Decomposition. In *Submission to the thirty-fourth annual conference of the Association for the Advancement of Artificial Intelligence (AAAI)*, 2020.

[22] Jeremy Charlier, Radu State, and Jean Hilger. Predicting Sparse Clients' Actions with CPOPT-Net in the Banking Environment. In *The 32nd Canadian Conference on Artificial Intelligence (Canadian AI)*, 2019.

[23] Jeremy Charlier, Eric Falk, Radu State, and Jean Hilger. User-Device Authentication in Mobile Banking using ApHeN for PARATUCK2 Tensor Decomposition. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 886–894. IEEE, 2018.

[24] Jeremy Charlier, Gaston Ormazabal, Radu State, and Jean Hilger. MQLV: Optimal Policy of Money Management in Retail Banking with Q-learning. In *Workshop on MIning DAta for financial applicationS (MIDAS) in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECML-PKDD)*, 2019.

[25] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[26] Frédéric Chazal and Bertrand Michel. An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists. *arXiv preprint arXiv:1710.04019*, 2017.

[27] Gene H Golub. Cf van loan. *Matrix computation, Baltimore, MD, USA*, 1989.

[28] J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3), 1970.

[29] Pentti Paatero. A weighted non-negative least squares algorithm for three-way 'PARAFAC' factor analysis. *Chemometrics and Intelligent Laboratory Systems*, 38(2), 1997.

[30] Brett W Bader, Richard A Harshman, and Tamara G Kolda. Temporal analysis of semantic graphs using ASALSAN. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007.

[31] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[32] Geoffrey E Hinton and Richard S Zemel. Autoencoders, minimum description length and Helmholtz free energy. In *Advances in neural information processing systems*, pages 3–10, 1994.

[33] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[34] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems*, pages 899–907, 2013.

[35] Cédric Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

[36] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

[37] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[38] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein Auto-Encoders. *arXiv preprint arXiv:1711.01558*, 2017.

[39] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological Persistence and Simplification. *Discrete & Computational Geometry*, 28(4):511–533, 2002.

[40] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

[41] Olivier Bousquet, Sylvain Gelly, Ilya Tolstikhin, Carl-Johann Simon-Gabriel, and Bernhard Schoelkopf. From optimal transport to generative modeling: the vegan cookbook. *arXiv preprint arXiv:1705.07642*, 2017.

[42] Afra Zomorodian and Gunnar Carlsson. Computing Persistent Homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

[43] Robert Ghrist. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

[44] Frédéric Chazal, Vin de Silva, Marc Glisse, and Steve Oudot. *The Structure and Stability of Persistence Modules*. SpringerBriefs in Mathematics. Springer International Publishing, 2016.

[45] Gurjeet Singh, Facundo Mémoli, and Gunnar E Carlsson. Topological methods for the analysis of high dimensional data sets and 3D object recognition. In *SPBG*, pages 91–100, 2007.

[46] Afra Zomorodian. Fast construction of the Vietoris-Rips complex. *Computers & Graphics*, 34(3):263–271, 2010.

[47] Monica Nicolau, Arnold J Levine, and Gunnar Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, page 201102826, 2011.

[48] Pek Y Lum, Gurjeet Singh, Alan Lehman, Tigran Ishkanov, Mikael Vejdemo-Johansson, Muthu Alagappan, John Carlsson, and Gunnar Carlsson. Extracting insights from the shape of complex data using topology. *Scientific reports*, 3:srep01236, 2013.

[49] Paul Bendich, James S Marron, Ezra Miller, Alex Pieloch, and Sean Skwerer. Persistent homology analysis of brain artery trees. *The annals of applied statistics*, 10(1):198, 2016.

[50] Kelin Xia and Guo-Wei Wei. Persistent homology analysis of protein structure, flexibility, and folding. *International journal for numerical methods in biomedical engineering*, 30(8):814–844, 2014.

[51] Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Unbalanced optimal transport: geometry and Kantorovich formulation. *arXiv preprint arXiv:1508.05216*, 2015.

[52] Matthias Liero, Alexander Mielke, and Giuseppe Savaré. Optimal entropy-transport problems and a new Hellinger–Kantorovich distance between positive measures. *Inventiones mathematicae*, 211(3):969–1117, 2018.

[53] Herbert Robbins and Sutton Monro. A stochastic approximation method. In *Herbert Robbins Selected Papers*, pages 102–109. Springer, 1985.

[54] Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. In *Advances in Neural Information Processing Systems*, pages 3440–3448, 2016.

[55] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*. Springer, 2010.

[56] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[57] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[58] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Rmsprop: Divide the gradient by a running average of its recent magnitude. *Neural networks for machine learning, Coursera lecture 6e*, 2012.

