

FINREC16

Proceedings of the

2nd International Workshop on Personalization & Recommender Systems in Financial Services



JUNE 16, 2016 | BARI, ITALY

Edited by

Giovanni Semeraro, Cataldo Musto, Mathias Bauer

Organized by



UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

DIPARTIMENTO DI
INFORMATICA

Università degli Studi di Bari 'Aldo Moro'

Department of Computer Science

Bari

Italy

Giovanni Semeraro, Cataldo Musto, Mathias Bauer, editors

Proceedings of the 2nd International Workshop on Personalization & Recommender
Systems in Financial Services

June 16, 2016, Bari, Italy

Chairs

Giovanni Semeraro, University of Bari, Italy

Mathias Bauer, mineway GmbH, Germany

Program Committee

Mathias Bauer, mineway GmbH, Germany

Shlomo Berkovsky, NICTA, Australia

Paul Blazek, cyLEDGE, Vienna, Austria

Robin Burke, DePaul University, IL, USA

Kuan-Ta Chen, Academia Sinica, Taiwan

Marco de Gemmis, University of Bari, Italy

Tommaso di Noia, Polytechnic of Bari, Italy

John O'Donovan, University of California Santa Barbara, CA, USA

Alessandro Giuliani, University of Cagliari, Italy

Dietmar Jannach, TU Dortmund, Germany

Pasquale Lops, University of Bari, Italy

Cataldo Musto, University of Bari, Italy

Eetu Mäkelä, Aalto University, Finland

Fedelucio Narducci, University of Bari, Bari, Italy

Giovanni Semeraro, University of Bari, Italy

Juha Tiihonen, Aalto University, Finland

Markus Zanker, Alpen-Adria-Universitaet Klagenfurt, Austria

David Zibriczky, Budapest University of Technology and Economics, Hungary

Proceedings Chair

Cataldo Musto, University of Bari, Italy

Organizational Support

Vincenzo Tamburrano, Italy

Preface

Personalization and recommendation technologies provide the basis for applications tailored to the needs of individual users. These technologies play an increasingly important role for financial service providers, in addition to several firms of the digital economy.

According to a recent paper (N. Leavitt, [A Technology that Comes Highly Recommended](#), IEEE Computing now, April 8, 2013), recommender systems technology “is used by shopping websites such as Amazon, which receives about 35 percent of its revenue via product recommendations.

It is also used by coupon sites like Groupon; by travel sites to suggest flights, hotels, and rental cars; by social-networking sites such as LinkedIn; by video sites like Netflix to recommend movies and TV shows, and by music, news, and food sites to suggest songs, news stories, and restaurants, respectively.

Even financial-services firms recently began using recommender systems to provide alerts for investors about key market events in which they might be interested”.

According to a [Bloomberg Business news](#) appeared on March 2015, funds run by robots account for 400 Billion Dollars.

The aim of this workshop was to bring together researchers and practitioners working in financial services related areas in order to: (1) understand and discuss open research challenges, (2) provide an overview of existing technologies, and (3) provide a basis for information exchange between industry and academia.

Giovanni Semeraro, Mathias Bauer

Content

<i>Paper</i>	<i>Page</i>
A cognitive business is a business that thinks Pietro Leo, invited talk	1-2
Recommender Systems meet Finance: A literature review Dàvid Zibriczky	3-10
Forecasting out-of-the-ordinary financial events Marco Brambilla, Davide Greco, Sara Marchesini, Luca Marconi, Mirjana Mazuran, Martina Morlacchi Bonfanti, Alessandro Negrini and Letizia Tanca	11-15
Online User Behavioural Modeling with Applications to Price Steering Van Tien Hoang, Vittoria Cozza, Marinella Petrocchi and Rocco De Nicola	16-21
Application of Constraint-based Technologies in Financial Services Recommendation Alexander Felfernig, invited talk	22-23
The Art of Spending and Recommendations in Personal Finance Leonid Ivonin, Mark Perry and Sriram Subramanian	24-28
NormaSearch: a Big Data application for financial services Ylenia Maruccia, Giovanni Pansini, Gloria Polimeno and Felice Vitulano	29-32
A Smart Financial Advisory System exploiting Case-Based Reasoning Luigi Portinale, Giorgio Leonardi, Paolo Artusio and Marco Valsania	33-40

A cognitive business is a business that thinks

Pietro Leo¹

Abstract. The talk will start and go around this claim: “Could a business think?” and mainly explore what cognitive computing is starting to provide an added value as diverse industries. Main focus will be how cognitive computing is impacting financial services and how it could contribute to reshape that industry

¹ IBM. email: pietro_leo@it.ibm.com

Recommender Systems meet Finance: A literature review

Dávid Zibriczky^{1,2}

Abstract. The present work overviews the application of recommender systems in various financial domains. The relevant literature is investigated based on two directions. First, a domain-based categorization is discussed focusing on those recommendation problems, where the existing literature is significant. Second, the application of various recommendation algorithms and data mining techniques is summarized. The purpose of this paper is providing a basis for further scientific research and product development in this field.

