ARISTOTLE UNIVERSITY OF THESSALONIKI
FACULTY OF SCIENCES
DEPARTMENT OF INFORMATICS

MASTER THESIS

PREDICTION METHODS FOR TIME-EVOLVING
DYADIC PROCESSES

MYRSINI NTEMI

SUPERVISOR:
Prof. CONSTANTINE KOTROPOULOS

Thessaloniki 2017
## Contents

1 Introduction .......................... 1

2 Time-Series with dynamic volatility 3
   2.1 Overview ................................ 3
   2.2 Netflix prize ............................ 4
   2.3 Stock market ............................. 4

3 Collaborative Filtering 6
   3.1 Overview ............................... 6
   3.2 Challenges .............................. 8
   3.3 Collaborative filtering via matrix factorization .... 9
   3.4 Problem formulation ..................... 10
   3.5 Probabilistic approach ................... 12

4 Background 14
   4.1 Gaussian distribution ................... 14
   4.2 Variational Bayesian methods ............ 20
   4.3 Brownian motion ........................ 23

5 Collaborative Kalman Filter 26
   5.1 Introduction to Kalman filter ............... 26
   5.2 Collaborative Kalman filtering ............... 28
   5.3 Variational Inference ..................... 32
      5.3.1 Variational inference and drift ........... 32
      5.3.2 Variational inference model ............... 32
      5.3.3 Variational distribution calculation ....... 33
      5.3.4 Updating the approximate distributions .... 34
   5.4 Geometric Brownian motion inference ........... 35
      5.4.1 Prediction of new data .................. 41
6 Experimental results

6.1 Movie rating prediction
   6.1.1 Data
   6.1.2 Initialization
   6.1.3 Evaluation

6.2 Stock price prediction
   6.2.1 Data
   6.2.2 Initialization
   6.2.3 Evaluation

7 Conclusion
Dedicated to my father,
Dimitris.
Abstract

User preferences and stock prices evolve through time. Capturing their changes is crucial in order to make accurate recommendations and stock market predictions. This thesis deals with dynamic statistical models applied to time-evolving dyadic processes, such as movie rating and stock price prediction. Specifically, a collaborative Kalman filter (CKF) is applied in order to model user-item interactions as latent locations that shift spatially with respect to a multidimensional Brownian motion. For this purpose, a time evolving drift parameter is learnt, which successfully captures stock price volatility over time. Variational approximation is applied to overcome intractability problems in posterior inference. For movie rating prediction, experiments have been conducted on data from Netflix prize competition. For stock price prediction AMEX, NYSE, and NASDAQ historical data from 1962 to 2017 have been used. Experimental results show that the resulting model outperforms several state-of-the-art methods in movie rating prediction. In this task, a root mean squared error of 0.7212 is achieved. In stock market, the learned Brownian motion drift parameter tracks closely stock price dynamic volatility. This fact leads to accurate stock price behaviour estimation. The average root mean squared error in stock price range of 11.840 – 155.750, is 5.500.
Μέθοδοι Πρόβλεψης για Χρονοεξελισσόμενες Δυαδικές Χρονοσειρές


Στην παρούσα διπλωματική εργασία μελετήθηκαν και αναπτύχθηκαν στατιστικά μοντέλα, τα οποία λαμβάνουν υπ’ όψιν τη δυναμική πληροφορία των δεδομένων. Τα μοντέλα εφαρμόστηκαν σε εξελισσόμενες στο χρόνο δυαδικές διεργασίες, όπως η πρόβλεψη αξιολόγησης ταινιών και η πρόβλεψη μελλοντικών τιμών των μετοχών. Συγκεκριμένα, εφαρμόζονται συνεργατικά φίλτρα, σύμφωνα με τα οποία οι χρήστες και οι ταινίες θεωρούνται σημεία σε έναν κοινό λανθάνοντα χώρο. Αυτή η διαδικασία στηρίζεται στην παραγοντοποίηση πινάκων. Η πρόβλεψη για την εκάστοτε δυάδα σε κάθε βήμα χρόνου, διεξάγεται μέσω του υπολογισμού του εσωτερικού γινομένου των διανυσμάτων των χρηστών και των ταινιών, το οποίο λαμβάνεται υπ’ όψιν τη δυναμική πληροφορία των δεδομένων. Τα μοντέλα εφαρμόστηκαν σε εξελισσόμενες στο χρόνο δυαδικές διεργασίες, όπως η πρόβλεψη αξιολόγησης ταινιών και η πρόβλεψη μελλοντικών τιμών των μετοχών. Συγκεκριμένα, εφαρμόζονται συνεργατικά φίλτρα, σύμφωνα με τα οποία οι ταινίες και οι χρήστες θεωρούνται σημεία σε έναν κοινό λανθάνοντα χώρο. Αυτή η διαδικασία στηρίζεται στην παραγοντοποίηση πινάκων. Η πρόβλεψη για την εκάστοτε δυάδα σε κάθε βήμα χρόνου, διεξάγεται μέσω του υπολογισμού του συστηματικού γινομένου των διανυσμάτων των χρηστών και των ταινιών, το οποίο αποτελεί παράμετρο μιας κατανομής. Στις κλασικές τεχνικές παραγοντοποίησης πινάκων, οι θέσεις των διανυσμάτων των χρηστών και των ταινιών στον λανθάνοντα χώρο, θεωρούνται σταθερές. Η συγκεκριμένη διαδικασία είναι στατική εφόσον δεν λαμβάνει υπ’ όψιν μια πιθανή αλλαγή στις προτιμήσεις των χρηστών στο χρόνο και γενικότερα τη δυναμική που υπάρχει στα δεδομένα. Αντίθετα στις μεθόδους που εφαρμόστηκαν οι προαναφερθείσες θέσεις αντιμετωπίζονται ως κινούμενες στο χρόνο. Συνεπώς τα συγκεκριμένα διανύσματα είναι συναρτήσεις του χρόνου. Η κίνηση των διανυσμάτων στο χώρο
γίνεται σύμφωνα με μια πολυδιάστατη κίνηση Brown. Για το λόγο αυτό το μοντέλο μαθαίνει επίσης και μια επιπλέον παράμετρο, την παράμετρο ολίσθησης (drift parameter). Η παράμετρος ολίσθησης συλλαμβάνει τη μεταβλητότητα των λανθανόντων διανυσμάτων σε μία μονάδα χρόνου. Οι δυναμικές ιδιότητες του μοντέλου είναι παρόμοιες με εκείνες του υλιτρου Kalman και για το λόγο αυτό καλείται Συνεργατικό Φίλτρο Kalman (Collaborative Kalman Filter, CKF). Σε κάθε βήμα χρόνου η τρέχουσα κατάσταση των λανθανόντων διανυσμάτων είναι ίση με την προηγούμενη τους κατάσταση με επιπρόσθετο θόρυβο. Συγκεκριμένα, τα διανύσματα κάθε κατάστασης σχετίζονται ακολουθών μια κατανομή Gauss, με μέσο διάνυσμα το λανθάνον διάνυσμα της τρέχουσας κατάστασης και πίνακα συμμεταβλητότητας ανάλογο της παραμέτρου μετατόπισης κατάστασης και της χρονικής διαρκείας μεταξύ των δύο καταστάσεων.

Για να αποκτηθεί η εκ των προτέρων κατανομή της επόμενης κατάστασης, εφαρμόζεται περιθωριοποίηση (marginalization) σε σχέση με την τρέχουσα. Μετά την εμφάνιση μιας νέας παρατήρησης, υπολογίζεται η εκ των υστέρων κατανομή. Η εκ των υστέρων κατανομή είναι μια πολυμεταβλητή κανονική κατανομή, η οποία χρησιμοποιείται στη συνέχεια για τον υπολογισμό της εκ των προτέρων κατανομής της επόμενης κατάστασης κ.ο.κ. Τα μέσα διανύσματα και οι πίνακες συμμεταβλητότητας των κατανομών των λανθανόντων διανυσμάτων είναι επίσης δυναμικά εξελισσόμενες ποσότητες οι οποίες υπολογίζονται εκ νέου σε κάθε βήμα χρόνου.

Η παράμετρος ολίσθησης μπορεί να θεωρηθεί υπερπαράμετρος των κατανομών αυτών. Είναι υπεύθυνη για το βαθμό της κίνησης των διανυσμάτων σε κάθε βήμα του χρόνου. Για παράδειγμα, εάν η παράμετρος ολίσθησης λαμβάνει μεγάλες τιμές, τα λανθάνοντα διανύσματα καλύπτουν μεγάλες αποστάσεις στο χώρο, ώστε να εφαρμόζονται κατάλληλα στο παρατηρηθέν διάνυσμα, αλλά ταυτόχρονα απορρίπτεται πληροφορία για τις προηγούμενες καταστάσεις. Αντίθετα, εάν η παράμετρο ολίσθησης λάβει μικρές τιμές, συγκρατείται η προηγούμενη πληροφορία και η εκ των υστέρων κατανομή συγκεντρώνεται στο μέσο διάνυσμα.

Για την πρόβλεψη της βαθμολογίας χρησιμοποιείται παλινδρόμηση διατεταγμένων υποδειγμάτων (ordered probit regression). Η ευθεία των πραγματικών αριθμών διαιρείται σε 5 διαδοχικά διαστήματα, τόσα όσο είναι το πλήθος των αστεριών αξιολόγησης. Λαμβάνομε την πρόβλεψη για κάθε διάστημα, υπολογίζοντας σε ποιό από τα 5 διαστήματα ανήκει. Αυτό επιτυγχάνεται μέσω της εκτίμησης μέγιστης πιθανοφάνειας.

Για την προσέγγιση των εκ των υστέρων κατανομών, εφαρμόζεται μέθοδος μεταβολών μέσου πεδίου (mean-field variational method), ώστε οι υπολογισμοί να καταστούν διαχειρίσιμοι. Σύμφωνα με την εκ νόμου διαδικασία, η εκ των υστέρων κατανομή εκάστοτε λανθάνοντος διανύσματος προσεγγίζεται από μια (προσεγγιστική) κατανομή. Η τελευταία, προέρχεται από την ελαχιστοποίηση
ση της απόστασης Kullback-Leibler μεταξύ αυτής και της πραγματικής εκ των υστέρων κατανομής σε ένα συγκεκριμένο βήμα χρόνου. Για κάθε λανθάνον διάνυσμα προσδιορίζεται η οικογένεια κατανομής, στην οποία ανήκει η προσεγγιστική κατανομή. Αυτό επιτυγχάνεται μεγιστοποιώντας την αναμενόμενη λογαριθμική από κοινού πιθανοφάνεια (log joint likelihood) και αποκλείοντας κάθε φορά την υπό εξέταση προσεγγιστική κατανομή.

Το τμήμα στο οποίο ανήκει η εκάστοτε δυάδα και κατά συνέπεια η βαθμολογία ταινιών από τον χρήστη περιγράφεται από μια λανθάνουσα μεταβλητή για την οποία επιλέγεται μια συναγωγή μεταβολών (variational inference), το μέσο διανύσματος και το πίνακα συμμεταβλητότητας των λανθανόντων διανυσμάτων των χρηστών και των ταινιών. Συγκεκριμένα περιγράφεται από μια αποκομμένη κανονική κατανομή (truncated normal distribution).

Δεδομένης της δυναμικής συμπεριφοράς του μοντέλου, οι προσεγγιστικές κατανομές των λανθανόντων διανυσμάτων, θεωρούνται ότι εξελίσσονται επί της στάσης του χρόνου. Για το λόγο αυτό, σε κάθε βήμα χρόνου πραγματοποιούνται ανανέώσεις στις παραμέτρους των κατανομών. Σε ένα τρέχον βήμα χρόνου, η μέση τιμή είναι ίση με το εσωτερικό γινόμενο των εκ των προτέρων κατανομών των λανθανόντων διανυσμάτων χρηστών και ταινιών. Στη συνέχεια ακολουθεί η ανανέωση της λανθάνουσας μεταβλητής που περιγράφει την βαθμολογία, του μέσου διανύσματος και του πίνακα συμμεταβλητότητας των εκ των υστέρων κατανομής των λανθανόντων διανυσμάτων των χρηστών και των ταινιών.

Σε κάθε βήμα χρόνου, η παράμετρος ολίσθησης της πολυδιάστατης κίνησης Brown, προσεγγίζεται μέσω της Συναγωγής Μεταβολών (Variational Inference). Θεωρούμε μοναδικές παραμέτρους μετατόπισης για κάθε λανθάνον διάνυσμα και εφαρμόζεται σημειακή εκτίμηση. Οι ανανεώσεις των παραμέτρων ολίσθησης διεξάγονται χρησιμοποιώντας μια εκτίμηση Taylor δεύτερης τάξης σε σχέση με το πιο πρόσφατα εκτιμημένο σημείο, το οποίο είναι και η τελευταία τιμή της παραμέτρου ολίσθησης που παρατηρήθηκε. Η προαναφερθέντα ανανέωση ταυτίζεται με την ιδιοανάλυση του εκ των υστέρων πίνακα συμμεταβλητότητας των λανθανόντων διανυσμάτων της προηγούμενης παρατήρησης.

Στην περίπτωση της πρόβλεψης βαθμολογίας ταινιών, τα πειράματα διεξήχθησαν στο σύνολο δεδομένων του Netflix, το οποίο απαρτίζεται από συνολικά 100,000,000 βαθμολογίες ταινιών, που καλύπτουν την περίοδο 1999-2006. Επικρατεί το σύστημα βαθμολογίας 5 αστέρων, με βαθμολογίες 1 μέχρι 5. Το σύνολο περιεχεί πληροφορία για 17,770 ταινίες και 480,000 χρήστες. Τα δεδομένα υποβλήθηκαν σε προεπεξεργασία. Η τελική χρονοσειρά απαρτίζεται από τετράδες ταινία-χρήστη-βαθμολογία-ημερομηνία, διατεταγμένες χρονολογικά. Ακολουθούν αρχικοποίηση των δεδομένων και η εφαρμογή της μεθόδου. Η αξιολόγηση των αποτελεσμάτων έγινε μέσω του υπολογισμού της ρίζας του μέσου τετραγωνικού σφάλματος (root-mean-square error, RMSE). Τα αποτελέσματα συγκρίθηκαν με καθιερωμένες μεθόδους, όπως η Επίκαιρη Μπαεζυανή Μέθοδος Μεταβολών - Αναμενόμενη Τιμή - Μεγιστοποίηση (Online Variational
Bayes Expectation Maximization), πιθανοτική παραγοντοποίηση πινάκων, πιθανοτική παραγοντοποίηση πινάκων κατά Bayes. Η μέθοδος του συνεργατικού φίλτρου Kalman απέφερε τα βέλτιστα αποτελέσματα με RMSE = 0.7212 (α-στέρια). Η δυνατότητα του μοντέλου να συλλαμβάνει τη δυναμική πληροφορία των προτιμήσεων των χρηστών οδήγησε σε εμφανώς καλύτερα αποτελέσματα εν συγκρισει με τα υπόλοιπα στατικά μοντέλα.

