ABSTRACT

YANG, RUIXIN. Algorithms for Data-Driven Business Intelligence with Applications to Revenue Optimization and Financial Forecasting. (Under the direction of Dr. Nagiza F. Samatova).

Business Intelligence (BI) has emerged as an increasingly important area within the support of data ecosystem and machine intelligence. BI systems, widely developed by researchers from both academia and industry, are now used extensively in numerous domains, including commerce, management, smart health, security, and financial market. In particular, algorithms for BI have gained substantial interest due to their abilities to discover new patterns and provide actionable insights. Thus, such algorithms have been adopted by many business applications such as recommendation systems in E-Commerce, predictive modeling in information systems, and marketing analysis. However, in a rapidly changing business and financial environment, traditional algorithms, such as frequent itemset mining, association rule mining, support vector machines, and autoregressive-moving average models could no longer meet the modern demands of different business goals. Therefore, in this work, we focus on developing more effective data-driven BI, especially for E-Commerce and financial markets analysis. We use revenue optimization and financial forecasting as exemplar applications and anticipate our newly developed algorithms can be applied to a variety of BI tasks.

In this dissertation, we first propose algorithms for revenue optimization in real time utility-based recommendation. Recommender Systems (RS) in E-commerce are typically used to suggest products to online shopping customers, and now play a key role in product marketing strategies for major online retailers, such as Walmart Inc. and Amazon.com, Inc. The main goal of such systems is to predict likely future customer desires and to trigger

purchases through the timely provision of product recommendations. Therefore, RS have become indispensable tools for both customers and retailers. However, most existing RS recommend products from customers' point of view (i.e. likelihood of purchase by customers) but ignore one of the most important business goals: the optimization of revenue. Consequently, there is an increasing need to learn utility patterns online and provide near real-time utility-based recommendations. To address these challenges, we first define the utility of recommendation sets and formulate the problem of real time utility-based recommendation. We observe that online transaction streams are usually accompanied with flow fluctuation, and based on that observation, we propose an Adaptive Online Top-K (RAOTK) high utility itemsets mining model to guide the utility-based recommendations. Additionally, three variants of this algorithm are described and we provide a structural comparison of the four algorithms with discussions on their advantages and limitations. Moreover, to make our model more personalized, we also take the buying power of customers into account and propose a simple but effective method to estimate the consumers' willingness to pay. Finally, extensive empirical results on real-world datasets show that the proposed methodology works effectively and outperforms several state-of-the-art models by average 20 percent.

In our second component, inspired by the concept of utility, we develop an intelligent weighted fuzzy time series model for financial markets forecasting. It is well-known that financial forecasting is an extremely challenging task given the complex, nonlinear nature of financial market systems. To address this challenge, we present an intelligent weighted fuzzy time series model, which builds upon our sine-cosine adaptive human learning optimization (SCHLO) algorithm to search for optimal parameters for forecasting. New weighted operators that consider frequency based chronological order and stock volume are analyzed, and SCHLO is integrated to determine the effective intervals and weighting factors.

Furthermore, a novel short-term trend repair operation is developed to complement the final forecasting process. Finally, the proposed model is applied to four world major trading markets: the Dow Jones Index (DJI), the German Stock Index (DAX), the Japanese Stock Index (NIKKEI), and Taiwan Stock Index (TAIEX). Experimental results show that our model is consistently more accurate than the state-of-the-art models. The easy implementation and effective forecasting performance suggest that our proposed model could be a favorable market application prospect.

Finally, to further improve forecasting performance, we focus on combining Empirical Mode Decomposition (EMD) with the weighted fuzzy time series model. Although financial indices can be observed as single time series, they are actually an aggregation of many sub-series generated by different institutions or individuals. Thus, creating forecasting models based only on a single observed series may encounter bottlenecks. Therefore, we present a new Intelligent Hybrid Weighted Fuzzy (IHWF) time series model based on EMD. EMD is a time series processing technique to extract the possible modes of various kinds of institutional and individual investors and traders, embedded in a given time series. Subsequently, the proposed weighted fuzzy time series method, incorporates chronological order-based frequency and Neighborhood Volatility Direction (NVD) components, is analyzed and integrated with Adaptive Sine-Cosine Human Learning Optimization (ASCHLO) to determine the optimal universe discourse, intervals and weights. Finally, we evaluate our model on actual 15 years trading data from TAIEX, and compare our findings to the well-known forecasting models. The results show that the proposed method outperforms the compared models in terms of one-day-ahead prediction accuracy.

Algorithms for Data-Driven Business Intelligence with Applications to Revenue Optimization and Financial Forecasting

by
Ruixin Yang

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Science

Raleigh, North Carolina

2019

APPROVED BY:

Dr. Dennis R. Bahler	Dr. Kemafor Anyanwu Ogan
Dr. Steffen Heber	Dr. Nagiza F. Samatova Chair of Advisory Committee

DEDICATION

To my parents and my wife, for their endless support, encouragement and love.

BIOGRAPHY

Ruixin Yang was born in Chengdu, China in 1988. He received his bachelor and master degrees in engineering from Shanghai University, China, in 2011 and 2014, respectively. In August 2014, he began his Ph.D. in Computer Science at North Carolina State University, under the guidance of Dr. Nagiza F. Samatova. His research interests include business intelligence, quantitative analysis, evolutionary algorithms and applied data science.

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the support, understanding, encouragement and guidance from many people in my graduate life.

First and foremost, I would like to express my sincere gratitude to my advisor Dr. Samatova, for the continuous support of my Ph.D. study and research, for her wisdom, enthusiasm, and academic guidance during every critical step I took toward becoming an independent researcher. Next, I would also like to thank my committee members for their valuable time and comments that improved the presentation and contents of this dissertation.

Additionally, I would like to thank the other graduate students of Dr. Samatova's research group: Mingyang, Xu, Paul Jones, Steve Harenberg, Stephen Ranshous, Changsung Moon and Shiou Tian Hsu, for their helpful discussions and commends of this dissertation and associated publications.

Finally, I would especially like to thank my family for their endless love and support over the years. My wife, Haoqi Ni has been extremely supportive of me throughout this entire process and finally help me get to this point. And I would express a deep sense of gratitude to my parents for their continued support and encouragement, without whom I would never have enjoyed so many opportunities.

This material is based upon work supported in whole or in part with funding from the Laboratory for Analytic Sciences (LAS). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the LAS and/or any agency or entity of the United States Government.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER

1

INTRODUCTION

1.1 Background

Business Intelligence (BI) has emerged as an increasingly important area within the support of data ecosystem and machine intelligence [4]. Base on a report from Stratistics MRC, the Global BI market is expected to grow from $15.64 billion in 2016 to reach $29.48 billion by 2022 with a CAGR of 11.1% [169]. According to an IBM Global CIO study, the collective voice of more than 2,500 chief information officers (CIOs) worldwide points to BI and analytics as the top visionary plan for enhancing their enterprises' competitiveness [84]. BI, traditionally as an umbrella term, has been used to describe a bunch of concepts and

methods to improve business decision making, ranging from standard reports to highly sophisticated advanced statistics and also includes databases, underlying architectures, tools and applications [112, 150, 199]. The major objectives of BI are to enable organizations or companies to become smarter, work smarter, and make better decisions for conducting appropriate analyses and performing actions [104, 180, 200]. BI systems, widely developed by researchers from both academia and industry, are now used extensively in numerous domains, including commerce, management, smart health, security and financial market [11, 43, 46, 62, 123, 129, 149, 162, 176].

In particular, data-driven method, an advanced knowledge discovery technique of BI, has gained substantial interest worldwide in the past decade, due to new patterns, discoveries and insights can be obtained from various sources of business data. Thus, the data-driven method has been adopted by many business applications such as recommendation in E-Commerce, predictive modeling in information systems and marketing analysis. Cnudde et al. [50] increased the user loyalty using spatial behavioral user data. Lee et al. [105] adopted a multi-agent-based system for the profitable customers segmentation. Abbasi [1] presented a meta-learning framework to enhance financial fraud detection. Hahn et al. [74] provided an integrated data models on applications of in-memory analytics in the field of supply chain management. Huang et al. [80] mined interesting resource allocation rules from event log to support business process management. Wu [201] Applied frequent itemset mining to identify a small itemset that satisfies a large percentage of orders in a warehouse. Ge et al. [63] designed a taxi business intelligence system using route recommendation. Shambour et al. [161] proposed a trust-semantic fusion-based recommendation approach for e-business application. Ballings et al. [17] presented an ensemble of deep belief network combined with support vector regression for time series forecasting. Wang et al. [186] forecasted stock market indexes using principle component

analysis and stochastic time neural networks. However, in a rapidly changing business and financial environment, traditional data-driven methods, such as frequent itemset mining [25], association rule mining [125], support vector machine [171], regression model of a time series [210], autoregressive moving average model [156] could no longer meet the modern demands of different business goals [22, 31, 33, 77, 111, 155, 164, 176, 204, 206]. Therefore, in this work, we focus on developing more efficient data-driven methods for BI, especially in terms of E-Commerce and financial market analysis. We use revenue optimization and marketing forecasting as exemplar applications but believe our newly developed algorithms can be broadly applied to various tasks.

1.2 Related Work

1.2.1 Recommender Systems for E-commerce

Recent years have witnessed the unparalleled success of ecommerce. For example, Amazon, the online retailer giant, has become the fifth biggest U.S.-listed company with $350 billion market cap [98]. Recommender Systems (RS) have played a vital role and will continue to deeply influence e-commerce. They enable customers to quickly discover potential purchases without wasting their time with many related searches, and they allow retailers to boost their profits. With this increasing importance for both customers and retailers, RS have attracted considerable attention from both industry and academia. Industries are using these systems in new ways [109], and academics are researching how to optimize them even further. A recommender system usually includes three steps: 1). acquiring preference from customers' input data; 2). generating the recommendation using proper techniques; 3). finally presenting the recommendation results to customers [196]. Current

approaches include collaborative filtering (CF) [51], content-based filtering (CBF) [20], and hybrid approaches [119]. These techniques usually ingest product ratings received from customers and then suggest new products of potential interest [3]. Collaborative filtering, which provides recommendations by the levels of similarity of other consumers' preferences, is one of the most widely used algorithms in recommender systems [3, 53, 158]. This technique helps the consumer make a better and easier decision and also can choose a product based on the consumers' preference from a large pool of candidates. More recently, Larrain et al. [103] made an extensive evaluation and comparison of time-aware recommendation methods, addressing the problems of when to use the temporal information and how to use weight the similarity between users and items by exploring the effect of different time-decay functions. Loni et al. [120] demonstrated that Factorization Machines can exploit additional representations of information inherent in the user-item matrix to improve recommendation performance. Chen et al. [34] proposed a heterogeneous evolutionary clustering to gather users with similar interest into the same cluster based on CF. Polato and Aiolli [148] used a boolean kernel to alleviate the sparsity issue in CF contexts. Zhou et al. [217] introduced bias and confidence weights into CF and to further improve the stability and accuracy of online CF. Wang et al. [194] used a hybrid user similarity model for CF to evaluate the user similarity comprehensively and objectively. Kant et al. [88] investigated a centroid selection approach for k-means clustering algorithm to solve the sparsity problem of CF. Karabadji et al. [90] improved the CF approach by using an evolutionary multi-objective optimization-based method to pull up a group of profiles that maximizes both similarity with the active user and diversity between its members which gained the performances in terms of both accuracy and diversity. Bellogin [21] proposed a new perspective to analyze the user behavior by finding other users who have similar sequential patterns instead of focusing only on similar ratings in the items. Another

widely used approach is content-based filtering (CBF) which exploits suitable items to users based on the descriptions of items and user preferences [165]. A number of recent studies in literature investigated CBF [24, 52, 89, 135, 146, 165, 203, 205]. Since CBF algorithms employ profile information (such as purchases, queries and ratings) of active users, thus can generate accurate recommendations even in the situation where CF systems encounter new item and sparsity problems [3, 28, 211]. And therefore, many researchers [8, 9, 15, 18, 70, 92, 138, 144, 157] have chosen hybrid ways to achieve better performance.

1.2.2 High Utility Pattern Mining

In many real-world data-driven applications, data mining techniques are first adopted to extract interesting and useful patterns from databases for supporting better decision making. During past two decades, two fundamental tasks, frequent itemset mining (FIM) [5, 25, 68, 76] and association rule mining (ARM) [6, 127], have received significant attention for revealing these relationships. The goal of frequent itemset mining is to find items that co-occur in a database above a given frequency threshold, without considering the quantity or weight such as profit of the items [56]. For example, in market analysis, frequent itemset mining from a transaction database refers to the discovery of the itemsets which frequently appear together in the transactions [177]. And association rules are mined in two phases. In the first phase, itemsets that frequently co-occur in transactions are mined. Subsequently, in the second phase, rules are discovered from the generated frequent itemsets, considering whether an item appeared in a transaction or not [99]. Although traditional ARM has been successfully used in several businesses and scientific applications [76, 114, 136, 145], it can only reveal relationships, impacted by the factor of frequency (presence or absence), between items in databases. In other words, it does not consider other factors for evaluating

patterns such as the price, utility, importance, weight, or cost of items or itemsets that are less frequent but more profitable. In reality, most business applications often expect more discovery in specification of item attributes (e.g. demands, profits and margins) to get interesting patterns to plan ahead. For example, a retail business may be interested in identifying its most valuable products (products who contribute significant profit to the company). These are the items, may be sold in a full priced or high margin, but absent from many transactions because not every customer would buy these items. Therefore, it is not enough to discover more attractive patterns for decision maker only considering items' purchase quantities or occurrence frequencies [58]. To address this limitation of traditional ARM and FIM techniques, high-utility itemset mining (HUIM) [32] was proposed. Basically, utility is a measure of how "useful" (i. e., "profitable", "interested" or "import") an item is. The utility of an item is based on two factors, i.e., local utility and external utility. The local utility refers to importance of the items in the database, like the quantity of the item sold in the transaction and the external utility means the importance of distinct item, like a profit table [117, 177]. An item/itemset is called a high utility item/itemset if its utility is larger than or equal to a minimum utility threshold user specified; otherwise, it is called a low utility itemset. HUIM is an important task since it is useful to discover highly profitable and interesting patterns and then more efficient and strategic business decisions can be adapted to increase sales and profit. Thus, HUIM is essential to a wide range of applications such as retail chain-stores [118], website click streaming analysis [116] and biomedical applications [178]. Recently, many HUIM algorithms have been proposed to gain a better performance. Liu et al. [118] introduced the concept of transaction weighted utility (TWU) and mines high utility itemsets in a two-phase algorithm. Yao and Hamilton [101] identified several mathematical properties of utility constraints and utilized two pruning strategies, utility upper bound and support upper bound, for HUIM. Lan et al. [101]

proposed a level-wise mining approach to find efficiently high utility itemsets in databases and then they adopted an indexing mechanism to speed up the execution and reduce the memory requirement in the mining process [102]. Ahmed et al. [7] presented a tree-based candidate pruning technique, HUC-Prune, to capture important utility information of the candidate patterns which can avoid the level-wise candidate generation process by adopting a pattern growth approach. Tseng et al. [178] proposed two algorithms, UP-Growth and UP-Growth+, where high utility itemsets is maintained in a tree-based data structure named utility pattern tree such that candidate itemsets can be generated efficiently with only two scans of database. Liu and Qu [116] introduced a new data structure called utility lists to store both the utility information about an itemset and the heuristic information for pruning the search space. Liu et al. [115] proposed a high utility itemset growth approach that works in a single phase without generating candidates. The algorithm enumerates itemsets by prefix extensions, prunes search space by utility upper bounding, and maintains original utility information in the mining process by a novel data structure. Zida et al. [218] introduced an array-based utility counting technique and utilized transaction merging and database projection techniques to efficiently mine HUIs.

1.2.3 Intelligent Optimization Algorithm

Recently, with the intensive research in engineering, finance, medicine and other fields, optimization problems have become more and more complex and challenging [75]. Thus, methods for solving these problems have been a widely researched topic [134, 160]. The traditional optimization methods are mainly based on gradient algorithm and linear or nonlinear programming, such as the gradient descent method [67], Newton method [195], quasi Newton optimization [66] and geometric programming [27]. Although these algo-

rithms are easy to implement and also provide some useful strategies for the optimization problem of the ideal models [106], they heavily rely on the initial starting point, the topology of the feasible region and the surface associated with objective functions and thus easy to fall into local optimum [57]. Besides, when solving large-scale complex optimization problems, due to the difficulty and high cost of computation of gradient information [94], gradient based algorithms are usually inefficient and cannot get ideal results.

To break through the bottleneck of the traditional optimization method, more recently, various meta-heuristics have been developed to deal with complex optimization problems, inspired by physical and biological systems. For instance, genetic algorithms [49] mimic the evolution of living organisms with genetic operators such as the reproduction, crossover and mutation. Simulated annealing [183] resembles the cooling process of molten metals through annealing. Ant Colony Optimization (ACO) [54] takes inspiration from the foraging and path making behaviors of ants. Particle swarm optimization (PSO) [95] copies the behaviors of birds searching for food in a collaborative way. Artificial fish-swarm algorithm [184] imitates the fish-swarm behaviors like the praying behavior, swarming behavior and chasing behavior. Immune algorithm [209] emulates the defense process of the immune system against its invaders in a biological body. Bees Swarm Optimization [91] models the behaviors of honey bees collecting, processing and advertising of nectars. Compared with traditional gradient-based algorithms, these nature inspired algorithms can tackle NP-hard problems more effectively and efficiently with the advantages of broad applicability, flexibility and ease of implementation. Therefore, meta-heuristic design has been drawing increasing attention from researchers, and novels algorithms have been presented in the latest decade such as the biogeography-based optimization algorithm [163], harmony search (HS) [126], the hunting search algorithm [139], the chemical reaction optimization algorithm [100] and the gravity algorithm [113]. However, most of these algorithms are

originally designed to solve the continuous or discrete problems which cannot be directly applied to binary-coding problems such as feature selection and knapsack problems, and therefore various binary variants like the discrete binary PSO algorithms [96, 131], discrete binary differential evolution algorithms [142, 187, 202], binary HS [65, 189] and binary gravitational search algorithm [151] are proposed to extend the applications of algorithms. Compared with real-coded or discrete-coded optimization algorithms, only a few binary-coded algorithms have been proposed in recent years while more and more complicated binary-coding problems arise. On the other hand, for many engineering problems with low precision like the design of controllers [188], binary-coding algorithms may perform better than real-coded algorithms in terms of search accuracy and the convergence speed as infinite search space is mapped into finite solution space. In addition, binary algorithms can be easily used to solve hybrid-coded problems in which real variables, discrete variables and binary variables are included, which is an advantage for engineering applications.

1.2.4 Fuzzy Time Series Forecasting

Modeling and forecasting financial markets provide key information for risk management, portfolio construction and financial decision making. However, financial markets are notoriously complex, dynamic, nonlinear, non-stationary, nonparametric, noisy and chaotic systems [2], which can be influenced by many factors including economic cycle, government policies, and investor psychology [215]. Therefore, forecasting such systems is extremely challenging, and this area has attracted considerable worldwide attention both in research and financial communities. Although many techniques [37, 108] have been developed for forecasting financial markets, building an efficient model is still an attractive proposition since even a tiny improvement in forecasting accuracy may have a huge impact on decision

making for portfolio and risk managers. Therefore, researchers continue to strive to discover ways to maximize the accuracy of financial market forecasting.

From a conventional statistical point of view, Box and Jenkins [26] established the Auto-Regressive Moving Average (ARMA) model, which combines a moving average with a linear difference equation. Although ARMA can make preliminary forecasts, it is restricted by certain requirements on the data, such as linear stationarity. To handle non-stationary datasets, the Auto-Regressive Integrated Moving Average (ARIMA) model [26], was proposed under the assumption of linearity between variables. However, conventional time series methods only deal with linear forecasting models and variables must obey a statistical normal distribution. In order to overcome this limitation, computational intelligence techniques, from a non-statistical perspective, have been applied to forecasting financial markets. Schumaker and Chen [159] Dai et al. [47] used support vector machines to forecast stock indices, and de Oliveira et al. [140] and Huck [82] proposed forecasting models using an artificial neural network. However, since the native financial markets include both nonlinear and vague data [166–168], fuzzy logic is utilized to describe daily observations and has recently been standing out as an important approach [172].

Thus, researchers are now showing an increasing interest in using Fuzzy Time Series (FTS) models for financial market forecasting [36, 154] as well as many real-word financial applications [10, 152]. The concept of FTS was first proposed by Song and Chissom to forecast university enrollment. In a financial context, Teoh et al. [173] presented a model incorporating trend-weighting into the FTS models and applied their model to the forecasting of empirical stock markets. Chen and Chang [38] handled fuzzy rules by clustering algorithms and assigned different weights to clusters and applied their model to forecast the Taiwan Stock Index(TAIEX). Then, Chen and Chen [40] continued to propose a method for forecasting TAIEX based on fuzzy variation groups as well as main factor and secondary

factor. By using high-order fuzzy logical relationships, Chen [39] also presented an approach to forecast the TAIEX and the inventory demand based on FTS. Zhou et al. [216] developed a portfolio optimization model combining information entropy and FTS to forecast stock exchange in Chinese financial market. Avazbeigi et al. [13] applied three variable FTS for forecasting the automobile production. Uslu et al. [181] considered the recurrence number of the fuzzy relations in the defuzzification stage of FTS and applied their method to spot gold forecast. Park et al. [143] developed bivariate fuzzy time series to predict TAIEX and South Korea's index. Wei et al.[198] forecasted the trend of TAIEX stock by combining a linear model and moving average technical index. Chen and Chen [38] introduced the probabilities of trends and proposed a modified fuzzy forecasting model with fuzzy-trend logical relationship. Rubio et al.[154] proposed weighted FTS model based on the weights of chronological-order and trend-order to forecast the future performance of stock market indices. Although those approaches are slightly successful for the financial market forecast, in the lately financial markets with high volatility, accurate forecasting tends to be more and more difficult and challenging due to the rapid changes of trading rules and management systems.

CHAPTER

2

REAL TIME UTILITY-BASED RECOMMENDATION FOR REVENUE OPTIMIZATION VIA AN ADAPTIVE ONLINE TOP-K HIGH UTILITY ITEMSETS MINING MODEL

2.1 Introduction

The mainstream recommendation techniques usually ingest product ratings received from customers and then suggest new products of potential interest. However, few of these approaches attempt to directly optimize business revenue— instead optimizing 'proxy' factors, such as customer purchase likelihood. Optimizing revenue directly is a more challenging task than predicting customer behavior since it involves a larger range of factors, such as real time price, customer interest trend, retailers' sale and so on. Therefore, the economic utility of a product is important to take into account because it directly influences the demand, and price, of each product.