1 INTRODUCTION

Recommender Systems [63] are information filtering and decision supporting systems that present items in which the user is likely to be interested in a specific context. We consider *users* the active entities that perform *interactions* (e.g. viewing, purchasing, rating, etc.) in the system. We call *items* the objects with which the user can interact (e.g. products, movies, songs, etc.). The parameter setting that characterizes the environment (e.g. time, device, location) is defined as *context*; furthermore, we consider the actual preferences (e.g. filters, rules, item types) as *constraints* of the recommendations. Both users and items can be described by *metadata* (e.g. age, gender for users; genre, price for items). Recommender systems apply several data mining algorithms such as popularity-based methods, collaborative- [67] and content-based filtering [58], hybrid techniques [9], knowledge-based methods [79, 24] or case-based reasoning [74] depending on the characteristics of the domain, the quality of available data and the business goals.

Recommendation services offer several level of personalization, starting from manually defined "editorial picks" to complex context-aware hybrid solutions. Businesses often mix various types of carousels in the same page to cover diversified collection of recommendations. Although the majority of the recommender algorithms focuses on capturing user preferences, non-personalized techniques can also be considered as building blocks of a complex service (e.g. first carousel shows personalized recommendations, the second one contains the most popular items in the last week).

Recommender systems appeared in the mid-1990s, however, they are receiving significant attention since the Netflix Prize [3]. Nowadays, recommender systems are applied in a very broad scale of domains [48] such as movies (Netflix), books (Amazon) or music (Spotify). Generally speaking, recommender systems are useful in any domains, where a significant amount of choice exists in the system and users are interested in just a small portion of items.

Compared to the subjects of conventional recommender systems, financial products usually require a long-term significant financial commitment as their utility is not realized immediately depending

on several external factors (like market returns, governmental regularizations, currency, etc.); furthermore, expert knowledge is necessary to judge which one is a good choice. In order to reduce the risk of such a choice, users tend to formulate stricter expectations to these products than to conventional e-commerce ones, thus applying a recommender system in financial domains is a challenging task. Users typically protect their personal data, which is especially true for financial services, causing privacy risk issues in recommender systems [61, 17] and requiring more complex alternative personalization methods. As privacy issues are significant in financial services, personal metadata and individual transactional data are often missing, which causes user cold-start problem for recommender systems.

From a business perspective, a common challenge that several financial institutions are facing is the lack of an intelligent decision support system [13]. As sales activities of financial products requires expert knowledge, recommender systems offer great benefits for financial services by either improving the efficiency of sales representatives or automatizing decision making process for the clients. Therefore, a significant demand is observed for these decision support systems.

In this literature review, we investigate the existing application of recommender system techniques focusing on the financial domains. First, we perform domain-based categorization, distinguishing the most developed fields; then we discuss the applications in less developed financial domains. Second, we summarize the most often applied recommender system methods and additional techniques that are indirectly used for recommendations.

2 DOMAIN-BASED REVIEW

In our terminology, a financial domain is a specific area of finance that can be properly identified, modeled and developed based on its specific properties. For example, we consider stocks and portfolios as two different domains in this context, because in the first case an individual stock should be recommended, but in the second one a composition of financial assets should be selected, which is a different recommendation scenario. Based on the work of Burke and Ramezani [10], a domain can be characterized by the following aspects: (1) *heterogeneity* that captures the diversity of items' properties in a domain, (2) *churn* that characterizes the level of novelty and expected lifespan of the items, (3) *interaction style* that describes how the users are able to express their preference, (4) *preference stability* that characterizes the degree of variation of user preferences over time, (5) *risk* that determines the expected tolerance of the users for false recommendations and (6) *scrutability* that refers to the demand for explanation of recommendations.

In the following subsections, we propose a categorization of scientific contribution in financial services considering these properties. First, we introduce the applications in online banking systems and

¹ Department of Finance, Budapest University of Technology and Economics, Hungary, email: zibriczky@finance.bme.hu

² ImpressTV, Hungary, email: david.zibriczky@impressstv.com

we discuss two general-purpose multi-domain solutions. Second, we walk through on well-defined financial products such as loans, insurance policies and riders, real estate and stocks. Third, we introduce the standard portfolio selection problem and we discuss various techniques of personalized asset allocation. Finally, we collect other less studied domains.