[59] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

[60] Steve Y. Oudot. *Persistence Theory: From Quiver Representations to Data Analysis*. Number 209 in Mathematical Surveys and Monographs. American Mathematical Society, 2015.

[61] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 454–463. IEEE, 2000.

[62] Shafie Gholizadeh, Armin Seyeditabari, and Wlodek Zadrozny. Topological signature of 19th century novelists: Persistent homology in text mining. *Big Data and Cognitive Computing*, 2(4):33, 2018.

[63] Anthea Monod, Sara Kališnik, Juan Ángel Patiño-Galindo, and Lorin Crawford. Tropical sufficient statistics for persistent homology. *arXiv preprint arXiv:1709.02647*, 2017.

[64] Jeremy Charlier et al. PHom-GeM. https://github.com/dagrate/phomgem, 2019.

[65] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001.

[66] Richard A Harshman. Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis. 1970.

[67] Richard A Harshman and Margaret E Lundy. Uniqueness proof for a family of models sharing features of Tucker's three-mode factor analysis and PARAFAC/CANDECOMP. *Psychometrika*, 61(1), 1996.

[68] Stephen Wright and Jorge Nocedal. Numerical optimization. *Springer Science*, 35(67-68), 1999.

[69] Alon Gonen and Shai Shalev-Shwartz. Faster SGD using sketched conditioning. *arXiv preprint arXiv:1506.02649*, 2015.

[70] Mert Pilanci and Martin J Wainwright. Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence. *SIAM Journal on Optimization*, 27(1):205–245, 2017.

[71] Yurii Nesterov et al. Gradient methods for minimizing composite objective function, 2007.

[72] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654, 2014.

[73] Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. *arXiv preprint arXiv:1802.09568*, 2018.

[74] Evrim Acar, Daniel M Dunlavy, and Tamara G Kolda. A scalable optimization approach for fitting canonical tensor decompositions. *Journal of Chemometrics*, 25(2), 2011.

[75] Richard A Harshman. Models for analysis of asymmetrical relationships among n objects or stimuli. In *First Joint Meeting of the Psychometric Society and the Society of Mathematical Psychology, Hamilton, Ontario, 1978*, 1978.

[76] Evrim Acar, Tamara G Kolda, and Daniel M Dunlavy. All-at-once optimization for coupled matrix and tensor factorizations. *arXiv preprint arXiv:1105.3422*, 2011.

[77] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755), 1999.

[78] Max Welling and Markus Weber. Positive tensor factorization. *Pattern Recognition Letters*, 22(12):1255–1261, 2001.

[79] Rasmus Bro. *Multi-way analysis in the food industry: models, algorithms, and applications*. PhD thesis, Københavns UniversitetKøbenhavns Universitet, LUKKET: 2012 Det Biovidenskabelige Fakultet for Fødevarer, Veterinærmedicin og NaturressourcerFaculty of Life Sciences, LUKKET: 2012 Institut for FødevarevidenskabDepartment of Food Science, LUKKET: 2012 Kvalitet og Teknologi-Quality & Technology, 1998.

[80] Jeremy Charlier, Radu State, and Jean Hilger. Non-Negative PARATUCK2 Tensor Decomposition Combined to LSTM Network for Smart Contracts Profiling. In *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 74–81. IEEE, 2018.

[81] Takanori Maehara, Kohei Hayashi, and Ken-ichi Kawarabayashi. Expected tensor decomposition with stochastic gradient descent. In *AAAI*, pages 1919–1925, 2016.

[82] Matthew Brand. Fast online SVD revisions for lightweight recommender systems. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 37–46. SIAM, 2003.

[83] Mustansar Ali Ghazanfar and Adam Prugel. The advantage of careful imputation sources in sparse data-environment of recommender systems: Generating improved SVD-based recommendations. *Informatica*, 37(1), 2013.

[84] Dheeraj kumar Bokde, Sheetal Girase, and Debajyoti Mukhopadhyay. Role of matrix factorization model in collaborative filtering algorithm: A survey. *CoRR, abs/1503.07475*, 2015.

[85] Ioannis T Christou, Emmanouil Amolochitis, and Zheng-Hua Tan. AMORE: design and implementation of a commercial-strength parallel hybrid movie recommendation engine. *Knowledge and Information Systems*, 47(3):671–696, 2016.

[86] Howal Sadanand, Desai Vrushali, Nerlekar Rohan, Mote Avadhut, Vanjari Rushikesh, and Rananaware Harshada. Movie recommender engine using collaborative filtering. In *Smart Computing and Informatics*, pages 599–608. Springer, 2018.