In addition, existing online retailers, such as Amazon, Walmart and eBay, generate millions of transactions as well as a large number of user generated contents in a real-time manner. Furthermore, retailers now prefer to hold limited-time sales events to attract customers, such as Black Friday, in which up to 426 items were sold per second on Amazon at the end of 2013 [182]. This suggests that models need to learn from streaming data, not just from large amounts of offline data. Thus, the rapidly-growing popularity of RS, combined with the massive volumes of transaction flows, poses an urgent need for recommendation models that can be learned in real time.

There have been some existing methods that address the task of revenue-based recommendation [14, 121]. Although these methods include reasonable strategies that recommend items based on revenue-optimization, they have considerable limitations. Specifically, [14] focuses on static scenarios that are not useful in e-commerce due to its rapidly-changing nature. [121] typically needs a lot of prior knowledge, such as user's adoption rates for each item over time. And also, few existing methods are able to process streaming information in order to provide real time recommendations that are now becoming essential [61].

Therefore, to address these challenges, we focus on the real time utility model for revenue optimization since this model can truly reflect the integrated information towards revenue, rather than only optimizes users' adoption rate as previous works did. Specifically, we first define the problem of utility-recommendation towards revenue optimization. Then we set out a new research challenge: *real-time utility-based recommendation*. The major difference between our recommendation problem and the problems tackled in prior studies is that we take into account the retailer's overall revenue, and we do this on streaming data. Also, since in dynamic e-commerce environments, different potential recommendation itemsets could earn different revenues for the retailer, and even the same itemsets could generate different levels of revenue at different times of year (e.g., during the holiday season), it is very difficult for traditional RS to tackle these problems. Therefore, we propose an Adaptive Online Top-K high utility itemsets mining model (RAOTK) that extracts the utility information just using real-time transaction flows, and monitors Top-K high utility itemsets to guide the recommendations towards optimal revenue. Furthermore, we observe that only considering utility may reduce the customers' product adoption likelihood in some circumstances (since high-utility purchases may be less common than everyday purchases). As such, we derive three algorithm variants, which we call Adaptive Online Top-K HUIs mining with Ratings (RAOTK-R), Adaptive Online Top-K HUIs mining with Frequency (RAOTK-F) and Hybrid Adaptive Online Top-K HUIs mining with ratings and frequency (RHAOTK). Together, these give us a comprehensive set of utility-based recommendation algorithms. Finally, to improve accuracy and make our model more personalized to individual customers, we take buying power of customers into consideration and propose a simple but effective method to estimate the Online consumers' Willingness to Pay (OWP). By adopting OWP, the performance of the proposed framework is substantially enhanced.

We propose a framework for real time utility-based recommendation. The major contri-

butions of this work are summarized as follows:

- We propose a real time utility-based recommendation method, which considers transaction flow fluctuation (Section 2.2.2). We also propose an adaptive online Top-K high utility itemsets mining model (Section 2.3.1) to guide the utility-based recommendations in a real-time manner.
- We propose four efficient algorithms (RAOTK, RAOTK-R, RAOTK-F and RHAOTK) for optimizing utility-based recommendations and we provide a structural comparison of the four algorithms with discussion on their advantages and limitations (Section 2.3.2).
- We propose a simple but effective method to estimate the online consumers' willingness to pay (OWP), which accounts for buying power of customers, making our model more personalized (Section 2.3.4).
- We perform extensive experimental comparisons on three real-world datasets (Amazon, Foodmart and Chainstore). The results show that our algorithms (in particular RHAOTK) can make real-time recommendations that help optimize revenue (Section 2.4).

2.2 Formulation

In this section, we give the definition of real time utility-based recommendation and also introduce the concept of online transactions streams with flow fluctuation. Unlike most RS that utilize databases (e.g., of user content) to build models, we require only a transaction stream both for learning the model and for providing utility-based recommendations in real-time.

2.2.1 Problem Definition

As stated, the goal of real time utility-based recommendation is to optimize the total expected revenue. Let $U = \{u_i\}$ for $i = 1, 2, ..., n$ be a set of users in an online e-commerce network. User u_i makes revenue contribution r_i to the retailer. The value r_i can be divided into two parts: *target revenue* and *elastic revenue* as follows:

$$r_i = r_i^T + r_i^E \tag{2.1}$$

where r_i^T and r_i^E represent the target and elastic components of the revenue, respectively. Target revenue denotes the revenue derived from a user's *rigid demand* [73]. For example, if a user already knows the exact item that he or she wants to buy (for instance, an iPhone 6s Gray 64gb), the target revenue would be achieved by a direct purchase made by him or her without needing a recommendation. Clearly, since this is a rigid demand from the user, this demand is seldom impacted by recommendations from RS. On the other hand, elastic revenue refers to all other revenue types, such as revenue from purchases that accompany the main purchase (e.g., accessories for the iPhone). Elastic revenue also includes purchases from random browsing (e.g., sales on front page) and purchases inspired by other types of recommendation (e.g., unrelated items that other customers also bought). Thus, for an online retailer with n users, the total revenue TR from these users can be obtained by summing the revenue for each user. We therefore have:

$$TR = \sum_{j=1}^{n} r_j \tag{2.2}$$

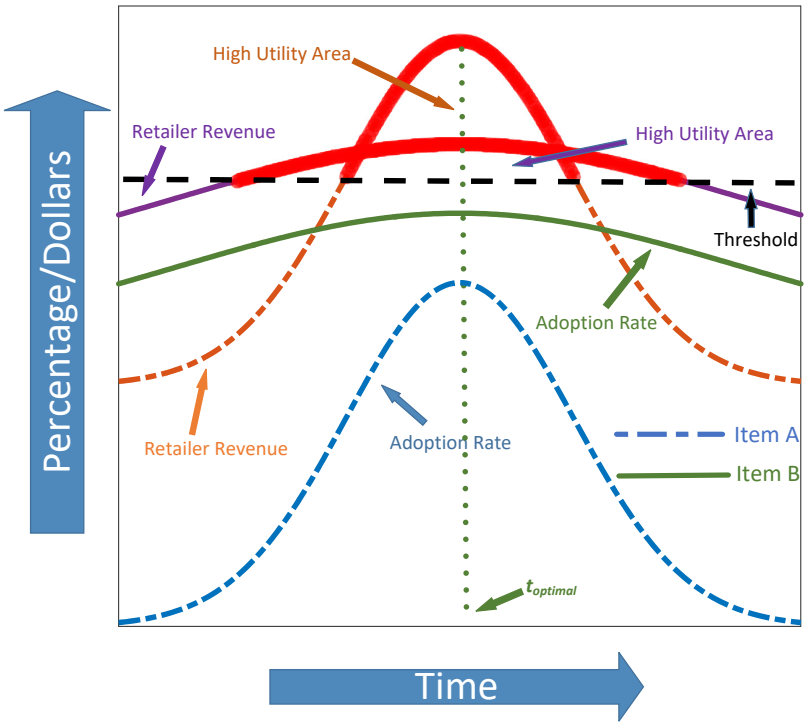


Figure 2.1 Utility distribution for two different items with distinct adoption rate curves that vary over time

The revenue generated from each item (IR) depends on each item's utility (IU) and user's adoption rate (AR), which can be formulated as Eq. 2.3.

$$IR = IU \times AR \tag{2.3}$$

Figure 2.1 illustrates the revenues generated from two items (or products) over time. We assume that IU is relatively constant with time, and so the IR is proportional to AR. AR usually varies with time roughly in the form of a normal distribution [93]. This implies that, for most items, there is a 'golden' revenue period. In this chapter, we call this a *high utility area*. The dashed lines (Item A) correspond to a product with lower average adoption

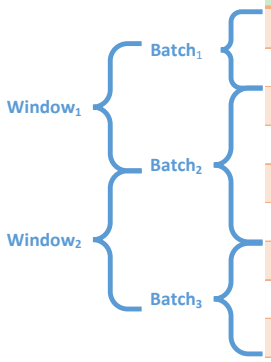
rate while the solid lines (Item B) represent a higher adoption rate item. Even though the adoption rate of Item B is always higher than that of Item A, the high utility area (above a fixed threshold) of Item A is larger than that of Item B at a given time. For intuitive understanding here, $t_{optimal}$, labeled in Figure 2.1, represents revenue will be optimal for each item. Our formulation of real time utility-based recommendations allows us to present items to users at the right time to obtain the optimal total revenue TR across all users.

2.2.2 Data Model

Before discussing the proposed approaches in detail, it is important to introduce the type of data we will be working with. Unlike most RS, we use online transaction streams to perform the utility-based recommendation. The transaction stream is defined as $TS = T_1, T_2, ..., T_m$, where T_m is the most recent of m transactions. $I = I_1, I_2, ..., I_n$ is a set of n distinct items such that for each transaction $T_k \in TS$ is a subset of we and each item $we_j \in we$ has two positive values: one is $q(I_j, T_k)$, which we call *internal utility* (e.g., quantity) of T_k and the other is $p(I_j, T_k)$ called *external utility* (e.g., price). Thus, we have $\forall T_k \in TS, T_k = (I_j, q(I_j, T_k)|I_j \in I)$.

Figure 2.2 shows an example of a data stream containing nine transactions. Inside each transaction, each letter represents an item and is associated with a purchase quantity (i.e., the internal utility). The corresponding item utilities are shown in Table 2.1.

In a real time environment, transactions are generated constantly [219]. To better analyse these transactions, we group several transactions into one batch and, in this way, we discretize the continuous transaction stream. For example, if an online retailer receives 2, 2, 1, 4 and 3 transactions in a single 5-second window, we treat them all as one batch that includes 12 transactions. Therefore, we now have successive batches $B_i = T_k, T_{k+1}, ..., T_n$ arriving at uniform time intervals. To analyze recent batches for trends, we introduce the



Transaction ID	Transaction	Transaction Utility
T_1	(A,1) (C,2) (D,2)	10
T_2	(A,2) (C,1) (E,4) (F,6) (G,3)	53
T_3	(A,4) (B,1) (C,5) (D,3)	26
T_4	(B,1) (C,4) (D,6) (E,2) (G,5)	42
T_5	(F,8)	40
T_6	(A,3) (B,1) (C, 4)	14
T_7	(B,1) (C,1) (D,1) (E,2) (F,1) (G, 1)	21
T_8	(D,1) (E,5) (G,4)	26
T_9	(C,3) (G,3)	9

Figure 2.2 An example of transaction stream

Table 2.1 Item utility

Item	A	B	C	D	E	F	G
Utility	2	4	1	3	3	5	2

concept of window. Each window W_i includes l most recent batches which can be formulated as $W_i = B_s, B_{s+1}, ..., B_{s+l-1}$. As new batch are most indicative of current trends, we remove the oldest batch in the window w_i every time a new batch arrives (Figure 2.2).

Definition 2.2.1. *(utility of an item) The utility of item $i_j \in we$ in transaction $t_k \in TS$ is denoted by $u(i_j, t_k)$ and defined as*

$$u(i,t) = p(i_j, TS) \times q(i_j, t_r) \tag{2.4}$$

Definition 2.2.2. *(utility of an itemset in a transaction) The utility of an itemset X in a transaction T_k is denoted by $u(X, t_k)$ and defined as*

$$u(X,T) = \sum_{x \in X \wedge X \subseteq T_k} u(X, T_k) \tag{2.5}$$

Definition 2.2.3. *(utility of a transaction) The utility of a transaction T_k is denoted by tu(T) and defined as*

$$t\,u(T) = \sum_{i \in T_k} u(i, T_k) \tag{2.6}$$

Definition 2.2.4. *(utility of an itemset in a transaction stream TS) The utility of an itemset X in a transaction stream TS is denoted as $u(X)$ and defined by*

$$u(X) = \sum_{T_k \in TS \wedge X \subseteq T_k} u(X, T_k) \tag{2.7}$$

Definition 2.2.5. *(Transaction-weighted utilization) The transaction-weighted utilization of an itemset X is the sum of the transaction utilities of all the transactions containing X, which is donated by $t\,w\,u(X)$ and defined as*

$$t\,w\,u(X) = \sum_{T_k \in TS \wedge X \subseteq T_k} t\,u(T_k) \tag{2.8}$$

Then, $u(D, T_1) = 2 \times 2 = 4; u((BG), T_7) = 1 \times 4 + 1 \times 2 = 6$; $t\,u(T_8) = 3 \times 1 + 3 \times 5 + 2 \times 4 = 26$; $t\,w\,u(AC) = t\,u(T_1) + t\,u(T_2) + t\,u(T_6) = 10 + 23 + 14 = 47$ and $u(F) = 2 \times 8 = 16$.

In real world applications, transaction streams are always subject to fluctuation [27] — day and night, weekday and weekend, holiday and working days, as well as seasonal peaks and troughs. These are all components of transaction stream fluctuation (TSF). Figure 2.3 shows two common forms of TSF. As can be seen from the figure, transactions streams vary considerably across time periods. Our method takes this into account.

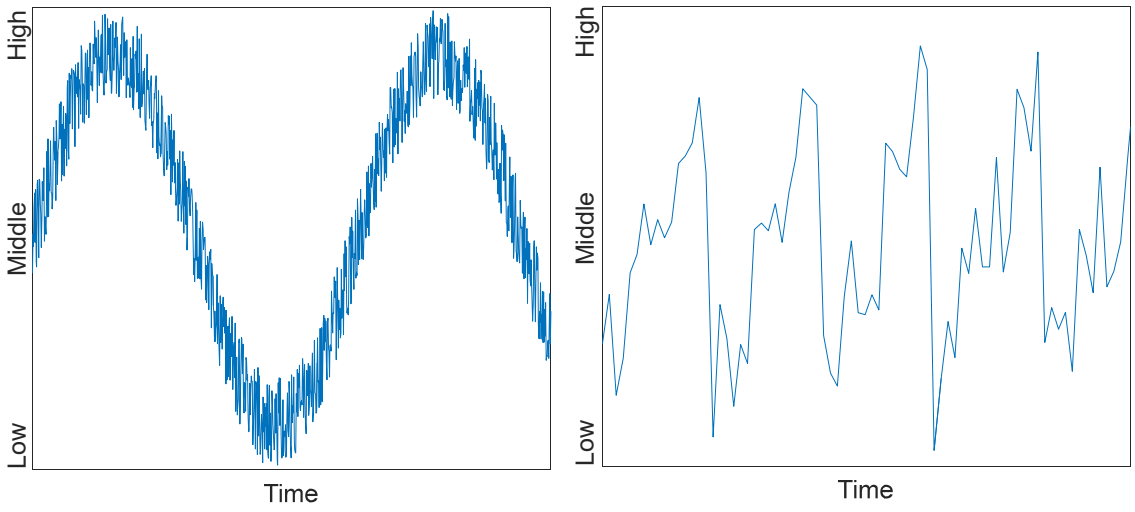


Figure 2.3 Examples of transaction streams fluctuation

2.3 Approach

To tackle the real time utility-based recommendation problem, we first build upon an adaptive Top-K high utility itemsets (HUIs) mining model to capture utility information from the transaction stream with TSF. We then propose several recommendation strategies efficiently towards the revenue optimization. Finally, we investigate variants that incorporate user ratings and frequency of item sales.

2.3.1 Creating an Adaptive Model

As Eq. 2.3 shows, the item revenue depends on two factors: item utility and user's adoption rate. Usually, the adoption rate is not known for new items, which gives us a cold-start problem [213]. However, since transactions are generated automatically when users place orders, it is possible to estimate both IR and AR through transactions streams using standard Top-K high utility itemsets mining methods such as T-HUDS [219] and TOPK-SW [121].

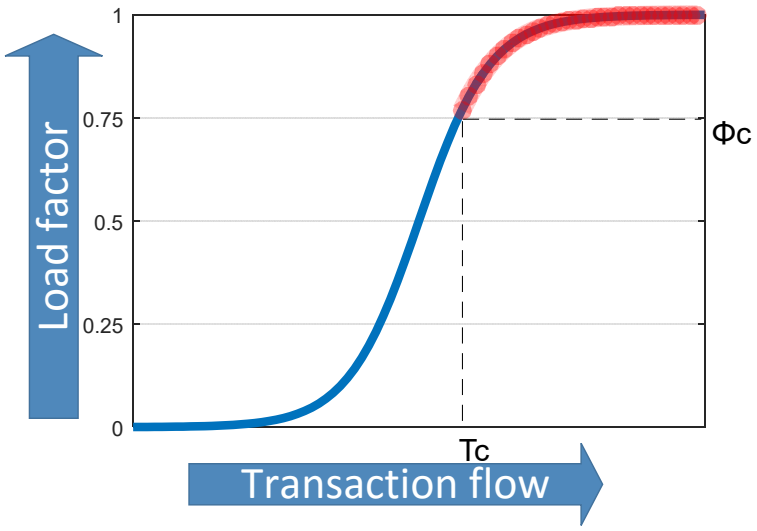


Figure 2.4 The curve of transaction flow transformation

Definition 2.3.1. *(Top-K high utility itemset). An itemset X is called Top-K high utility itemset (Top-K HUI) in transaction stream if f there are less than k itemsets whose utilities are larger than u(X).*

In general, Top-K high utility itemsets mining methods include three main steps (the details can be found in [219] and TOPK-SW [121]):

(1) Tree construction (e.g., HUDS-tree in T-HUDS method);

(2) Mining Top-K HUIs on a window;

(3) Tree update (e.g., insert and remove transactions in the new batches into the tree).

Thus, it is possible to obtain utility information from transaction streams in real-time using standard Top-K high utility itemsets mining methods such as T-HUDS [219] and TOPK-SW [121]. However, these approaches do not take stream fluctuation into account (and they are only methods for generating HUIs, not for producing recommendations). T-HUDS is designed to handle low-volume transaction streams and use minimal memory, whereas TOPK-SW is better able to process high-volume streams (but it is very memory-

intensive due to its internal tree structures). In our approach, we switch between these two algorithms according to the normalized threshold function in Eq. 2.9,

$$\Phi = \frac{1}{1 + e^{-\lambda(T-\delta)}} \tag{2.9}$$

where T is the number of transactions in a certain window and λ, δ are control parameters. By using the normalized threshold, we can easily transform transaction flow rate t a *load factor* Φ (in the range 0 to 1). This makes it simple to monitor whether the transaction flow is heavy or light. Figure 2.4 shows this transformation and Φ_c is the threshold (or critical) value of Φ that separate heavy and light flows. If Φ is smaller than Φ_c, we consider the transaction flow is in a normal or light condition and thus we utilize T-HUDS to mine HUIs while, if Φ is greater than Φ_c, we consider the system to be experiencing a heavy transaction flow and TOPK-SW is adopted instead.

2.3.2 The RAOTK Model and Variants

In this section, we first propose an efficient algorithm named RAOTK (utility-based Recommendation based on AOTK) for revenue optimization and then describe three algorithm variants: RAOTK-R (RAOTK with Ratings), RAOTK-F (RAOTK with Frequency) and RHAOTK (Hybrid RAOTK with ratings and frequency). Each of these includes a different strategy to optimize revenue.

2.3.2.1 RAOTK

We first describe our utility-based recommendation algorithm called RAOTK (pseudo-code in Algorithm 2.1). Our objective is to find a recommendation strategy that presents items/itemsets in an optimal time manner to optimize the revenue that each user con-

tributes. Since the RS may face a cold start [69] problem when insufficient data (e.g., on user's preferences) is available, our method exacts utility information just through the transaction stream by adopting the proposed AOTK model. In a nutshell, the algorithm starts with recommendation sets (R) being empty ($\oslash$) and, in each iteration, it adds to R the items/itemsets that provide the potentially largest revenue. Unlike classical deterministic characteristics of recommendation algorithms that may reduce the performance of recommendation [61], to enhance accuracy for the cold start situation, we employ a probability algorithm in our method. First, we propose the idea of *itemsets normalization by utility* and then we use a *weighted random selection* process for generating recommendation candidates.

The RAOTK algorithm also extends this to make use of user's purchase history (PH) which is usually easy to obtain in practice, and allows us to achieve better performance via a personalization strategy. We use PH in two ways: firstly, we combine it with the Top-K HUIs pool and, secondly, we insert PH into the transaction stream to generate personalized recommendations.

Itemsets normalization by utility (INU). Let $TKI = HUI_1, HUI_2, ..., HUI_k$ be a set of Top-K HUIs and $TKU = TKU_1, TKU_2, ..., TKU_k$ be a set of the utilities of corresponding HUIs. We then have:

$$||TKU_N|| = \sum_{i=1}^{k} TKU_i \tag{2.10}$$

where TKU_N is total utility of TKI. The normalization of each HUI in TKI can be derived from Eq. 2.11:

$$\hat{TKU}_i = \frac{TKU_i}{||TKU_N||} \quad (i \in 1, 2, ..., k) \tag{2.11}$$

Clearly, $\hat{HUI}_i \in [0,1]$ and $\sum_{i=1}^{k} \hat{HUI}_i = 1$

Weighted random selection (WRS). Let p be a random number between 0 and 1 and $\zeta = \zeta_1, \zeta_2, ..., \zeta_m$ be an increasing sequence of $\hat{HUI}$ as in definition above.

Definition 2.3.2. *(HUI increasing sequence). A sequence* $< \hat{HUI} >_k$ *of terms of* ζ *is increasing iff:* $\forall j, k \in \zeta : j < k \Rightarrow \hat{HUI}_j < \hat{HUI}_k$

Thus, starting from 0 and ending at 1, we can get a set of continuous intervals $CI = CI_1, CI_2, ..., CI_k$ composed of ζ where CI_i can be represented as:

$$CI_i = [\sum_{m=1}^{i-1} \zeta_m, \sum_{m=1}^{i} \zeta_m] \quad (i \in 2, ..., k) \tag{2.12}$$

with $CI_1 = [0, \zeta_1]$. Then the recommendation candidate will be decided by the value p as in Eq. 2.13. This means that, if p falls in CI_i, the corresponding HUI_i will be selected.

$$R^+ = HUI_i | p \in CI_i \tag{2.13}$$

where R^+ is the candidate selected into the recommendation set R.

Personalized strategy (PS). When a user's PH is available, RAOTK first links PH with TKI. If there exist items/itemsets in PH that are also items/itemsets in TKI, the corresponding HUI will be selected directly. More formally, let ph be a subset of PH, and the recommendation candidate will be decided as Eq. 2.14:

$$R^+ = \arg\max\{TKU | (\exists ph \in PH \vee ph \subset TKI)\} \tag{2.14}$$

where k_p is the length of PH. Furthermore, besides linking TKI, RAOTK also combines PH with the transaction stream TS. First, we randomly select an item named ph in the user's purchase history and then Eq. 2.14 shows how RAOTK generates recommendation

candidates. Note that, here we also adopt a probability method to guide the algorithm's choice of personalized strategy.

Definition 2.3.3. *(utility of an itemset in a transaction stream TS) The utility of an itemset X in a transaction stream TS is denoted as $u(X)$ and defined by*

$$u(X) = \sum_{T_k \in TS \wedge X \subseteq T_k \wedge I_m \subseteq X} p(I_m, TS) \times q(I_m, T_k) \tag{2.15}$$

$$\begin{aligned} R^+ &= \arg\max\{U(X, TS_i) | (\exists ph_j \in PH \vee ph_j \subset TS_i)\} \\ &(j = 1, 2, ..., k_p; l = 1, 2, ..., k_p) \end{aligned} \tag{2.16}$$

For each recommendation, RAOTK determines which strategy should be adopted through the use of a control parameter: *utility entry rate* P_u. If a random number (in the range 0-1) $p_r < P_u$, then the weighted random selection will be chosen, otherwise RAOTK will use the personalized strategy to generate candidates.