2.1 Online banking and multi-domain solutions

By the rapid growth of information technology, the banking industry changed significantly in the last decade. With the spreading of on-line payment solutions in various devices, a massive online data flow appeared in bank systems centralizing data from multiple domains. Banks are forced to change technologies that is capable to handle big data and exploit business value from the massive information flow. Yahyapour [84] and Asosheh et al. [2] investigate the introduction of recommender systems into Iranian banking system using Technology Acceptance Model. Based on the results of their questionnaire, there is a significant willingness to introduce such a solution in banking systems, which primarily depends on perceived ease of use, usefulness and the bank's attitude.

In order to exploit the value of contextual information of transactional data, Gallego and Huecas [30] and Vico and Huecas [81] developed context-aware recommender prototypes. Based on credit card using history and geolocation data, they implemented a clustering-based method that provides personalized recommendation about money spending opportunities close to the user. They find high user satisfaction of using such a solution; however, they also consider the importance of privacy issues. Fano and Kurth [22] introduce a concept of interactive management tool that assists in personal resource (money) allocation. For the optimization of this objective, they propose an algorithm, which considers expenses, financial goals and time of attainment. Yu [86] introduces a prototype of online personal finance management tool, which is capable to provide insurance planning, asset allocation and investment recommendation. Overall, a number of works are published for banking sector; however, all of them seem to be non-production concept only.

Felfernig et al. [27, 26] present two general-purpose knowledge-based recommender systems with intelligent user interface, which can be flexibly applied on various financial products. The authors prefer knowledge-based algorithms over the conventional collaborative- and content-based filtering, because they can be applied more efficiently in multi-criteria-based financial decisions. For those cases, when no results can be shown for a multi-constraint setting, Felfernig and Stettinger [28] propose a constraint diagnosis and repairing technique.

Related to online banking and multi-domain solutions, the products are basically heterogeneous. The churn rate depends on the type of items accessed by these systems; however, we consider it low in banking environment. As these solutions offer interactive user interfaces, the interactions are explicit. We argue that the user preference is unstable, because it strongly depends on the actual goal of the user. These systems focus on money management and spending opportunities, thus we identify high risk and significant demand for explanation.

2.2 Loan

A *loan* is lending money from one entity (individual or organization) to another one with specified conditions. Under a loan product, we mean a debt with a promissory note specifying the amount of money

borrowed, the interest rate and the dates of payment. In this domain, the recommendation problem is finding the right product of the loan company for the borrower, which both satisfies his financial needs and will be likely to be paid back by the borrower. Felfernig et al. [25] propose a real-time constraint-based recommender application that supports sales between the representatives and consumers focusing on loan recommendation problem.

Microfinance is a type of banking service that supports low-income individuals and groups, who would otherwise have no opportunity to borrow money. In the last couple of years, the peer-to-peer (P2P) lending became popular, in which individuals or groups have opportunity to invest money by lending to another parties using a P2P lending marketplace. In this context, the recommendation task is to find an appropriate pairing between the lenders and individuals who need loans. Choo et al. [15] propose a maximum-entropy-based recommendation method to solve this problem using the dataset of Kiva P2P lending marketplace. Lee et al. [43] also developed a solution for Kiva, using collaborative filtering techniques for finding a fair pairing of microfinance. Significant work is published by Guo et al. [37], who formulate an instance-based credit risk assessment model for evaluating risk and return of each individual loan. San Miguel et al. [66] introduce a P2P loan recommendation method via social network. They design a data framework architecture, which is capable to integrate both public and private data dealing with privacy issues. Bhaskar and Subramanian [7] introduce an adaptive recommender system that assists microfinance institutions. They discuss the impact and limitations of such a system in an Indian case study.

Based on the properties of this domain, we argue that loans are less heterogeneous; however, we distinguish between basic loan products and microfinance solutions. We think that the churn rate for conventional products is low, but for microfinance is typically higher. The interaction type is explicit for both opportunities and the individual transactions are rare. We argue that the preference of a user is unstable, because the demand for loans can change by personal financial status. Loans are definitely risky products; therefore, the explanation of recommendations is required.

2.3 Insurance

In the insurance domain, an *insurance policy* is a contract between the insurer and the insured (policyholder). For an initial payment (premium), the insurer takes obligation to pay compensation for insured if loss caused by perils under the terms of policy. As standard policies have little room for customization, *insurance riders* are introduced to extend benefits that is purchased separately from the basic policy. Both insurance policy and insurance rider can be the object of personalized recommendation problem.

Mitra et al. [52] discuss a high-level concept of recommending both insurance policies and riders. In their short paper, they summarize the potential business benefits of introducing recommender systems in this domain. For insurance policy recommendation, Rahman et al. [60] implemented a real-time web-based application. They apply a case-based reasoning algorithm to support insurance sale agents to offer the most suitable policies for their clients. Another real-time cloud- and web-based application was developed by Abbas et al. [1], which recommends health insurance policies. The system applies multi-attribute utility-based theory that finds the most similar products to the preference of the user based various criteria (e.g. premium, co-pay, co-insurance, benefits). Life insurance recommendation problem is also investigated by Gupta and Jain [38]. Their short paper discusses the application of association rule mining for

such problem focusing on cold-start problem; however, it does not publish empirical results or architectural description. Rokach et al. [65] investigate the main domains of recommender systems comparing them to the insurance sector highlighting the main differences. In their work, they apply a basic item-to-item collaborative-filtering-based method as a possible solution for the recommending insurance riders.