Chapter 1

Introduction

People through decades have relied on their peers’ recommendations and perspective to support decisions and discover new material. Research has led to advanced computer-based systems providing the opportunity to expand the set of people from whom users can obtain recommendations. At the same time, in dyadic processes [1], the input to the prediction system is a dyad, such as the pair user-movie. The objective is to anticipate an estimation of an observed value related to the dyad. Each dyad is characterized only by a unique identifier. Once there is not any further information available about dyads, latent features have to be learned. Collaborative filtering is a special case of dyadic prediction, where the objective is to predict the ratings that are related to pairs. Collaborative filtering is referred to the process of filtering information by utilizing the assessments of other individuals. It is based on the idea that individuals who concurred in their assessment of specific things in the past are probably going to concur again in future assessments. For example, user rating systems, such that of Netflix, Amazon, and YouTube utilize their users’ preferences in movies, products, or various videos, respectively. These preferences could be either ratings in five star rating system, likes and dislikes, etc. Another application is the stock price prediction, based on collaborative filtering technique, in which historical data used in order to make future predictions.

Recommendation systems based on collaborative filtering confront a few challenges: The first challenge is the huge volume of given data (large scale, big data) . The second one is that the matrix which includes the ratings, may be sparse. Usually a small portion of movies in the dataset are rated by each user. These challenges are been treated satisfactorily by matrix factorization (MF). Some of the most successful approximations of collaborative filtering models are based on matrix factorization techniques. In these methods, users and objects are mapped into the same latent space, such that their interac-
tions are represented as inner products in this joint latent factor space [2]. In these approaches, the locations of users and items in their common latent space are considered to be fixed in time. Thus, the aforementioned procedure is a stationary process and does not take into account a possible change in users’ preferences through time and dynamic information generally and this is a crucial parameter in order to produce a consistent and more efficient model.

In this thesis, we study, apply, and develop statistical models that take into consideration dynamic information. In these dynamic collaborative techniques, the locations of users and items, evolve temporally. Specifically, they drift along time according to a multidimensional Brownian motion [3]. For this reason, the model also learns a drift parameter. In stock price prediction, due to the nature of the problem, the Brownian motion drift parameter is not static, but a dynamic one, in order to capture the price volatility alternation. For this purpose, the model has to learn a dynamic probability distribution on latent user and object locations. To maintain tractability, a variational inference approximation of posterior calculations is applied. Since the model appears to have similar properties with those of the Kalman filters [4], the method is referred to as *Collaborative Kalman Filter* [5]. In the following chapters, there is a detailed mathematical presentation of all methods studied and applied, which constitute the resulting model.
Chapter 2

Time-Series with dynamic volatility

2.1 Overview

Time series is referred to as a process comprising random variables, collected sequentially and ordered by time. Time series analysis consists of two main subcategories: univariate and multivariate time-series analysis. A univariate time-series is a time series consisting of measurements of a single variable through time. When the observations occur at equal time intervals, the sequence is a discrete-time time series. In contrast, continuous-time time series is a stochastic process where measurements are not sampled in discrete points in time, but instead time is considered to be continuous. On the other hand, multivariate time-series analysis consists of measurements applied to groups of variables, in order to examine their interactions through time. The general models of these two categories of time-series analysis are:

- **Univariate.** Let a univariate time-series be $Y_1, \ldots, Y_T$. In this case, $T$ measurements on a single variable have been made at $T$ discrete time points.

\[
Y_t = g(t) + \phi_t, \ t = 1, \ldots, T,
\]

where $g(t)$ denotes the signal and it is a deterministic function of time. The stochastic sequence $\phi_t$, is usually referred to as noise process.

- **Multivariate.** Assume the time series consists of $n$ variables, $\{y_{1t}\}, \ldots, \{y_{nt}\}$. A multivariate time series is the $(n \times 1)$ vector time series $\{Y_t\}$, where the $i^{th}$ row of $\{Y_t\}$ is $\{y_{it}\}_{t=1}^T$. That is, for any time $t$, $Y_t = (y_{1t}, \ldots, y_{nt})'$.
Volatility, refers to the variability observed in a component of a time series over time. Volatility, in financial economics, usually illustrates the conditional standard deviation of components that follow a random Wiener process in a diffusion model in continuous-time domain [6]. Volatility is not constant, but time-varying.

2.2 Netflix prize

Netflix, is a company which provides an on-line subscription service, which enables individuals to rent movies, by paying a fixed month to month charge. The users are also able to create ordered lists of films and series they wish to watch in future. In 2006, Netflix announced a competition which became widely known as the Netflix prize competition. It provided a dataset which contained approximately 100,000,000 ratings, of roughly 480,000 users and 18,000 different movie titles. The collected data obtained from a long time period, i.e., 1998-2005. The ratings lay within the range [1, 5] stars. The identity of users was omitted. The challenge was the development of a system more accurate than their existing one, which was called Cinematch.

An accurate rating prediction system was very crucial to be developed, because if subscribers failed to find movies that interested and engaged them, they tended to abandon the service.

Successful approximations encounter on the dynamic aspect of prediction. That is, observed data are used to infer about unobserved data, with respect to model’s variability over time. Dynamic effects turn out to have much importance than static ones. This arises from the fact that each user and movie potentially goes through a distinct series of changes in their characteristics: a user’s taste and behaviour may evolve over time. In this more realistic setting than static ones, users and movies are organized into matrices and their (user-item) interactions are modeled as functions of time. Users for example, they can be modeled as vectors moving in state space according to a particular volatility measure (i.e. Brownian motion). The user-item interactions, include dynamic information. In Chapter 3, there is a further analysis on such data structures governed by temporal dynamics.

2.3 Stock market

Stock market is one of the most essential fields of market economy. Investors are able to buy or sell financial assets or shares of ownership in corporations, enabling the financial capital to be accessible. Through this procedure,
investors are capable of possibly gaining money by profiting from companies’ future prosperity. On the other hand, profit increment can be achieved by buying shares and right before their underestimation, vending them. The return of their initial investment and much more a profit achievement, is not always possible. This happens, because of stock price constant fluctuation through time. The underlying risk lies in a possible fall of stock price below its initial value at which it was bought. Consequently, its critical for the appropriate decision to be taken. A model which is able to make accurate predictions on the next price of a stock, based on historical stock prices, may reinforce each investor’s decision and lead to risk minimization and profit increment.

Due to the nature of such data, historical stock prices can be represented as a time series. Return volatility is, of course, crucial to financial economics. A characteristic feature of economic and financial time series is volatility clustering into periods of high and of low volatility occur in the data. Typically, the changes between these periods do not exhibit any systematic patterns and seem best modeled as occurring randomly [7].

A baseline method which is an efficient and widely applied one to time series analysis, is the Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) Process [8]. Through this method, volatility estimation is applied and thus, stock price prediction is achieved. Time-varying volatility is assumed to govern the stock prices and not a constant one. A GARCH model objective is to minimize prediction errors in a current time step, by utilizing information from previous time steps, i.e., return residuals. Since then, there have been many attempts to capture stock prices’ volatility through a wide range of methods.
Chapter 3

Collaborative Filtering

3.1 Overview

The term Collaborative Filtering (CF) was first used by the developers of an email filtering system called “Tapestry” [9], who believed that information filtering can be more effective when humans are involved in the filtering process. This system accumulated the contents of messages for future use. Metadata related to authors and readers were stored as well. Another provided feature to clients, was that of storing comments about messages. Queries consisting of text, annotations and semantic information, could be applied by clients.

After Trapestry emergence, researchers around the world recognized the potential of this field. Maltz and Ehrlich [10] developed an active collaborative filtering recommender system in which individuals, who find documents that are interested in, send “pointers” about the particular ones, to their partners. Pointers are annotations that include a link to the “pointed” document and relevant information about its content. This procedure provides a prior knowledge about the document. Thus, recipients are able to decide if the particular document is close to their interests without reading it.

In automated collaborative filtering systems, historical user preferences are utilized in order to relate each user to those with corresponding interests. Early automated collaborative filtering systems included GroupLens, a system developed to help people to find articles they would like within a huge stream of available articles [11], Ringo, which makes personalized recommendations for music albums and artists [12], and Bellcore’s Video Recommender in movie domain [13]. Early collaborative filtering algorithms for recommendation systems utilized the association inferences, which exhibited very high time complexity and very poor scalability. Recent methods instead, like probabilistic matrix factorization [14] and singular value decomposition
techniques (SVD) [15], use matrix operations in order to make predictions.

Some recent state-of-the-art methods apply deep learning techniques and parallel computing, in order to deal with the basic problems of collaborative filtering. Manipulation of the enormous amount of data, is provided by a method [16] which implements user-based algorithms within a MapReduce framework, on Hadoop\(^1\). This method provides a solution to the problems related to scalability of collaborative filtering. Another well performing method, is the Neural Collaborative Filtering (NCF) [17]. Its concept is similar to matrix factorization approaches, but instead of utilizing inner products of users and items latent vectors, a neural architecture is used. A function describing user-item interactions is learned by a multi-layer perceptron. Evidences imply that the more the layers, the better recommendation. In [18], a Keyword-Aware Service Recommendation (KASR) method is proposed, in order to offer efficient personalized recommendations. This is a user-based collaborative filtering approach, which utilizes keywords as indicators of users' taste. It was implemented on Hadoop. A clustering based collaborative filtering method is proposed [19] in the e-commerce area. This approach is based on a Latent Class Regression model (LCRM), which takes into consideration textual information from reviews and ratings on the products. Clusters of reviewers are constructed, according to their preferences and \(k\)-nearest neighbours is applied in order to achieve efficient recommendation of products. Another method utilizing online reviews is proposed in [20], which actually combines user preferences and opinions as well. The difference between each user opinion and each user rating is being analysed in order to construct a model providing efficient recommendations. In [21], a collaborative filtering approach is suggested, which applies real-time web mining (i.e., Twitter). Products and users are related to specific textual information existing in their reviews. Textual information is also used in Collaborative Topic Regression (CTR) model [22] along with other users similar tastes, in order to recommend scientific articles. This method arise from the combination of probabilistic topic modeling and latent factor based collaborative filtering. Review texts are also utilized in Deep Cooperative Neural Networks (DeepCoNN) [23]. This system is constructed by two parallel neural networks. The first is responsible for learning users conduct from the reviews they write. The second one is responsible for learning item attributes from reviews written for them. These two layers collaborate and result in the modeling of user-item interactions.

As mentioned previously, in order to establish recommendations, CF systems need to relate the fundamental entities: items and users. To facilitate

\(^1\)Hadoop is a distributed cloud computing platform.
such a comparison, two fundamental approaches exist: the neighborhood approach and the latent factor approach. The first one, focuses on item-item or user-user interactions. An item-item approach models the preference of a user to an item by examining items similar the particular one, which this user has rated. Latent factor models, such as matrix factorization, contain an alternative approach. They transform both items and users to the same latent factor space. The latent space tries to explain ratings by characterizing both products and users with factors automatically inferred from user feedback. Another usual more specific categorization of CF techniques is as follows [24]:

- **user-based** approaches. The goal is to find the nearest neighbours of each individual according to their preferences. Then, the user rating prediction on a single movie, is based on the closeness of nearest neighbours with respect to their ratings.

- **model-based** approaches. This category, consist mainly of approaches based on clustering. Users with similar tastes are aggregated into groups. The users preferences within a cluster are utilized to predict another’s preference on a new item.

- **item-based** approaches, that associate to each item its set of nearest neighbours, and then predict a user rating on an item employing the ratings of the nearest neighbours of the item considered.

### 3.2 Challenges

Through the last years the available information dramatically increases and the e-commerce users have raised challenges for existing recommender systems, which are more difficult to be addressed. Some important challenges are [25]:

- **Scalability.** The main challenge is to achieve scalability of the recommender system in order to match and accommodate any number of users and content volume available. To deal with the increasing number of users as well as items, computations in several recommender systems increase as well. However, the computations may produce inaccurate results and get computationally expensive [26]. In addition, a quick response is demanded in a plethora of frameworks. Recommendations should be provided to all users, regardless of their interactions and previously given ratings. For this purpose, the collaborative filtering method should be highly scalable. Due to the vast amount of
users, these procedures are highly computationally expensive. However, widely known web companies, i.e., Twitter, use clusters of computers in order to distribute the computations needed to produce recommendations [27].

- **Accuracy.** Recommender systems that are able to interact with users and learn their preferences are essential in order to provide accurate recommendations. This fact creates serious concerns over the user interface and the interaction model. The precondition for accurate recommendations to be created can be provided only when accurate data is available. Accordingly, collecting reliable data without the noise in user preference data sets becomes a priority. At the same time, privacy should not be violated by recommender systems, but this also depends on internet cache [28].

- **The Cold-start Problem.** This problem arises along with the introduction of new users, new items, or new systems, when recommendation is not possible as very little information is available about the user or no rating is available for the product [29]. This is difficult to tackle, since it is not possible to find similar users or to build a content-based profile without previous preferences of a user. As such, research in this area has primarily focused on effectively selecting items to be rated by a user so as recommendation performance rapidly increases with the least user feedback. In this setting, classical techniques from active learning [30] can be leveraged to address item selection.

- **Data sparsity.** Sparsity of data is one of the most crucial factors frequently encountered by most of the recommender systems. Data is collected when users sign up for the first time on the website or through a survey. Nevertheless, sparsity of data occurs when most often the users don’t rate most of the items and the ratings of the items remain sparse [31]. Many researchers attempt to overcome this challenge.

### 3.3 Collaborative filtering via matrix factorization

Matrix factorization (MF) approaches, in their majority, are based on the concept of mapping users and items into a latent feature space. The observed ratings are approximated via the dot products of the latent feature vectors of users and items. Recently, these models have gained popularity and researchers’ interest. Thanks to their attractive accuracy, scalability
and the ability to handle high levels of sparsity in data. A common example of methods in information retrieval is SVD, because it is capable of identifying latent semantic factors [32]. Nevertheless, due to the large number of missing values, incomplete matrices arise and SVD-based models face difficulties. The strict choice of few known entries should be avoided, because it is possible to be led to overfitting. This problem can be handled via methods relied on imputation, which fills in missing ratings and makes the rating matrix dense. However, these methods can become computationally expensive as they significantly increase the amount of input data and perform badly in terms of accuracy, due to new data created. Other well known MF-based collaborative filtering based are Non-Negative Matrix Factorization [33], Bayesian Probabilistic Matrix Factorization [34], and Mixed Membership Matrix Factorization [35].

3.4 Problem formulation

Assume a set of users $M$ and a set of items $N$, which are organized into matrices $\mathbf{U} \in \mathbb{R}^{K \times M}$ and $\mathbf{W} \in \mathbb{R}^{K \times N}$, respectively. Each column of matrix $\mathbf{U}$ ($\mathbf{u}_i$) represents user location in a latent space. The columns of $\mathbf{W}$ ($\mathbf{w}_j$) represent item (e.g., movie) location in the same latent space. An incomplete matrix $\mathbf{Z} \in \mathbb{R}^{M \times N}$ contains all the ratings that users have assigned to items. Then, the latent features should be discovered. Specifically, our task is to estimate two matrices, $\mathbf{U}$ for users and $\mathbf{W}$ for items, which are modeled to have much lower rank, such that the ratings are predicted through the relationship:

$$\mathbf{Z} \approx \mathbf{U}^T \mathbf{W}. \quad (3.1)$$

In MF approaches, $K$ is responsible for the balance between capturing user preferences and overfitting [36]. Thus, it is crucial to define its value properly.