Algorithm 2.1 RAOTK

Input: Transaction stream TS, recommendation set size K_r, user set U and purchase history PH, utility entry rate P_u

Output: Recommendation set R

$R \leftarrow \oslash$ and call algorithm AOTK to generate HUIs;

foreach $u \in U$ **do**

  while $|R| < K_r$ **do**

    if $PH \in \oslash$ **then**

      INU and WRS as 2.3.2.1

    else

      foreach $ph \in PH$ **do**

        PS for TKI as 2.3.2.1

      $p_r \leftarrow rand(0,1)$ **if** $p_r < P_u$ **then**

        INU and WRS as 2.3.2.1

      else

        random select $ph \in PH$ PS for TS as 2.3.2.1

2.3.2.2 RAOTK-R

The second algorithm that we propose is RAOTK-R (pseudo-code in Algorithm 2.2). It incorporates ratings into RAOTK to improve the performance of utility-based recommendation. We first give a definition of the utility based rating and then describe several strategies that efficiently enable RAOTK to make use of rating information towards revenue optimization.

Utility-based rating. Generally, in an online e-commerce environment, users participate in shopping experiences and share feedback by evaluating products with numerical ratings [97]. Although high ratings usually indicate a high user adoption rate for certain

items, from the perspective of utility, high rating doesn't guarantee high revenue for retailer. Thus, we define utility-based rating (UR) as Eq. 2.17 to take both utility and rating into consideration:

$$UR = U \times \Re \tag{2.17}$$

where U is item utility and $\Re$ is predicted ratings. We also use the following equation to normalize UR:

$$\hat{UR}_i = \frac{UR_i}{max\{UR\}} \quad (i \in 1,2,...,k) \tag{2.18}$$

UR-guided selection strategy. Instead of only utilizing the weighted random selection strategy, RAOTK-R also uses UR to generate recommendation candidates with a probability method. Let $\Psi = \Psi_1, \Psi_2, ..., \Psi_m$ be the decreasing sequence of $\hat{UR}$ as definition above.

Definition 2.3.4. *(UR decreasing sequence). A sequence* $< \hat{UR} >_k$ *of terms of* Ψ *is decreasing iff:* $\forall j,k \in \Psi : j < k \Rightarrow \hat{UR}_j > \hat{UR}_k$

Then, for a certain random probability, RAOTK-R generates candidates as in Eq. 2.19:

$$R^+ = \arg\max\{\hat{UR}\} \tag{2.19}$$

Modified personalized strategy (MPS) based on UR. Unlike RAOTK that just uses utility for generating candidates when combining PH with TS, RAOTK-R adopts UR instead to guide the selection as in Eq. 2.20 (compared to Eq. 2.14):

$$R^+ = \arg\max\{\hat{UR} \,|(\exists ph \in PH \vee ph \subset TS)\} \tag{2.20}$$

Similar to RAOTK, RAOTK-R also probabilistically determines the strategy to take. The difference is that, since RAOTK-R has a unique UR-based selection strategy, we then use

another control parameter: *TS entry rate* P_t (where $P_t < P_u$) for selection. If $P_u < p_r < P_t$, UR-based strategy will be adopted and if $p_r > P_t$, RAOTK will use personalized strategy to generate candidates over transaction stream.

2.3.2.3 RAOTK-F

Similar to rating-based methods, frequency is another widely used index in RS [64]. Thus, we propose a third algorithm named RAOTK-F that includes sales frequency information in utility-based recommendations. We first give a definition of utility-based frequency and then describe how this modifies the algorithm.

Utility-based frequency and UF guided selection strategy. A *frequent item* in e-commerce is one that users often buy (in a given period of time). It is reasonable to recommend frequent items since the probability that users will buy them is higher than for infrequent items. However, from the perspective of utility, high frequency doesn't guarantee high revenue for the retailer. Thus, we define utility-based frequency (UF) in Eq. 2.21 to take both utility and frequency into consideration:

$$UF = U \times \Im \tag{2.21}$$

where U is the item utility and $\Im$ is the item sales frequency. We also use the following equation to normalize UF:

$$\hat{UF_i} = \frac{UF_i}{max\{UF\}} \quad (i \in 1,2,...,n) \tag{2.22}$$

The remainder of the process is similar to that for RAOTK-R, but guided by UF instead of UR.

Algorithm 2.2 RAOTK-R

Input: Transaction stream TS, recommendation set size K_r, user set U, purchase history PH, predicted ratings $\Re$, utility entry rate P_u, TS entry rate P_t

Output: Recommendation set R

$R \leftarrow \oslash$ and call algorithm AOTK to generate HUIs;

foreach $u \in U$ **do**

- **while** $|R| < K_r$ **do**
  - **if** $PH \in \oslash$ **then**
    - $p \leftarrow rand(0,1)$ **if** $p_r < P_u$ **then**
      - INU and WRS as 2.3.2.1
    - **else**
      - UR strategy as 2.3.2.2
  - **else**
    - **foreach** $ph \in PH$ **do**
      - MPS for TKI as 2.3.2.2
    - $p_r \leftarrow rand(0,1)$ **if** $p_r < P_u$ **then**
      - INU and WRS as 2.3.2.1
    - **else if** $P_u < p_r < P_t$ **then**
      - UR strategy as 2.3.2.2
    - **else**
      - random select $ph \in PH$ and MPS for TS as 2.3.2.2

2.3.2.4 RHAOTK

The above two algorithm variants either utilize rating or frequency to guide the utility-based recommendation. Since, for some circumstances, due to insufficient information or misleading feedback provided by users, some predicted ratings do not reflect the true

preferences of users, thus, if combined with frequency, the performance may be improved. For example, assume a certain item is recommended to the user according to its predicted rating but the sales frequency of this item is low. As a result, it may be hard for the user to accept this recommendation. Conversely, high sales frequency does not mean high predicted rating for a particular user. Thus, it makes sense to combine rating and frequency in a hybrid utility-based recommendation scheme. As such, we propose a fourth algorithm named RHAOTK that includes both. We first give a definition of hybrid utility evaluation (HUE) and then describe the modified strategies adopted in RHAOTK.

Hybrid utility evaluation and HUE-guided selection strategy. Based on the above insight and methods to get utility-based rating (Eq. 2.17) and utility-based frequency (Eq. 2.21), we propose an idea to evaluate utility in a hybrid manner:

$$HUE = \hat{UR} \times \hat{UF} \tag{2.23}$$

where $\hat{UR}$ is the utility-based rating and $\hat{UF}$ is the utility-based frequency. We also use the following equation to normalize HUE:

$$\hat{HUE}_i = \frac{HUE_i}{max\{HUE\}} \quad (i \in 1,2,...,n) \tag{2.24}$$

HUE guided selection strategy. In RHAOTK, to utilize the hybrid information from utility, rating and frequency, we modify Eq. 2.25 as the following in selection procedure.

$$R+ = \arg\max_{N_R}\{\hat{HUE}\} \tag{2.25}$$

which means the items with top NR number of $\hat{HUE}$ candidates will be selected. Also

regarding to personalized strategy, when combines PH with TS, RHAOTK adopts HUE to guide the candidates generations as Eq. 2.26.

$$R^{+} = \arg\max\{\hat{HUE} \,|(\exists ph_j \in PH \vee ph_j \subset TS_i)\} \quad (j = 1,2,...,k_p; l = 1,2,...,k_p) \tag{2.26}$$

To summarize, the four proposed algorithms can be visualized by a 3-dimensional graphical representation (Figure 2.5). As can be seen from figure, three item attributes (utility, rating and frequency) are treated as 3 base coordinates. Based on this coordinate system, RAOTK, represented by the brown vector, only considers utility information, while RAOTK-R, represented by the green vector, combines utility with rating. Similarly, RAOTK-F, represented by the magenta vector takes both utility and frequency into consideration. Finally, RHAOTK, represented by the cyan vector includes all 3 types of information.

2.3.3 Algorithm Complexity

For RAOTK, the INU and WRS operation has the same time complexity $O(log(TKI))$ and thus if $PH \in \oslash$, the complexity is $O(U \times k_r \times log(TKI))$, otherwise, the complexity is $O(U \times ((k_r + P_u + PH) \times log(TKI) + (1 - P_u) \times log(TS)))$. And for the three variants, the UR, UF and HUE has the time complexity $O(log(UR))$, $O(log(UF))$ and $O(log(HUE))$, respectively. Here we only show the complexity of RAOTK-R due to the similarity. On the condition of $PH \in \oslash$, the complexity is $O(U \times k_r \times (P_u \times log(TKI) + (1 - P_u) \times log(UR)))$, otherwise, the complexity is $O(U \times ((k_r + P_u + PH) \times log(TKI) + (1 - P_u - P_t) \times log(UR) + P_t \times log(TS)))$.

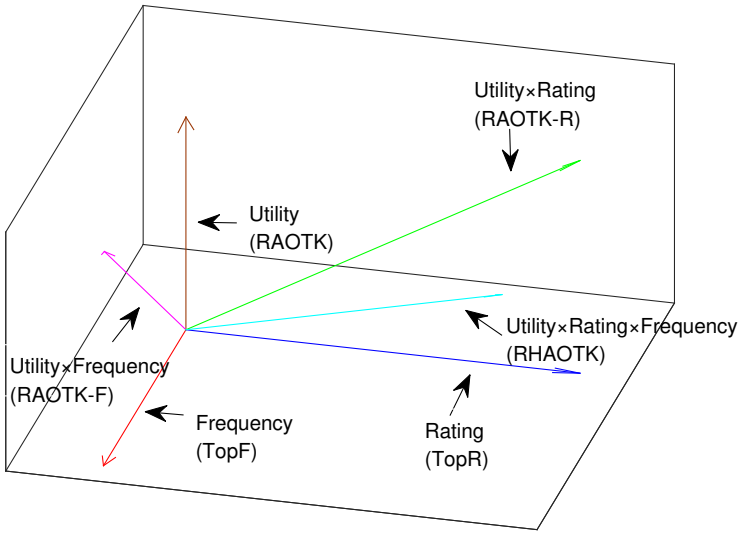


Figure 2.5 3D representation of the algorithms

2.3.4 Online user's Willingness to Pay (OWP) Estimation Model

To better employ the proposed methods for utility-based recommendation, in this section, we propose an Online user's Willingness to Pay (OWP) estimation model. Through this model, we can approximately estimate user's buying power to avoid some ineffective recommendations. Generally, our model compares average *user's buying power* in transaction streams with the current user's buying power and then a coefficient named UWP will be calculated accordingly to determine each *user's willingness to pay*. More formally, let APC be the *average price of a category* and UBP be the user's buying power - these are represented

in Eqs. 2.27 and 2.28, respectively.

$$APC(m) = \frac{\sum_{i=1}^{k} GP_i | GP_i \in C_m}{||GP||} \quad (i = 1,2,...,k;\ m = 1,2,...,M) \tag{2.27}$$

$$UBP(m,n) = \frac{\sum_{j=1}^{l} IP_j | (IP_j \in C_m \wedge IP_j \in PH_n)}{||IP||} \quad (j = 1,2,...,l;\ m = 1,2,...,M;\ n = 1,2,...,N) \tag{2.28}$$

where $APC(m)$ means the average price of category m (C_m) and $UBP(m,n)$ represents the n^{th} user's buying power for category m. Also GP and IP are the *global payment* (that all others users have paid for the category) and *individual payment* (that the current user has paid for this category). So, the degree of the n^{th} user's willingness to pay (UWP) for category m is given by:

$$UWP(m,n) = \frac{UBP(m,n)}{APC(m)} \quad (m = 1,2,...,M;\ n = 1,2,...,N) \tag{2.29}$$

To locate the position user stands in the whole online population, here we use Figure 2.6. for fuzzy classification. As can be seen from figure, we divide the UWP into five categories in our OWP model and then based on this classification we optimize our utility-based recommendation. More specifically, here we use neighbor rule considering the fuzzy characteristics of OWP model. For example, when a user is tagged as "Middle-valuation", then we will not recommend the items with lowest and highest price category (similar definition

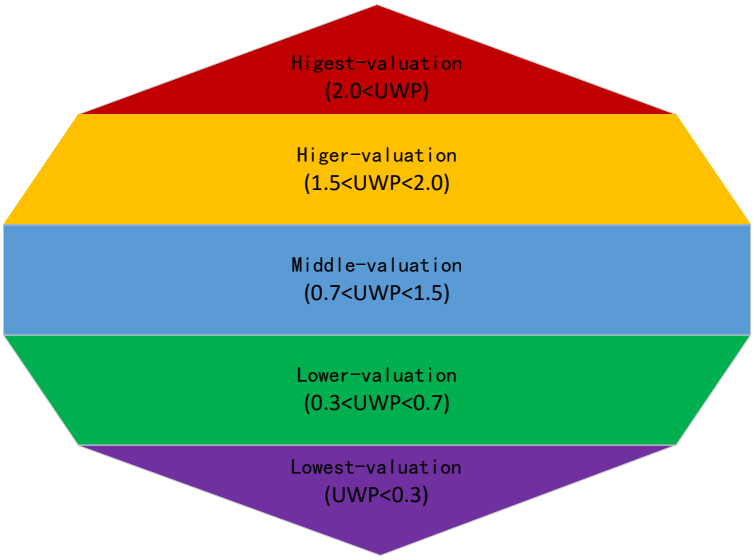


Figure 2.6 Real time utility distribution

as UWP category). Therefore, when making the recommendation decision, OWP can act as a role of filter to avoid ineffective recommendation.

2.4 Empirical Evaluation

In this section, we first describe the datasets used for evaluation as well as our experimental setup. Then, we conduct the process of parameter optimization and finally discuss the results.

2.4.1 Datasets

We used 3 open-source datasets for the evaluation:

- Amazon [130]—this dataset consists of text-content reviews and item metadata that includes descriptions, price, brand and co-purchasing info (such as “bought together”

and “also bought”). The dataset includes 142.8 million reviews and 9.4 million products spanning May 1996 - July 2014. We selected two very distinct categories (Electronics and Clothing) and, for each category, we select the 10,000 most popular items July 2010 – July 2014. We used a total of 100,000 transactions for each category.

- Foodmart [179]—this dataset, acquired from the Microsoft FoodMart 2000 database, contains sales records collected through a series of supermarkets and grocery stores. Each record includes: product, price, unit sale, purchase time, customer and store name. The data includes 269,720 transactions associated with 1559 products.

- ChainStore [147]—this dataset, obtained from NU-Mine-Bench 2.0, consists of 1,112,949 transactions of a chain supermarket in California. The data contains product, price and unit sale information, and includes a total of 46,086 items.

We chose these datasets in order to test our algorithms on three different real-world scenarios: (1) where purchase history and ratings are available (Amazon), (2) where only purchase history but no ratings are available (Foodmart), and (3) where neither are available (ChainStore). We assume that, in the real-world, a transaction stream will always be available, but user reviews/ratings and purchase history may not.

2.4.2 Experimental Settings

We compared our algorithms with the following state of the art algorithms for revenue optimization: G-Greedy [122], a recently proposed dynamic algorithm based on ratings to maximize revenue; TopR (Top-rating) recommends to every user the k items with highest predicted rating using MyMediaLite [59]; TopF (Top-frequency) recommends to every user the k items with highest sales frequency [174]. To verify if the statement in [59] can be also

applied to e-commerce, we include a baseline that generates random recommendations [59]. As per the real-world scenarios described in Section 2.4.2, we were not able to test all these algorithms against all the datasets (G-Greedy cannot work yet without ratings). In all test cases, we are looking to optimize the Mean Total Revenue (MTR) achieved by recommendation as follows:

$$MTR = \frac{\sum_{k=1}^{N_T} \sum_{j=1}^{N_U} \sum_{i=1}^{N_M} p(I_i, TS) \times q(I_i, TS)}{N_T} \tag{2.30}$$

where N_T is the number of tests; N_U represents the size of user set and N_M is the number of matched items (accepted recommendations). We randomly select 10000 successive transactions in two categories of Amazon datasets for parameters study. Based on the results, we select $P_u = 0.4$ for ROATK one parameter set: P_t =0.6 and P_u =0.2 for all three variants. For transaction stream fluctation (TSF) setting, we generated a sinusoid to simulate fluctuation with maximum 1000 batches and each package included a random number of transactions ranging from 1 to 5. Here, Φ_c was set to 0.75 (roughly 1:3 of busy/idle time) and we generated Top 200 HUIs for each transaction stream and 50 recommendation candidates (no repeated recommendations) for each user. Note that the proposed algorithms are probability-based, so we ran all the tests 10 times and averaged the results.

2.4.3 Parameter Analysis

RAOTK depends on one control parameter: the utility entry rate P_u, while RAOTK-R, RAOTK-F and RHAOTK are each influenced by two parameters: the TS entry rate P_t and the utility entry rate (modified as UR, UF, and HUE), P_u. To verify the performance under different parameters, both categories of the Amazon dataset (Electronics and Clothing) were used.

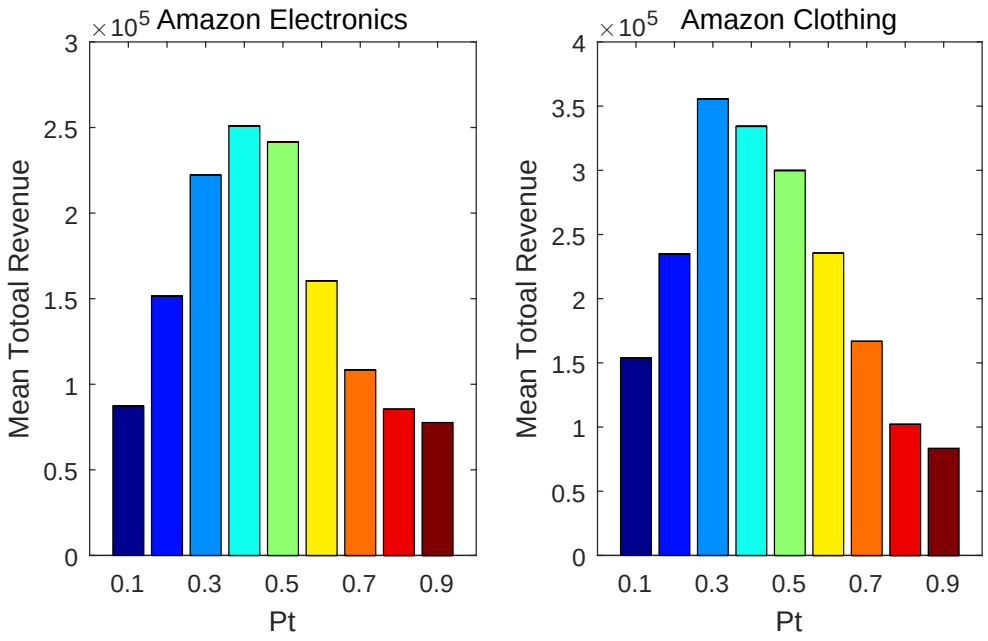


Figure 2.7 The results of RAOTK under different p_t

In all cases, we are looking to optimize MTR achieved by recommendation as in Eq. 2.30. Figure 2.7-2.10 show the MTR results for RAOTK for a range of values of P_t and also for other three variants using different P_t and P_u values. The results shown RAOTK gains best performance when the value of P_u is between 0.3 and 0.5. This indicates that selecting too many HUIs regardless of user's preferences may reduce performance. Based on this, we select $P_u = 0.4$ for testing ROATK. For RAOTK-R, RAOTK-F and RHAOTK, the optimal value for P_t was located between 0.6 and 0.7 and P_u was in the range 0.2 to 0.3. This indicates that a roughly even balance of UR/UF/HUE algorithms, as compared to the personalized pool, gives best performance. In this chapter, although we can set the parameters for each algorithm separately to achieve the best performance, to be more generalizable, we only use one parameter set: P_t =0.6 and P_u =0.2 for all three algorithms.

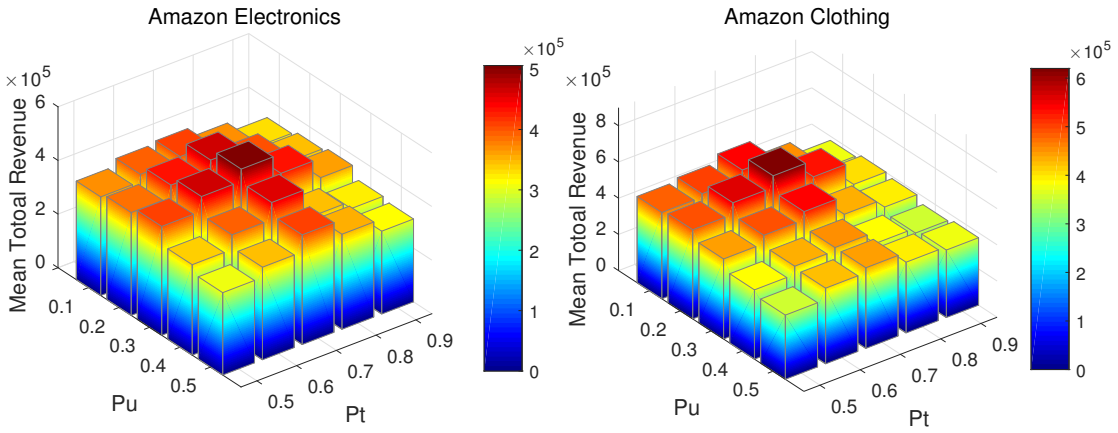


Figure 2.8 The results of RAOTK-R under different p_t and p_u

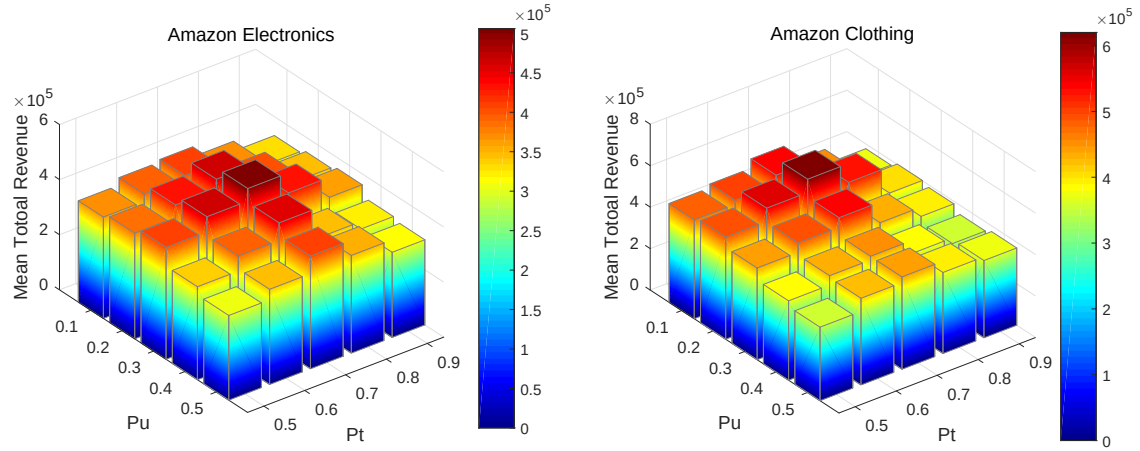


Figure 2.9 The results of RAOTK-F under different p_t and p_u

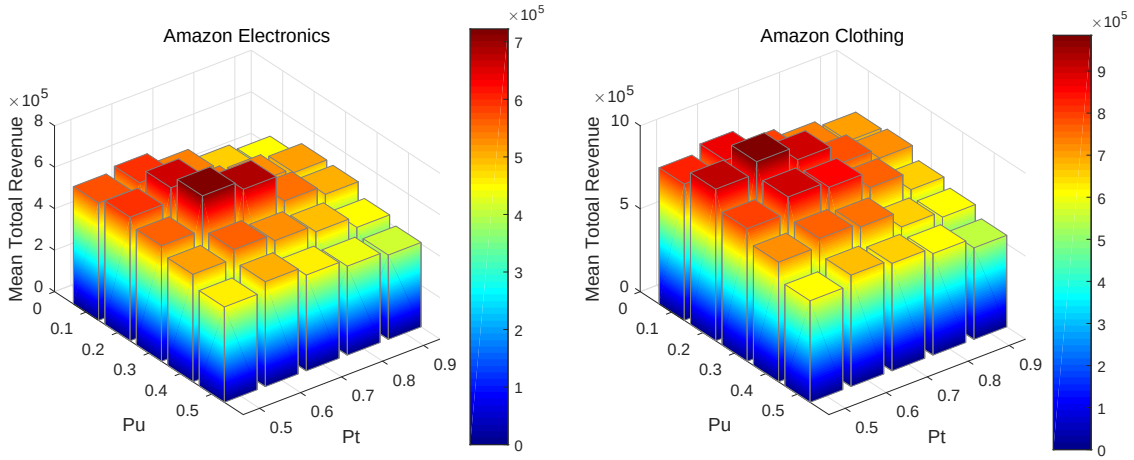


Figure 2.10 The results of HRAOTK under different p_t and p_u

2.4.4 Results and Analysis

To verify the revenue optimization ability and scalability, the proposed algorithms were applied to three datasets listed in Section 2.4.1. Experimental results, based on optimization of the MTR metric, are shown in Figure 2.11 (note that due to the scenario setting, OWP didn't apply for the ChainStore dataset).