Based on the study published by Rokach et al. [65], the insurance domain is quite small, the interactions are indirect and the attention span of the users is low; therefore, the size and quality of available dataset is low. The items are typically complex, the constraints of users are high; however, they have little expertise. We consider this domain homogeneous with low item churn rate. We think that the user preference is more stable for insurance than loans; however, we also note that a user is likely not to be interested in a same product after contracting one. Insurance products are less risky than loans, but the demand for explanation is still high.

2.4 Real estate

Real estate is a property consisting of the land, its natural resources and the buildings on it. The purchase of real estate is a rare and expensive transaction, which may be undertaken for investment or for personal residence. Therefore, buyers pay special attention to find the proper choice considering several various preferences, which leads to a multi-criteria decision problem. In this review, we primarily consider real estate as a type of investment.

The application of recommender systems in real estate domain has relatively weak literature, relevant papers were presented in the last five years only. One of the most significant contribution is published by Yuan et al. [87]. They propose a combination of ontological structure and case-based reasoning for real estate recommendation problem; furthermore, they implement a web-based application with map visualization interface. Daly et al. [18] introduce a transportation time calculator to extend conventional metadata of real estate. In their work, they also propose a method to find the trade-off between multi-criteria. Wang et al. [82] apply a simple similarity-based collaborative-filtering method for personalized ranking of real estate; however, their data were collected by questionnaires. Quantitative and qualitative criteria for decision making is investigated by Ginevičius et al. [33], who present a study about the application of recommender systems for real estate management. Another study is published by Kafi et al. [42], which discusses a "fuzzification" method on the metadata of real estate and the implementation of their solution, called Fuzzy Expert System.

As real estate can be described by the same well-defined features (e.g. price, size, rooms), this domain is homogeneous. We argue that the churn rate of items is significant, because items usually become unavailable after a purchase. The interactions can be both implicit (e.g. browsing) and explicit (e.g. purchasing), we argue that browsing data is frequent but purchase events are quite rare. We consider the preference of a user stable; however, it can change over the time in long term. Purchasing real estate is expensive and risky transaction, thus proper explanation is required.

2.5 Stocks

A *stock* is a type of security, which represents ownership in a company and claims on its assets, earnings and dividends. Stocks are traded in stock market, where the prices are controlled by traders' bids (buy price) and offers (sell price). They are held to gain profit

on both dividends and the difference of selling-buying price. As stock market can be volatile depending on economic events and market news, the estimation of future profit (utility) is very challenging task. Interpreting the recommendation problem in this context, those profitable stocks should be recommended to the investor that meet his risk-aversion preference and trading behavior.

2.5.1 Non-personalized stock recommendation

The application of decision support systems in stock market has significant literature. Most of the contributions focus on improving the accuracy of predicting future returns (or trends) [89, 47, 14], providing buy/sell signals [16, 83, 12] or introducing automatic trading solutions [19, 40]; however, majority of these papers ignore the personalization factor. Nonetheless, global ranking of available stocks can be considered as non-personalized recommendations. A number of papers pointed out on the observation that groups have greater knowledge than individuals and they can provide better market predictions, calling it the "wisdom of crowds" [39, 80, 36]. Eickhoff and Muntermann [20] present significant correlation between the prediction power of stock analysts and a set of social media users. Stephan and Von Nitzsch [75] report that individuals cannot beat the market substantially; however, inexperienced investors can take benefits from online communities. Several works consider the application of natural language processing methods on financial news [70, 69, 32, 49] and social networks texts [64, 4]. A comprehensive review about techniques of opinion mining and sentiment analysis is published by Ravi and Ravi [62].

2.5.2 Personalized stock recommendation

In order to provide personalized recommendations, individual information is required about the investor; however, explicit user preferences are not available in most of the cases. One way to overcome this difficulty is providing a user interface, where investor can specify his preferences. An early solution was implemented by Liu and Lee [45], which offers a set of features for analyzing and picking stocks based on preferences specified by the investor. Yoo et al. [85] propose a graphical user interface, which calculates personalized recommendations based on Moving Average Convergence Divergence (MACD) indicator and user interactions. Seo et al. [72] introduce a management tool that applies multiple agents to collect information about the stocks and provides stock recommendations based on what the investor is holding. Chalidabhongse and Kaensar [11] design a framework, which uses stochastic technical indicator on stock returns. The solution considers both explicit preferences and user interactions for personalized recommendations.