The interaction between user $i$ and item $j$ is represented by the dot product between user and item vectors, $z_{ij} = \mathbf{u}_i^T \mathbf{w}_j$. The minimization of the regularized squared error over the set of observed ratings, is a popular approach to estimate unobserved data in matrix $\mathbf{Z}$ [2]:

$$\mathcal{F} = \min_{\mathbf{U}, \mathbf{W}} \sum_{i,j \in \mathbf{Z}} (z_{ij} - \mathbf{u}_i^T \mathbf{w}_j)^2 + \lambda_1 \|\mathbf{U}\|_F^2 + \lambda_2 \|\mathbf{W}\|_F^2 \quad (3.2)$$

where $\lambda_1$, $\lambda_2$ are regularization parameters for user and item vectors and $\|\cdot\|_F$ denote the Frobenius norm of a matrix, respectively. A model like this, is closely related to SVD. The learning procedure is based on the exploitation of last observed ratings. In order to predict unknown ratings, generalization
on previously observed ones should be done. To avoid overfitting in the observed data, the last two regularization terms are introduced in (3.2), whose magnitudes should strictly be controlled. The regularization terms \( \lambda_1 \) and \( \lambda_2 \) are usually determined by cross-validation techniques. There are two basic approaches to minimizing Equation (3.2), \textit{stochastic gradient descent} and \textit{alternating least squares}.

- **Stochastic gradient descent.** For every given training example, \( z_{ij} \) is predicted and the resulting error is calculated as

  \[
  e_{ij} = z_{ij} - u^T_i w_j. 
  \]  

  (3.3)

  Subsequently, the parameters are updated as [37]:

  \[
  \frac{\partial F}{\partial u_i} = 2e_{ij} \frac{\partial e_{ij}}{\partial u_i} + 2\lambda_1 u_i = 2e_{ij}(-w_j) + 2\lambda_1 u_i \Rightarrow u_i \leftarrow u_i - \gamma \frac{\partial F}{\partial u_i} = u_i + \gamma (e_{ij}w_j - \lambda_1 u_i) 
  \]  

  (3.4)

  \[
  \frac{\partial F}{\partial w_j} = 2e_{ij} \frac{\partial e_{ij}}{\partial w_j} + 2\lambda_2 w_j = 2e_{ij}(-u_i) + 2\lambda_2 w_j \Rightarrow w_j \leftarrow w_j - \gamma' \frac{\partial F}{\partial w_j} = w_j + \gamma' (e_{ij}u_i - \lambda_2 w_j). 
  \]  

  (3.5)

  The meta-parameters \( \gamma \) and \( \gamma' \) represent the step sizes. They are determined by cross-validation as well.

- **Alternating least squares.** \( u_i \) and \( w_j \) are unknown up to this point and this results in the fact that Equation (3.2) is not convex. In order to solve this problem, it has to become to a quadratic form. This can be done by keeping the set of vectors \( U \) fixed and solving for each item \( w_j \). Say that \( F \) in (3.2) is rewritten as:

  \[
  F = ||Z - U^T W||_F^2 + \lambda_1 ||W||_F^2 
  \]

  \[
  = tr[(Z - U^T W)(Z^T - W^T U)] + \lambda_2 tr(WW^T) 
  \]  

  (3.6)

  If we set the partial derivative with respect to \( W \) equal to zero, we obtain

  \[
  \frac{\partial F}{\partial W} = \frac{\partial}{\partial W} tr[(Z - U^T W)(Z^T - W^T U)] + \lambda_2 tr(WW^T) = 0 \iff 
  \frac{\partial}{\partial W} tr[ZZ^T - U^T WZ^T - ZW^T U + U^T WW^T U + \lambda_2 tr(WW^T) = 0
  \]  

  (3.7)
If we use the property of trace, which says that the trace of a matrix and the trace of its transpose are equal, and the cyclic property, we obtain

\[-U^T Z^T - U^T Z^T + 2WU^T + 2\lambda_2 W = 0 \iff -2U^TZ^T + (UU^T + \lambda_2 I)2W = 0\]

(3.8)

Then if we solve with respect to \( W \), we obtain

\[W = (UU^T + \lambda_2 I)^{-1}U^TZ^T\]

(3.9)

Reversely, if item vectors \( W \) are kept fixed it can be solved for each user \( u_i \), always in a least-squares manner,

\[\frac{\partial F}{\partial U} = \frac{\partial}{\partial U} \text{tr}[(Z - U^TW)(Z^T - W^TU)] + \lambda_1 \text{tr}(UU^T) = 0 \iff \frac{\partial}{\partial U} \text{tr}[ZZ^T - U^TWW^T - ZW^TU + U^TWW^TU] + \lambda_1 \text{tr}(UU^T) = 0 \iff -WZ^T - WZ^T + 2WW^TU + 2\lambda_1 U = 0\]

(3.10)

If we solve with respect to \( U \), we obtain

\[(WW^T + \lambda_1 I)U = WZ^T \iff U = (WW^T + \lambda_1 I)^{-1}WZ^T\]

(3.11)

### 3.5 Probabilistic approach

According to this approach, instead of approximating the entries in matrix \( Z \) as indicated in Section 3.4, the goal is to estimate their distribution. Again, the result depends on the dot product of user and item matrices, \( U \) and \( W \)

\[z_{ij} \sim p(u_i^T w_j).\]

(3.12)

where \( p() \) denotes a probability density function. Matrices \( U \) and \( W \) have to be learned and then the prediction for the unobserved entry \( i, j \) in matrix \( Z \), \( z_{ij} \), is accomplished by utilizing the distribution function \( p(u_i^T w_j) \).

For example, in the case of normal distribution, \( z_{ij} \sim \mathcal{N}(u_i^T w_j, \sigma^2) \), with mean \( u_i^T w_j \) and variance \( \sigma^2 \), we have defined a family of distributions. The goal then is to use observed data to make a statement about the parameters governing our model. This can be done by statistical inference [38]. We
can approximate matrices $U$ and $W$ through maximum likelihood inference. This method obtains values of model parameters that define a distribution that is most likely to have resulted in the observed data. If $A_{ui}$ is the set containing all the items user $i$ has rated, the update of $u_i$ is obtained as follows [5]. Starting from the pdf of the observations,

$$p(z_{ij}|u_i, w_j, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(z_{ij} - u_i^T w_j)^2\right\}.$$  (3.13)

We calculate the log likelihood function

$$-\log L = -\log \prod_{j \in A_{ui}} p(z_{ij}|u_i, w_j, \sigma) =$$

$$- \sum_{j \in A_{ui}} \log\left(\frac{1}{\sigma \sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(z_{ij} - u_i^T w_j)^2\right\}\right) =$$

$$- \sum_{j \in A_{ui}} \left\{-\log \sigma - \frac{1}{2} \log(2\pi) - \frac{1}{2\sigma^2}(z_{ij} - u_i^T w_j)^2\right\} =$$

$$|A_{ui}|(\log \sigma + \frac{1}{2} \log(2\pi)) + \frac{1}{2\sigma^2} \sum_{j \in A_{ui}} (z_{ij} - u_i^T w_j)^2.$$  (3.14)

To minimize the negative log likelihood, we equate its gradient with respect to $u_i$ with 0 and we arrive at:

$$\frac{\partial L}{\partial u_i} = \frac{1}{2\sigma^2} \sum_{j \in A_{ui}} 2(z_{ij} - u_i^T w_j) \frac{\partial L}{\partial u_i} (z_{ij} - u_i^T w_j) = 0$$

$$\sum_{j \in A_{ui}} (z_{ij} - u_i^T w_j)(-w_j) = 0 \iff$$

$$\sum_{j \in A_{ui}} z_{ij} w_j = \sum_{j \in A_{ui}} w_j w_j^T u_i \Rightarrow$$

$$u_i = (\sum_{j \in A_{ui}} w_j w_j^T)^{-1}(\sum_{j \in A_{ui}} z_{ij} w_j).$$  (3.15)

This update utilizes all the objects $w_j$ rated by user $u_i$. The same procedure is applied to update each item $w_j$ using all users rated that particular item. The fundamental problem in these models is that they do not evolve through time and they are unable to capture temporal information.

A solution to this problem is provided by Brownian motion [3], which enables the system to model dynamic information about users $u_i$ and items $w_j$. In the following chapters, an extensive analysis on this topic will be made.
Chapter 4

Background

4.1 Gaussian distribution

The normal distribution which is also referred to as the Gaussian distribution, is the most important category of probability distributions for continuous variables. For a single variable $x$, univariate Gaussian distribution is defined as:

$$
N(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp \left\{-\frac{1}{2\sigma^2}(x - \mu)^2 \right\} \tag{4.1}
$$

where $\mu$ denotes the mean, while $\sigma^2$ denotes the variance. The extension of univariate Gaussian distribution, when instead of examining a variable, a vector $x \in \mathbb{R}^D$ is briefly reviewed next [39]. This probability distribution is also called multivariate Gaussian distribution,

$$
N(x|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp \left\{-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu) \right\} \tag{4.2}
$$

where $\mu$ is the mean vector, and $\Sigma$ is a $D \times D$ covariance matrix. The expression

$$
\Delta^2 = (x - \mu)^T \Sigma^{-1}(x - \mu) \tag{4.3}
$$

is also called squared Mahalanobis distance and defines ellipsoidal equidistance contours. The axes defining the ellipsoidal contours, correspond to the eigenvectors of the covariance matrix.

$$
\Sigma u_i = \lambda_i u_i \tag{4.4}
$$

where $u_i$ denotes the $i^{th}$ eigenvector corresponding to the $i^{th}$ eigenvalue, $\lambda_i$, $i = 1, \ldots, D$. The eigenvectors define an orthonormal basis,

$$
u_i^T u_j = \delta_{ij} \tag{4.5}\]
where $\delta_{ij}$ denotes a Kronecker delta. Let us express the covariance matrix $\Sigma$ as follows,

$$\Sigma = \sum_{i=1}^{D} \lambda_i u_i u_i^T \iff (4.6)$$

$$\Sigma^{-1} = \sum_{i=1}^{D} \frac{1}{\lambda_i} u_i u_i^T. \quad (4.7)$$

Then, we can obtain $\Delta$

$$\Delta = \sum_{i=1}^{D} \frac{y_i^2}{\lambda_i} \quad (4.8)$$

where $y_i = u_i^T(x - \mu)$, transformed variables into a new coordinate system. The vector $y = (y_1, \ldots, y_D)^T$ is defined as

$$y = U(x - \mu) \quad (4.9)$$

where $U$ is a matrix having as rows the eigenvectors $u_i^T$. Matrix $U$ satisfies $UU^T = I$. Thus, in this new coordinate system, $y_i$ follows the Gaussian distribution. The Jacobian matrix $J$ can be utilized in order to derive the probability density function when we move from $x$ to $y$ coordinate system. It can be shown that the $ij$-th element of $J$ is

$$J_{ij} = \frac{\partial x_i}{\partial y_j} = U_{ji}. \quad (4.10)$$

That is, $J = U^T$. Due to the orthonormality property and using $|U^T| = |U|$, we obtain,

$$|J|^2 = |U^T|^2 = |U^T||U| = |U^T U| = |I| = 1 \Rightarrow |J| = 1. \quad (4.11)$$

The determinant of the covariance matrix is given by

$$|\Sigma|^{1/2} = \prod_{j=1}^{D} \lambda_j^{1/2}. \quad (4.12)$$

Hereafter, the Gaussian distribution becomes

$$p(y) = p(x)|J| = \prod_{j=1}^{D} \frac{1}{(2\pi \lambda_j)^{1/2}} \exp \left\{- \frac{y_j^2}{2\lambda_j}\right\} \quad (4.13)$$

where $D$ denotes the number independent univariate Gaussian distributions. Therefore, coordinate transformations, i.e., rotations and shifts, are achieved.
via the eigenvectors. In the new coordinate system, the distribution integral calculation can be

\[ \int_{-\infty}^{\infty} p(y) dy = \prod_{j=1}^{D} \int_{-\infty}^{\infty} \frac{1}{(2\pi \lambda_j)^{1/2}} \exp \left\{ -\frac{y_j^2}{2\lambda_j} \right\} dy_j = 1. \]  

(4.14)

**Conditional Gaussian distributions.** One set conditioned on another set is said to be Gaussian, if these sets are jointly normally distributed. Let \( \mathbf{x} \in \mathbb{R}^D \), have a Gaussian distribution \( \mathcal{N}(\mathbf{x} | \mathbf{\mu}, \Sigma) \). This vector can be partitioned into two subvectors \( \mathbf{x}_a \) and \( \mathbf{x}_b \),

\[ \mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix}. \]  

(4.15)

The mean vector \( \mathbf{\mu} \) and the covariance matrix \( \Sigma \) are also divided into two subsets,

\[ \mathbf{\mu} = \begin{pmatrix} \mathbf{\mu}_a \\ \mathbf{\mu}_b \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}. \]  

(4.16)

The resulting precision matrix \( \Lambda \), which is equal to \( \Sigma^{-1} \), is given by:

\[ \Lambda = \begin{pmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{pmatrix}. \]  

(4.17)

Subsequently, the problem of calculating \( p(\mathbf{x}_a | \mathbf{x}_b) \) is addressed

\[ p(\mathbf{x}_a | \mathbf{x}_b) = \frac{p(\mathbf{x}_a, \mathbf{x}_b)}{p(\mathbf{x}_b)}. \]  

(4.18)

The application of the partitions of vector \( \mathbf{x} \), mean vector \( \mathbf{\mu} \), covariance matrix \( \Sigma \), and precision matrix \( \Lambda \) leads to the following expression:

\[ -\frac{1}{2}(\mathbf{x} - \mathbf{\mu})^T \Sigma^{-1}(\mathbf{x} - \mathbf{\mu}) = \\
-\frac{1}{2}(\mathbf{x}_a - \mathbf{\mu}_a)^T \Lambda_{aa} (\mathbf{x}_a - \mathbf{\mu}_a) + \frac{1}{2}(\mathbf{x}_b - \mathbf{\mu}_b)^T \Lambda_{bb} (\mathbf{x}_b - \mathbf{\mu}_b) - \frac{1}{2}(\mathbf{x}_a - \mathbf{\mu}_a)^T \Lambda_{ab} (\mathbf{x}_b - \mathbf{\mu}_b) - \frac{1}{2}(\mathbf{x}_b - \mathbf{\mu}_b)^T \Lambda_{ba} (\mathbf{x}_a - \mathbf{\mu}_a). \]  

(4.19)

Since (4.19) is a function of \( \mathbf{x}_a \) and \( \mathbf{x}_b \), the first priority in order to derive \( p(\mathbf{x}_a | \mathbf{x}_b) \), is to determine the mean vector \( \mathbf{\mu}_{a|b} \) and covariance matrix \( \Sigma_{a|b} \). This can be accomplished without integration.
If we set the terms that are independent of \( x \) to be constant, the exponent term of a normal distribution, in general, becomes,

\[
-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu) = -\frac{1}{2}x^T \Sigma^{-1}x + x^T \Sigma^{-1}\mu + C
\]  