It can be seen that RHAOTK outperforms other algorithms on all datasets. Specifically, it outperforms G-Greedy by about 20% revenue gain and also the RAOTK-R and RAOTK-F algorithms about 50% increase. Compared to TopR and TopF methods, the performance of RHAOTK is approximately 2-3 times better. It is interesting to notice that, although [61] mentioned that random news recommendations can sometimes perform better than those from recommender systems, the results here, show that this is not the case. In addition, note that even for the different categories in the Amazon dataset, the performance of RAOTK-R

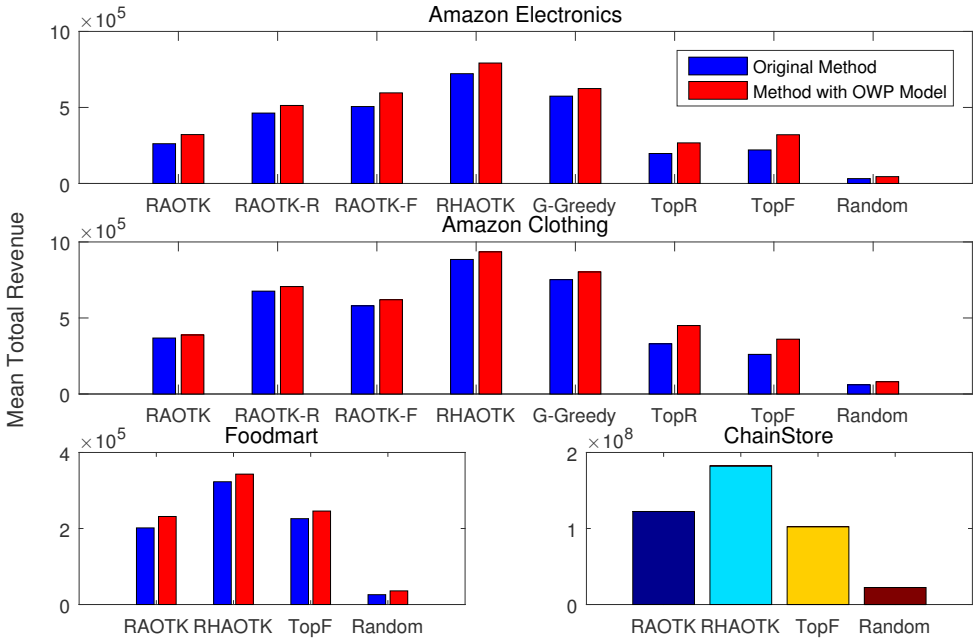


Figure 2.11 Mean total revenue obtained from all datasets

and RAOTK-F differ due to user's different preferences. For example, people may like to use the same type of electronics but have very different clothing preferences. Besides, the results also show the effectiveness (10-20% gain compared to original method) of the proposed OWP model in different cases.

2.5 Related Work

Most research into recommender systems in e-commerce has focused on models that guide users towards products based on user's predict ratings [45]. Generally, RS algorithms can be classified into content-based (CBF) or collaborative filtering (CF) methods. Those techniques usually use the ratings received from users on the products and then suggest new products of potential interest [3]. However, these methods are all from the perspective

of customers but ignore one of the most important business goals: the optimization of revenue.

Recently, there have been some existing methods that address the task of revenue recommendation [14], [35], [48], [122]. Although these methods have good strategies to recommend items based on revenue, [14], [35], [48] focus on static scenarios that have limited value in real-world e-commerce due to its capricious nature. While [122] also often needs prior knowledge of user's adoption rates for each item over time and thus cannot be applied to our case. We have argued that the ability to generate real-time recommendations is becoming increasingly critical [61], and that the challenge of real time utility-based recommendation is not well studied in the context of revenue optimization.

2.6 Conclusions

In this chapter, we have addressed a new research challenge - that of real time utility-based recommendation for revenue optimization, and proposed four algorithm variants to solve this challenge. We combine an adaptive online Top-K high utility itemsets mining model with rating and frequency information. In addition, to make our algorithms more personalized, we also take the buying power of customers into consideration and then propose a simple but effective method to estimate the consumers' willingness to pay. Finally, we conduct extensive experiments on Amazon, Foodmart and ChainStore datasets to show the effectiveness of the proposed algorithms.

CHAPTER

3

AN INTELLIGENT WEIGHTED FUZZY TIME SERIES MODEL BASED ON A SINE-COSINE ADAPTIVE HUMAN LEARNING OPTIMIZATION ALGORITHM AND ITS APPLICATION TO FINANCIAL MARKETS FORECASTING

3.1 Introduction

During the last decade, the world stock market cap has soared by 133% (from 28 trillion to 65 trillion), especially in emerging markets, e.g., the market cap of China and India has exploded by 1479% and 639%, respectively [85]. Thus, driven by the market power, stock market prediction plays an increasingly important and even crucial role in economic decisions [72]. However, since the financial markets are complex and nonlinear systems which can be influenced by various factors, such as economic cycle and government policies, it is extremely challenging to do an accurate stock forecasting, and thus has attracted considerable worldwide attention, in both the research and financial community. Subsequently, a variety of forecasting approaches have been developed [36, 83, 86, 133, 170, 175, 185, 197, 207]. Among these approaches, researchers have recently shown a renewed interest in using Fuzzy Time Series (FTS) models for stock forecasting [36, 40, 154, 193], as FTS possesses the universal approximation property in nature and has thus been successfully used in many financial applications in the real world [10, 60, 128, 152]. The concept of FTS was first proposed by Song and Chissom to forecast university enrollment. In a financial context, Teoh et al. [173] presented a model that incorporated trend-weighting into the FTS models, and applied their model to the forecasting of empirical stock markets. Chen and Chen [40] proposed a method for forecasting the Taiwan Stock Index (TAIEX) based on FTS and fuzzy variation groups. By using high-order fuzzy logical relationships, Chen et al. [39] also presented an approach to forecast the TAIEX and inventory demand based on FTS. Zhou et al. [216] developed a portfolio optimization model combining information entropy and FTS to forecast stock exchange in the Chinese financial market. Uslu et al. [181] considered the recurrence number of the fuzzy relations in the defuzzification stage of FTS, and applied their method to spot gold forecast. Rubio et al.[154] proposed weighted FTS model based

on the weights of chronological-order and trend-order to forecast the future performance of stock market indices.

Although these approaches show reasonable success for stock data forecasting, they have several drawbacks. On one hand, it is known that the length of intervals, as well as weighting factor, influence forecast accuracy in FTS, which leads to many models that are very sensitive of parameters settings in real world applications [154]. On the other hand, most models only depend on the trend indicated by the fuzzy logic group, but ignore the large influence of the short-term trend developed by neighbor series. In addition, few methods consider the stock volume, one of the most important metrics in stock analysis, which may also weaken the forecast accuracy, since stock volume is closely related with the trend.

In this chapter, the initial goal is to address the above weaknesses of existing FTS models. To this end, we first develop and deploy a novel evolutionary algorithm, Sine-Cosine adaptive Human Learning Optimization (SCHLO), to optimize the FTS and required parameters automatically and globally. Human Learning Optimization (HLO), which was presented recently by [190], is an evolutionary computation method, and has been successfully used to resolve optimization problems in different applications [30, 191, 192]. Thanks to the simple but effective structure of HLO, it does not need the gradient information in its optimization procedure, which is very suitable for the FTS optimization. Second, new weighted operators that consider the frequency based chronological order and stock volume are analyzed and integrated with SCHLO for optimization. Finally, a novel short-term trend repair operation is developed to complement the final forecasting process.

The rest of this chapter is organized as follows. In Section 3.2, we briefly review the basic concepts of a fuzzy time series forecasting model [37]. In Section 3.3, we introduce the concepts of standard HLO techniques proposed by Wang et al [190] and present the

proposed SCHLO. In Section 3.4, we propose a new FTS forecasting approach based on optimal partitions of intervals and hybrid weighting factors, obtained concurrently by SCHLO. In Section 3.5, we analyze the experimental results of the proposed FTS forecasting method and several state-of-the-art weighted fuzzy time series methods, when applied to forecasting several stock indices (Dow Jones Index (DJI), German Stock Index (DAX), Japanese Stock Index (NIKKEI) and Taiwan Stock Index (TAIEX)). Finally, the conclusions are discussed in Section 3.6.

3.2 Preliminaries

In this section we review some basic concepts, including fuzzy sets, fuzzy time series, fuzzy-trend logical relationship groups, and a fuzzy time series forecasting model.

3.2.1 Fuzzy Time Series

Song and Chissom [166–168] first introduced the concept of a fuzzy time series, based on the fuzzy set theory [208].

Definition 3.2.1. *[208] Let $U = \{u_1, u_2, ..., u_n\}$ be the universe of discourse. The fuzzy set A in the universe of discourse U can be defined as follows:*

$$A = f_A(u_1)/u_1 + f_A(u_2)/u_2 + ... + f_A(u_n)/u_n \tag{3.1}$$

where f_A is the membership function of the fuzzy set A, $f_A : U \rightarrow [0,1]$, and $f_A(u_i)$ denotes the degree of membership of u_i belonging to the fuzzy set A, with $f_A(u_i) \in [0,1]$ and $1 \leq i \leq n$.

Definition 3.2.2. *[167] Let $Y(t)(t = ..., 0, 1, 2, ...)$, a subset of R (set of real numbers) be the*

universe of discourse in which fuzzy sets $f_i(t)(i=1,2,...)$ are defined. If F(t) is a collection of $f_i(t)(i=1,2,...)$, then F(t) is called a fuzzy time series on Y(t)(t=...,0,1,2,...).

Definition 3.2.3. *[167] Let $F(t-1)=A_i$ and $F(t)=A_j$. The fuzzy logical relationship between $F(t-1)$ and $F(t)$ is referred by $A_i \rightarrow A_j$, where A_i is called the left-hand side and A_j is the right-hand side of the fuzzy logical relationship.*

Definition 3.2.4. *[37] Suppose there are fuzzy logical relationships with the same left-hand side A_i such that $A_i \rightarrow A_{j1}, A_i \rightarrow A_{j2}, ..., A_i \rightarrow A_{jn}$. These fuzzy logical relationships can be grouped into a fuzzy logical relationship group, shown as $A_i \rightarrow A_{j1}, A_{j2}, ..., A_{jn}$.*

Definition 3.2.5. *[154] Let $F(t-1)=A_i$ and $F(t)=A_j$, for $i, j=1,2,...,m$, which is denoted as $A_i \rightarrow A_j$. Then $k=j-i$ is the jump associated to this fuzzy logical relationship.*

3.2.2 Fuzzy Time Series Forecasting Model

Over the past decade, many fuzzy time-series models have been proposed by following Song and Chissom's Definitions [166–168]. Among these models, Chen [37] proposed a model by applying simplified arithmetic operations in forecasting algorithms, replacing the complicated max-min composition operation introduced by Song and Chissom, and thus is more effective. Since we will modify Chen's model to a new weighted FTS, we first illustrate its procedure, as follows:

Step 1 Partition the universe of discourse. First, according to the min and max values in the dataset, define the D_{min} and D_{max} variables and choose two arbitrary positive numbers, D_1 and D_2, to partition the universe of discourse, i.e., $U=\{D_{\min}-D_1, D_{\max}+D_2\}$. Then divide this universe into equal length intervals, i.e., $u_1, u_2, ..., u_m$.

Step 2 Define the fuzzy sets and fuzzify the historical data. Using the defined sub intervals in former step, the fuzzy sets are defined as Eq. 3.2 and each historical datum in time series is fuzzified its corresponding fuzzy set accordingly.

$$A_i = \frac{0}{u_1} + \frac{0}{u_2} + ... + \frac{0.5}{u_{i-1}} + \frac{1}{u_i} + \frac{0.5}{u_{i+1}} + ... + \frac{0}{u_m}, i = 1,2,3,...m. \tag{3.2}$$

Step 3 Establish fuzzy logic relationships (FLRs) and fuzzy logic relationship groups (FLRGs). FLRs are grouped based on the current states of the data according to Definition 3.2.4.

Step 4 Calculate the forecast values. Let $F(t) = A_i$. If there is only one fuzzy logical relationship in the sequence, and $A_i \rightarrow A_j$, then $F(t+1)$, the forecast output, equals A_j. If $A_i \rightarrow A_1, A_2, ..., A_k$, then $F(t+1)$ equals $A_1, A_2, ..., A_k$. If there are no fuzzy relationship groups, then the forecast output equals A_i.

Step 5 Defuzzify. If the forecast $F(t+1) = A_{i1}, A_{i2}, ..., A_{ik}$, the forecast values at time t+1 are calculated as

$$Final(t+1) = \frac{\sum_{j=1}^{k} M_{ij}}{k} \tag{3.3}$$

where $M_{i1}, M_{i2}, ..., M_{ij}$ are the defuzzified values of $A_{i1}, A_{i2}, ..., A_{ik}$, respectively, and $Final(t+1)$ is the final forecast.

3.3 Sine-Cosine Adaptive Human Learning Optimization Algorithm

Recently, the human learning optimization algorithm [190], inspired by human learning mechanisms, has been successfully applied to real world applications, in large part due to its excellent global search ability and robustness to various problems [30, 191, 192]. In this section, we first introduce the basic techniques of HLO, and then propose a sine-cosine adaptive mechanism to further improve the performance of HLO. In HLO, three learning operators, i.e., the random learning operator, individual learning operator, and social learning operation, are used to yield new candidates to search for the optima, which simulates the human learning process. The detailed processes can be found in [190]. The binary-coding framework is adopted in HLO, and consequently an individual in HLO is represented by a binary string as shown in Eq. 3.4,

$$x_i = \left[\begin{array}{ccccccc} x_{i1} & x_{i2} & \cdots & x_{ij} & \cdots & x_{iM} \end{array} \right], x_{ij} \in \{0,1\}, 1 \leq i \leq N, 1 \leq j \leq M \tag{3.4}$$

where x_i denotes the i-th individual, N is the size of population, and M is the dimension of the solution. Each bit of a binary string is initialized as "0" or "1" randomly, which stands for a basic element of the knowledge or skill that people want to learn and master. After generating N individuals, an initial population is obtained, shown in Eq. 3.5, for solving

problems.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1j} & \cdots & x_{1M} \\ x_{21} & x_{22} & \cdots & x_{2j} & \cdots & x_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ x_{i1} & x_{i2} & \cdots & x_{ij} & \cdots & x_{iM} \\ \vdots & \vdots & & \vdots & & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nj} & \cdots & x_{NM} \end{bmatrix} \tag{3.5}$$

3.3.1 Random Learning Operator

To emulate the phenomena of randomness in human learning, a simplified random learning operator was developed for HLO, shown in Eq. 3.6.

$$x_{ij} = Rand(0,1) = \begin{cases} 0, 0 \leq rand() \leq 0.5 \\ 1, else \end{cases} \tag{3.6}$$

3.3.2 Individual Learning Operator

An individual knowledge database (IKD) is used to store personal best experiences, used to mimic and implement the individual learning of humans, shown in Eqs. 3.7-3.8

$$IKD = \begin{bmatrix} ikd_1 \\ ikd_2 \\ \vdots \\ ikd_i \\ \vdots \\ ikd_N \end{bmatrix}, 1 \leq i \leq N \tag{3.7}$$

$$ikd_i = \begin{bmatrix} ikd_{i1} \\ ikd_{i2} \\ \vdots \\ ikd_{ip} \\ \vdots \\ ikd_{iL} \end{bmatrix} = \begin{bmatrix} ik_{i11} & ik_{i12} & \cdots & ik_{i1j} & \cdots & ik_{i1M} \\ ik_{i21} & ik_{i22} & \cdots & ik_{i2j} & \cdots & ik_{i2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ ik_{ip1} & ik_{ip2} & \cdots & ik_{ipj} & \cdots & ik_{ipM} \\ \vdots & \vdots & & \vdots & & \vdots \\ ik_{iL1} & ik_{iL2} & \cdots & ik_{iLj} & \cdots & ik_{iLM} \end{bmatrix}, 1 \leq p \leq L \tag{3.8}$$

where ikd_i denotes the IKD of person i, L is the pre-defined number of solutions saved in the IKD, and ikd_{ip} stands for the p-th best experience of person i. When HLO conducts individual learning, it generates new solutions based on the knowledge in the IKD, following Eq. 3.9

$$x_{ij} = ikd_{ipj} \tag{3.9}$$

3.3.3 Social Learning Operator

To emulate the process of learning from sharing experiences, the social knowledge data (SKD) is used to preserve the knowledge of the population, as shown in Eq. 3.10

$$SKD = \begin{bmatrix} skd_1 \\ skd_2 \\ \vdots \\ skd_q \\ \vdots \\ skd_H \end{bmatrix} = \begin{bmatrix} sk_{11} & sk_{12} & \cdots & sk_{1j} & \cdots & sk_{1M} \\ sk_{21} & sk_{22} & \cdots & sk_{2j} & \cdots & sk_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ sk_{q1} & sk_{q2} & \cdots & sk_{qj} & \cdots & sk_{qM} \\ \vdots & \vdots & & \vdots & & \vdots \\ sk_{H1} & sk_{H2} & \cdots & sk_{Hj} & \cdots & sk_{HM} \end{bmatrix}, 1 \leq q \leq H \tag{3.10}$$

where H is the size of the SKD and skd_q is the q-th solution in SKD. Based on the knowledge in the SKD, HLO can perform social learning, as shown in Eq. 3.11, to generate better solutions in the search process.

$$x_{ij} = sk_{qj} \tag{3.11}$$

In summary, HLO emulates three human learning operators (random, individual, and social) to yield new solutions and search for the optima. These learning operators are based on the knowledge stored in the individual knowledge data (IKD) and individual knowledge data (SKD), and can be further integrated and operated to improve learning as shown in

Eq. 3.12

$$x_{ij} = \begin{cases} Rand(0,1), 0 \leq rand() \leq pr \\ ik_{ipj}, pr < rand() \leq pi \\ sk_{qj}, else \end{cases} \tag{3.12}$$

where pr is the probability of random learning, and the values of $(pi - pr)$ and $(1 - pi)$ represents the probabilities of performing individual learning and social learning, respectively.

3.3.4 Sine-cosine Adaptive Mechanism

In this chapter, we observe that the individual and social learning parameter pi as well as random learning parameter pr are extraordinarily important, since they directly determine the balance between exploration and exploitation, and consequently influence the performance of the algorithm. However, it is not easy to set the optimal parameters of meta-heuristics, as they usually depend on the problems. Moreover, it is very hard to get rid of local optima in some types of complex optimization problems. Thus, it is very critical to determine when to increase (decrease) pi and pr for the purpose of increasing (decreasing) the exploration/exploitation ability. As a result, the monotonous linear adaptive mechanisms are not that effective. Therefore, we propose a new adaptive mechanism based on sine-cosine to dynamically strengthen the search efficiency and relieve the effort of the parameter setting, as follows

$$pr = pr_{\text{mid}} + pr_{\text{mid}} \times \cos\left(Ite + r\right) \tag{3.13}$$

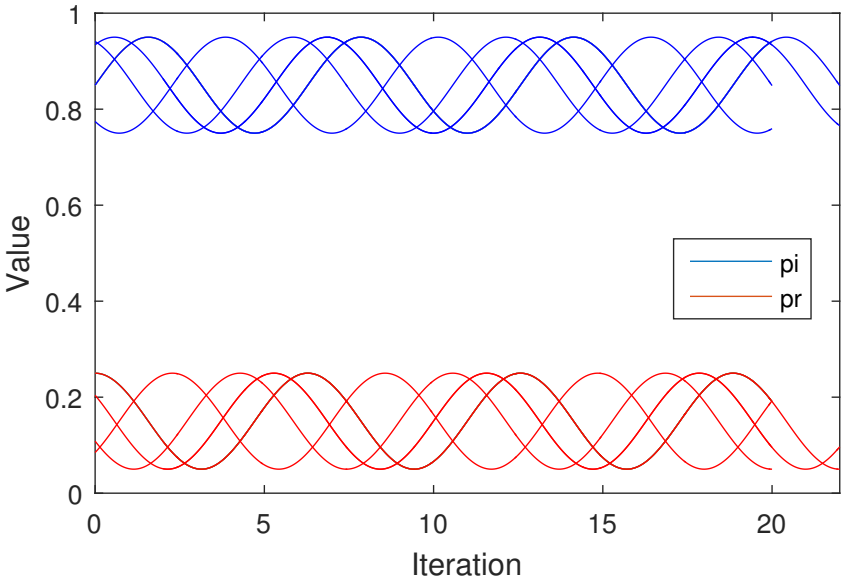


Figure 3.1 Adaptive mechanism for pi and pr

$$pi = pi_{\text{mid}} + pr_{\text{mid}} \times \sin(Ite + r) \tag{3.14}$$

$$r = \begin{cases} 0, 0 \leq rand() \leq 0.5 \\ 1, else \end{cases} \tag{3.15}$$

where pr_{mid} and pi_{mid} are the middle point of fluctuation of pr and pi, respectively, and Ite is the current iteration number. Fig 1 shows the change of the parameters. In this way, various combinations of pr and pi are generated, so that the SCHLO is able to intelligently handle the complex optimization cases.