[87] Xia Ning and George Karypis. Slim: Sparse linear methods for top-n recommender systems. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 497–506. IEEE, 2011.

[88] Defu Lian, Zhenyu Zhang, Yong Ge, Fuzheng Zhang, Nicholas Jing Yuan, and Xing Xie. Regularized content-aware tensor factorization meets temporal-aware location recommendation. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 1029–1034. IEEE, 2016.

[89] Shenglin Zhao, Michael R Lyu, and Irwin King. Aggregated temporal tensor factorization model for point-of-interest recommendation. In *International Conference on Neural Information Processing*, pages 450–458. Springer, 2016.

[90] Tianhang Song, Zhaohui Peng, Senzhang Wang, Wenjing Fu, Xiaoguang Hong, and S Yu Philip. Based cross-domain recommendation through joint tensor factorization. In *International Conference on Database Systems for Advanced Applications*, pages 525–540. Springer, 2017.

[91] Zhou Zhao, Ruihua Song, Xing Xie, Xiaofei He, and Yueting Zhuang. Mobile query recommendation via tensor function learning. In *IJCAI*, volume 15, pages 4084–4090, 2015.

[92] Yang Hu, Xi Yi, and Larry S Davis. Collaborative fashion recommendation: A functional tensor factorization approach. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 129–138. ACM, 2015.

[93] Hancheng Ge, James Caverlee, and Haokai Lu. Taper: A contextual tensor-based approach for personalized expert recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 261–268. ACM, 2016.

[94] Faisal M Almutairi, Nicholas D Sidiropoulos, and George Karypis. Context-aware recommendation-based learning analytics using tensor and coupled matrix factorization. *IEEE Journal of Selected Topics in Signal Processing*, 11(5):729–741, 2017.

[95] Guoyong Cai and Weidong Gu. Heterogeneous context-aware recommendation algorithm with semi-supervised tensor factorization. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 232–241. Springer, 2017.

[96] Evangelos E Papalexakis, Christos Faloutsos, and Nicholas D Sidiropoulos. ParCube: Sparse parallelizable CANDECOMP-PARAFAC tensor decomposition. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(1):3, 2015.

[97] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 233–240. ACM, 2016.

[98] Hao Wu, Zhengxin Zhang, Kun Yue, Binbin Zhang, Jun He, and Liangchen Sun. Dual-regularized matrix factorization with deep neural networks for recommender systems. *Knowledge-Based Systems*, 2018.

[99] Xian Wu, Baoxu Shi, Yuxiao Dong, Chao Huang, and Nitesh Chawla. Neural tensor factorization. *arXiv preprint arXiv:1802.04416*, 2018.

[100] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[101] Yongxin Yang and Timothy Hospedales. Deep multi-task representation learning: A tensor factorisation approach. *arXiv preprint arXiv:1605.06391*, 2016.

[102] Yi Zuo, Jiulin Zeng, Maoguo Gong, and Licheng Jiao. Tag-aware recommender systems based on deep neural networks. *Neurocomputing*, 204:51–60, 2016.

[103] Kuo-You Peng, Sheng-Yu Fu, Yu-Ping Liu, and Wei-Chung Hsu. Adaptive runtime exploiting sparsity in tensor of deep learning neural network on heterogeneous systems. In *Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS), 2017 International Conference on*, pages 105–112. IEEE, 2017.

[104] Geoff Skinner. Cyber security for younger demographics: A graphic based authentication and authorisation framework. In *Region 10 Conference (TENCON), 2016 IEEE*. IEEE, 2016.

[105] MNMI Adeka, Kelvin OO Anoh, M Ngala, Simon Shepherd, E Ibrahim, I Elfergani, A Hussaini, J Rodriguez, and Raed A Abd-Alhameed. Africa: cyber-security and its mutual impacts with computerisation, miniaturisation and location-based authentication. 2017.

[106] Haifeng Li and Xiaowei Zhu. Face recognition technology research and implementation based on mobile phone system. In *Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016 12th International Conference on*. IEEE, 2016.

[107] Nisha Chhabra and Richa Dutta. Low quality iris detection in smart phone: A survey. 2016.

[108] Riccardo Spolaor, QianQian Li, Merylin Monaro, Mauro Conti, Luciano Gamberini, and Giuseppe Sartori. Biometric authentication methods on smartphones: A survey. *PsychNology Journal*, 14(2-3), 2016.

[109] Mary Theofanos, Simson Garfinkel, and Yee-Yin Choong. Secure and usable enterprise authentication: Lessons from the field. *IEEE Security & Privacy*, 14(5), 2016.