Some of the works assume that user attributes and individual user transactions are available in the data set. Yujun et al. [88] propose a stock recommender algorithm based on big order net inflow. They argue that using just big orders underscores low-valued stocks and reduce computational requirement for advanced algorithms. They introduce a fuzzy-based method, which recommends stocks that were selected by similar users. Taghavi et al [78] propose a concept of classical recommender system for ranking stocks. In their work, they combine hybrid techniques with various information collector agents. Although their concept is quite close to conventional recommender systems in e-commerce, they do not publish empirical results. The application of standard collaborative-filtering methods is also investigated by Sayyed et al. [68]; however, they present a preliminary concept only.

2.5.3 Characteristics of stock market

Due to its variability over time, stock market is more difficult to characterize than previous domains. We argue that stocks are heterogeneous, because they represent companies from various sectors. The churn rate is low, because companies leave stocks exchange very rarely. Considering bidding and trading transactions, the interaction style is rather implicit with very high volume. We argue that the user preference is unstable, because it is strongly driven by news and the ever changing global economy. Recommending stocks is very risky; therefore, a particular good explanation is required; however, it is a quite challenging task.

2.6 Asset allocation and portfolio management

A *portfolio* is a composition of finite number financial assets with various weights. It is well observed phenomena, that diversification reduces the risk of an investment, because the specific risk of each component become insignificant; therefore, portfolios offers better risk-return tradeoff than individual stocks. The technique of portfolio composition is often called *asset allocation*. In this context, the recommendation tasks are selecting assets and estimating their optimal weights in portfolio meet individual preferences and risk-aversion.

2.6.1 Modern Portfolio Theory

One of the most well-known portfolio selection model (Modern Portfolio Theory, MPT) was published by Markowitz [50]. His model can be interpreted as a two-step recommendation problem. First, well-diversified portfolios offer the best risk-return tradeoff for every risk level, these set of portfolios are the object of recommendation. Second, an investor is modeled by his risk-aversion utility function, which scores every investment opportunity based on risk and expected return. Investors select those portfolios that maximize his utility function. The practical drawback of this theoretical model is finding efficient portfolios requires complex calculation and estimating the individual utility function itself is challenging task.

Based on MPT, several works are published for asset allocation [73, 8]; however, the first concepts of automated solutions appears in the early 2000s. Elton and Gruber [21] argues that investors often make irrational decisions; therefore, automatized recommendations are advantageous for preventing irrational portfolio selections. Sycara et al. [77] present an overview of the application of intelligent agents in portfolio management. They highlight the specificity of this domain such as heterogeneity of information, dynamic change of environment, time-dependency and cost-constraints. Several researchers extend MPT by fuzzy techniques for modeling risk-aversion [90], estimating risk of portfolios [6] and composing optimal portfolios [23, 57]. For generating efficient portfolios, Nanda et al. [55] integrate a stock clustering method, Raei and Jahromi [59] apply two types of multi-criteria decision methods. Although the aforementioned works propose various type of sophisticated portfolio weighting methods, they are just non-personalized models.

2.6.2 Personalized portfolio selection

Musto et al. [54, 71, 53] propose a case-based reasoning methodologies for asset allocation, which consider user metadata for personalization. In their work, recommended portfolios are calculated based on what similar users selected applying various combining strategies. The authors provide empirical results of neighbor selection- and asset

allocation methods in terms of average yield and intra-list-diversity of portfolios. Garcia-Crespo et al. [31] and Gonzalez-Carrasco et al. [34] introduce a fuzzy model that transforms the ontology of investor (education, age, income, risk-aversion, etc.) and the ontology of portfolio (market risk, interest rate, liquidity, returns, etc.) to a unified bi-dimensional matrix, where dimensions are psychological and social behavior features. Portfolios are recommended based on the distance of investor and portfolio models. The authors also discuss the architecture the solution and compare the value of applied accuracy measures with other domains. Beraldi et al. [5] present a decision support system for assisting strategic asset allocation using stochastic optimization method. In their solution, an investor can define his strategy by setting its parameters (initial cash, period, type of assets and currency). Based on these criteria, portfolios are generated maximizing the tradeoff between expected final wealth, Conditional Value at Risk and risk aversion parameter. The authors provide a detailed high-level architecture and performance measurement of their solution.

2.6.3 Characteristics of portfolio management

As portfolios can contain various assets, the portfolio management is heterogeneous. Although the churn rate may vary by the type of domains, we consider it low, because the assets are purchased for long-term investment. On the other hand, portfolios are basically unique and they always change if reallocation is performed. Assuming an interactive user interface, the interaction type is explicit, because investors can specify both their preferences or the desired weight of assets in portfolios. The stability of user preference may vary over time, but it is less unstable than stock exchange, because portfolios are typically composed for long-term investment. The risk of such investment is still high and explanation is desired in this domain.