3.4 A Novel Weighted Fuzzy Time Series Method Based on SCHLO

In this section we propose a novel weighted FTS forecasting method based on SCHLO (WFTS-SCHLO). Specifically, we first propose several modifications for the basic FTS method, by introducing three types of weights. These weights not only consider the frequency based chronological order, but also take the stock volume into consideration. In addition, based on our observation that there is a big influence of the short-term trend developed by neighbor series, we further propose a novel short-term trend repair operation to complement the forecasting process. Then, SCHLO is utilized to find the optimal partitions of intervals in the universe of discourse and the optimal weight factors simultaneously, based on the historical training data. Next, using the optimal settings obtained by SCHLO, the proposed model fuzzifies the historical data on each trading day into fuzzy sets, and establishes fuzzy logic relationship groups. Finally, based on the fuzzified historical testing data, the model performs defuzzification and uses the obtained optimal weight factors to calculate the forecast outputs.

3.4.1 The Proposed Weighted FTS Model

In our model, the trapezoidal fuzzy numbers [208] with midpoints of intervals [37] are used to analyze and derive the forecast values.

Definition 3.4.1. *A fuzzy number A is said to be a trapezoidal fuzzy number,* $A=(a,b,c,d)$,

if its membership function has the following form:

$$\mu_{\tilde{A}}(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \frac{d-x}{d-c}, & c \le x \le d \\ 0, & x > d \end{cases} \tag{3.16}$$

Definition 3.4.2. *Let $A = (a_1, b_1, c_1, d_1)$ and $B = (a_2, b_2, c_2, d_2)$ be trapezoidal fuzzy numbers, and let λ be a real number. Then,*

$$A \oplus B = (a_1 + a_2, b_1 + b_2, c_1 + c_2, d_1 + d_2) \tag{3.17}$$

$$\lambda A = \begin{cases} (\lambda a_1, \lambda b_1, \lambda c_1, \lambda d_1) \ \lambda \ge 0 \\ (\lambda d_1, \lambda c_1, \lambda b_1, \lambda a_1) \ \lambda \le 0 \end{cases} \tag{3.18}$$

Assume there are m intervals, which are $u_1 = [d_1, d_2]$, $u_2 = [d_2, d_3]$, ... , $u_{m-1} = [d_{m-1}, d_{m-2}]$, $u_m = [d_m, d_{m+1}]$. The fuzzification process is as shown in Eq. 3.19 and the standard FLRs and FLRGs are determined according to Definitions 3.2.3 and 3.2.4

$$\text{fuzzify}(F(t)) = A_i \ if \ u(A_i) = \max[\ u(A_z)] \ for \ all \ z \tag{3.19}$$

where where $z = F(1), F(2), ..., F(t)$ is the datum at time t; and $u(A_z)$ is the degree of membership of $F(t)$ under A_z.

Following the above preparations of FTS, we next turn to the proposed weighting rules that consider the frequency based chronological order and the stock volume. Generally,

in our model, the forecast of stock is based on the trend of FTS in two aspects. One is pattern trend (PT), and the other is long-term trend (LT). Specifically, for PT, suppose we have observed the stock data $F(t) = s_t$ for $t = 1, 2, ..., N$, and they are all assigned to a fuzzified number A_i (e.g., $fuzzify(F(N)) = A_n$) Our task is to predict $F(N+1)$. After prediction, we will check if the same fuzzified number exists in the historical data, if it is equal to A_n, and where the trend continues afterwards. As for LT, it means the model will not focus on $F(N)$ but on the whole history data since as economy, the stock has its own cycle [16]. The long-term trend, LT, will now focus on $F(N)$, but instead on the entirety of the historical data, as, similar to the economy, stocks have their own cycle. Thus, we scan all the $FLRs$ and determine what the current cycle is, and what follows. Since the frequency, chronological order, and stock volume all have a large influence over the trend in stock time series data, we propose the following weight factors for getting an accurate PT and LT.

3.4.1.1 PT Weight Factor

Let $F(t_0) \rightarrow A_p$ and $F(t_0+1) \rightarrow A_q$, with $V(t_0)$ and $V(t_0+1)$, where V stands for stock volume, have three relationships, $t_0 \rightarrow t_0+1$, $V(t_0) \rightarrow V(t_0+1)$ and $A_p \rightarrow A_q$. The weight associated with this FLR, based on the two data points, is defined as follows:

$$w_d = V(t_0) * V(t_0+1) * sqrt[t_0 + (t_0+1)] \tag{3.20}$$

Considering there may be lots of data in a logical relationship, $A_p \rightarrow A_q$, the total weight associated with this FLR is calculated as

$$w_r = \sum_j w_d \tag{3.21}$$

where j is the number of data combinations that fall into the same FLR. On the other side, A_p can also constitute many $FLRs$, namely $A_p \rightarrow A_{k1}$, $A_p \rightarrow A_{k2}$,..., $A_p \rightarrow A_{ko}$, which forms PT. Therefore, we have the PT weight as:

$$w_{PT}^{k} = \frac{w_r^k}{\sum_o w_r} \tag{3.22}$$

where o is the number of $FLRs$ A_p constitutes.

3.4.1.2 LT Weight Factor

Inspired by the jump theory in [154], here we use a jump metric together with the proposed w_d to measure the LT. Let $\delta = p - q$ be the jump between a FLR. Obviously, δ can be positive, negative, and zero (representing a rise, fall, and square position in stock context). Similar to PT, the total weight associated with δ is calculated as

$$w_{\Delta} = \sum_c w_d \tag{3.23}$$

where c is the number of data combinations formed from the same jump. Accordingly, The LT weight is calculated as

$$w_{LT}^{k} = \frac{w_{\Delta}^k}{\sum_b w_{\Delta}} \tag{3.24}$$

where b is the number of deltas.

Based on the proposed PT and LT weight factors, we have a rough picture of the current trend and are able to do the forecast. However, we find it is not enough to only take PT and LT into account, since these two metrics are not closely related to the short-term trend. In other words, although they have considered the chronological order, there is still the possibility that the movement of a stock is largely influenced by a short period, only within

a few days (e.g, the stock may continue to drop after a nose dive caused by "black swan event"). Therefore, the short-term trend should be carefully addressed. To this end, we propose a short-term trend repair operation to further improve the forecast performance.

3.4.1.3 Short-term Trend Repair Operation

Let s_l be the length of a short-term period we observed before the last observed data. Thus, we have the sequence $F(t-s_l)$, $F(t-s_l+1)$,..., $F(t-1)$, $F(t)$ and the corresponding fuzzified numbers A. Here, we use a differential strategy to recognize the short-term trend as follows.

$$\begin{aligned} &Diff(s_l) = A_{(t-s_l+1)} - A_{(t-s_l)} \\ &Diff(s_{l-1}) = A_{(t-s_l+2)} - A_{(t-s_l+1)} \\ &..... \\ &Diff(2) = A_{(t-1)} - A_{(t-2)} \\ &Diff(1) = A_{(t)} - A_{(t-1)} \end{aligned} \tag{3.25}$$

Then, we have a differential sequence $DS = [Diff(s_l), Diff(s_l-1), ..., Diff(2), Diff(1)]$ for this short-term period, and to simulate three closing quotations (rise, fall, and square position), DS is simplified to the following form:

$$DS_c = \begin{bmatrix} cmp\big(Diff(s_l),0\big), cmp\big(Diff(s_l-1),0\big) \\ ,\ldots, cmp\big(Diff(2),0\big), cmp\big(Diff(1),0\big) \end{bmatrix} \tag{3.26}$$

where cmp is the compare operation with 0, and it only returns -1, 0 and 1. The short-term trend is now represented by DS_c, and the model will learn from historical data to output

the result of this trend. Using a similar step, we have:

$$\begin{aligned} DS_{c1} &=> cmp\left(Diff(0)_1, 0\right) \\ DS_{c2} &=> cmp\left(Diff(0)_2, 0\right) \\ &... \\ DS_{cn} &=> cmp\left(Diff(0)_n, 0\right) \end{aligned} \tag{3.27}$$

where $Diff(0)$ is the differential result between the fuzzified numbers A of next days data and the last observed data in sequence. In other words, it is the output of the current differential sequence. Since some DSs in the historical data may be the same, the model groups all the same DSs by simply adding the corresponding output, and finally the model gets the comprehensive trend results.

3.4.2 Fuzzy Forecast Outputs

In the forecasting step, we propose to use a linear combination of the above weight strategies and repair operation, as shown in Eq. 3.28

$$\widetilde{A}_{t+1} = \alpha\widetilde{A}_{PT} + \beta\widetilde{A}_{LT} + \gamma\widetilde{A}_{RO} \ (\alpha+\beta+\gamma=1) \tag{3.28}$$

where

$$\widetilde{A}_{PT} = w^1_{PT}A_{k1} \oplus w^2_{PT}A_{k2} .... \oplus w^o_{PT}A_{ko} \tag{3.29}$$

$$\widetilde{A}_{LT} = w^1_{LT}A_{t+\Delta 1} \oplus w^2_{LT}A_{t+\Delta 2} .... \oplus w^b_{LT}A_{t+\Delta b} \tag{3.30}$$

$$\widetilde{A}_{RO} = A_{t+Diff} \tag{3.31}$$

Here, t means the time stamp of last observed data.

3.4.3 Algorithm of the weighted fuzzy-trend time series method based on SCHLO

In this section, SCHLO is utilized to find the optimal partitions of intervals in the universe of discourse and the optimal weight factors simultaneously. The proposed weighted FTS forecasting method proceeds as follows:

Step 1 Let the universe of discourse U of the main factor be $[D_{min} - D_1, D_{max} + D_2]$, where D_{max} and D_{min} are the maximum value and the minimum value of the historical training data of the main factor, respectively; D_1 and D_2 are two proper positive real values to let the universe of discourse U cover the noise of the testing data of the main factor.

Step 2 Specify the control parameters of SCHLO, such as population size, pr_{mid}, and pi_{mid}. Randomly initialize the population, including interval length m and three weights α, β, γ for optimization. The fitness values are calculated and initial $IKDs$ and SKD are generated. Each individual does the following sub-steps to calculate the fitness value:

(a) Based on the generated interval length, define the fuzzy sets on U and fuzzify the historical data as Eq. 3.19.

(b) Establish $FLRs$ from the fuzzy time series, and $FLRs$ and $FLRGs$ according to Definitions 3.2.3 and 3.2.4.

(c) Defuzzify and calculate the forecast outputs. Specifically, the algorithm first calculates PT and LT weight factors and derives the $\widetilde{A}_{PT}$ and $\widetilde{A}_{LT}$ according

to Eq. 3.29 and Eq. 3.30, respectively. Then, short-term trend repair operation is performed to get $\widetilde{A}_{RO}$. Based on the obtained $\widetilde{A}_{PT}$, $\widetilde{A}_{LT}$, and $\widetilde{A}_{Ro}$ SCHLO generates α, β, γ. The model can calculate $\widetilde{A}_{t+1}$ according to Eq. 3.28. Finally, the one-step forecast will be the middle point of the interval-valued mean of $\widetilde{A}_{t+1}$.

(d) Calculate the fitness value, i.e. the root mean square error (RMSE), as follows:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{t_{\max}} (F_t - R_t)^2}{t_{\max}}} \tag{3.32}$$

where t_{max} denotes the number of trading days of the historical training data, F_t, denotes the forecast value of the validation datum on trading day t, and R_t is the actual value of the historical validation datum on trading day t.

Step 3 Improvise new individuals. The new solutions are yielded by performing the adaptive pr and pi rule and the learning operators shown in Eqs. 3.12-3.15.

Step 4 Calculate the fitness of new candidates. The fitness value of new individuals is computed as steps 2.1-2.4.

Step 5 Update the $IKDs$ and SKD according to the fitness of the new individuals.

Step 6 Check the termination criterion. If the termination criterion is not met, steps 3 — 5 will be repeated to keep searching for the optimal parameters, otherwise go to the next step.

Step 7 Based on the obtained optimal parameters, the model performs steps 2.1 — 2.4 to calculate the final forecast on the test datasets.

3.5 Real-World Experiments and Results

3.5.1 Stock Market Data and Experimental Settings

To evaluate the forecast accuracy of the proposed model, we analyze data sets from four major world trading markets (DJI, DAX, NIKKEI and TAIEX) obtained from Yahoo Finance (https://finance.yahoo.com/). For every stock market index, we work with daily values collected over three years, from January 2014 to December 2016. Data from 2014 and 2015 are used for training and validation, and data from 2016 is for testing. Similar to mainstream financial forecast methods, a daily rolling horizon strategy is applied. Specifically, this strategy preserves the size of each training set by eliminating the first observation and by including the last observed quotes. Thus, the FLRs and FLGs, as well as all weighting factors, are then re-evaluated. To compare the performance of the WFTS-SCHLO, four state-of-the-art fuzzy time series methods with various strategies [42, 154, 193, 207] as well as a standard HLO based WFTS (WFTS-HLO) are used for comparison. The performance is measured by RMSE, Eq. 3.32. All of the parameters for the baseline models are set to those reported in the initial publications. For our model, WFTS-SCHLO, the population size is 30, pi_{mid} is set to 0.1, and pr_{mid} is set to 0.9. The number of iterations is fixed at 300. All implementations are in Python, and programs were run on a PC with an Intel Core CPU i7-4790 @3.60GHz and 16 GB RAM.

3.5.2 Results and Analysis

Fig. 2 shows the time series of the daily observations and the daily forecasts provided by WTFS-SCHLO for the year 2016. Table 3.1 shows that WTFS-SCHLO consistently outperforms the baseline methods, having the lowest RMSE for its predictions in every market,

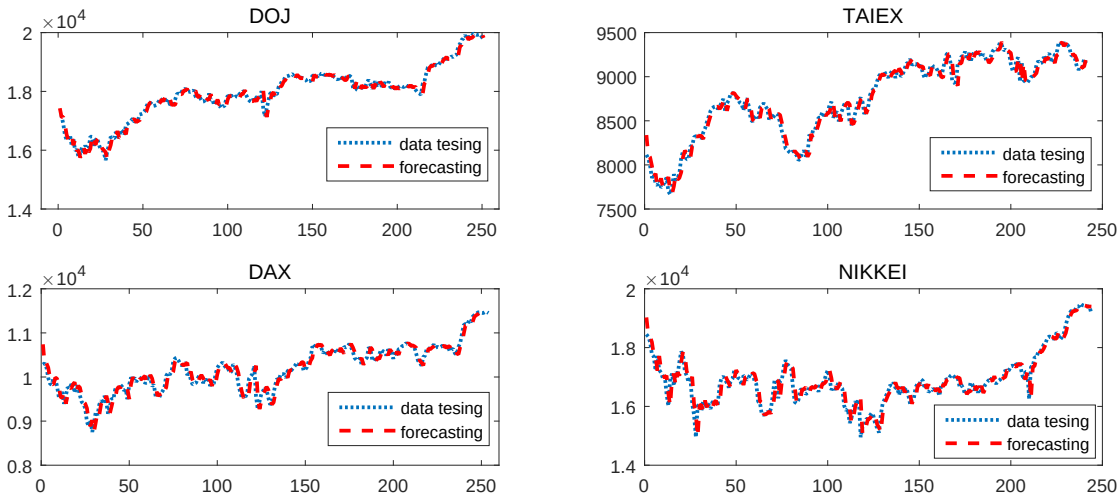


Figure 3.2 Time series of the daily observations and the daily forecasts provided by WTFS-SCHLO for the different markets in year 2016

which means our model has the best forecasting accuracy. In order to analyze the forecast error achieved by each method, a statistical analysis was carried out. Table 3.2 shows the adjusted p-values of the pairwise statistical comparisons of the RMSE forecast errors, performed through the paired t-test at Holm's adjustment. The statistical analysis reveals that there are significant differences between our model and the other forecasting methods, in favor of the proposed model.

Table 3.1 The Forecast performance (RMSE) of four stock markets

Markets \ Models	Yu model [207]	Cheng model [42]	Wang model [193]	Rubio model [154]	WTFS-HLO	WTFS-SCHLO
DOJ	174.81	172.51	468.36	141.59	139.06	136.53
DAX	137.23	138.74	335.29	132.92	130.09	127.53
NIKKEI	267.37	262.04	514.18	284.77	259.40	252.29
TAIEX	82.17	82.84	199.35	74.54	72.91	69.46

Table 3.2 Adjusted p-values of the pairwise comparison of the RMSE forecast errors between WTFS-SCHLO with different models

Markets \ Models	Yu model [207]	Cheng model [42]	Wang model [193]	Rubio model [154]	WTFS-HLO
DOJ	6.41e-3	0.002	4.54e-12	0.021	0.038
DAX	0.008	1.84e-3	5.07e-10	0.017	0.029
NIKKEI	0.012	0.019	7.30e-11	5.25e-3	0.034
TAIEX	0.026	0.015	4.81e-10	0.031	0.042

3.6 Conclusions

In this chapter, an intelligent weighted fuzzy time series model is presented for financial market forecasting. The model utilizes a sine-cosine adaptive human learning optimization algorithm to search for the optimal parameters, improving forecasting performance. New weighted operators that consider frequency based chronological order and stock volume are analyzed and integrated with SCHLO to determine the effective universe discourse and intervals. Furthermore, a novel short-term trend repair operation is developed to complement the final forecasting process. Finally, the proposed model is applied to data from four major world trading markets, the Dow Jones Index, the German Stock Index, the Japanese Stock Index, and Taiwan Stock Index. The experimental results demonstrate that the proposed model consistently outperforms its counterparts, in terms of accuracy. The easy implementation and effective forecasting performance suggest our proposed model could be a favorable market application prospect.

CHAPTER

4

AN INTELLIGENT AND HYBRID WEIGHTED FUZZY TIME SERIES MODEL BASED ON EMPIRICAL MODE DECOMPOSITION FOR FINANCIAL MARKETS FORECASTING

4.1 Introduction

Previous research has shown that the volatility of financial market is often influenced by different investment behaviors between various institutions and/or individual investors [212] such as psychological differences [137], investment experiences [132], short and long-term expectations [71]. and even different interpretation of government policies [12]. Therefore, although the observation of the financial market is a single time series, it can be a mixture of several sub-series generated by different institutions or individuals. Thus, creating forecasting models based only on a single observed series is not enough and may lead to bottlenecks. On the other hand, most models only depend on the historical trend indicated by the fuzzy logic group but ignore the big influence of the volatility direction indicated by neighborhood. In addition, many existing models are very sensitive to parameters settings such as the length of interval and thus forecast accuracy will be much reduced with inappropriate parameter settings in various real-world applications [41].

Therefore, to address the above weakness of existing FTS models, a novel Intelligent and Hybrid Weighted Fuzzy (IHWF) time series model is proposed in this chapter. We first adopt Empirical Mode Decomposition (EMD) [79] to extract the possible modes of various kinds of institutional and individual investors embedded in an observed time series. Then, we propose a New Weighted FTS (NWF) that considers both chronological-order based frequency and the neighborhood volatility direction (NVD). The NWF will perform the proper forecasting according to various sub-series extracted by EMD and all the predicted results are aggregated as the final forecast. Finally, to optimize the NWF automatically and globally, a novel evolutionary algorithm, Adaptive Sine-Cosine Human Learning Optimization (ASCHLO), is developed and deployed. To the best of our knowledge, no existing FTS models decompose the financial time series to analyze the inner forecast

process. The solution framework designed in this research is unique and significantly improves the forecast accuracy.

4.2 Preliminaries

This section briefly reviews the relevant studies on empirical mode decomposition, fuzzy sets and fuzzy time series.

4.2.1 Empirical Mode Decomposition

Empirical Mode Decomposition, introduced by Huang et al. [79] is an adaptive decomposition method for nonlinear and non-stationary time series data using the Hilbert-Huang transform (HHT). The EMD is based on the simple assumption that any signal consists of different components, thus any data may have different coexisting modes of oscillations at the same time. Given this, the time series can be decomposed into a group of zero mean and quasi-periodic signals. Each component is called an intrinsic mode function (IMF), and this is added to a residue series that represents the different scales of the original time series. Together, these form the adaptive and physical basis of the data as given in Eq. 4.1:

$$x(t) = \sum_{i=1}^{m} c_i(t) + r(t) \tag{4.1}$$

where $x(t)$ is the original time series data, each $c_i(t)$ represents the ith IMF, and $r(t)$ is the residual component. The detailed procedure of EMD is briefly explained in Algorithm 4.1.

The EMD components (IMFs and residue signal) represent different characteristics of the original signal. Fig. 1, an example of the decomposed TAIEX from 2004, demonstrates this difference. As can be seen from the figure, each component has different characteristics

Algorithm 4.1 Empirical Mode Decomposition

1: Identify all extrema (local maxima $x_{max}[n]$ and minima $x_{min}[n]$) of the time series $x(t)$
2: Connect $x_{max}[n]$ by a cubic spline to define the upper envelope, $U(t)$, and $x_{min}[n]$ by a second cubic spline to define the lower envelope, $L(t)$
3: Calculate mean of the envelope $m(t)$, i.e., $m(t) = (U(t) + L(t))/2$
4: Compute the difference to obtain an IMF candidate $h(t) = x(t) - m(t)$.
5: Check if $h(t)$ satisfies the condition of an IMF property, i.e., the envelopes are symmetric with respect to zero mean under certain criteria
6: if $h(t)$ is an IMF, take $h(t)$ as the ith IMF, and replace $x(t)$ with the residual $R_i(t) = x(t) - h(t)$. Otherwise, repeat Steps 2 to 5 until a valid IMF function $h(t) is found$.

which implies that the market is driven by different investment modes at same time. For example, IMF1 may represent High-Frequency Trading mode while residual could represent a Long-Term Investment mode.

4.2.2 Fuzzy Time Series

Song and Chissom [166–168] first introduced the concept of a fuzzy time series, based on the fuzzy set theory [208].

Definition 4.2.1. *[208] Let $U = \{u_1, u_2, ..., u_n\}$ be the universe of discourse. The fuzzy set A in the universe of discourse U can be defined as follows:*

$$A = f_A(u_1)/u_1 + f_A(u_2)/u_2 + ... + f_A(u_n)/u_n \tag{4.2}$$

where f_A is the membership function of the fuzzy set A, $f_A : U \rightarrow [0,1]$, and $f_A(u_i)$ denotes the degree of membership of u_i belonging to the fuzzy set A, with $f_A(u_i) \in [0,1]$ and $1 \leq i \leq n$.