(4.20)

where \( C \) is the constant term, mentioned before. This simplification method is also known as completing the square, which is applied to the quadratic forms in order to manipulate them.

\[
-\frac{1}{2}x_a^T \Lambda_{aa} x_a 
\]

In (4.19), the second-order term w.r.t. \( x_a \) can be exploited in order to find the precision matrix, i.e., linear terms \( x_a^T \{\Lambda_{aa} \mu_a - \Lambda_{ab} (x_b - \mu_b)\} \) with respect to \( x_a \). Thus, we obtain covariance, precision matrix, and the mean vector respectively.

\[
\mu_{a|b} = \Sigma_{a|b} \{\Lambda_{aa} \mu_a - \Lambda_{ab} (x_b - \mu_b)\} = \mu_a - \Lambda_{aa}^{-1} \Lambda_{ab} (x_b - \mu_b). 
\]  

(4.22)

By exploiting the following partitioning of a matrix, we can obtain the following identity for the partitioning of its inverse:

\[
\begin{pmatrix}
A & B \\
C & D
\end{pmatrix}^{-1} =
\begin{pmatrix}
M & -MBD^{-1} \\
-D^{-1}CM & D^{-1} + D^{-1}CBM^{-1}
\end{pmatrix}
\]

(4.23)

where \( M = (A - BD^{-1}C)^{-1} \) is known as the Schur complement of the matrix in the left-hand side of (4.23) w.r.t. \( D \). Using the definition of the precision matrix, (4.21) and (4.23), we arrive at the following closed-form expressions for the definition of the conditional mean vector and conditional covariance matrix of \( p(x_a|x_b) \):

\[
\mu_{a|b} = \mu_a + \Sigma_{ab} \Sigma_{bb}^{-1} (x_b - \mu_b)
\]

\[
\Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab} \Sigma_{bb}^{-1} \Sigma_{ba}.
\]  

(4.24)

Because \( \Sigma_{ba} = \Sigma_{ab}^T \), the above expression becomes:

\[
\Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab} \Sigma_{bb}^{-1} \Sigma_{ab}^T.
\]  

(4.25)

**Marginal Gaussian distributions.** It is well known that:

\[
p(x_a) = \int p(x_a, x_b) dx_b.
\]  

(4.26)
In order to evaluate \( p(x_a) \), covariance matrix and mean vector have to be derived again, in a similar manner as for the conditional Gaussian distributions we locate the terms involving \( x_b \) and complete the square:

\[
-\frac{1}{2}x^T_b \Lambda_{bb} x_b + x^T_b \{ \Lambda_{bb} \mu - \Lambda^T_{ab}(x_a - \mu_a) \} = \sum_a \mu_a = \sum_a \left( \Lambda_{aa} - \Lambda_{ab} \Lambda^{-1}_{bb} \Lambda_{ba} \right) \mu_a + C \Rightarrow \Sigma_a = \left( \Lambda_{aa} - \Lambda_{ab} \Lambda^{-1}_{bb} \Lambda_{ba} \right)^{-1} \}
\]

The integral of the first term is recognized to be the reciprocal of the normalization constant \( K \), which guarantees that the integral below a pdf equals 1. This constant depends on the determinant of \( \Lambda^{-1}_{bb} \).

\[
\int \exp\left\{ -\frac{1}{2}(x_b - \Lambda^{-1}_{bb} m)^T \Lambda_{bb} (x_b - \Lambda^{-1}_{bb} m) \right\} dx_b. \quad (4.28)
\]

If we complete the square for the last term in (4.27) inserting the definition of \( m \), we obtain an expression, which depends on \( x_a \),

\[
-\frac{1}{2}x^T_a (\Lambda_{aa} - \Lambda_{ab} \Lambda^{-1}_{bb} \Lambda_{ba}) x_a + x^T_a (\Lambda_{aa} - \Lambda_{ab} \Lambda^{-1}_{bb} \Lambda_{ba})^{-1} \mu_a + C \Rightarrow \\
\Sigma_a = \left( \Lambda_{aa} - \Lambda_{ab} \Lambda^{-1}_{bb} \Lambda_{ba} \right)^{-1} \quad (4.29)
\]

Therefore, the marginal pdf is expressed in terms of \( \mu_a \) and \( \Sigma_{aa} \).

**Gaussian variables and Bayes theorem.** Suppose that we are given a Gaussian marginal distribution \( p(x) \) and a Gaussian conditional distribution \( p(y|x) \). The mean vector of the conditional distribution depends on \( x \), but the covariance matrix does not. Firstly, the aforementioned distributions are:

\[
\begin{align*}
p(x) & = \mathcal{N}(x|\mu, \Lambda^{-1}) \\
p(y|x) & = \mathcal{N}(y|Ax + b, L^{-1})
\end{align*}
\]

where \( x \in \mathbb{R}^M, y \in \mathbb{R}^D \) and \( A \in \mathbb{R}^{D \times M} \). The last term \( b \) refers to the mean of the conditional distribution, while \( \mu \) refers to the marginal distribution. \( \Lambda, L \) are the precision matrices of marginal and conditional distributions, respectively. Let us define

\[
z = \begin{pmatrix} x \\ y \end{pmatrix}.
\]
Then,
\[ \ln p(z) = \ln p(x) + \ln p(y|x) \]
\[ = -\frac{1}{2} (x - \mu)^T \Lambda (x - \mu) - \frac{1}{2} (y - Ax - b)^T L (y - Ax - b) + C \]  
(4.33)

where \( C \) is a constant. The next step is to define the precision matrix. From the second-order terms in (4.33) we can obtain,
\[ -\frac{1}{2} x^T (\Lambda + A^T L A) x - \frac{1}{2} y^T L y + \frac{1}{2} y^T L A x + \frac{1}{2} x^T A^T L y = \]
\[ -\frac{1}{2} \begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} \Lambda + A^T L A & -A^T L \\ -L A & L \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \]
\[ -\frac{1}{2} z^T R z \]  
(4.34)

where \( R \) is the precision matrix. The covariance matrix is \( \Sigma = R^{-1} \) and it is defined as:
\[ \Sigma = \begin{pmatrix}
\Sigma_x \\ \Lambda^{-1}
\end{pmatrix}
\begin{pmatrix}
\Sigma_{xy} \\ \Lambda^{-1} A^T
\end{pmatrix}
\begin{pmatrix}
\Sigma_y \\ \Lambda^{-1} A
\end{pmatrix}
\begin{pmatrix}
\Sigma_{xy}^T \\ \Lambda^{-1} A^T L
\end{pmatrix}.
\]  
(4.35)

Gathering the first-order terms of (4.33) we obtain
\[ x^T \Lambda \mu - x^T A^T L b + y^T L b = \begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} \Lambda \mu - A^T L b \\ L b \end{bmatrix} \]  
(4.36)

In order to define the mean, we apply the same technique as in conditional and marginal distributions. By completing the square we arrive at:
\[ \mathbb{E}[z] = R^{-1} \begin{bmatrix} \Lambda \mu - A^T L b \\ L b \end{bmatrix} = \begin{bmatrix} \mu \\ A \mu + b \end{bmatrix}. \]  
(4.37)

Finally, we can define the mean and covariance matrix of \( p(y) \) as
\[ \mathbb{E}\{y\} = A \mu + b \]  
(4.38)
\[ \text{cov}\{y\} = L^{-1} + A \Lambda^{-1} A^T \]  
(4.39)

Respectively, the mean and covariance of \( p(x|y) \) are defined as
\[ \mathbb{E}\{x|y\} = (\Lambda + A^T L A)^{-1} (A^T L (y - b) + L \mu), \]  
(4.40)
\[ \text{cov}\{x|y\} = (\Lambda + A^T A L)^{-1}. \]  
(4.41)
The distribution \( p(x) \) can be regarded as the prior distribution of \( x \). Accordingly, distribution \( p(x|y) \) can be considered the posterior. The form of distributions \( p(y) \) and \( (x|y) \) becomes,

\[
p(y) = \mathcal{N}(y|\Lambda \mu + b, L^{-1} + A \Lambda L \Lambda^T) \tag{4.42}
\]

\[
p(x|y) = \mathcal{N}(x|\Sigma \{A^T L (y - b) + A \mu\}, \Sigma), \tag{4.43}
\]

\[
\Sigma = (\Lambda + A^T L A)^{-1} \tag{4.44}
\]

### 4.2 Variational Bayesian methods

The parameters approximation in complex Bayesian models is usually related to approximation of intractable integrals. In order to solve such problems, variational approximations are applied. The objective is to approximate a conditional over the latent variables given the observed variables. The solution to this problem is provided by optimization. The main idea is to use a family of densities over the latent variables [40] and seek the specific distribution family that is closest to the posterior we are interested in, according to Kullback-Leibler (KL) divergence.

Let us define the hidden variables \( z = z_{1:n} \), the observations \( x = x_{1:n} \), and some fixed parameters \( \theta \). The posterior distribution is

\[
p(z|x, \theta) = \frac{p(z, x|\theta)}{\int_z p(z|x|\theta)}. \tag{4.45}
\]

The calculation of the posterior is impossible in many cases. An example is the Bayesian mixture of Gaussians. In this case, after omitting the parameters that are fixed, the posterior distribution becomes

\[
p(\mu_{1:K}, z_{1:n}|x_{1:n}) = \frac{\prod_{k=1}^K p(\mu_k) \prod_{i=1}^n p(z_i)p(x_i|z_i, \mu_{1:K})}{\int_{\mu_{1:K}} \sum_{z_{1:n}} \prod_{k=1}^K p(\mu_k) \prod_{i=1}^n p(z_i)p(x_i|z_i, \mu_{1:K})} \tag{4.46}
\]

where \( \mu_k \sim \mathcal{N}(0, \tau^2) \) for \( k = 1, \ldots, K \) are the means. A solution for the numerator is feasible. However, the denominator consists of \( K^n \) terms and the bigger value \( n \) takes the more intractable the problem becomes. The solution is provided through variational approximation.

KL approach is based on fitting a Gaussian to the desired distribution. The KL divergence is given by,

\[
KL(q||p) = \mathbb{E}_q \left[ \log \frac{q(Z)}{p(Z|x)} \right] \tag{4.47}
\]
where \( \mathbf{Z} \) denotes a random vector. Actually KL divergence cannot be minimized. However, a function which is equal to KL, up to an additive constant can be utilized. This function is known as the evidence lower bound (ELBO) function. ELBO function can be obtained via the application of Jensen’s inequality for concave functions, such as the logarithm [41]:

\[
f(\mathbb{E}[\mathbf{X}]) \geq \mathbb{E}[f(\mathbf{X})],
\]

(4.48)

\[
\log p(\mathbf{x}) = \log \int_z p(\mathbf{x}, z) dz = \log \int_z p(\mathbf{x}, z) \frac{q(z)}{q(z)} dz = \log \left( \mathbb{E}_q \left[ \frac{p(\mathbf{x}, \mathbf{Z})}{q(\mathbf{Z})} \right] \right) \geq \mathbb{E}_q[\log p(\mathbf{x}, \mathbf{Z})] - \mathbb{E}_q[\log q(\mathbf{Z})] \triangleq \text{ELBO}(q)
\]

(4.49)

In (4.49) the first term is the expected logarithm of the joint probability, while the second term is the entropy of the variational distribution, which depends on latent variables \( \mathbf{Z} \).

The next step is to restrict \( \mathbf{Z} \), in a way that belongs in a tractable distribution family. Then, the maximization of ELBO is equivalent to the minimization of the KL divergence between the approximating distribution and the true posterior as is shown next.

We wish to obtain parameters that produce as tight bounds as possible of the marginal likelihood of observations \( \mathbf{x} \). Solving this maximization problem is equivalent to finding the proper distribution family that is closest to the true posterior [42]:

\[
\text{KL}(q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{x})) = \mathbb{E}_q \left[ \log \frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{x})} \right] = \mathbb{E}_q[\log q(\mathbf{Z})] - \mathbb{E}_q[\log p(\mathbf{Z}|\mathbf{x})] = -(\mathbb{E}_q[\log p(\mathbf{Z}, \mathbf{x})] - \mathbb{E}_q[\log q(\mathbf{Z})]) + \log p(\mathbf{x}) = -\text{ELBO}(q) + \log p(\mathbf{x})
\]

(4.50)

where ELBO denotes the evidence lower bound. The logarithm of the marginal distribution \( \log p(\mathbf{x}) \), is also referred to as (log) evidence and it can be expressed with respect to ELBO as,

\[
\log p(\mathbf{x}) = \text{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) + \text{ELBO}(q)
\]

(4.51)
The variational bound arises from the fact that $\text{KL}(\cdot) \geq 0$ [43]. That is,

$$\log p(x) \geq \text{ELBO}(q).$$  \hspace{1cm} (4.52)

The premise is that ELBO approximates well the marginal distribution. Therefore, it can be used as a criterion to select the optimal model.

In order to simplify the optimization process, a simple distribution family have to be chosen. One of the simplest distribution families, is the **mean-field variational family**. Hidden variables in this form of distribution, are mutually independent. Let this family is factorized as,

$$q(z_1, \ldots, z_m) = \prod_{j=1}^{m} q_j(z_j)$$  \hspace{1cm} (4.53)

As we see, the latent variables are partitioned into sets and factorization can be applied in each set of variables. Afterwords, the goal is to optimize the ELBO function over (4.53). For this purpose, a coordinate ascent algorithm can be applied [44]. This is an iterative optimization method, according to which all variational distributions are kept fixed except the one to be optimized. The procedure is the following:

1. Let us apply the chain rule of probabilities, in order to decompose

$$p(z_1:m, x_{1:n}) = p(x_{1:n}) \prod_{j=1}^{m} p(z_j | z_1:(j-1), x_{1:n})$$  \hspace{1cm} (4.54)

2. The entropy of the variational distribution is decomposed

$$\mathbb{E}[\log q(Z_{1:m})] = \sum_{j=1}^{m} \mathbb{E}_j[\log q(Z_j)],$$  \hspace{1cm} (4.55)

where $\mathbb{E}_j[\cdot]$ is the expectation w.r.t $q(Z_j)$

3. ELBO decomposition is the next step

$$\mathcal{L} = \log p(x_{1:n}) + \sum_{j=1}^{m} \left\{ \mathbb{E}[\log p(Z_j | Z_{1:(j-1)}, X_{1:n})] - \mathbb{E}_j[\log q(Z_j)] \right\}$$  \hspace{1cm} (4.56)
4. Let us elaborate the term inside the sum in (4.56). We have:

\[
\mathbb{E}_{q}[\log p(Z_j|Z_{-j}, X_{1:n})] = \int \prod_i q_i(z_i) \log p(z_j|z_{-j}, x_{1:n}) dz_j dZ_{-j}
\]

\[
= \int dq_j(z_j) \int dZ_{-j} \prod_{i \neq j} q_j(z_j) \log p(z_j|z_{-j}, x_{1:n})
\]

\[
= \int dq_j(z_j) \mathbb{E}_{-j}[\log p(z_j|z_{-j}, x_{1:n})]
\]

(4.57)

because the term \( \prod_{i \neq j} q_j(z_j) \) is recognized to be the conditional pdf when the \( j \)th latent variable is excluded. Let us separate the \( k \)th term in (4.56):

\[
L_k = \int q(z_k) \mathbb{E}_{-k}[\log p(z_k|Z_{-k}, X)] dz_k - \int q_k(z_k) \log q_k(z_k) dz_k
\]

(4.58)

5. In addition the constraint \( \int q_k(z_k) dz_k = 1 \) should be satisfied. Let \( \xi \) be the Lagrange multiplier. If we set the derivative of the Lagrangian w.r.t. \( q_k(z_k) \) equal to 0 we obtain:

\[
\frac{d \hat{L}}{dq(z_k)} = \mathbb{E}_{-k}[\log p(z_k|Z_{-k}, X)] - \log q(z_k) - 1 + \xi = 0
\]

(4.59)

6. Solving for \( q_k(z_k) \) and inserting the expression to the constraint, we obtain the following expression:

\[
q_k^*(z_k) \propto \exp \{\mathbb{E}_{-k}[\log p(z_k|Z_{-k}, X)]\},
\]

(4.60)

This is an iterative update procedure for each \( q(z_k) \), which leads ELBO to a convergence to a local minimum. The algorithm is efficient, because the analysis is not applied on the whole data set before improving the global variational parameters. Additionally, the required computations at each iteration are about a single local context [44].