2.7 Other financial domains

In this subsection, we discuss the financial domains that have weak literature in recommender systems. We mention only the most significant differences in characteristics from the aforementioned domains.

An emerging domain of investment opportunities is venture finance. *Venture capital* is a type of private equity that is offered for startup companies as seed funding. This kind of investment is typically risky, but expects high returns on promising companies. As companies typically need only a few rounds of funding, the item churn is high in this case. The goal in this domain is to find an advantageous matching between the venture capital firms and their investment partners. Related to this problem, Stone et al. [76] published a relevant work focusing on the application of collaborative filtering. They report that the domain is characterized by extremely sparse long-tailed data, thus the efficient use of conventional recommender system methods is challenging. Continuing their work, Zhao et al. [91] investigate diversification techniques in this field. The authors propose 5 algorithms for ranking startups and a quadratic portfolio weight optimization method considering risk-aversion levels.

Stock fund is a fund that principally invests in stocks. The composition of stock fund is defined by fund manager focusing on a certain sector or a level of risk. Due to its diversification level, stock funds are less risky than stocks; however, they often cannot be traded in stock market thus the amount of transactions is low. Matsatsinis and Manarolis [51] introduce a hybrid application for stock fund recommendation problem. To reduce the sparsity issues, they propose the combination of collaborative filtering and multi-criteria decision

analysis. Lacking individual real data on transactions, they evaluate the proposed model on simulated investment behavior.

Jannach and Bundgaard-Joergensen [41] apply knowledge-based techniques to design a web-based advisory tool to improve the completeness of a *business plans*. In this context, the personalization of related questions is considered as a type of recommendation problem. The application also provides a summary of financials, level of completeness and aggregated advices. The risk of recommendation is low and the explanation is not critical in this case.

3 METHOD-BASED REVIEW

In this section, we categorize relevant scientific contributions based on the applied methodologies. First, we walk through the standard recommendation methods such as collaborative-filtering, content-based filtering, knowledge- and case-based recommender systems. Second, we discuss various hybrid techniques and additional data mining and machine learning methods that indirectly applied for recommendation problems in financial services. Further domain-related studies, architectures and user interface designs are not discussed in this section.

3.1 Collaborative filtering

One of the most often used technique in recommender systems is *collaborative filtering (CF)* [67]. As this method require interactions only, it can be applied in various domains. Collaborative filtering is able to extract latent behavioral pattern in transactional data that cannot be modeled by metadata; therefore, collaborative filtering methods usually have higher accuracy than metadata-based methods. On the other hand, their efficiency strongly depends on the sparsity of data and the novelty of items (cold-start problem); furthermore, it is quite challenging to explain the output of CF algorithms, which is a strong disadvantage for risky financial domains.

Among collaborative filtering-based solutions, the majority of works apply item-based nearest-neighbor methods for recommending insurance riders [65], real estate, [82] and venture capital [76]. We also find preliminary concept of the application of similarity-based recommendations for stock market [68]. Lee et al. [43] apply matrix factorization for Bayesian personalized ranking in micro-finance services. They propose a fairness-aware optimization with stochastic gradient descent (SGD). A significant contribution is published by Zhao et al. [91], who propose five different collaborative-filtering methods for venture capital domain. CF is also applied in several other hybrid methods; however, we discuss those in a later section.

3.2 Content-based filtering

Content-based filtering (CBF) [58] recommends items based on the metadata of items in user history and other available items; therefore, this method requires metadata and individual interactions only. CBF algorithms can cope with the cold start problem and their recommendations are easy to explain by meta words; however, the models strongly rely on the quality of metadata and they are usually less accurate than collaborative filtering methods.

We find that the metadata-based recommendation problem is usually associated with *multiple-criteria decision analysis (MCDA)* [29]. Due to the complexity of real estate selection problem, MDCA models are often applied in that field. Ginevičius et al. [33] propose a model that handles quantitative and qualitative criteria for real estate

management. Daly et al. [18] presents housing recommender system, which considers not just the metadata of a home, but the transportation opportunities to the user specified locations. A metadata-based solution for peer-to-peer lending is proposed by San Miguel et al. [66]; however, it is different from the conventional content-based filtering. The authors introduce a framework that capable to represent user data in vector-based- and semantic user models. We conclude that pure metadata-based methods are not typical in financial domains.