Definition 4.2.2. *[167] Let $Y(t)(t = ..., 0, 1, 2, ...)$, a subset of R (set of real numbers) be the universe of discourse in which fuzzy sets $f_i(t)(i = 1, 2, ...)$ are defined. If F(t) is a collection of $f_i(t)(i = 1, 2, ...)$, then F(t) is called a fuzzy time series on Y(t)(t=...,0,1,2,...).*

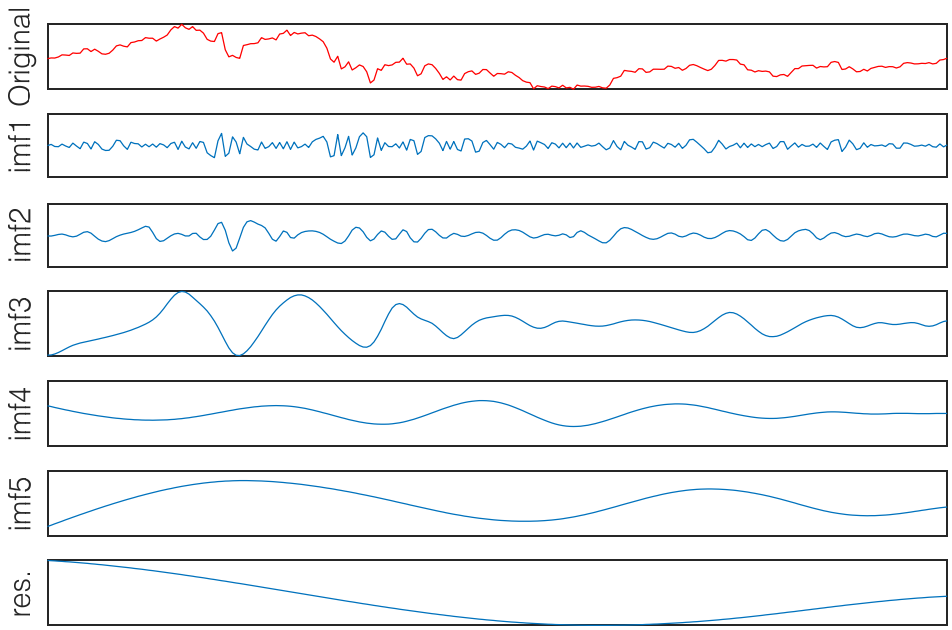


Figure 4.1 The IMFs and residual of TAIEX from 2004 decomposed by EMD

Definition 4.2.3. *[167] Let $F(t-1)=A_i$ and $F(t)=A_j$. The fuzzy logical relationship between $F(t-1)$ and $F(t)$ is referred by $A_i \rightarrow A_j$, where A_i is called the left-hand side and A_j is the right-hand side of the fuzzy logical relationship.*

Definition 4.2.4. *[37] Suppose there are fuzzy logical relationships with the same left-hand side A_i such that $A_i \rightarrow A_{j1}, A_i \rightarrow A_{j2}, ..., A_i \rightarrow A_{jn}$. These fuzzy logical relationships can be grouped into a fuzzy logical relationship group, shown as $A_i \rightarrow A_{j1}, A_{j2}, ..., A_{jn}$.*

Definition 4.2.5. *[154] Let $F(t-1)=A_i$ and $F(t)=A_j$, for $i, j = 1, 2, ..., m$, which is denoted as $A_i \rightarrow A_j$. Then $k = j - i$ is the jump associated to this fuzzy logical relationship.*

4.3 Adaptive Sine-cosine Human Learning Optimization Algorithm

Human Learning Optimization (HLO), which was presented recently by [190], is an evolutionary computation method inspired by the human learning mechanisms, and has been successfully used to resolve optimization problems in different applications [30, 191, 192]. Due to the simple but effective structure of HLO, it does not need the gradient information in its optimization procedure, which is very suitable for the FTS optimization. In this section, we first briefly introduce the basic techniques of HLO and then propose the adaptive sine-cosine mechanism to further improve the performance of HLO.

In HLO, three learning operators - a random learning operator, an individual learning operator, and a social learning operator - are used to yield new candidates to search for the optima. These attempt to mimic human learning processes, as described in more detail in [190]. The learning operators are based on the knowledge stored in the individual knowledge data (IKD) and social knowledge data (SKD), and can be further integrated and operated to improve learning as shown in Eq. 4.3

$$x_{ij} = \begin{cases} Rand(0,1), 0 \leq rand() \leq pr \\ ik_{pj}, pr < rand() \leq pi \\ sk_{qj}, else \end{cases} \tag{4.3}$$

where ik and sk are individuals in IKD and SKD. pr is the probability of random learning, and the values of $(pi-pr)$ and $(1-pi)$ represents the probabilities of performing individual learning and social learning, respectively.

Adaptive Sine-cosine Mechanism Based on our observation, the parameters pi and

pr are extremely important. Since the search balance between the exploration and exploitation are directly determined by these two parameters, they consequently influence the performance of the algorithm. However, in basic HLO, the value of the two parameters are set with fixed value by trial and error which may have a big impact on performance, especially for various types of complex optimization problems. Therefore, we propose a new adaptive mechanism based on sine-cosine to dynamically strengthen the search efficiency and relieve the effort of the parameter setting as Eqs. (4.4-4.5),

$$pr = pr_{\text{mid}} + pr_{\text{mid}} \times \sin(Ite) \tag{4.4}$$

$$pi = pi_{\text{mid}} + pr_{\text{mid}} \times \cos(Ite + 1) \tag{4.5}$$

where pr_{mid} and pi_{mid} are the mid-points of fluctuation of pr and pi, respectively, and Ite is the current iteration number. In this way, various combinations of pr and pi are generated, so that ASCHLO is able to intelligently handle complex optimization cases.

4.4 The Intelligent and Hybrid Weighted Fuzzy Time Series Model Based on ASCHLO

We propose a novel intelligent and hybrid weighted fuzzy time series model (IHWF) based on ASCHLO. We first apply the EMD technique to decompose the original financial time series into several sub-series and then introduce a concept of the neighborhood volatility direction (NVD). By integrating two types of weights, we modify and improve the basic FTS model. These weights not only consider the chronological-order based frequency but also take

NVD into consideration. Subsequently, we use this weighted FTS to develop a forecasting model for each sub-series, and to make the corresponding prediction for possible modes extracted from original observations. In this process, ASCHLO is globally utilized to optimize the associated parameters such as the partition of intervals, the weight factors both in NVD and forecast process. Finally, the final forecasting result of the original time series is calculated by aggregating all the forecasting results of extracted modes obtained through appropriate IHWF model.

4.4.1 The Proposed Weighted FTS Model

In our model, the trapezoidal fuzzy numbers [208] with midpoints of intervals [37] are used to analyze and derive the forecast values.

Definition 4.4.1. *A fuzzy number A is said to be a trapezoidal fuzzy number, $A=(a,b,c,d)$, if its membership function has the following form:*

$$\mu_{\widetilde{A}}(x)=\begin{cases} 0, & x<a \\ \frac{x-a}{b-a}, & a\le x\le b \\ 1, & b\le x\le c \\ \frac{d-x}{d-c}, & c\le x\le d \\ 0, & x>d \end{cases} \tag{4.6}$$

Definition 4.4.2. *Let $A=(a_1,b_1,c_1,d_1)$ and $B=(a_2,b_2,c_2,d_2)$ be trapezoidal fuzzy numbers, and let λ be a real number. Then,*

$$A\oplus B=(a_1+a_2,b_1+b_2,c_1+c_2,d_1+d_2) \tag{4.7}$$

$$\lambda A = \begin{cases} (\lambda a_1, \lambda b_1, \lambda c_1, \lambda d_1) \ \lambda \geq 0 \\ (\lambda d_1, \lambda c_1, \lambda b_1, \lambda a_1) \ \lambda \leq 0 \end{cases} \tag{4.8}$$

Assume there are m intervals, which are $u_1 = [d_1, d_2]$, $u_2 = [d_2, d_3]$, ..., $u_{m-1} = [d_{m-1}, d_{m-2}]$ and $u_m = [d_m, d_{m+1}]$. The fuzzification process is as shown in Eq. 4.9 and the standard $FLRs$ and $FLRGs$ are determined according to Definitions 4.2.3 and 4.2.4

$$\text{fuzzify}(F(t)) = A_i \ if \ u(A_i) = \max[\ u(A_z)] \ for \ all \ z \tag{4.9}$$

where $z = F(1), F(2), ..., F(t)$ is the datum at time t; and $u(A_z)$ is the degree of membership of $F(t)$ under A_z.

Following the above background on FTS, we introduce the proposed weighting rules considering the neighborhood volatility direction and chronological-order based frequency. In our model, all forecasts are based on two aspects of FTS: one is neighborhood volatility direction (NVD) , and the other is long-term trend (LT) analysis.

4.4.1.1 Neighborhood Volatility Direction

Suppose we have observed stock data $F(t) = s_t$ for $t = 1, 2, ..., N$ and they are all assigned to a fuzzified number A_i (e.g., $fuzzify(F(N)) = A_n$), let $F(t_0) \rightarrow A_p$ and $F(t_0+1) \rightarrow A_q$. When predicting $F(N+1)$, traditional fuzzy logical groups $(FLRs)$ only consider the historical trend based on A_n. For example, if A_n constitutes several $FLRs$ in a historical time series, i.e., $A_n \rightarrow A_{k1}, A_n \rightarrow A_{k2}, ..., A_n \rightarrow A_{kp}$, the output of the forecast will be either the mean or the median of these intervals. However, this inference has two deficiencies which may negatively impact the forecast performance: firstly, if A_n does not belong to any $FLRs$ (e.g., the index reaches record highs or lows), this inference will be invalid; secondly, the

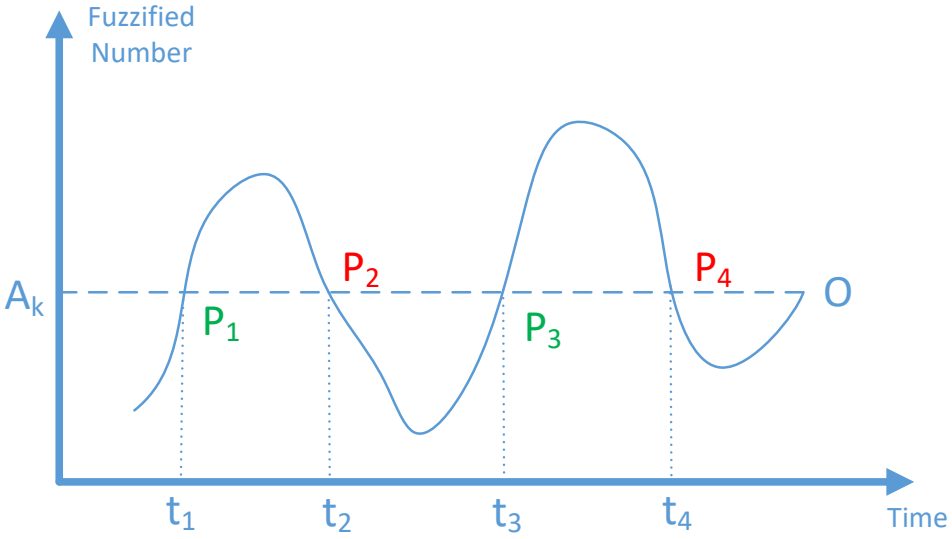


Figure 4.2 Illustration of NVD

model ignores short-term trends at the current point. To illustrate this, consider the time series in Fig. 2 - we now have $F(N)$ (labeled as point O) and need to forecast $F(N+1)$. Assume there are four historical points (i.e., P_1, P_2, P_3, P_4 in Fig. 2), which have the same fuzzified number as O - traditional methods will then take all four points into consideration. However, financial markets usually have their own inertial trend [23] and from Fig. 2, O is currently in an up-trend, indicated by its neighborhood. So it is not appropriate to take P_2 and P_4, which could have negative impact, into account. Therefore, to address these deficiencies, we propose the concept of neighborhood volatility direction.

Definition 4.4.3. *Let $F(t) = A_{k1}, F(t-1) = A_{k2}, ..., F(t-n) = A_{kn}$, where A_{kn} is fuzzified number and n is a small positive integer. The neighborhood volatility direction for $F(t)$ is denoted as $NVD(t) = [A_{k1}, A_{k2}, ..., A_{kn}]$.*

Definition 4.4.4. *Suppose i, j are two data points in a certain time series, let $NVD(i) = [A_{k1}, A_{k2}, ..., A_{kn}]$ and $NVD(j) = [A'_{k1}, A'_{k2}, ..., A'_{kn}]$. The NVD distance ($D_{NVD}$) is defined as*

follows:

$$D_{NVD} = |A_{k1} - A'_{k1}| + \ldots + |A_{kn} - A'_{kn}| \tag{4.10}$$

In the proposed method, the top-N historical points with the shortest distance will be determined and weighted for the current forecast by D_{NVD}. The weight associated with these points is chronological-order, as follows:

$$w^k_{\mathrm{NVD}} = \frac{t_k}{\sum_n t} \tag{4.11}$$

where t_k means the time stamp of kth point selected.

4.4.1.2 Long-Term Trend

Inspired by jump theory in [154], here we also use jump metric together with chronological-order to measure the long-term trend:

Definition 4.4.5. *[154] Let $F(t-1) = A_i$ and $F(t) = A_j$, for $i, j = 1, 2, 3\ldots$, which is denoted as $A_i \rightarrow A_j$. Then $k = j - i$ is the jump associated to this fuzzy logical relationship.*

Let Δ be the jump between a FLR. Obviously, Δ can be negative, zero and positive (represents rise, square and fall in stock context). Similar to NVD, the total weight associated with Δ is calculated as:

$$w_\Delta = \sum\nolimits_{\mathrm{m}} t \tag{4.12}$$

where m is the number of data combinations constitute the same jump. Accordingly, the LT weight is calculated as:

$$w^q_{LT} = \frac{w^q_\Delta}{\sum_p w_\Delta} \tag{4.13}$$

where p is the number of jumps in time series.

4.4.2 Fuzzy Forecast Outputs

In the forecasting step, for each IMF, we propose to use a linear combination of the above weight strategies as shown in Eq. 4.14

$$\widetilde{A}_{t+1} = \alpha \widetilde{A}_{NVD} + \beta \widetilde{A}_{LT} \ (\alpha + \beta = 1) \tag{4.14}$$

where

$$\widetilde{A}_{NVD} = w^1_{NVD} A_{k1} \oplus w^2_{NVD} A_{k2} \oplus .... \oplus w^o_{NVD} A_{kn} \tag{4.15}$$

$$\widetilde{A}_{LT} = w^1_{LT} A_{t+\Delta 1} \oplus w^2_{LT} A_{t+\Delta 2} \oplus .... \oplus w^q_{LT} A_{t+\Delta p} \tag{4.16}$$

and where t is the time stamp of most recently observed data. Similar as IMFs, the final forecast is calculated as Eq. 4.17

$$\widetilde{F}_{t+1} = \gamma \widetilde{F}_{aggr} + (1-\gamma)\widetilde{F}_{LT} \tag{4.17}$$

where $\widetilde{F}_{aggr}$ is aggregated results of all IMFs and $\widetilde{F}_{LT}$ is calculated similar as $\widetilde{A}_{LT}$ based on actual observed data.

4.4.3 Algorithm for IHWF based on ASCHLO

In this section, ASCHLO is globally utilized to find the optimal partitions of intervals in the universe of discourse and the optimal weight factors including the number of top-N points selected (NTP), NVD weights and LT weights simultaneously. The proposed model is now presented as follows:

Step 1: Apply EMD to decompose the original time series $x(t)$ to IMF components $c_i(t)(i =$

1, 2, 3, ..., m) and one residual component $r(t)$.

Step 2: For each IMF and residual component, let the universe of discourse U be $[D_{min} - D_1, D_{max} + D_2]$, where D_{max} and D_{min} are the maximum and the minimum values of the historical training data; D_1 and D_2 are two positive real values to allow the universe of discourse U to cover the noise of the testing data.

Step 3: Specify the control parameters of ASCHLO, such as population size, pr_{mid}, and pi_{mid}. Randomly initialize the population, including interval length l and three weights, i.e., NTP, α, β and γ for optimization. The fitness values are calculated and initial $IKDs$ and SKD are generated. Each individual performs the following sub-steps to calculate the fitness value:

(a) Based on the generated interval length, define the fuzzy sets on U and fuzzify the historical data as Eq. 4.9.

(b) Establish $FLRs$ from the fuzzy time series, and $FLRs$ and $FLRGs$ according to Definitions 4.2.3 and 4.2.4.

(c) Defuzzify and calculate the forecast outputs. Specifically, the algorithm first calculates NVD and LT weight factors and derives the $\widetilde{A}_{NVD}$ and $\widetilde{A}_{LT}$ according to Eq. 4.15 and Eq. 4.16, respectively. Based on the obtained $\widetilde{A}_{NVD}$ and $\widetilde{A}_{LT}$, as well as NTP, α, β and γ generated by ASCHLO. The model can calculate $\widetilde{A}_{t+1}$ according to Eq. 4.14. Finally, the one-step forecast will be the middle point of the interval-valued mean of $\widetilde{A}_{t+1}$.

(d) Calculate the fitness value, i.e. the root mean square error (RMSE), as follows:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{t_{\max}} (F_t - R_t)^2}{t_{\max}}} \tag{4.18}$$

where t_{max} denotes the number of trading days of the historical training data, F_t, denotes the forecast value of the validation datum on trading day t, and R_t is the actual value of the historical validation datum on trading day t.

Step 4: Improvise new individuals. New solutions are yielded by performing the adaptive pr and pi rule and the learning operators shown in Eqs. 4.3-4.5.

Step 5: Calculate the fitness of new candidates. The fitness value of new individuals is computed as steps 3.a—3.d.

Step 6: Update the $IKDs$ and SKD according to the fitness of the new individuals.

Step 7: Check the termination criterion. If the termination criterion is not met, steps 3—6 will be repeated to keep searching for the optimal parameters, otherwise go to the next step.

Step 8: Based on the obtained optimal parameters, the model performs steps 3.a—3.d to calculate the final forecast on the test datasets.

Step 9: Ultimately, the forecasting results of all extracted IMFs and residual component obtained by appropriate IHWF model are aggregated as the final forecast for the original financial time series using Eq. 4.1 and Eq. 4.17.

Table 4.1 The comparison of per-year RMSE and average RMSE for different models when forecasting the TAIEX from November to December (1990 to 2004). Overall, our model (IHWF) obtains the lowest average RMSE, and is the highest ranked model.

Year \ Models	Chen [37]	Yu [207]	Lee [107]	Huarng [81]	Egrioglu [55]	Cai [29]	Chen [40]	Cheng [44]	Wang [193]	Zhang [214]	IHWF
1990	249.31	227.15	278.94	199.65	182.42	187.10	172.89	**168.77**	232.58	176.25	173.12
1991	80.49	61.27	91.62	54.96	49.43	**39.58**	72.87	46.63	75.12	43.70	42.58
1992	60.05	67.75	78.59	61.03	50.61	39.37	43.44	45.53	64.30	40.35	**39.10**
1993	110.84	105.39	122.48	117.62	104.34	**101.26**	103.21	105.30	121.07	102.94	101.78
1994	112.07	135.29	141.13	88.34	78.50	76.32	78.63	77.17	128.96	69.38	**67.83**
1995	79.85	70.22	77.75	64.20	62.29	56.05	66.66	50.34	74.12	52.99	**47.61**
1996	54.64	54.31	59.82	52.95	51.33	49.45	59.75	53.19	55.43	**51.16**	52.30
1997	148.29	133.04	166.70	135.99	129.51	123.98	139.68	131.45	147.62	113.54	**112.16**
1998	167.84	151.27	184.55	136.46	132.07	118.41	124.44	115.89	166.94	**114.93**	115.05
1999	149.80	142.01	165.39	131.85	125.47	102.34	115.47	**100.74**	152.11	102.77	109.36
2000	176.57	191.31	277.82	**121.78**	156.58	131.53	123.62	125.62	225.82	150.39	129.58
2001	148.66	167.79	171.02	149.97	113.20	112.59	123.85	113.04	128.01	112.33	**110.25**
2002	101.40	75.11	112.45	89.13	85.95	60.33	71.98	62.94	92.07	**66.41**	68.85
2003	74.35	66.47	128.45	53.38	65.67	51.54	58.06	**51.16**	132.88	52.66	51.23
2004	82.32	72.34	75.58	67.11	61.14	**50.33**	57.73	54.25	58.40	54.95	54.15
Average RMSE	119.76	114.71	137.11	101.62	96.56	86.67	94.15	86.80	123.69	86.98	**84.99**
Ranking	9	8	10	7	6	2	5	3	11	4	1

4.5 Real-World Experiments and Results

4.5.1 Stock Market Data and Experimental Settings

To evaluate the forecast accuracy of the proposed IHWF model, the actual trading data of Taiwan Capitalization Weighted Stock Index (TAIEX) is used as benchmark. We work with daily values collected over 15 years, from January 1990 to December 2004 and also adopt the ten-month/two-month split for training/testing. That is, the historical data of the TAIEX of each year from January to October is used as the training set and the data from November to December is used as the test set. This is the same split used in existing fuzzy forecasting methods [29, 40, 44, 55, 214]. We compare the performance of the proposed IHWF model with ten well-known fuzzy time series methods using a variety of strategies [29, 37, 40, 44, 55, 81, 107, 193, 207, 214]. The performance is evaluated using RMSE as in Eq. 4.18.

All of the parameters for the baseline models are set to those reported in the initial publications. For our model, the population size is 30 and the NVD length is 2. pi_{mid} and pr_{mid} are set to 0.9 and 0.1 respectively. The number of iterations is fixed at 50. All implementations are in Python, and experiments were run on a PC with an Intel Core CPU i7-4790 @3.60GHz and 16 GB RAM.