4.3 Brownian motion

Brownian motion refers to the random motion that fluid particles follow. This mechanism is the result of the irregular repeated collisions of particles. It is named after botanist Robert Brown who in 1827 observed this phenomenon
in pollen grains lying in water. In 1905, Albert Einstein confirmed this theory [45]. The first approach to the Brownian motion as a stochastic process was developed by Norbert Wiener in 1918, who made the assumption that probabilistic laws governing the paths, which liquid particles follow [46]. Thus, the Brownian motion, is also referred to as Wiener process and is usually denoted by $W(t)$.

**Standard Brownian motion.** It is a stochastic process $\{W(t) : t \in [0, \infty)\}$ satisfying the properties:

- $W(t) \sim \mathcal{N}(0, t), \forall t \geq 0$,
- $W(0) = 0$,
- $\forall n, 0 \leq t_0 < t_1 < \cdots < t_n < \infty$, the increments $W(t_1) - W(t_0), W(t_2) - W(t_1), \ldots, W(t_n) - W(t_{n-1})$ are independent,
- for $s \leq t$, $W(t) - W(s) \sim W(t - s) - W(0) = W(t - s) \sim \mathcal{N}(0, t - s)$, the increments are stationary, normally distributed with zero mean and variance $t - s$,
- the Brownian motion paths are continuous, $\forall \omega \in \Omega, t \mapsto W(t)(\omega)$

In standard Brownian motion, the mean is constant, i.e., zero, and does not vary.

**Brownian Motion with drift.** A random variable $X(t)$ follows a Brownian motion with drift at time $t \mapsto dX = \alpha dt + \sigma dW(t)$. (4.61)

This is a stochastic differential equation (SDE), where $\alpha, \sigma, dW(t)$ denote the drift, the volatility, and the increment of standard Brownian motion, respectively. If $\alpha$ and $\sigma$ are known constants and $\sigma > 0$, then (4.61) describes an arithmetic Brownian motion. If we integrate (4.61), we obtain

$$X(t) = x_0 + \alpha t + \sigma W(t)$$

where $x_0 = X(0)$ is the initial value. $X(t)$ is normally distributed and the mean and variance of the distribution are defined as

$$\mathbb{E}[X(t)] = X(0) + \alpha t \quad \text{and} \quad \text{Var}[X(t)] = \sigma^2 t.$$  

Therefore, the drift has a constant rate. The probability density function is

$$f(t, x) = \frac{1}{\sigma \sqrt{2\pi t}} e^{-\frac{(x - x_0 - \alpha t)^2}{2\sigma^2 t}}.$$  

24
Geometric Brownian motion with drift. Changes occur in random process $dX(t)$ are related to the value $X(t)$ admits in the current state. This is described by a rate of changes and the according stochastic differential equation

$$
\frac{dX(t)}{X(t)} = \alpha dt + \sigma dW(t) \Rightarrow dX(t) = \alpha X(t) dt + \sigma X(t) dW(t)
$$

(4.66)

where the drift term $\alpha X(t)$ and volatility term $\sigma X(t)$ are linearly related to the current state value $X(t)$. In this case, the magnitude of the drift term is proportional to $X(t)$, and not a constant one. If we apply Itô’s lemma in SDE of (4.66), we obtain the solution

$$
X(t) = e^{\log x_0 + \mu t + \sigma W(t)}
$$

$$
= x_0 e^{\mu t + \sigma W(t)}
$$

(4.67)

where

$$
\mu = \alpha - \frac{1}{2} \sigma^2.
$$

(4.68)

Therefore, $X(t)$ has a log-normal distribution with mean, variance, and probability density function defined as

$$
E(X(t)) = x_0 e^{\mu t},
$$

(4.69)

$$
Var(X(t)) = x_0^2 e^{2\mu t}(e^{\sigma^2 t} - 1),
$$

(4.70)

$$
f(t,x) = \frac{1}{\sigma x \sqrt{2\pi t}} e^{-\frac{(\log x - \log x_0 - \mu t)^2}{2\sigma^2 t}}.
$$

(4.71)

Geometric Brownian motion models are applied in stock price prediction, in order to obtain non-negative values.
Chapter 5

Collaborative Kalman Filter

5.1 Introduction to Kalman filter

A wide range of problems utilize the Kalman filter. It is a highly efficient estimator, also referred to as Linear Quadratic Estimator (LQE). It is playing an increasingly important role in computer science and engineering as well. The Kalman filter is usually applied to stochastic estimations from noisy measurements. Its name comes from Rudolph Kalman. In 1960, Rudolph Kalman published his paper on the linear filtering problem [47].

The Kalman filter, is basically an arrangement of equations. If some particular conditions are met, the error covariance of the state estimate is minimized, through optimization. Estimations are accomplished by prediction and correction. Specifically, estimations are carried out for the current state at time $t$. In the next state, information from the previous one is used with measurements including noise. Thus, knowledge about the next state is obtained, prior to accessing it. This knowledge defines the a-priori estimates. It is a form of feedback.

The mathematical equations are categorized into measurement and time update equations. In time evolving processes we are interested in the latter. Usually both of them are combined to form a “predictor-corrector” model.

Through time-update equations, the state in current time step is propagated to the next time step, along with the error covariance estimate. This defines an a priori estimate of the next state.

The goal is to set up a recursion for the state $x_k \in \mathbb{R}^d$ at time step $k$ described by [4]:

$$x_k = Ax_{k-1} + Bu_k + w_{k-1}. \quad (5.1)$$

The actual measurement $z \in \mathbb{R}^p$ is defined as

$$z_k = Hx_k + v_k. \quad (5.2)$$
The mutually independent random vectors \( w, v \) denote the process and measurement noise, respectively. These vectors follow the normal probability distribution:

\[
p(w) \sim \mathcal{N}(0, Q) \\
p(v) \sim \mathcal{N}(0, R)
\]  

(5.3)

where \( Q \in \mathbb{R}^{d \times d} \) is the process noise covariance and \( R \in \mathbb{R}^{p \times p} \) is the measurement noise covariance. In Equation (5.1) the \( d \times d \) matrix \( A \) provides association between the previous time step \( k - 1 \) and the current step \( k \). The association between the input \( u \in \mathbb{R}^l \) and the state \( x \) is provided by the \( d \times l \) matrix \( B \). The \( p \times d \) matrix \( H \) associates the measurement \( z \) to the state.

At step \( k \), \( \hat{x}^- \in \mathbb{R}^d \) is defined as the \textit{a priori} state estimate. Let \( \hat{x} \in \mathbb{R}^d \) denote \textit{a posteriori} estimate of this state, given the measurement \( z_k \). The \textit{a priori} and the \textit{a posteriori} error estimations are given by

\[
e^-_k \equiv x_k - \hat{x}^- \\
x_k \equiv x_k - \hat{x}.
\]  

(5.4)

The \textit{a priori} and \textit{a posteriori} error covariance estimates are formed as

\[
P^-_k = \mathbb{E}[e^-_k e^{-T}_k] \\
P_k = \mathbb{E}[e_k e^T_k].
\]  

(5.5)  

(5.6)

The goal is to derive an equation in order to compute the \textit{a posteriori} state estimation \( \hat{x}_k \),

\[
\dot{x}_k = \hat{x}^-_k + K(z_k - H\hat{x}^-_k)
\]  

(5.7)

where the term \((z_k - H\hat{x}^-_k)\) is referred to as the residual. It is equivalent to the difference between the actual measurement and the predicted measurement. The matrix \( K \) is the gain, which is chosen in such a way as to minimizes the \textit{a posteriori} error covariance estimate (5.6). It weighs the difference of measurements mentioned before. A popular form of matrix \( K \) that minimizes (5.6) is

\[
K_k = P^-_k H^T(P^-_k H^T + R)^{-1} = \frac{P^-_k H^T}{HP^-_k H^T + R}
\]  

(5.8)

where it is obvious that as the \textit{a priori} error covariance estimation approaches zero, the gain \( K \) weighs the residual less heavily. Conversely, as the error covariance of the measurement \( R \) approaches zero, the gain \( K \) weights the residual more heavily. Herewith, as \( R \) decreases, \( z_k \) becomes more accurate, while \( H\hat{x}^-_k \) is less accurate. Besides, as \( P^-_k \) decreases, \( z_k \) becomes less accurate, while \( H\hat{x}^-_k \) becomes more accurate [4].
In order to propagate the state and covariance estimations to the next state, the time update equations as mentioned in Section 5.1, are formed as

\[
\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{5.9}
\]

\[
P_k^- = AP_{k-1}A^T + Q \tag{5.10}
\]

where \(A, B\) are those appearing in (5.1).

### 5.2 Collaborative Kalman filtering

In a user/item setting we can consider the locations of users \(u_i\) and items \(w_j\) as being in a latent state space and specifically move within it, because temporal changes may occur in user preferences. The predictions for each dyad in every time step is carried out by utilizing the dot product of \(u_i\) and \(w_j\) as a parameter to a distribution. In the dynamic setting, user \(u_i\) and item \(w_j\) vectors are functions of time.

Kalman filter extends matrix factorization to dynamic setting. In continuous-time framework, it provides the ability to model a sequence of observed vectors \(y_n \in \mathbb{R}^p\) as linear functions of latent state vectors \(w_n \in \mathbb{R}^d\) along with some noise [5]. Specifically, the current state of these vectors is equivalent to the previous one, with additional noise. If the vector \(w_0\) is said to have a Gaussian prior distribution, then for each next state \(n = 1, \ldots, N\) and zero-mean noise, these vectors are formed as

\[
w_{n+1} | w_n \sim N(w_n, \alpha I) \tag{5.11}
\]

\[
y_{n+1} | w_{n+1} \sim N(Aw_{n+1}, \sigma^2 I). \tag{5.12}
\]

In (5.11), \(w_{n+1}\) is the latent vector of the next state. The Gaussian distribution has a mean vector equal to the current state, \(w_n\) and covariance matrix \(\alpha I\). In (5.12) \(y_{n+1}\) denotes the observed vector in state \(n + 1\). The Gaussian distribution has a mean vector equal to \(Aw_{n+1}\) and a covariance matrix equal to \(\sigma^2 I\).

To obtain the prior distribution of the current state, marginalisation on the previous state is required. Then, in real-time prediction models, when an observation occurs, the current state posterior should be calculated. These steps ensure that the model is able to capture latent state vector motion.

Assume that the latent state vectors \(w_n\) in the current state are said to have multivariate normal distribution \(w_n \sim N(\mu_n, \Sigma_n)\), with mean \(\mu_n\) and covariance \(\Sigma_n\). To obtain next state \(w_{n+1}\) distribution, we marginalize with respect to the current one:

\[
p(w_{n+1}) = \int_{\mathbb{R}^d} p(w_{n+1} | w_n)p(w_n)dw_n = N(w_{n+1} | \mu_n, aI + \Sigma_n). \tag{5.13}
\]
In (5.13), $\alpha$ is a drift parameter, which reflects the volatility of $w_n$ in one unit of time. In continuous-time problems, in order to define a Brownian motion on $w_n$, we utilize $\Delta t_n \alpha I$ instead of $\alpha I$. $\Delta t_n$ stands for the time difference between two events. The transition from one state to the next one, depends on this time difference.

Subsequently, after the occurrence of a new observation, the calculation of the posterior is done as follows:

$$p(w_{n+1} \mid y_{n+1}) \propto p(y_{n+1} \mid w_{n+1})p(w_{n+1}) = \mathcal{N}(w_{n+1} \mid \mu_{n+1}, \Sigma_{n+1})$$ (5.14)

where the covariance matrix and the mean vector are defined as:

$$\Sigma_{n+1} = \left( \frac{A^T A}{\sigma^2} + B_n^{-1} \right)^{-1}$$ (5.15)

$$\mu_{n+1} = \Sigma_{n+1} \left( \frac{A^T y}{\sigma^2} + B_n^{-1} \mu_n \right)$$ (5.16)

where $B_n = \alpha I + \Sigma_n$. The posterior of $w_{n+1}$ then can be used to obtain the prior of the next state ($w_{n+2}$) and so on.

Output prediction. Assume the ratings are real numbers. Accordingly the real line $R$ is divided into as many segments as the number of classes. For example, if a five-star rating system is given, the real line is partitioned into five segments. Let the partition for class $k$ be

$$I_k = (l_k, r_k)$$ (5.17)

where $l_k < r_k, r_k = l_{k+1}, l_k = r_{k-1}$ and $k = 1, \ldots, m$, with $m$ denoting the number of segments. These segments are ordered. We obtain the output prediction for each dyad $(i, j)$ at time $t$, $z_{ij}[t] \in \{1, \ldots, m\}$, by drawing from a Gaussian distribution $\mathcal{N}(\langle u_i[t], w_j[t] \rangle, \sigma^2)$ and discovering in which region it belongs to. $\langle u_i[t], w_j[t] \rangle = u_i[t]^T w_j[t]$ denotes the inner product between its two vector arguments. This consideration is related to probit models and specifically to ordered probit regression [48]. In ordered probit models, the dependent variable takes potential values that have a natural ordering. The goal is to estimate the probability of the dependent variable falling into a specific category. This is usually accomplished by applying maximum likelihood estimation.