[110] Xavier Bultel, Jannik Dreier, Matthieu Giraud, Marie Izaute, Timothée Kheyrkhah, Pascal Lafourcade, Dounia Lakhzoum, Vincent Marlin, and Ladislav Motá. Security analysis and psychological study of authentication methods with PIN codes. In *IEEE 12th International Conference on Research Challenges in Information Science (RCIS 2018)*, 2018.

[111] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[112] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[113] Klaus Schittkowski. NLPQLP: A new Fortran implementation of a sequential quadratic programming algorithm for parallel computing. *Report, Department of Mathematics, University of Bayreuth*, 2001.

[114] Rokach Lior et al. *Data mining with decision trees: theory and applications*, volume 81. World scientific, 2014.

[115] Taehwan Kim, Yisong Yue, Sarah Taylor, and Iain Matthews. A decision tree framework for spatiotemporal sequence prediction. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 577–586. ACM, 2015.

[116] Leo Breiman. *Classification and regression trees*. Routledge, 2017.

[117] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017.

[118] David H Hubel and Torsten N Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3), 1959.

[119] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, and Gerald Penn. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012.

[120] Yi Zheng, Qi Liu, Enhong Chen, Yong Ge, and J Leon Zhao. Time series classification using multi-channels deep convolutional neural networks. In *International Conference on Web-Age Information Management*. Springer, 2014.

[121] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[122] Charles George Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, 1970.

[123] Roger Fletcher. A new approach to variable metric algorithms. *The computer journal*, 13(3):317–322, 1970.

[124] Donald Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of computation*, 24(109):23–26, 1970.

[125] David F Shanno. Conditioning of quasi-Newton methods for function minimization. *Mathematics of computation*, 24(111):647–656, 1970.

[126] Magnus Rudolph Hestenes and Eduard Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS Washington, DC, 1952.

[127] Aric Hagberg, Alex Kent, Nathan Lemons, and Joshua Neil. Credential hopping in authentication graphs. In *2014 International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*. IEEE Computer Society, Nov. 2014.

[128] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[129] Igor Halperin. QLBS: Q-learner in the Black-Scholes (-Merton) worlds. *arXiv preprint arXiv:1712.04609*, 2017.

[130] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[131] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637–654, 1973.

[132] Robert C Merton. Theory of rational option pricing. *The Bell Journal of economics and management science*, pages 141–183, 1973.

[133] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[134] Richard Stuart Sutton. Temporal credit assignment in reinforcement learning. 1984.

[135] Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards*. PhD thesis, King's College, Cambridge, 1989.

[136] R Williams. A class of gradient-estimation algorithms for reinforcement learning in neural networks. In *Proceedings of the International Conference on Neural Networks*, pages II–601, 1987.

[137] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[138] Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, and Yann LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3626–3633, 2013.

[139] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[140] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double Q-learning. In *AAAI*, volume 2, page 5. Phoenix, AZ, 2016.

[141] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[142] Paul Wilmott. *Paul Wilmott on quantitative finance*. John Wiley & Sons, 2013.

[143] John C Hull. *Options futures and other derivatives*. Pearson Education India, 2003.

[144] Oldrich Vasicek. An equilibrium characterization of the term structure. *Journal of financial economics*, 5(2):177–188, 1977.

[145] John N Tsitsiklis and Benjamin Van Roy. Analysis of temporal-diffference learning with function approximation. In *Advances in neural information processing systems*, pages 1075–1081, 1997.

[146] Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995.

[147] Sascha Lange and Martin Riedmiller. Deep auto-encoder neural networks in reinforcement learning. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2010.

[148] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[149] Hado V Hasselt. Double Q-learning. In *Advances in Neural Information Processing Systems*, pages 2613–2621, 2010.

[150] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. *arXiv preprint arXiv:1806.06923*, 2018.

[151] Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. *arXiv preprint arXiv:1707.06887*, 2017.

[152] Paweł WawrzyńSki and Ajay Kumar Tanwani. Autonomous reinforcement learning with experience replay. *Neural Networks*, 41:156–167, 2013.

[153] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *ICML*, 2014.

[154] David Balduzzi and Muhammad Ghifary. Compatible value gradients for reinforcement learning of continuous deep policies. *arXiv preprint arXiv:1509.03005*, 2015.

[155] Nicolas Heess, Gregory Wayne, David Silver, Timothy Lillicrap, Tom Erez, and Yuval Tassa. Learning continuous control policies by stochastic value gradients. In *Advances in Neural Information Processing Systems*, pages 2944–2952, 2015.

[156] Susan A Murphy. A generalization error for Q-learning. *Journal of Machine Learning Research*, 6(Jul):1073–1097, 2005.

[157] Santander. Santander product recommendation. https://www.kaggle.com/c/santander-product-recommendation/data, 2016.