4.5.2 Results and Analysis

Table 4.1 shows a comparison of per-year RMSE and the average RMSE for different methods when forecasting the TAIEX from 1990 to 2004. We also include the ranking of different methods in the bottom row. From the above comparisons, we can observe that the proposed IHWF method does not always outperform the baselines for each year, but it does best in

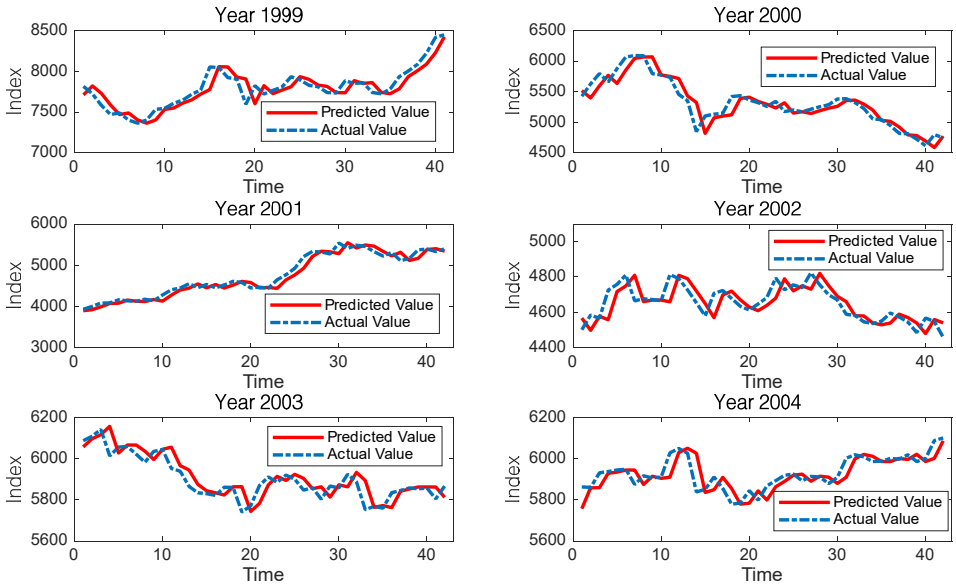


Figure 4.3 The actual daily quotes and the forecasts provided by IHWF for TAIEX from November to December (1999 to 2004). Time is measured in working days.

one third of the years (5 out of 15). These kinds of mixed results are very common in the field of stock market prediction because different drivers and forces are influencing the stock market during a given time period (for instance, 'bear' and 'bull' markets). However, the long time span over which we performed our experiments does provide evidence for the efficacy of our proposed method, which leads to the lowest average RMSE shown in Table 4.1. Finally, Figure 4.3 shows time series of daily forecasts provided by IHWF alongside the actual daily observations for the year 1999—2004. We surmise that the IHWF model is more effectively addressing the hysteresis problem of stock index forecasting, especially when the market exhibits cycles of boom (e.g., Year 2001) and bust (e.g., Year 2000). Although it is very hard to do the accurate forecast in some periods with more volatile (especially in a very short time span), our model can still capitalize the market trend which may help investors do a better decision making.

4.6 Conclusions

A new intelligent hybrid weighted fuzzy time series model has been presented for financial market forecasting. The model uniquely combines empirical mode decomposition with a novel weighted fuzzy time series method and is enhanced by an adaptive sine-cosine human learning parameter optimization method. The proposed algorithm considers chronological-order based frequency and, in addition, a neighborhood volatility direction is analyzed and integrated with ASCHLO in order to determine the effective universe discourse, intervals and weights. The proposed model is evaluated against actual trading data from the TAIEX index from 1990 to 2004. Our experimental results demonstrate that the method addresses the hysteresis problem of stock market forecasting, and outperforms its counterparts over the long term, in terms of RMSE. The combination of a relatively simple implementation and effective forecasting performance suggests that our proposed model is suitable for next-generation market forecasting applications.

CHAPTER

5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

This dissertation has presented several efficient data-driven methods for BI, especially in terms of E-Commerce and financial market analysis. We started by motivating this work from the observations that few of current recommendation approaches attempt to directly optimize business revenue — instead optimizing 'proxy' factors, such as customer purchase likelihood.Optimizing revenue directly is a more challenging task than predicting customer behavior since it involves a larger range of factors, such as real time price, customer interest trend, retailers' sale and so on. Therefore, the economic utility of a product is important to

take into account because it directly influences the demand, and price, of each product.

To do so, in Charter 2, we focus on the real time utility model for revenue optimization since this model can truly reflect the integrated information towards revenue, rather than only optimizes users' adoption rate as previous works did. Specifically, we first define the problem of utility-recommendation towards revenue optimization. Then we set out a new research challenge: real-time utility-based recommendation. To solve this challenge, we combine an adaptive online Top-K high utility itemsets mining model with rating and frequency information. In addition, to make the algorithms more personalized, we also take the buying power of customers into consideration and then propose a simple but effective method to estimate the consumers' willingness to pay. Finally, we conduct extensive experiments on Amazon, Foodmart and ChainStore datasets to show the effectiveness of the proposed algorithms.

Inspired by the concept of utility, in Chapter 3, we develop an intelligent weighted fuzzy time series model for financial markets forecasting.we first develop and deploy a novel evolutionary algorithm, Sine-Cosine adaptive Human Learning Optimization (SCHLO), to optimize the FTS and required parameters automatically and globally. Human Learning Optimization (HLO). Thanks to the simple but effective structure of HLO, it does not need the gradient information in its optimization procedure, which is very suitable for the FTS optimization. Second, new weighted operators that consider the frequency based chronological order and stock volume are analyzed and integrated with SCHLO for optimization. Finally, a novel short-term trend repair operation is developed to complement the final forecasting process. we analyze the experimental results of the proposed FTS forecasting method and several state-of-the-art weighted fuzzy time series methods, when applied to forecasting several stock indices (Dow Jones Index (DJI), the German Stock Index (DAX), the Japanese Stock Index (NIKKEI), and Taiwan Stock Index (TAIEX). Experimental results

show that our model is consistently more accurate than the baseline methods. The easy implementation and effective forecasting performance suggest our proposed model could be a favorable market application prospect.

In Chapter 4, we notice that although the observation of the financial market is a single time series, it can be a mixture of several sub-series generated by different institutions or individuals. Thus, creating forecasting models based only on a single observed series is not enough and may lead to bottlenecks. On the other hand, most models only depend on the historical trend indicated by the fuzzy logic group but ignore the big influence of the volatility direction indicated by neighborhood. Therefore, to address the above weakness of existing FTS models, a novel Intelligent and Hybrid Weighted Fuzzy time series model is proposed in this chapter. We first adopt Empirical Mode Decomposition to extract the possible modes of various kinds of institutional and individual investors embedded in an observed time series. Then, we propose a New Weighted FTS that considers both chronological-order based frequency and the neighborhood volatility direction. The NWF will perform the proper forecasting according to various sub-series extracted by EMD and all the predicted results are aggregated as the final forecast. Finally, to optimize the NWF automatically and globally, a novel evolutionary algorithm, Adaptive Sine-Cosine Human Learning Optimization , is developed and deployed. To the best of our knowledge, no existing FTS models decompose the financial time series to analyze the inner forecast process. The solution framework designed in this research is unique and significantly improves the forecast accuracy.

5.2 Future Work

Business intelligence, especially driven by the data, is one of the most attractive fields of interdisciplinary research. There are many directions that could be expanded upon the work we presented in this dissertation.

5.2.1 Scope for Revenue Optimization

In this dissertation, we optimize the revenue based on recommender system. Thus, a future study might consider the further improvement of recommender system for the revenue purpose. Although recommendation for e-commerce now has been much more comprehensive, we still have several challenges such as context-aware and interaction limitations. [78, 153]. Cross-domain recommendation is a trend of next generation of commendation. For example, the appropriate recommendations are not only based on the browsing history, purchase history or product utility information but also related to the email conversation, social network activities or even weather changes. Therefore, how to connect these domains and deign the algorithms which can efficiently do the recommendation for revenue optimization based on these multi-domain data should be a very interesting future work.

In addition to recommender system, some researchers have shown their interests towards revenue optimization from the perspective of dynamic pricing model [19], sequential purchase contexts [141] and resource allocation [87]. Therefore, depends on the demands of revenue optimization, another future work might consider the hybrid model with heterogeneous perspectives such as economic pricing model or heuristic approach together to achieve some synergy between different mechanisms for the objective of revenue optimization.

5.2.2 Scope for Financial Forecasting

The current proposed fuzzy time series model is based on the pure time series data. As we know, on the one hand, the financial markets nowadays can be influenced or quickly act upon the emergencies or breaking news [124]. On the other hand, advanced internet technologies allows us to incorporate previously unexplored sources, such as text, image and even video data from social media or social network, into modern forecasting models [110]. Obviously, unlike the pure time series data, we need to handle huge volumes of different types of data. Considering the fast development of various neural network and deep learning techniques nowadays, it is very promising to introduce these techniques to our model. Also, with the help of the powerful GPU computation resources, the researchers could not only build a hybrid model incorporating heterogeneous data but can also track the dynamic changes of financial market more efficiently and also explore the potential power of the plenty history data of different types.

Besides, the global financial markets are becoming more interactional. For example, if the stock market of USA slumps, then the Chinese market next day would open lower. Also, the crude oil market may still have some impact of the stock market. Therefore, another future study might consider incorporating multiple markets to train the model using various inputs.

In addition, although the financial market changes dynamically, it still has its intrinsic economic rules. For example, no mater how the market fluctuates, it always see-saws from bull and bear market with the periodicity. Thus, we may consider to introduce some economical knowledge for improving the robustness of the model.

Finally, in Chapter 4, we introduce the basic EMD to decompose the time series and the forecast performance is influenced by the results of EMD. Thus, another interesting future

work might be creating heuristic or approximation methods for EMD which can adaptively optimize the IMFs and residual for a better decompose results, especially for the financial markets. In this way, the forecast results may be much improved.

BIBLIOGRAPHY

[1] Abbasi, A. et al. “Metafraud: a meta-learning framework for detecting financial fraud”. *Mis Quarterly* (2012), pp. 1293–1327.

[2] Abu-Mostafa, Y. S. & Atiya, A. F. “Introduction to financial forecasting”. *Applied Intelligence* **6**.3 (1996), pp. 205–213.

[3] Adomavicius, G. & Tuzhilin, A. “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions”. *IEEE transactions on knowledge and data engineering* **17**.6 (2005), pp. 734–749.

[4] Agarwal, R. & Dhar, V. *Big data, data science, and analytics: The opportunity and challenge for IS research*. 2014.

[5] Agrawal, R., Srikant, R., et al. “Fast algorithms for mining association rules”. *Proc. 20th int. conf. very large data bases, VLDB*. Vol. 1215. 1994, pp. 487–499.

[6] Agrawal, R. et al. “Mining association rules between sets of items in large databases”. *Acm sigmod record*. Vol. 22. 2. ACM. 1993, pp. 207–216.

[7] Ahmed, C. F. et al. “Efficient tree structures for high utility pattern mining in incremental databases”. *IEEE Transactions on Knowledge and Data Engineering* **21**.12 (2009), pp. 1708–1721.

[8] Al-Hassan, M. et al. “A semantic enhanced hybrid recommendation approach: A case study of e-Government tourism service recommendation system”. *Decision Support Systems* **72** (2015), pp. 97–109.

[9] Al-Shamri, M. Y. H. & Bharadwaj, K. K. “Fuzzy-genetic approach to recommender systems based on a novel hybrid user model”. *Expert systems with applications* **35**.3 (2008), pp. 1386–1399.

[10] Alfonso, G. et al. “A fuzzy regression model based on finite fuzzy numbers and its application to real-world financial data”. *Journal of Computational and Applied Mathematics* **318** (2017), pp. 47–58.

[11] Arnott, D. et al. “Patterns of business intelligence systems use in organizations”. *Decision Support Systems* **97** (2017), pp. 58–68.

[12] Arrow, K. J. & Kruz, M. *Public investment, the rate of return, and optimal fiscal policy*. Vol. 1. Routledge, 2013.

[13] Avazbeigi, M. et al. “Choosing the appropriate order in fuzzy time series: A new N-factor fuzzy time series for prediction of the auto industry production”. *Expert Systems with Applications* **37**.8 (2010), pp. 5630–5639.

[14] Azaria, A. et al. “Movie recommender system for profit maximization”. *Proceedings of the 7th ACM conference on Recommender systems*. ACM. 2013, pp. 121–128.

[15] Balabanović, M. & Shoham, Y. “Fab: content-based, collaborative recommendation”. *Communications of the ACM* **40**.3 (1997), pp. 66–72.

[16] Balcilar, M. et al. “Common cycles and common trends in the stock and oil markets: Evidence from more than 150years of data”. *Energy Economics* **61** (2017), pp. 72–86.

[17] Ballings, M. et al. “Evaluating multiple classifiers for stock price direction prediction”. *Expert Systems with Applications* **42**.20 (2015), pp. 7046–7056.

[18] Basu, C. et al. “Recommendation as classification: Using social and content-based information in recommendation”. *Aaai/iaai*. 1998, pp. 714–720.

[19] Bayliss, C. et al. “Dynamic pricing for vehicle ferries: Using packing and simulation to optimize revenues”. *European Journal of Operational Research* **273**.1 (2019), pp. 288–304.

[20] Belkin, N. J. & Croft, W. B. “Information filtering and information retrieval: Two sides of the same coin?” *Communications of the ACM* **35**.12 (1992), pp. 29–38.

[21] Bellogín, A. & Sánchez, P. “Collaborative filtering based on subsequence matching: A new approach”. *Information Sciences* **418** (2017), pp. 432–446.

[22] Bertsimas, D. & Thiele, A. “Robust and data-driven optimization: Modern decision-making under uncertainty”. *INFORMS tutorials in operations research: models, methods, and applications for innovative decision making* **3** (2006).

[23] Bilias, Y. et al. “Portfolio inertia and stock market fluctuations”. *Journal of Money, Credit and Banking* **42**.4 (2010), pp. 715–742.

[24] Blanco-Fernández, Y. et al. “Exploring synergies between content-based filtering and spreading activation techniques in knowledge-based recommender systems”. *Information Sciences* **181**.21 (2011), pp. 4823–4846.

[25] Borgelt, C. “Frequent item set mining”. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2**.6 (2012), pp. 437–456.

[26] Box, G. E. et al. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[27] Boyd, S. et al. "A tutorial on geometric programming". *Optimization and engineering* **8**.1 (2007), p. 67.

[28] Burke, R. "Hybrid recommender systems: Survey and experiments". *User modeling and user-adapted interaction* **12**.4 (2002), pp. 331–370.

[29] Cai, Q. et al. "A new fuzzy time series forecasting model combined with ant colony optimization and auto-regression". *Knowledge-Based Systems* **74** (2015), pp. 61–68.

[30] Cao, J. et al. "Application of multi-objective human learning optimization method to solve AC/DC multi-objective optimal power flow problem". *International Journal of Emerging Electric Power Systems* **17**.3 (2016), pp. 327–337.

[31] Cao, L. "Domain-driven data mining: Challenges and prospects". *IEEE Transactions on Knowledge and Data Engineering* **22**.6 (2010), pp. 755–769.

[32] Chan, R. et al. "Mining high utility itemsets". *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE. 2003, pp. 19–26.

[33] Chen, C. P. & Zhang, C.-Y. "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data". *Information Sciences* **275** (2014), pp. 314–347.

[34] Chen, J. et al. "Evolutionary heterogeneous clustering for rating prediction based on user collaborative filtering". *Swarm and Evolutionary Computation* **38** (2018), pp. 35–41.

[35] Chen, L.-S. et al. "Developing recommender systems with the consideration of product profitability for sellers". *Information Sciences* **178**.4 (2008), pp. 1032–1048.

[36] Chen, M.-Y. & Chen, B.-T. "A hybrid fuzzy time series model based on granular computing for stock price forecasting". *Information Sciences* **294** (2015), pp. 227–241.

[37] Chen, S. "Forecasting enrollments based on fuzzy time series". *Fuzzy sets and systems* **81**.3 (1996), pp. 311–319.

[38] Chen, S.-M. & Chang, Y.-C. "Multi-variable fuzzy forecasting based on fuzzy clustering and fuzzy rule interpolation techniques". *Information sciences* **180**.24 (2010), pp. 4772–4783.

[39] Chen, S.-M. & Chen, C.-D. “Handling forecasting problems based on high-order fuzzy logical relationships”. *Expert Systems with Applications* **38**.4 (2011), pp. 3857–3864.

[40] Chen, S.-M. & Chen, C.-D. “TAIEX forecasting based on fuzzy time series and fuzzy variation groups”. *IEEE Transactions on Fuzzy Systems* **19**.1 (2011), pp. 1–12.

[41] Chen, S.-M. & Phuong, B. D. H. “Fuzzy time series forecasting based on optimal partitions of intervals and optimal weighting vectors”. *Knowledge-Based Systems* **118** (2017), pp. 204–216.

[42] Cheng, C.-H. et al. “Fuzzy time-series based on adaptive expectation model for TAIEX forecasting”. *Expert systems with applications* **34**.2 (2008), pp. 1126–1132.

[43] Cheng, H. et al. “An ontology-based business intelligence application in a financial knowledge management system”. *Expert Systems with Applications* **36**.2 (2009), pp. 3614–3622.

[44] Cheng, S.-H. et al. “Fuzzy time series forecasting based on fuzzy logical relationships and similarity measures”. *Information Sciences* **327** (2016), pp. 272–287.

[45] Cheung, K.-W. et al. “Mining customer product ratings for personalized marketing”. *Decision Support Systems* **35**.2 (2003), pp. 231–243.

[46] Chiang, R. H. et al. “Business intelligence and analytics education, and program development: A unique opportunity for the information systems discipline”. *ACM Transactions on Management Information Systems (TMIS)* **3**.3 (2012), p. 12.

[47] Dai, W. et al. “Incorporating feature selection method into support vector regression for stock index forecasting”. *Neural Computing and Applications* **23**.6 (2013), pp. 1551–1561.

[48] Das, A. et al. “Maximizing profit using recommender systems”. *arXiv preprint arXiv: 0908.3633* (2009).

[49] Davis, L. “Handbook of genetic algorithms” (1991).

[50] De Cnudde, S. & Martens, D. “Loyal to your city? A data mining analysis of a public service loyalty program”. *Decision Support Systems* **73** (2015), pp. 74–84.

[51] Delgado, J. & Ishii, N. “Memory-based weighted majority prediction”. *SIGIR Workshop Recomm. Syst. Citeseer*. 1999.

[52] Deng, S. et al. “Distributed content filtering algorithm based on data label and policy expression in active distribution networks”. *Neurocomputing* **270** (2017), pp. 159–169.

[53] Deshpande, M. & Karypis, G. “Item-based top-n recommendation algorithms”. *ACM Transactions on Information Systems (TOIS)* **22**.1 (2004), pp. 143–177.

[54] Dorigo, M. & Gambardella, L. M. “Ant colony system: a cooperative learning approach to the traveling salesman problem”. *IEEE Transactions on evolutionary computation* **1**.1 (1997), pp. 53–66.

[55] Egrioglu, E. et al. “Fuzzy time series forecasting method based on Gustafson–Kessel fuzzy clustering”. *Expert Systems with Applications* **38**.8 (2011), pp. 10355–10357.

[56] Erwin, A. et al. “Efficient mining of high utility itemsets from large datasets”. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2008, pp. 554–561.

[57] Fesanghary, M et al. “Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems”. *Computer methods in applied mechanics and engineering* **197**.33-40 (2008), pp. 3080–3091.

[58] Gan, W. et al. “A survey of incremental high-utility itemset mining”. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* ().

[59] Gantner, Z. et al. “MyMediaLite: a free recommender system library”. *Proceedings of the fifth ACM conference on Recommender systems*. ACM. 2011, pp. 305–308.

[60] García-Crespo, Á. et al. “SINVLIO: Using semantics and fuzzy logic to provide individual investment portfolio recommendations”. *Knowledge-Based Systems* **27** (2012), pp. 103–118.

[61] Garcin, F. et al. “Offline and online evaluation of news recommender systems at swissinfo. ch”. *Proceedings of the 8th ACM Conference on Recommender systems*. ACM. 2014, pp. 169–176.

[62] Gastaldi, L. et al. “Measuring the maturity of business intelligence in healthcare: Supporting the development of a roadmap toward precision medicine within ISMETT hospital”. *Technological Forecasting and Social Change* (2017).

[63] Ge, Y. et al. “A taxi business intelligence system”. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2011, pp. 735–738.

[64] Gedikli, F. & Jannach, D. “Recommending based on rating frequencies”. *Proceedings of the fourth ACM conference on Recommender systems*. ACM. 2010, pp. 233–236.

[65] Geem, Z. W. “Harmony search in water pump switching problem”. *International Conference on Natural Computation*. Springer. 2005, pp. 751–760.

[66] Gill, P. E. & Murray, W. “Quasi-Newton methods for unconstrained optimization”. *IMA Journal of Applied Mathematics* **9**.1 (1972), pp. 91–108.

[67] Govan, A. “Introduction to optimization”. *North Carolina State University, SAMSI NDHS, Undergraduate workshop*. 2006.

[68] Grahne, G. & Zhu, J. “Fast algorithms for frequent itemset mining using fp-trees”. *IEEE transactions on knowledge and data engineering* **17**.10 (2005), pp. 1347–1362.

[69] Graus, M. P. & Willemsen, M. C. “Improving the user experience during cold start through choice-based preference elicitation”. *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM. 2015, pp. 273–276.

[70] Gulzar, Z. et al. “PCRS: Personalized Course Recommender System Based on Hybrid Approach”. *Procedia Computer Science* **125** (2018), pp. 518–524.

[71] Guo, X. et al. *A Bayesian Approach to Excess Volatility, Short-term Underreaction and Long-term Overreaction during Financial Crises*. Tech. rep. Tinbergen Institute Discussion Paper, 2016.

[72] Guresen, E. et al. “Using artificial neural network models in stock market index prediction”. *Expert Systems with Applications* **38**.8 (2011), pp. 10389–10397.

[73] Guu, S.-M. “Properties of the multiple lot-sizing problem with rigid demand, general cost structures, and interrupted geometric yield”. *Operations research letters* **25**.2 (1999), pp. 59–65.

[74] Hahn, G. J. & Packowski, J. “A perspective on applications of in-memory analytics in supply chain management”. *Decision Support Systems* **76** (2015), pp. 45–52.

[75] Han, C.-W. & Park, J.-I. “Adaptive-Partitioning-Based stochastic optimization algorithm and its application to fuzzy control design”. *Hellenic Conference on Artificial Intelligence*. Springer. 2006, pp. 67–76.

[76] Han, J. et al. “Mining frequent patterns without candidate generation”. *ACM sigmod record*. Vol. 29. 2. ACM. 2000, pp. 1–12.

[77] Han, J. et al. *Data mining: concepts and techniques*. Elsevier, 2011.

[78] He, C. et al. “Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities”. *Expert Systems with Applications* **56** (2016), pp. 9–27.

[79] Huang, N. E. et al. “The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis”. *Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences*. Vol. 454. 1971. The Royal Society. 1998, pp. 903–995.

[80] Huang, Z. et al. “Mining association rules to support resource allocation in business process management”. *Expert Systems with Applications* **38**.8 (2011), pp. 9483–9490.

[81] Huarng, K. & Yu, T. H.-K. “Ratio-based lengths of intervals to improve fuzzy time series forecasting”. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **36**.2 (2006), pp. 328–340.

[82] Huck, N. “Pairs trading and outranking: The multi-step-ahead forecasting case”. *European Journal of Operational Research* **207**.3 (2010), pp. 1702–1716.

[83] Hung, J.-C. “Applying a combined fuzzy systems and GARCH model to adaptively forecast stock market volatility”. *Applied Soft Computing* **11**.5 (2011), pp. 3938–3945.