Likelihood model. As we mentioned, $u_i[t] \in R^d$ and $w_j[t] \in R^d$ are the latent state vectors at time $t$ for user $i$ and item $j$ respectively. The goal now is to calculate the probability that the dyad $(i, j)$ falls within the class $k$ or

$$P(z_{ij}[t] = k \mid u_i, w_j) = \int_{I_k} \mathcal{N}(y \mid \langle u_i[t], w_j[t] \rangle, \sigma^2)dy.$$ (5.18)
Let $y_{ij}[t]$ be a latent variable,

$$z_{ij}[t] \mid y_{ij}[t] = \sum_k k \, \mathbb{I}(y_{ij}[t] \in I_k)$$  \hspace{1cm} (5.19)$$

$$y_{ij}[t] \mid u_i, w_j \sim \mathcal{N}(\langle u_i[t], w_j[t] \rangle, \sigma^2)$$  \hspace{1cm} (5.20)$$

where $\mathbb{I}()$ denotes an indicator function. This approximation provides flexibility to user-item interactions, because they are allowed to interact multiple times through time.

**Prior distribution:** As mentioned above, the prior model is a latent multidimensional Brownian motion. In Brownian motion, the duration of time between two events should be defined. Thus, $\Delta_{u_i}^t$ and $\Delta_{w_j}^t$ represent the duration of time since the last event emergence for users and items, respectively. The posterior distributions in Kalman filters are in fact multivariate normal distributions, defined as

$$u_i[t - \Delta_{u_i}^t] \sim \mathcal{N}(\mu'_{u_i}[t - \Delta_{u_i}^t], \Sigma'_{u_i}[t - \Delta_{u_i}^t]),$$  \hspace{1cm} (5.21)$$

$$w_j[t - \Delta_{w_j}^t] \sim \mathcal{N}(\mu'_{w_j}[t - \Delta_{w_j}^t], \Sigma'_{w_j}[t - \Delta_{w_j}^t])$$  \hspace{1cm} (5.22)$$

where $t - \Delta_{u_i}^t$ denotes the time elapsed, since user $u_i$ was previously observed. The mean vector $\mu'$ and the covariance matrix $\Sigma'$ are also dynamically evolving quantities.

In (5.13), via marginalisation the prior distributions at time $t$ for user $u_i$ and item $w_j$, are obtained

$$u_i[t] \sim \mathcal{N}(\mu_{u_i}[t], \Sigma_{u_i}[t]),$$  \hspace{1cm} (5.23)$$

$$w_j[t] \sim \mathcal{N}(\mu_{w_j}[t], \Sigma_{w_j}[t]),$$  \hspace{1cm} (5.24)$$
where
\[
\mu[t] = \mu'[t - \Delta[t]] \tag{5.25}
\]
\[
\Sigma[t] = \Sigma'[t - \Delta[t]] + \Delta[t] \alpha \mathbf{I}. \tag{5.26}
\]

Hereafter, the parameters with a prime are related to posterior distributions. The term $\alpha \mathbf{I}$ represents the drift process. It is interpreted as the drift covariance matrix in one unit of time. This means that the procedure of transition from parameters of the latest update of posterior distribution to the parameters of the current update of the prior distribution, is accomplished through the addition of a small number to the diagonal of the posterior covariance matrix $\Sigma'[t - \Delta[t]]$.

**Drift.** The drift parameter $\alpha$ can be considered as a hyperparameter of the distributions. It is responsible for vector motion magnitude at each time step. For example, as $\alpha$ is getting big values, the vectors are able to move bigger distances in order to fit properly $\mathbf{y}$, but at the same time previous information is discarded. Reversely as $\alpha \to 0$ the vectors cover smaller distances, the previous information is retained and the posterior distribution is more concentrated at the mean. This results in converging to a specific point.

The drift parameter $\alpha$ is dynamically evolving as well. This is crucial in stock price prediction models, because the rate of price changes is not constant. In order to exclude negative values, we model $\alpha$ to be a geometric Brownian motion as $\alpha[t] = e^{a[t]}$, where $a$ is a Brownian motion,
\[
a[t] \sim \mathcal{N}(a[t - \Delta_a[t]], c \Delta_a[t]) \tag{5.27}
\]
with $t - \Delta_a[t]$ being the time elapsed since $a[t]$ was previously measured and $c$ being an extra drift parameter. If we expand (5.26) to this dynamic setting of $a$, instead of adding a fixed value to the diagonal of the posterior covariance matrix, we add the geometric Brownian motion drift parameter, to employ a stochastic integral in the transition between the posterior to the prior:
\[
\Sigma[t] = \Sigma'[t - \Delta[t]] + I \int_{t - \Delta_a[t]}^{t} e^{a[s]} ds. \tag{5.28}
\]
5.3 Variational Inference

5.3.1 Variational inference and drift

To ensure the fact that the Brownian motion \( \alpha \) is properly updated, a suitable approximation of the stochastic integral (5.28) is required [49], [50], [5], i.e.,

\[
\int_{t-\Delta^u_t}^{t} e^{a[s]}ds \approx e^{a[t]}\Delta^u_t
\]

(5.29)

where \( a[t] \) can either be unique for each user \( u_i \), or common (shared) to all users. To approximate the multidimensional Brownian motion, the first step is to draw the log drift value \( a[t] \) at time \( t \). Then, we consider the drift parameter to be fixed in the unobserved interval, in order to obtain an approximation to it [5].

\[
a[t] \sim \mathcal{N}(a[t] - \Delta^u_t, c\Delta^u_t),
\]

(5.30)

\[
\mathbf{u}[t] \sim \mathcal{N}(\mu_u[t] - \Delta^u_t, \Sigma_u[t - \Delta^u_t] + e^{a[t]}\Delta^u_t \mathbf{I}).
\]

(5.31)

Thus, the variational approximation is accomplished by a point estimation of the Brownian motion \( a[t] \).

5.3.2 Variational inference model

The posterior of an event at time \( t \) based on the latent state variables is

\[
p(u_i[t], w_j[t], y_{ij}[t] | z_{ij}[t], a_{u_i}, a_{w_j}) \propto
p(z_{ij}[t] | y_{ij}[t]) p(y_{ij}[t] | u_i[t], w_j[t]) p(u_i[t] | a_{u_i}) p(w_j[t] | a_{w_j}).
\]

(5.32)

The fundamental problem in posterior calculation is intractability. To handle such problem the mean-field variational method is applied to approximate properly the posterior [51]. This procedure enables us to conduct tractable calculations. The posterior (5.32) can be approximated through a factorized distribution i.e.,

\[
q(u_i[t], w_j[t], y_{ij}[t]) = q(u_i[t]) q(w_j[t]) q(y_{ij}[t]).
\]

(5.33)

Within this procedure, the posterior distribution of each variable is being approximated by distribution \( q(\cdot) \) assuming independent variables. This can be obtained by minimizing the Kullback-Leibler (KL) divergence between \( q \) and the true posterior distribution at time \( t \). This is simply equivalent to maximizing the following variational objective function [42]:

\[
\mathcal{L}_t = \mathbb{E}_q[\ln p(z_{ij}[t], u_i[t], w_j[t], y_{ij}[t])] - \mathbb{E}_q[\ln q]
\]

(5.34)

Optimization with respect to the parameters of distribution \( q \) is required in order to converge to a local maximum, because \( \mathcal{L}_t \) is a non-convex function.
5.3.3 Variational distribution calculation

In order to elaborate further the objective function (5.34), the expectation of the log joint likelihood should be added to the entropy of each distribution \( q \). In this manner, we should define the distribution family of each distribution \( q \). This could be accomplished by exponentiating the expected log joint likelihood and excluding the \( q \) distribution, which is under investigation, \[ q_i \propto \exp\{\mathbb{E}_{q_i}[\ln p(\cdot)]\}. \] (5.35)

The hidden data distribution \( q(y_{ij}) \) can be calculated through,

\[ q(y_{ij}) \propto \exp\{\ln p(z_{ij} \mid y_{ij}) + \mathbb{E}_q[\ln p(y_{ij} \mid u_i, w_j)]\} \] (5.36)

where, the time notation have been omitted. At this point, the notion of class as mentioned in Section 5.2 should be inserted. The distribution of the output \( i, j \) at time \( t \), \( z_{ij}[t] \), is:

\[ p(z_{ij}[t] \mid y_{ij}[t]) = \mathbb{I}(y_{ij}[t] \in \mathcal{I}_{z_{ij}[t]}). \] (5.37)

Accordingly, for the hidden data \( y_{ij} \) the distribution \( p(y_{ij}[t]) \) at time \( t \), is defined on the interval \( \mathcal{I}_{z_{ij}[t]} \), associated to class \( z_{ij}[t] \),

\[ q(y_{ij}[t]) = \mathcal{T}\mathcal{N}_{\mathcal{I}_{z_{ij}[t]}}(y_{ij}[t] \mid \langle \mathbb{E}_q[u_i[t]], \mathbb{E}_q[w_j[t]] \rangle, \sigma^2) \] (5.38)

where \( \mathcal{T}\mathcal{N}(\cdot) \) denotes a truncated normal distribution with mean \( \langle \mathbb{E}_q[u_i[t]], \mathbb{E}_q[w_j[t]] \rangle \) and variance \( \sigma^2 \), because the values of \( y_{ij} \) lie within an interval.

To find optimal distributions of the remaining variables the same procedure is applied:

\[ q(u_i) \propto \exp\{\ln p(u_i) + \mathbb{E}_q[\ln p(y_{ij} \mid u_i, w_j)]\}. \] (5.39)

It can be shown that \( q(u_i) \) is a multivariate Gaussian,

\[ q(u_i[t]) = \mathcal{N}(u_i[t] \mid \mu'_{u_i[t]}, \Sigma'_{u_i[t]}). \] (5.40)

Similarly for \( w_j \):

\[ q(w_j) \propto \exp\{\ln p(w_j) + \mathbb{E}_q[\ln p(y_{ij} \mid u_i, w_j)]\} \] (5.41)

\[ q(w_j[t]) = \mathcal{N}(w_j[t] \mid \mu'_{w_j[t]}, \Sigma'_{w_j[t]}). \] (5.42)
5.3.4 Updating the approximate distributions

Since our model has dynamic behaviour, the approximate distribution \( q(\cdot) \) of each variable should evolve as well. For this purpose, updates at each time step should be derived:

- **\( q(y_{ij}[t]) \) coordinate ascent update.** The first step in updating the approximate distributions, is to update their parameters. The mean parameter at time \( t \) update is given by,

\[
m_{ij}[t] = \langle \mathbb{E}_q[u_i[t]], \mathbb{E}_q[w_j[t]] \rangle
\]

(5.43)

where \( \langle \mathbb{E}_q[u_i[t]], \mathbb{E}_q[w_j[t]] \rangle \) is the inner product of prior means of user \( i \) and item \( j \), respectively. The outcome for \( y_{ij}[t] \) depends on the interval in which its expected value falls. Assume that this interval consists of two boundaries with respect to \( z_{ij}[t] \); a left one \( l_{z_{ij}}[t] \) and right one \( r_{z_{ij}}[t] \).

Let the definition of interval magnitude arise from

\[
\beta_{ij}[t] = l_{z_{ij}}[t] - m_{ij}[t], \quad \gamma_{ij}[t] = r_{z_{ij}}[t] - m_{ij}[t]
\]

(5.44)

Then, the expected value of \( y_{ij}[t] \) is given by,

\[
\mathbb{E}_q[y_{ij}[t]] = m_{ij}[t] + \sigma \frac{\phi(\beta_{ij}[t]) - \phi(\gamma_{ij}[t])}{\Phi(\gamma_{ij}[t]) - \Phi(\beta_{ij}[t])}
\]

(5.45)

where \( \phi(\cdot) \) stands for probability density function (pdf) and \( \Phi(\cdot) \) for cumulative distribution function (cdf) of a standard normal distribution \( \mathcal{N}(0, 1) \).

- **The coordinate update of \( q(u), q(w) \).** The posterior covariance matrix \( \Sigma'_{u_i}[t] \) and the posterior mean vector \( \mu'_{u_i}[t] \) at time \( t \) should be updated as follows:

\[
\Sigma'_{u_i} = (\Sigma^{-1}_{u_i} + (\mu'_{w_j} \mu'^T_{w_j} + \Sigma'_{w_j})/\sigma^2)^{-1}
\]

(5.46)

\[
\mu'_{u_i} = \Sigma'_{u_i} (\mathbb{E}_q[y_{ij}] \mu'_{w_j}/\sigma^2 + \Sigma^{-1}_{u_i} \mu_{u_i})
\]

(5.47)

where \( \mu_{u_i}[t] \) and \( \Sigma_{u_i}[t] \) are prior parameters, defined as,

\[
\mu_{u_i}[t] = \mu'_{u_i}[t - \Delta_{u_i}^{[t]}]
\]

(5.48a)

\[
\Sigma_{u_i} \approx \Sigma'_{u_i}[t - \Delta_{u_i}^{[t]}] + \sigma^2_{u_i}[t] \Delta^{[t]}_{u_i}
\]

(5.48b)

The updates for the parameters of the posterior item distribution of \( q(w) \) are similar to (5.48a) and (5.48b).
5.4 Geometric Brownian motion inference

In order to infer the drift process in geometric Brownian motion, assuming individual drift parameters \( a \) for each user \( u \) and item \( w \), a point estimation procedure is applied. The updates are conducted by approximating the relevant terms in likelihood with a second-order Taylor expansion about the point \( a_{ui}[t - \Delta_{ui}^{[t]}] \), which is the last inferred value. Let us assume the second-order Taylor approximation of \( f(a_{ui}[t]) \):

\[
f(a_{ui}[t]) \approx f(a_{ui}[t - \Delta_{ui}^{[t]}]) + (a_{ui}[t] - a_{ui}[t - \Delta_{ui}^{[t]}]) f'(a_{ui}[t - \Delta_{ui}^{[t]}]) +
\frac{1}{2}(a_{ui}[t] - a_{ui}[t - \Delta_{ui}^{[t]}])^2 f''(a_{ui}[t - \Delta_{ui}^{[t]}])
\]

we are seeking to optimize the objective function (5.49) with respect to \( a_{ui}[t] \). Thus, we solve for

\[
f'(a_{ui}[t]) = \frac{df(a_{ui}[t])}{da_{ui}[t]} = 0 \iff
f'(a_{ui}[t - \Delta_{ui}^{[t]}]) + (a_{ui}[t] - a_{ui}[t - \Delta_{ui}^{[t]}]) f''(a_{ui}[t - \Delta_{ui}^{[t]}]) = 0 \iff
a_{ui}[t] - a_{ui}[t - \Delta_{ui}^{[t]}] = -\frac{f'(a[t - \Delta_{ui}^{[t]}])}{f''(a[t - \Delta_{ui}^{[t]}])} \iff
a_{ui}[t] = a_{ui}[t - \Delta_{ui}^{[t]}] - \frac{f'(a[t - \Delta_{ui}^{[t]}])}{f''(a[t - \Delta_{ui}^{[t]}])}.
\]