3.3 Knowledge-based recommendation

Knowledge-based recommender systems (KBRS) [79] focus on formalizing the knowledge about a domain based on its specificity, various constraints and ontology of items. The information about a user is usually collected by a knowledge acquisition interface, personalized recommendation is calculated based on the representation of knowledge about the user and available items. The advantage of knowledge-based methods is that the recommendations rely only on the domain-knowledge and constraints of the user preferences; furthermore, they are easy to be explained. On the other hand, the knowledge base itself should be built up and maintained, which can be a significant overhead in operating such an interactive decision support systems and the conflict should be resolved by heuristics when there is no matching item based on the actual constraints [28]. As knowledge-based methods are able to handle complex user preferences that is typical for financial domains, they can be potentially effective solutions assuming that the knowledge acquisition interface is implemented and knowledge about the domain is acquired. Felfernig et al. propose several solutions for recommending various financial products using constraint-based reasoning, which is a type of knowledge-based methods [27, 25, 26]. KBRS is also applied for personalizing questions of business plan analysis [41].

3.4 Case-based recommendation

Case-based recommender systems (CBRS) [46, 74] apply case-based reasoning (CBR) that solves the recommendation problem based on old similar cases. A case is defined in various ways (like product description, user preference, search criteria and outcome of case). CBRS relies on the first two step of case-based reasoning, which is (1) *retrieve* that finds relevant old cases to the current case and (2) *reuse* that applies the knowledge from relevant old cases. An actual case of the user is defined by user profile data or via interactive user interface. In order to find similar cases, similarity of attributes, collaborative patterns or knowledge of the domain are usually applied. On one hand, CBRS can be used for complex problems and it provides explainable recommendations. Based on Musto et al. [53], CBR has better properties than collaborative filtering for financial domains. On the other hand, these methods require a significant amount of data about the cases.

In financial domains, we find a number of case-based recommender systems. Rahman et al. [60] propose a CBR-based application for recommending insurance policies. Musto et al. [54, 71, 53] introduce case-based reasoning for portfolio recommendation. In their works, the authors also propose a diversification technique for weighting candidate solutions in revise step. Yuan et al. [87] introduce a real estate recommender that combines case-based reasoning and ontology of items. Guo et al. [37] applies instance-based method for peer-to-peer recommendation problem and employ kernel regression to find similarity weights of instances in the past.

3.5 Hybrid methods

We consider the combination of the different decision support methods as *hybrid method* [9]. Generally, hybrid recommenders benefit from the advantages of applied techniques, while their weaknesses are reduced. Hybrid methods can be more precise than conventional models; however, the efficient implementation of such solutions can be very difficult for complex problems.

We find hybrid solutions that incorporate credit card transactions in various domains to provide context-aware recommendations based on the location of the user [30, 81]. We argue that hybrid filtering is an efficient solution for cross-domain recommendation. Another hybrid application focuses on finding the most profitable stocks at a right time based on the investor preference [78]. They apply collaborative- and content-based filtering in algorithm level and social, economic and semantical agents in system level. CF and CBF is also combined by Mitra et al. [52] for recommending insurance product and by Choo et al. [15] for microfinancing. In order to reduce sparsity issues for stock fund recommendation, Matsatsinis and Manarolis [51] propose a combination of collaborative filtering and multi-criteria decision analysis.

There are a few applications of *association rule mining (ARM)* [44] in financial domains. A web-based hybrid association rule mining method is proposed for personalized recommendation of insurance products, which also deals with cold-start problem [38]. ARM is used in stock market for predicting trading-based relationships between stocks [56].

3.6 Complementary methods

In this section, we also discuss additional complementary techniques that are integrated to conventional recommender methods. We find that *fuzzy methods* are primarily introduced for stock market and asset allocation. Yujun et al. [88] introduce a fuzzy-based clustering for stock recommendations. A fuzzy-based transformation is introduced by Garcia-Crespo et al. [31] and Gonzalez-Carrasco et al. [34] for portfolio recommendation problem. Fuzzy-based expert systems are proposed for real-estate- [42] and portfolio recommendations [23, 35]. Several variations of fuzzy-based extensions of modern portfolio theory are introduced [90, 6, 57].

We find applications of *artificial neural networks (ANN)* for designing trading decision support systems [16] and extracting information from news [32]. In stock price forecasting, semantic methods are also considered for processing web texts [70] and emotions expressed in Twitter messages [64]. Based on our research, classification methods are usually applied for stock markets. *Support vector machines (SVM)* are used for incorporating information from financial news [69, 49], forecasting stock returns [89, 47] and providing stock buy/sell signals [83].

4 CONCLUSION

In this review, we have discussed the scientific contributions that were addressed to the recommendation problems in financial services in the last 15 years. We have performed a two-way investigation based on financial domains and applied recommendation techniques.