[84] *Insights from the 2011 IBM Global CIO Study*. `http://www-935.ibm.com/services/c-suite/cio/study/`.

[85] Iskyan, K. *China’s stock markets have soared by 1,479% since 2003*. Ed. by Insider, B. =http://www.businessinsider.com/world-stock-market-capitalizations-2016-11. 2016.

[86] Javedani Sadaei, H. & Lee, M. H. “Multilayer stock forecasting model using fuzzy time series”. *The Scientific World Journal* **2014** (2014).

[87] Jhang-Li, J.-H. & Chiang, I. R. “Resource allocation and revenue optimization for cloud service providers”. *Decision Support Systems* **77** (2015), pp. 55–66.

[88] Kant, S. et al. “LeaderRank based k-means clustering initialization method for collaborative filtering”. *Computers & Electrical Engineering* (2017).

[89] Kant, V. & Bharadwaj, K. K. “Enhancing recommendation quality of content-based filtering through collaborative predictions and fuzzy similarity measures”. *Procedia engineering* **38** (2012), pp. 939–944.

[90] Karabadji, N. E. I. et al. “Improving Memory-Based User Collaborative Filtering with Evolutionary Multi-Objective Optimization”. *Expert Systems with Applications* (2018).

[91] Karaboga, D. & Akay, B. “A survey: algorithms simulating bee swarm intelligence”. *Artificial intelligence review* **31**.1-4 (2009), p. 61.

[92] Kardan, A. A. & Ebrahimi, M. “A novel approach to hybrid recommendation systems based on association rules mining for content recommendation in asynchronous discussion groups”. *Information Sciences* **219** (2013), pp. 93–110.

[93] Kawamae, N. “Real time recommendations from connoisseurs”. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2015, pp. 537–546.

[94] Keerthi, S. S. “Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms”. *IEEE Transactions on Neural Networks* **13**.5 (2002), pp. 1225–1229.

[95] Kennedy, J. “Particle swarm optimization”. *Encyclopedia of machine learning*. Springer, 2011, pp. 760–766.

[96] Kennedy, J. & Eberhart, R. C. “A discrete binary version of the particle swarm algorithm”. *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*. Vol. 5. IEEE. 1997, pp. 4104–4108.

[97] Kim, Y. A. & Phalak, R. “A trust prediction framework in rating-based experience sharing social networks without a Web of Trust”. *Information Sciences* **191** (2012), pp. 128–145.

[98] Kingsbury, K. *Amazon Close to Topping Berkshire for Fifth Biggest U.S.-Listed Company by Market Cap*. Ed. by Company, D. J. `http://on.wsj.com/29wW9aj`. Accessed: 2016-07.

[99] Krishnamoorthy, S. “HMiner: Efficiently mining high utility itemsets”. *Expert Systems with Applications* **90** (2017), pp. 168–183.

[100] Lam, A. Y. & Li, V. O. “Chemical-reaction-inspired metaheuristic for optimization”. *IEEE Transactions on Evolutionary Computation* **14**.3 (2010), pp. 381–399.

[101] Lan, G.-C. et al. “An efficient gradual pruning technique for utility mining”. *International Journal of Innovative Computing Information and Control* **8**.7B (2012), pp. 5165–5178.

[102] Lan, G.-C. et al. “An efficient projection-based indexing approach for mining high utility itemsets”. *Knowledge and information systems* **38**.1 (2014), pp. 85–107.

[103] Larrain, S. et al. “Good times bad times: A study on recency effects in collaborative filtering for social tagging”. *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM. 2015, pp. 269–272.

[104] Larson, D. & Chang, V. “A review and future direction of agile, business intelligence, analytics and data science”. *International Journal of Information Management* **36**.5 (2016), pp. 700–710.

[105] Lee, J. H. & Park, S. C. “Intelligent profitable customers segmentation system based on business intelligence tools”. *Expert systems with applications* **29**.1 (2005), pp. 145–152.

[106] Lee, K. S. & Geem, Z. W. “A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice”. *Computer methods in applied mechanics and engineering* **194**.36-38 (2005), pp. 3902–3933.

[107] Lee, L.-W. et al. “Handling forecasting problems based on two-factors high-order fuzzy time series”. *IEEE Transactions on fuzzy Systems* **14**.3 (2006), pp. 468–477.

[108] Li, C. & Chiang, T.-W. “Complex neurofuzzy ARIMA forecasting—a new approach using complex fuzzy sets”. *IEEE Transactions on Fuzzy Systems* **21**.3 (2013), pp. 567–584.

[109] Li, H. et al. “Personalized TV Recommendation with Mixture Probabilistic Matrix Factorization”. *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM. 2015, pp. 352–360.

[110] Li, X. et al. “Text-based crude oil price forecasting: A deep learning approach”. *International Journal of Forecasting* (2018).

[111] Liao, S.-H. et al. “Data mining techniques and applications–A decade review from 2000 to 2011”. *Expert systems with applications* **39**.12 (2012), pp. 11303–11311.

[112] Lim, E.-P. et al. “Business intelligence and analytics: Research directions”. *ACM Transactions on Management Information Systems (TMIS)* **3**.4 (2013), p. 17.

[113] Lindsey, B. G. & Gerstein, G. L. “Two enhancements of the gravity algorithm for multiple spike train analysis”. *Journal of Neuroscience methods* **150**.1 (2006), pp. 116–127.

[114] Liu, B. et al. “Analyzing the subjective interestingness of association rules”. *IEEE Intelligent Systems and their Applications* **15**.5 (2000), pp. 47–55.

[115] Liu, J. et al. “Direct discovery of high utility itemsets without candidate generation”. *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE. 2012, pp. 984–989.

[116] Liu, M. & Qu, J. “Mining high utility itemsets without candidate generation”. *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM. 2012, pp. 55–64.

[117] Liu, Y. et al. “A fast high utility itemsets mining algorithm”. *Proceedings of the 1st international workshop on Utility-based data mining*. ACM. 2005, pp. 90–99.

[118] Liu, Y. et al. “A two-phase algorithm for fast discovery of high utility itemsets”. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2005, pp. 689–695.

[119] Liu, Z. et al. “A hybrid collaborative filtering recommendation mechanism for P2P networks”. *Future generation computer systems* **26**.8 (2010), pp. 1409–1417.

[120] Loni, B. et al. “’Free lunch’enhancement for collaborative filtering with factorization machines”. *Proceedings of the 8th ACM Conference on Recommender systems*. ACM. 2014, pp. 281–284.

[121] Lu, T. et al. “An Algorithm of Top-k High Utility Itemsets Mining over Data Stream.” *JSW* **9**.9 (2014), pp. 2342–2347.

[122] Lu, W. et al. “Show me the money: dynamic recommendations for revenue maximization”. *Proceedings of the VLDB Endowment* **7**.14 (2014), pp. 1785–1796.

[123] Lukić, J. et al. “A hybrid approach to building a multi-dimensional business intelligence system for electricity grid operators”. *Utilities Policy* **41** (2016), pp. 95–106.

[124] Luo, P. et al. “Measuring social influence for firm-level financial performance”. *Electronic Commerce Research and Applications* **20** (2016), pp. 15–29.

[125] Ma, B. L.W.H. Y. & Liu, B. “Integrating classification and association rule mining”. *Proceedings of the fourth international conference on knowledge discovery and data mining*. 1998.

[126] Mahdavi, M et al. “An improved harmony search algorithm for solving optimization problems”. *Applied mathematics and computation* **188**.2 (2007), pp. 1567–1579.

[127] Mannila, H. et al. “E cient algorithms for discovering association rules”. *KDD-94: AAAI workshop on Knowledge Discovery in Databases*. 1994, pp. 181–192.

[128] Marszałek, A & Burczyński, T. “Modeling and forecasting financial time series with ordered fuzzy candlesticks”. *Information sciences* **273** (2014), pp. 144–155.

[129] Maté, A. et al. “Empowering global software development with business intelligence”. *Information and Software Technology* **76** (2016), pp. 81–91.

[130] McAuley, J. et al. “Inferring networks of substitutable and complementary products”. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2015, pp. 785–794.

[131] Menhas, M. I. et al. “Comparative performance analysis of various binary coded PSO algorithms in multivariable PID controller design”. *Expert systems with applications* **39**.4 (2012), pp. 4390–4401.

[132] Menkhoff, L. et al. “Overconfidence, experience, and professionalism: An experimental study”. *Journal of Economic Behavior & Organization* **86** (2013), pp. 92–101.

[133] Merh, N. “Stock market forecasting”. *Journal of Information Technology Applications & Management* **19**.1 (2012), pp. 1–12.

[134] Mullen, R. J. et al. “A review of ant algorithms”. *Expert systems with Applications* **36**.6 (2009), pp. 9608–9617.

[135] Narducci, F. et al. “Concept-based item representations for a cross-lingual content-based recommendation process”. *Information Sciences* **374** (2016), pp. 15–31.

[136] Ng, R. T. et al. “Exploratory mining and pruning optimizations of constrained associations rules”. *ACM Sigmod Record*. Vol. 27. 2. ACM. 1998, pp. 13–24.

[137] Nofsinger, J. R. *The psychology of investing*. Routledge, 2017.

[138] Ochirbat, A. et al. “Hybrid occupation recommendation for adolescents on interest, profile, and behavior”. *Telematics and Informatics* (2017).

[139] Oftadeh, R et al. “A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search”. *Computers & Mathematics with Applications* **60**.7 (2010), pp. 2087–2098.

[140] Oliveira, F. A. de et al. “Applying Artificial Neural Networks to prediction of stock price and improvement of the directional prediction index–Case study of PETR4, Petrobras, Brazil”. *Expert Systems with Applications* **40**.18 (2013), pp. 7596–7606.

[141] Ozturk, O. C. & Karabatı, S. “A decision support framework for evaluating revenue performance in sequential purchase contexts”. *European Journal of Operational Research* **263**.3 (2017), pp. 922–934.

[142] Pampara, G. et al. “Binary differential evolution”. *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. IEEE. 2006, pp. 1873–1879.

[143] Park, J.-I. et al. “TAIFEX and KOSPI 200 forecasting based on two-factors high-order fuzzy time series and particle swarm optimization”. *Expert Systems with Applications* **37**.2 (2010), pp. 959–967.

[144] Pazzani, M. J. “A framework for collaborative, content-based and demographic filtering”. *Artificial intelligence review* **13**.5-6 (1999), pp. 393–408.

[145] Peng, M. et al. “Exploration of association rule mining for coding consistency and completeness assessment in inpatient administrative health data”. *Journal of Biomedical Informatics* (2018).

[146] PéRez-DíAz, N. et al. “SDAI: An integral evaluation methodology for content-based spam filtering models”. *Expert Systems with Applications* **39**.16 (2012), pp. 12487–12500.

[147] Pisharath, J. et al. “NU-MineBench 2.0”. *Department of Electrical and Computer Engineering, Northwestern University* (2005).

[148] Polato, M. & Aiolli, F. “Boolean kernels for collaborative filtering in top-N item recommendation”. *Neurocomputing* (2018).

[149] Polyvyanyy, A. et al. “Process querying: Enabling business intelligence through query-based process analytics”. *Decision Support Systems* **100** (2017), pp. 41–56.

[150] Raisinghani, M. S. *Business Intelligence in the Digital Economy: Opportunities, Limitations and Risks: Opportunities, Limitations and Risks*. Igi Global, 2003.

[151] Rashedi, E. et al. “BGSA: binary gravitational search algorithm”. *Natural Computing* **9**.3 (2010), pp. 727–745.

[152] Ravi, K. et al. “Fuzzy Formal Concept Analysis based Opinion Mining for CRM in Financial Services”. *Applied Soft Computing* (2017).

[153] Ricci, F. et al. “Recommender systems: introduction and challenges”. *Recommender systems handbook*. Springer, 2015, pp. 1–34.

[154] Rubio, A. et al. “Improving stock index forecasts by using a new weighted fuzzy-trend time series method”. *Expert Systems with Applications* **76** (2017), pp. 12–20.

[155] Rygielski, C. et al. “Data mining techniques for customer relationship management”. *Technology in society* **24**.4 (2002), pp. 483–502.

[156] Said, S. E. & Dickey, D. A. “Testing for unit roots in autoregressive-moving average models of unknown order”. *Biometrika* **71**.3 (1984), pp. 599–607.

[157] Salter, J. & Antonopoulos, N. “CinemaScreen recommender agent: combining collaborative and content-based filtering”. *IEEE Intelligent Systems* **21**.1 (2006), pp. 35–41.

[158] Schafer, J. B. et al. “Recommender systems in e-commerce”. *Proceedings of the 1st ACM conference on Electronic commerce*. ACM. 1999, pp. 158–166.

[159] Schumaker, R. P. & Chen, H. “Textual analysis of stock market prediction using breaking financial news: The AZFin text system”. *ACM Transactions on Information Systems (TOIS)* **27**.2 (2009), p. 12.

[160] Schwefel, H.-P. *Numerical optimization of computer models*. John Wiley & Sons, Inc., 1981.

[161] Shambour, Q. & Lu, J. “A trust-semantic fusion-based recommendation approach for e-business applications”. *Decision Support Systems* **54**.1 (2012), pp. 768–780.

[162] Shollo, A. & Kautz, K. “Towards an understanding of business intelligence”. *Australasian Conference on Information Systems*. Vol. 21. 2010.

[163] Simon, D. “Biogeography-based optimization”. *IEEE transactions on evolutionary computation* **12**.6 (2008), pp. 702–713.

[164] Solomatine, D. P. & Ostfeld, A. “Data-driven modelling: some past experiences and new approaches”. *Journal of hydroinformatics* **10**.1 (2008), pp. 3–22.

[165] Son, J. & Kim, S. B. “Content-based filtering for recommendation systems using multiattribute networks”. *Expert Systems with Applications* **89** (2017), pp. 404–412.

[166] Song, Q. & Chissom, B. S. “Forecasting enrollments with fuzzy time series—part I”. *Fuzzy sets and systems* **54**.1 (1993), pp. 1–9.

[167] Song, Q. & Chissom, B. S. “Fuzzy time series and its models”. *Fuzzy sets and systems* **54**.3 (1993), pp. 269–277.

[168] Song, Q. & Chissom, B. S. “Forecasting enrollments with fuzzy time series—part II”. *Fuzzy sets and systems* **62**.1 (1994), pp. 1–8.

[169] Strategymrc, ed. *Business Intelligence (BI) - Global Market Outlook (2016-2022)*. `http://www.strategymrc.com/report/business-intelligence-bi-market/`. 2016.

[170] Su, C.-H. & Cheng, C.-H. “A hybrid fuzzy time series model based on ANFIS and integrated nonlinear feature selection method for forecasting stock”. *Neurocomputing* **205** (2016), pp. 264–273.

[171] Suykens, J. A. & Vandewalle, J. “Least squares support vector machine classifiers”. *Neural processing letters* **9**.3 (1999), pp. 293–300.

[172] Talarposhti, F. M. et al. “Stock market forecasting by using a hybrid model of exponential fuzzy time series”. *International Journal of Approximate Reasoning* **70** (2016), pp. 79–98.

[173] Teoh, H. J. et al. “A hybrid multi-order fuzzy time series for forecasting stock markets”. *Expert Systems with Applications* **36**.4 (2009), pp. 7888–7897.

[174] Thanh Lam, H. & Calders, T. “Mining top-k frequent items in a data stream with flexible sliding windows”. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2010, pp. 283–292.

[175] Ticknor, J. L. “A Bayesian regularized artificial neural network for stock market forecasting”. *Expert Systems with Applications* **40**.14 (2013), pp. 5501–5506.

[176] Trieu, V.-H. “Getting value from Business Intelligence systems: A review and research agenda”. *Decision Support Systems* **93** (2017), pp. 111–124.

[177] Tseng, V. S. et al. “UP-Growth: an efficient algorithm for high utility itemset mining”. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2010, pp. 253–262.

[178] Tseng, V. S. et al. “Efficient algorithms for mining high utility itemsets from transactional databases”. *IEEE transactions on knowledge and data engineering* **25**.8 (2013), pp. 1772–1786.

[179] Tseng, V. S. et al. “Efficient algorithms for mining top-k high utility itemsets”. *IEEE Transactions on Knowledge and Data Engineering* **28**.1 (2016), pp. 54–67.

[180] Turban, E. et al. *Business intelligence: A managerial approach*. Pearson Prentice Hall Upper Saddle River, NJ, 2008.

[181] Uslu, V. R. et al. “A fuzzy time series approach based on weights determined by the number of recurrences of fuzzy relations”. *Swarm and Evolutionary Computation* **15** (2014), pp. 19–26.

[182] valentina.palladino. *Amazon sold 426 items per second in run-up to Christmas*. Ed. by Verge, T. =https://www.theverge.com/2013/12/26/5245008/amazon-sees-prime-spike-in-2013-holiday-season. 2013.

[183] Vasan, A & Raju, K. S. “Comparative analysis of simulated annealing, simulated quenching and genetic algorithms for optimal reservoir operation”. *Applied Soft Computing* **9**.1 (2009), pp. 274–281.

[184] Wang, C.-R. et al. “An improved artificial fish-swarm algorithm and its application in feed-forward neural networks”. *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*. Vol. 5. IEEE. 2005, pp. 2890–2894.

[185] Wang, J. et al. “Improved v-Support vector regression model based on variable selection and brain storm optimization for stock price forecasting”. *Applied Soft Computing* **49** (2016), pp. 164–178.

[186] Wang, J. & Wang, J. “Forecasting stock market indexes using principle component analysis and stochastic time effective neural networks”. *Neurocomputing* **156** (2015), pp. 68–78.

[187] Wang, L. et al. “A novel modified binary differential evolution algorithm and its applications”. *Neurocomputing* **98** (2012), pp. 55–75.

[188] Wang, L. et al. “An adaptive fuzzy controller based on harmony search and its application to power plant control”. *International Journal of Electrical Power & Energy Systems* **53** (2013), pp. 272–278.

[189] Wang, L. et al. “An improved adaptive binary harmony search algorithm”. *Information Sciences* **232** (2013), pp. 58–87.

[190] Wang, L. et al. “A simple human learning optimization algorithm”. *International Conference on Life System Modeling and Simulation and Intelligent Computing for Sustainable Energy and Environment*. Springer. 2014, pp. 56–65.

[191] Wang, L. et al. “A human learning optimization algorithm and its application to multi-dimensional knapsack problems”. *Applied Soft Computing* **34** (2015), pp. 736–743.

[192] Wang, L. et al. “An adaptive simplified human learning optimization algorithm”. *Information Sciences* **320** (2015), pp. 126–139.

[193] Wang, L. et al. “Effective intervals determined by information granules to improve forecasting in fuzzy time series”. *Expert Systems with Applications* **40**.14 (2013), pp. 5673–5679.

[194] Wang, Y. et al. “A hybrid user similarity model for collaborative filtering”. *Information Sciences* **418** (2017), pp. 102–118.

[195] Wedderburn, R. W. “Quasi-likelihood functions, generalized linear models, and the Gauss—Newton method”. *Biometrika* **61**.3 (1974), pp. 439–447.

[196] Wei, K. et al. “A survey of e-commerce recommender systems”. *Service systems and service management, 2007 international conference on*. IEEE. 2007, pp. 1–5.

[197] Wei, L.-Y. “A hybrid ANFIS model based on empirical mode decomposition for stock time series forecasting”. *Applied Soft Computing* **42** (2016), pp. 368–376.

[198] Wei, L.-Y. et al. “A hybrid ANFIS based on n-period moving average model to forecast TAIEX stock”. *Applied Soft Computing* **19** (2014), pp. 86–92.

[199] Williams, S. *Business intelligence strategy and Big Data analytics: a general management perspective*. Morgan Kaufmann, 2016.

[200] Wixom, B. H. et al. “Developing an enterprise business intelligence capability: The Norfolk Southern journey.” *MIS Quarterly Executive* **10**.2 (2011).

[201] Wu, C. “Applying frequent itemset mining to identify a small itemset that satisfies a large percentage of orders in a warehouse”. *Computers & operations research* **33**.11 (2006), pp. 3161–3170.

[202] Wu, C.-Y. & Tseng, K.-Y. “Topology optimization of structures using modified binary differential evolution”. *Structural and Multidisciplinary Optimization* **42**.6 (2010), pp. 939–953.

[203] Wu, M.-L. et al. “Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices”. *Expert Systems with Applications* **41**.6 (2014), pp. 2754–2761.

[204] Wu, X. et al. “Data mining with big data”. *IEEE transactions on knowledge and data engineering* **26**.1 (2014), pp. 97–107.

[205] Yang, S. et al. “Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive Statistical Relational Learning approach”. *Knowledge-Based Systems* **136** (2017), pp. 37–45.

[206] Yin, S. et al. “Data-based techniques focused on modern industry: An overview”. *IEEE Transactions on Industrial Electronics* **62**.1 (2015), pp. 657–667.

[207] Yu, H.-K. “Weighted fuzzy time series models for TAIEX forecasting”. *Physica A: Statistical Mechanics and its Applications* **349**.3 (2005), pp. 609–624.

[208] Zadeh, L. A. “Fuzzy sets”. *Information and control* **8**.3 (1965), pp. 338–353.

[209] Zandieh, M. et al. “An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times”. *Applied Mathematics and Computation* **180**.1 (2006), pp. 111–127.

[210] Zeger, S. L. “A regression model for time series of counts”. *Biometrika* **75**.4 (1988), pp. 621–629.

[211] Zenebe, A. & Norcio, A. F. “Representation, similarity measures and aggregation methods using fuzzy sets for content-based recommender systems”. *Fuzzy sets and systems* **160**.1 (2009), pp. 76–94.

[212] Zeng, X. et al. “Investment behavior prediction in heterogeneous information network”. *Neurocomputing* **217** (2016), pp. 125–132.

[213] Zhang, W. & Wang, J. “A collective bayesian poisson factorization model for cold-start local event recommendation”. *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM. 2015, pp. 1455–1464.

[214] Zhang, W. et al. “A multi-factor and high-order stock forecast model based on Type-2 FTS using cuckoo search and self-adaptive harmony search”. *Neurocomputing* **240** (2017), pp. 13–24.

[215] Zhong, X. & Enke, D. “Forecasting daily stock market return using dimensionality reduction”. *Expert Systems with Applications* **67** (2017), pp. 126–139.

[216] Zhou, R. et al. “A portfolio optimization model based on information entropy and fuzzy time series”. *Fuzzy Optimization and Decision Making* **14**.4 (2015), p. 381.

[217] Zhou, X. et al. “Confidence-weighted bias model for online collaborative filtering”. *Applied Soft Computing* (2017).

[218] Zida, S. et al. “EFIM: a fast and memory efficient algorithm for high-utility itemset mining”. *Knowledge and Information Systems* **51**.2 (2017), pp. 595–625.

[219] Zihayat, M. & An, A. “Mining top-k high utility patterns over data streams”. *Information Sciences* **285** (2014), pp. 138–161.