The function \( f(\cdot) \) in (5.49) could be the negative log-likelihood, which has to be minimized with respect to \( a_{ui}[t] \), i.e.,

\[
f(t) = -\ln p(u_t, a_{ui}[t]) = -\ln p(u_i[t]|a_{ui}[t - \Delta_{ui}^{[t]}]) - \ln p(a_{ui}[t]) \tag{5.51}
\]
The second term in (5.51) is rewritten as,

\[- \ln p(a_u[t]) = -\ln \mathcal{N}(a_u[t] \mid a_u[t - \Delta_{a_{u_i}}^{[i]}], c\Delta_{a_{u_i}}^{[i]}) \]

\[= -\ln \left\{ \frac{1}{\sqrt{2\pi c\Delta_{a_{u_i}}^{[i]}}} \exp \left\{ -\frac{1}{2c\Delta_{a_{u_i}}^{[i]}}(a_u[t] - a_u[t - \Delta_{a_{u_i}}^{[i]}])^2 \right\} \right\} \]

\[= \frac{1}{2} \ln(2\pi c\Delta_{a_{u_i}}^{[i]}) + \frac{1}{2c\Delta_{a_{u_i}}^{[i]}}(a_u[t] - a_u[t - \Delta_{a_{u_i}}^{[i]}])^2 \]

\[\propto \frac{1}{2c\Delta_{a_{u_i}}^{[i]}}(a_u[t] - a_u[t - \Delta_{a_{u_i}}^{[i]}])^2 \]  \hspace{1cm} (5.52)

The contribution of the second term in \( f' \) and \( f'' \) is

\[f''_1(a_u[t] - a_u[t - \Delta_{a_{u_i}}^{[i]}]) = \frac{1}{2c\Delta_{a_{u_i}}^{[i]}}(a_u[t] - a_u[t - \Delta_{a_{u_i}}^{[i]}])(-1)\]

\[= -\frac{1}{c\Delta_{a_{u_i}}^{[i]}}(a_u[t] - a_u[t - \Delta_{a_{u_i}}^{[i]}]) \]  \hspace{1cm} (5.53)

and

\[f''_2(a_u[t] - a_u[t - \Delta_{a_{u_i}}^{[i]}]) = \frac{1}{c\Delta_{a_{u_i}}^{[i]}} \]  \hspace{1cm} (5.54)

For the first term, we know [52]:

\[f_1() = -\ln p(\mathbf{u}_i[t] \mid a_u[t - \Delta_{a_{u_i}}^{[i]}]) \]

\[\leq -\mathbb{E}_{q(\theta)}\left[ \ln p(\mathbf{u}_i[t] \mid a_u[t - \Delta_{a_{u_i}}^{[i]}]) \right] + KL\{q(\theta) \parallel p(\theta, M)\} \]  \hspace{1cm} (5.55)

The inequality becomes equality if \( q(\theta) = p(\theta \mid D, M) \). That is the posterior distribution of the parameters given the data. Here, \( q(\theta) \) is the posterior distribution of \( \mathbf{u}_i[t] \).

Let us elaborate the term

\[-\mathbb{E}_{q(\theta)}\left[ \ln p(\mathbf{u}_i[t] \mid a_u[t - \Delta_{a_{u_i}}^{[i]}]) \right] = \]

\[-\mathbb{E}_q\left[ \ln \left\{ \frac{1}{\sqrt{\det(2\pi \Sigma_{a_{u_i}})}} \exp\left\{ -\frac{1}{2}(\mathbf{u}_i[t] - \mathbf{\mu}^\prime_{a_{u_i}}[t - \Delta_{a_{u_i}}^{[i]}])^T \Sigma_{a_{u_i}}^{-1}(\mathbf{u}_i[t] - \mathbf{\mu}^\prime_{a_{u_i}}[t - \Delta_{a_{u_i}}^{[i]}]) \right\} \right\} \right] \]  \hspace{1cm} (5.56)

where \( \Sigma_{a_{u_i}}[t] \approx \Sigma_{a_{u_i}}^\prime[t - \Delta_{a_{u_i}}^{[i]}] + e^{a_{u_i}[t - \Delta_{a_{u_i}}^{[i]}]} \Delta_{a_{u_i}}^{[i]} I \) is the prior covariance matrix at \( t \) and \( \mathbf{\mu}^\prime_{a_{u_i}}[t - \Delta_{a_{u_i}}^{[i]}] \) is the prior mean vector.
Then, (5.56) is explicitly written as

\[ f(t) = \frac{1}{2} \int \ln \det(2\pi \Sigma_{u_t}) \frac{1}{\sqrt{\det(2\pi \Sigma_{u_t}')}}. \]

\[ \exp \left\{ -\frac{1}{2} (u_t - \mu_{u_t})^T (\Sigma_{u_t}')^{-1} (u_t - \mu_{u_t}) \right\} du_t + \]

\[ \frac{1}{2} \int (u_t - \mu_{u_t} (t - \Delta_{u_t}'))^T \Sigma_{u_t}^{-1} (u_t - \mu_{u_t} (t - \Delta_{u_t}')). \]

\[ \frac{1}{\sqrt{\det(2\pi \Sigma_{u_t}')}} \exp \left\{ -\frac{1}{2} (u_t - \mu_{u_t} (t))^T (\Sigma_{u_t}')^{-1} (u_t - \mu_{u_t} (t)) \right\} du_t. \]

(5.57)

The first term in (5.57) is simply \( \frac{1}{2} \ln \det(2\pi \Sigma_{u_t}) \). Next, we elaborate the second term in (5.57)

\[ \frac{1}{2} \int (u_t - \mu_{u_t} (t - \Delta_{u_t}'))^T \Sigma_{u_t}^{-1} (u_t - \mu_{u_t} (t - \Delta_{u_t}')). \]

\[ \frac{1}{\sqrt{\det(2\pi \Sigma_{u_t}')}} \exp \left\{ -\frac{1}{2} (u_t - \mu_{u_t} (t))^T (\Sigma_{u_t}')^{-1} (u_t - \mu_{u_t} (t)) \right\} du_t. \]

(5.58)

Let

\[ z = (\Sigma_{u_t}')^{-1/2} (u_t - \mu_{u_t} (t)) \leftrightarrow \]

\[ (\Sigma_{u_t}')^{-1/2} z = u_t - \mu_{u_t} (t) \leftrightarrow \]

\[ u_t = \mu_{u_t} (t) + (\Sigma_{u_t}')^{-1/2} z. \]

(5.59)

Then

\[ dz = \det \left[ (\Sigma_{u_t}')^{-1/2} \right] du_t \leftrightarrow \]

\[ du_t = \det \left[ (\Sigma_{u_t}')^{1/2} \right] dz \leftrightarrow \]

\[ = \sqrt{\det(\Sigma_{u_t})} dz. \]

(5.60)

By replacing (5.59) and (5.60) into (5.58), we obtain

\[ \int (\mu_{u_t} (t) - \mu_{u_t} (t - \Delta_{u_t}')) + (\Sigma_{u_t}')^{1/2} z)^T \Sigma_{u_t}^{-1}. \]

(5.61)
The integrand of (5.61) is composed of four terms:

$$
\begin{align*}
& (\mu'_{u_i}[t] - \mu'_{u_i}[t - \Delta_u^{[l]}])^T \Sigma_{u_i}^{-1}(\mu'_{u_i}[t] - \mu'_{u_i}[t - \Delta_u^{[l]}]) \\
& + z^T (\Sigma'_{u_i})^{1/2} \Sigma_{u_i}^{-1}(\mu'_{u_i}[t] - \mu'_{u_i}[t - \Delta_u^{[l]}]) \Sigma_{u_i}^{-1} (\Sigma'_{u_i})^{1/2} z \\
& + (\mu'_{u_i}[t] - \mu'_{u_i}[t - \Delta_u^{[l]}])^T \Sigma_{u_i}^{-1}(\mu'_{u_i}[t] - \mu'_{u_i}[t - \Delta_u^{[l]}]) \\
& + z^T (\Sigma'_{u_i})^{1/2} \Sigma_{u_i}^{-1}(\Sigma'_{u_i})^{1/2} z. (5.62)
\end{align*}
$$

It can be shown that:

$$
\begin{align*}
& \int (\mu'_{u_i}[t] - \mu'_{u_i}[t - \Delta_u^{[l]}])^T \Sigma_{u_i}^{-1}(\mu'_{u_i}[t] - \mu'_{u_i}[t - \Delta_u^{[l]}]) \frac{1}{\sqrt{\det(2\pi I)}} \exp \left\{- \frac{1}{2} z^T z \right\} dz \\
= & \text{tr} (\Sigma_{u_i}^{-1}(\mu'_{u_i}[t] - \mu'_{u_i}[t - \Delta_u^{[l]}])(\mu'_{u_i}[t] - \mu'_{u_i}[t - \Delta_u^{[l]}])^T) (5.63)
\end{align*}
$$

and

$$
\begin{align*}
& \int z^T (\Sigma'_{u_i})^{1/2} \Sigma_{u_i}^{-1}(\Sigma'_{u_i})^{1/2} z \frac{1}{\sqrt{\det(2\pi I)}} \exp \left\{- \frac{1}{2} z^T z \right\} dz = \\
& \text{tr} ((\Sigma'_{u_i})^{-1} \Sigma_{u_i}) (5.64)
\end{align*}
$$

while integrating the second and the third term, the outcome is zero.

To sum up,

$$
\begin{align*}
f_i() &= \frac{1}{2} \ln \det(2\pi \Sigma_{u_i}) \\
& + \frac{1}{2} \text{tr} (\Sigma_{u_i}^{-1}(\mu'_{u_i}[t] - \mu'_{u_i}[t - \Delta_u^{[l]}])(\mu'_{u_i}[t] - \mu'_{u_i}[t - \Delta_u^{[l]}])^T) \\
& + \frac{1}{2} \text{tr} ((\Sigma'_{u_i})^{-1} \Sigma_{u_i})(5.65)
\end{align*}
$$

In the following, we need to identify which terms of (5.65) depend on $a_{u_i}[t - \Delta_u^{[l]}]$. Clearly, one such term is

$$
\Sigma_{u_i}[t] = \Sigma'_{u_i}[t - \Delta_u^{[l]}] + e^{a_{u_i}[t - \Delta_u^{[l]}]} \Delta_u^{[l]} I (5.66)
$$

Let $\Sigma'_{u_i}[t - \Delta_u^{[l]}] = Q \Lambda Q^T$, where $\Lambda = \text{diag}(\lambda_d)$ be the eigendecomposition of the posterior covariance matrix at the previous time instant. Then,

$$
\Sigma_{u_i} = Q \tilde{\Lambda} Q^T (5.67)
$$

where $\tilde{\Lambda} = \text{diag}(\tilde{\lambda}_d)$ with

$$
\tilde{\lambda}_d = (\lambda_d + e^{a_{u_i}[t]} \Delta_u^{[l]}), (5.68)
$$
where \( \lambda_d \) represents the \( d \)th eigenvalue of the posterior covariance matrix.

Another term is:

\[
(\Sigma'_{u_i})^{-1} = \Sigma^{-1}_{u_i} + \frac{(\mu'_{u_i} - \mu_{u_i}) (\mu'_{u_i} - \mu_{u_i})^T}{\sigma^2} \chi
\]

\[
= Q \hat{\Lambda}^{-1} Q^T + \chi
\]

where only the first term is of interest since the second term is a constant matrix with respect to the quantity of interest. It can be shown that

\[
\frac{\partial}{\partial a_{u_i}[t - \Delta^u_{u_i}]} \log \det(2\pi \Sigma_{u_i}) = \sum_d \frac{e^{a_{u_i}[t - \Delta^u_{u_i}] \Delta^u_{u_i}}}{\lambda_d + e^{a_{u_i}[t - \Delta^u_{u_i}] \Delta^u_{u_i}}}
\]

\[
= \sum_d \eta_d
\]

(5.70)

where

\[
\eta_d = \frac{e^{a_{u_i}[t - \Delta^u_{u_i}] \Delta^u_{u_i}}}{\lambda_d + e^{a_{u_i}[t - \Delta^u_{u_i}] \Delta^u_{u_i}}}
\]

(5.71)

For the second term of (5.65) we obtain

\[
\frac{\partial}{\partial a_{u_i}[t - \Delta^u_{u_i}]} \text{tr} \left( (\Sigma^{-1}_{u_i})^{-1} (\mu'_{u_i}[t] - \mu_{u_i}[t - \Delta^u_{u_i}]) (\mu'_{u_i}[t] - \mu_{u_i}[t - \Delta^u_{u_i}])^T \right) =
\]

\[
\frac{\partial}{\partial a_{u_i}[t - \Delta^u_{u_i}]} v^T \hat{\Lambda}^{-1} v =
\]

\[
\frac{\partial}{\partial a_{u_i}[t - \Delta^u_{u_i}]} \sum_d v_d^2 \frac{1}{\lambda_d + e^{a_{u_i}[t - \Delta^u_{u_i}] \Delta^u_{u_i}}}
\]

(5.72)

where

\[
v = Q^T (\mu'_{u_i}[t] - \mu_{u_i}[t - \Delta^u_{u_i}])
\]

(5.73)

and \( v_d \) denotes the \( d \)th element of \( v \). A careful examination of (5.73) reveals that

\[
v \approx Q^T (\Sigma'_{u_i} \Sigma^{-1}_{u_i} - I) \mu'_{u_i}[t - \Delta^u_{u_i}]
\]

(5.74)

where irrelevant terms were omitted. Therefore, (5.74) can be rewritten as

\[
v \approx (Q^T \Sigma'_{u_i} Q \hat{\Lambda}^{-1} Q^T - Q^T) \mu'_{u_i}[t - \Delta^u_{u_i}]
\]

\[
= (M \Lambda^{-1} - I) Q^T \mu'_{u_i}[t - \Delta^u_{u_i}]
\]

\[
\phi
\]

(5.75)

where

\[
\phi = (M \Lambda^{-1} - I) \phi.
\]
Accordingly the $d$-th element of $v$ is given by
\[
v_d \approx [(M\Lambda^{-1} - 1)\phi_d] = \left(\frac{M_{dd}}{\lambda_d} - 1\right)\phi_d + \sum_{l \neq d} \frac{M_{dl}\phi_l}{\lambda_l}\]
(5.76)

which also depends on $a_u[t - \Delta u]$. Returning back to (5.72)
\[
\frac{\partial}{\partial a_u[t - \Delta u]} \left(\sum_d v_d^2 \frac{\phi_d}{\lambda_d} + e^{a_u[t - \Delta u]} \Delta u \phi_d\right) = \sum_d \frac{\partial}{\partial a_u[t - \Delta u]} \left(\frac{v_d^2}{\lambda_d}\right) = \sum_d v_d \left(\frac{2 \frac{\partial}{\partial a_u[t - \Delta u]} \phi_d}{\lambda_d} - \frac{v_d \eta_d}{\lambda_d}\right)
(5.77)
\]

where $\eta_d$ is given by (5.71). Using (5.76), it can be shown that
\[
\frac{\partial v_d}{\partial a_u[t - \Delta u]} = -M_{dd} e^{a_u[t - \Delta u]} \Delta u \phi_d + \sum_{l \neq d} -M_{dl} \phi_l (e^{a_u[t - \Delta u]} \Delta u \phi_l) = - \left(\frac{M_{dd} \eta_d}{\lambda_d} + \sum_{l \neq d} \frac{M_{dl} \eta_l}{\lambda_l}\right) = - \sum_{l=1}^d M_{dl} \frac{\eta_l}{\lambda_l} \phi_l.
(5.78)
\]

By substituting (5.78) into (5.77) we arrive at
\[
\frac{\partial}{\partial a_u[t - \Delta u]} \left(\sum_d \frac{v_d^2}{\lambda_d}\right) = - \sum_d \frac{v_d}{\lambda_d} \left[2 \sum_l M_{dl} \frac{\eta_l}{\lambda_l} \phi_l + v_d \eta_d\right].
(5.79)
\]

The differentiation of the third term of (5.65) gives
\[
\frac{\partial}{\partial a_u[t - \Delta u]} \text{tr}[(\Sigma_u')^{-1} \Sigma_u] = \frac{\partial}{\partial a_u[t - \Delta u]} \text{tr}[(\Sigma_u^{-1} + \mathcal{X}) \Sigma_u] \approx \frac{\partial}{\partial a_u[t - \Delta u]} \text{tr}(\mathcal{I}) = 0
(5.80)
\]

where $\mathcal{X}$ is a matrix whose elements do not depend on $a_u[t - \Delta u]$. Therefore,
combining (5.65), (5.72), and (5.79), we obtain

\[
\frac{\partial}{\partial a_{ui}[t - \Delta_{ui}^t]} f_1() = \frac{1}{2} \left( \sum_d \eta_d - \sum_d \frac{v_d}{\lambda_d} \left[ 2 \sum_{l=1}^d M_{dl} \eta_l \phi_l + v_d \eta_d \right] \right)
\]

\[
= \frac{1}{2} \sum_d \left( \eta_d - \frac{v_d}{\lambda_d} v_d \eta_d \right) - \sum_d \frac{v_d}{\lambda_d} \sum_{l=1}^d M_{dl} \eta_l \phi_l.
\]

(5.81)

By combining (5.53) and (5.81), we obtain

\[
f' = \frac{\partial f}{\partial a_{ui}[t - \Delta_{ui}^t]} =
\]

\[
- \frac{1}{c \Delta_{ui}^t} (a_{ui}[t] - a_{ui}[t - \Delta_{ui}^t]) + \frac{1}{2} \sum_d \eta_d (1 - \eta_d) - \frac{1}{2} \sum_d \frac{v_d^2}{\lambda_d} \eta_d (1 - 2 \eta_d) - \sum_d \frac{v_d}{\lambda_d} \sum_{l=1}^d M_{dl} \eta_l \phi_l.
\]

(5.82)

Accordingly the second derivative is

\[
f'' = \frac{\partial f'}{\partial a_{ui}[t - \Delta_{ui}^t]} =
\]

\[
- \frac{1}{c \Delta_{ui}^t} + \frac{1}{2} \sum_d \eta_d (1 - \eta_d) - \frac{1}{2} \sum_d \frac{v_d^2}{\lambda_d} \eta_d (1 - 2 \eta_d)
\]

\[
+ \sum_d \eta_d \frac{v_d}{\lambda_d} \sum_{l=1}^d M_{dl} \phi_l \frac{\eta_l (1 - 2 \eta_l)}{\lambda_l}
\]

(5.83)

where

\[
\tilde{\lambda}_l = \lambda_l + e^{a_{ui}[r - \Delta_{ui}^t]} \Delta_{ui}^t
\]

(5.84)

### 5.4.1 Prediction of new data

Since this is a real-time prediction system, when a new observation \(y_{ij}[t]\) arises, \(q(y)\) is not required. This fact provides the flexibility in the updates of \(q(u)\) and \(q(w)\), because \(E_q[y_{ij}[t]]\) in (5.47) is replaced by the observed value \(y_{ij}[t]\). As mentioned before, in the Collaborative Kalman Filter, predictions rely heavily on posterior distributions calculations. This leads to an approximation of intractable integrals. A solution is provided by Monte Carlo methods [5]. Although, these methods are time consuming. To reduce time complexity, instead of utilizing samples from the distributions in order to
approximate expectations as in aforementioned methods, the prediction of a new entry can be accomplished by utilizing the means of prior distributions of $u_i$ and $w_j$. 
Chapter 6

Experimental results

6.1 Movie rating prediction

6.1.1 Data

Experiments were conducted on the Netflix data set, which consists of 100 million movie ratings spanning the period of 1999-2006. It is a five star movie rating system, thus, movies have ratings from 1 to 5. The set contains information about 17,770 movies and 480,000 users. The pre-processing of data was conducted on Python and C++, while inference was conducted on MATLAB. The dataset consists of 17,770 text files. Each one represents a movie and contains the user IDs that correspond to users who rated the particular movie, along with the rating and their respective date. The processed dataset contains all information provided in a single file in the form of quadruplets consisting of comma-separated values as

\{movieID, userID, rating, date\}.

All entries were sorted by date and organized into structured arrays. This is the time-series to be analysed. A unique ID was assigned to each user, for the sake of compatibility with the structures, through hashmap implementation (i.e., containers in MATLAB framework). Two separate structured arrays were constructed, one for user vectors and one for movie vectors. In order to avoid memory overflow, a batch algorithm was developed. Each batch consisted of 5,000 entries. Therefore, inference was applied on each batch and thereupon the memory was released.
6.1.2 Initialization

The latent vectors size was set to $d = 10$. The extra drift parameter $c$ was set to 0 ($c = 0$), so that the users have a shared drift parameter $a_u$. Movies share a drift parameter $a_w$ as well. The standard deviation was set to $\sigma = 1.76$, which defines the width of each of five classes which correspond to each rating star, yielding equal length sections. The mean of the priors of each user $\mathbf{u}_i$ and movie vector $\mathbf{w}_j$, was randomly initialized, such that their inner product $\mathbf{u}_i^T\mathbf{w}_j$ yields a value between 1 and 5. The time was initialized at the first date appearing in the data set. The covariance matrices were set equal to the identity matrix.

6.1.3 Evaluation

The evaluation of the ratings was conducted via root-mean-square error (RMSE) measure:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{k}(\hat{y}_i - y_i)^2}{k}}$$  \hspace{1cm} (6.1)

where $\hat{y}_i, y_i, k$, denote the predicted rating, the real rating, and the number of instances (e.g., trials), respectively. Comparison against state-of-the-art algorithms was undertaken.

1. Online VB-EM
2. Probabilistic Matrix Factorization (PMF) [14]
3. Bayesian Probabilistic Matrix Factorization (BPMF) [34]

Table 6.1: Performance comparison with respect to RMSE.

<table>
<thead>
<tr>
<th>Method</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMF</td>
<td>1.0204</td>
</tr>
<tr>
<td>BPMF</td>
<td>1.0182</td>
</tr>
<tr>
<td>Online VB-EM</td>
<td>0.8740</td>
</tr>
<tr>
<td>CKF</td>
<td>0.7212</td>
</tr>
</tbody>
</table>

The calculation of RMSE in dynamic and in static models is different. In dynamic models, we do not use a test set, but we conduct predictions for every rating in the data set before having access to particular rating in order to update the model. In static models, a classic evaluation is applied, through the usage of test sets. As we see in Table 6.1, CKF performed best. The ability of the model to capture dynamic information of user preferences led to more consistent results than static models.


6.2 Stock price prediction

6.2.1 Data

For stock price prediction, opening values from AMEX, NASDAQ, and NYSE exchange spanning the period 1962-2017, were used. Data were gathered from Yahoo Finance through the development and application of a Python parser, which was also applied to process properly the stock data. The processed stock data were gathered in a text file in triplets consisting of comma-separated values as

\{stock\_name, date, opening\_value\}

All entries were organized and ordered in the same way as in movie rating prediction. In this procedure, user-item pairs do not exist, but instead \(y_{ij}[t]\) is treated as a continuous value (i.e., stock price). Accordingly, the concept of partitions, needed in Netflix is not applicable and there is only one state global vector \(w_{j=1}\).

6.2.2 Initialization

The latent state vector size was set to \(d = 5\). Each stock has its own drift Brownian motion \(a_u[t]\), which is learned by the model. Initially, \(a\) was drawn randomly from a normal distribution, with mean \(-6\) and standard deviation \(5 \times 10^{-2}\). At each time step, the inferred drift parameter \(a_u[t]\) is appended in a list, in order to become accessible in a next time step. The extra drift parameter for stocks \(c_u\) was set to \(c_u = 5 \times 10^{-2}\). The extra drift for the global state vector \(w\) was set to 0 (i.e., \(c_w = 0\)). This led the model to learn a converging value of \(a_w[t] \rightarrow -11.7\) \([5]\). Standard deviation was set at \(\sigma = 0.01\), a small value, because stock prices exhibit strong volatility through time and a close tracking is demanded.

6.2.3 Evaluation

In the following figures, there is a comparison between the real opening stock prices, the predicted ones and the drift parameter of each one of them. Experimental results on BP, Coca-Cola and Pfizer stock prices are shown below,
Figure 6.1: BP oil company real stock prices along with predicted prices and the corresponding inferred drift parameter.
As can be seen, Figure 6.1a illustrates the given historical stock prices of BP oil company. The price is partially volatile, mainly in the starting period, the middle period, and at the end. Intense fluctuations occur in early 1980s and early 1990s, where the price increases rapidly achieving the maximum price of 147.125$. In 1979, stock price falling is related to the Iranian revolution. The early 1990s recession was triggered by the Iraqi invasion of Kuwait. In early 2000s, a global recession emerged, due to the increase of oil price [53]. In Figure 6.1b the respective predicted prices are depicted. This plot is approximately identical to 6.1a. This fact demonstrates the high performance of the algorithm. At each time step the predicted opening value is very close to the real opening stock value. Figure 6.1c shows the learned Brownian motion drift parameter of BP stock. The drift parameter captures the volatility of the stock price along time. Specifically the rate of change around peaks.
Figure 6.2: Coca-Cola beverage company historical stock prices along with predicted prices and the corresponding inferred drift parameter.
In Figure 6.2a, we see the given historical stock prices of Coca-Cola. The stock price exhibits high volatility over the whole time domain. A clear pattern cannot be inferred. The price takes its highest value in late 1960s, which is $155.75. In early 1970s, the price decreases dramatically, ending up with the minimum value of $28.875 in the mid-1970s. This is the result of the competition with PepsiCo, which at that time exceeded Coca-Cola sales by far. In mid-1970s, PepsiCo conducted an experiment, in which potential consumers experienced a blind taste test of Coca-Cola and PepsiCo [54]. The results showed that consumers in their majority, preferred the taste of PepsiCo. These results were published, and that had a negative impact on Coca-Cola sales. In order to gain back popularity, in April 1985 Coca-Cola introduced the “New Coke”, promoting a new concept, including the name and the flavour. Consumer’s reacted negatively to this change. This led to the falling of Coca-Cola stock price, as can be seen in Figure 6.2a. Figure 6.2b shows the predicted prices, which again follow consistently the real historical prices. In Figure 6.2c the respective learned Brownian motion drift parameter is illustrated. The drift parameter track closely the motion of Coca-Cola stock price. We can see that the high volatility, the peaks, and the falls, are equivalent to those of the historical data depicted in Figure 6.2a.
Figure 6.3: Pfizer pharmaceutical company historical stock prices along with predicted prices and the corresponding inferred drift parameter.
Figure 6.3a represents the historical prices of Pfizer pharmaceutical company. The variance increases in the first half of the time domain. The maximum value of 149.187$ is reached in 1997, when the company introduced new groundbraking medication pills, i.e, “Sildenafil”, “Lipitor”, etc. Pfizer’s stock price in that year, increased by 150% [55]. In early 2000s the price decreases. From then and on, the stock price rate of change is apparently lower in comparison to the first half of its overall period. We notice in Figure 6.3b, the prediction performance, which is again sufficiently highly accurate. Once again, the inferred drift parameter of Pfizer stock, especially after mid-1970s track real prices adequately. The high fluctuations around the peaks and the falls are being captured consistently. The stock price prediction accuracy is summarized in Table 6.2.

<table>
<thead>
<tr>
<th>Stock</th>
<th>RMSE</th>
<th>Price range(USD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coca-Cola</td>
<td>3.5529</td>
<td>28.875-155.75</td>
</tr>
<tr>
<td>BP</td>
<td>8.3794</td>
<td>27.25-147.125</td>
</tr>
<tr>
<td>Pfizer</td>
<td>4.176</td>
<td>11.84-149.187</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>5.5004</td>
<td>11.84-155.75</td>
</tr>
</tbody>
</table>

Table 6.2: Stock prices prediction accuracy.

Table 6.2 reports the prediction results measured by RMSE metric. RMSE is low with respect to the price range of the stocks. This proves that an accurate stock price prediction method has been developed. The latent state space, consisting of learned latent vectors $u_i$ and $w_j$, efficiently represents the information of the given data. The learned Brownian motion drift parameter reflects proportional changes in prices. An important factor of accurate predictions, is the dynamic nature of the drift parameter, which allows latent vectors to move in latent state space and capture the dynamic changes of stock prices.
Chapter 7
Conclusion

In this thesis a prediction method is developed based on collaborative Kalman filter for time-evolving dyadic processes, i.e., movie rating prediction and stock price prediction. The model expands matrix factorization techniques applied to collaborative filtering. Users and objects are represented as latent vectors in the same state space. Instead of having static locations in this latent space, they are considered to have a dynamic behaviour. The dynamic behaviour is being learned by a geometric multidimensional Brownian motion drift parameter. In stock price prediction, this parameter also evolves through time in order to capture stock price dynamic fluctuations. In order to learn the dynamically evolving distributions of latent state vectors, variational inference was applied. In movie rating prediction, experiments were conducted on the Netflix data set. The comparison against state-of-the-art algorithms indicates that CKF performed best, by achieving the lowest RMSE of 0.7212. In stock price prediction, stock data were obtained by AMEX, NASDAQ, and NYSE exchange. The results show that the learned time-evolving drift parameter allows latent state vectors to capture each stock’s changing volatility through time.

Our future work will concentrate on developing dynamic models for time-series prediction, based on Particle filters. Extensions of Kalman filter will be applied in order to achieve even more accurate dynamical estimations.
Bibliography


[11] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and
John Riedl. GroupLens: an open architecture for collaborative filtering

[12] Upendra Shardanand and Pattie Maes. Social information filtering: al-
gorithms for automating word of mouth. In Proc. SIGCHI Conf. Human
Factors in Computing Systems, pages 210–217. ACM Press/Addison-

[13] Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. Recom-

[14] Andriy Mnih and Ruslan R. Salakhutdinov. Probabilistic matrix factor-
ization. In Advances in Neural Information Processing Systems, pages

[15] Huseyin Polat and Wenliang Du. SVD-based collaborative filtering with

[16] Zhi-Dan Zhao and Ming-Sheng Shang. User-based collaborative-
filtering recommendation algorithms on hadoop. In Proc. 3rd Int.
Conf. WKDD’10 Knowledge Discovery and Data Mining, pages 478–

[17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-
Steering Committee, 2017.

[18] Shunmei Meng, Wanchun Dou, Xuyun Zhang, and Jinjun Chen. Kasr:
A keyword-aware service recommendation method on mapreduce for big
data applications. IEEE Transactions on Parallel and Distributed Sys-

[19] Li Chen and Feng Wang. Preference-based clustering reviews for
augmenting e-commerce recommendation. Knowledge-Based Systems,