Considering the domains, our finding is the following. Banking institutes have a significant willingness to introduce decision support systems; however, we find just concepts for that problem. There is a great support for personalizing peer-to-peer lending than conventional loan services. Although insurance domain is small, we find

a decent number of applications recommending both insurance policies and riders. There are a few papers dealing with real estate recommendation; a decent part of them is empirical study only. There is a huge literature dealing with stock market. A significant part of publications focuses on predicting stock prices and providing buy/sell signals; however, these methods are non-personalized. Several works introduce interactive user interface for managing stocks, but only a few number of papers propose machine learning methods for personalized stock recommendation. We also find a significant literature for asset allocation. On the basis of modern portfolio theory, several methods are introduced to find efficient portfolios for various risk-aversion levels; however, the personalization is realized in selecting risk level only. Some of the works apply machine learning methods to compose personalized portfolios based on individual attributes. Furthermore, we present promising applications of recommender systems for venture finance, stock funds and business plan-related questionnaire.

Several domains can be characterized by homogeneous products; however, we argue that stock exchange, portfolio management and multi-domain solutions are rather heterogeneous. The item churn rate is basically low among the financial domains, except for real estate, where the offers are available until only one transaction by nature. Assuming that user interface is provided, the interaction style is explicit, otherwise implicit data or user profile metadata can be used only. We find that the preference stability is various in these domains depending on individual financial status and the changes of global market. As the object of recommendations are usually related to money spending transactions, we consider all financial domains; therefore, the demand for proper explanation about the recommendations is significant.

Based on our method-based analysis, we conclude that collaborative filtering is applied in various domains where the product itself is well-defined; however, it is limited to handle complex recommendation problems. We find a small number of applications using pure content-based filtering. Due to the specificity of financial domains, multiple-criteria decision analysis and case-based reasoning has significant advantage over collaborative- and content-based filtering. Assuming that a well designed user interface is available, knowledge-based methods has great benefits for assisting personalization problems. We find several hybrid methods combining collaborative- and content-based filtering, we argue that application of association rules is less significant. Investigating other methods, we find that fuzzy techniques are basically applied for portfolio selection problem; furthermore, artificial neural networks and support vector machines are typically used in stock market decision systems.

Summarizing our work, we state that an extensive work is being in progress for investigating applications of recommendation systems in financial services; however, there remain several unexploited opportunities in this field for both scientific research and product development.

ACKNOWLEDGEMENTS

I am grateful to ImpressTV for funding this research and supporting me by flexible work hours. I thank Mihály Ormos from Budapest University of Technology and Economics for giving valuable advices for literature review. Least but not least, I am thankful to the organizers of 2nd International Workshop on Personalization and Recommender Systems in Financial Services for extending the deadline of submission, providing valuable feedbacks and encouraging me to finish this paper.

REFERENCES

- [1] Assad Abbas, Kashif Bilal, Limin Zhang, and Samee U Khan, 'A cloud based health insurance plan recommendation system: A user centered approach', *Future Generation Computer Systems*, **4344**, 99–109, (feb 2015).
- [2] Abbas Asosheh, Sanaz Bagherpour, Nima Yahyapour, and A Asosheha, 'Extended Acceptance Models for Recommender System Adaption, Case of Retail and Banking Service in Iran', *WSEAS Transactions on Business and Economics*, **5**(5), 189–200, (2008).
- [3] James Bennett and Stan Lanning, 'The netflix prize', in *Proceedings of KDD cup and workshop*, volume 2007, p. 35, (2007).
- [4] Janek Benthaus and Roman Beck, 'It's more about the Content than the Users! The Influence of Social Broadcasting on Stock Markets', in *European Conference on Information Systems*, (2015).
- [5] P Beraldi, A Violi, and F De Simone, 'A decision support system for strategic asset allocation', *Decision Support Systems*, **51**(3), 549–561, (jun 2011).
- [6] J D Bermudez, J V Segura, and E Vercher, 'A fuzzy ranking strategy for portfolio selection applied to the Spanish stock market, 2007'.
- [7] Tarun Bhaskar and Gopi Subramanian, 'Loan recommender system for microfinance loans: Increasing efficiency to assist growth', *J Financ Serv Mark*, **15**(4), 334–345, (mar 2011).
- [8] Michael J Brennan, Eduardo S Schwartz, and Ronald Lagnado, 'Strategic asset allocation', *Journal of Economic Dynamics and Control*, **21**(89), 1377–1403, (jun 1997).
- [9] Robin Burke, 'Hybrid recommender systems: Survey and experiments', *User modeling and user-adapted interaction*, **12**(4), 331–370, (2002).
- [10] Robin Burke and Maryam Ramezani, 'Matching Recommendation Technologies and Domains', *Recommender Systems Handbook*, **54**, 367–386, (2011).
- [11] T H Chalidabhongse and C Kaensar, 'A Personalized Stock Recommendation System using Adaptive User Modeling, 2006'.
- [12] Ya-Wen Chang Chien and Yen-Liang Chen, 'Mining associative classification rules with stock trading data: A GA-based method', *Knowledge-Based Systems*, **23**(6), 605–614, (aug 2010).
- [13] Hilary Cheng, Yi-Chuan Lu, and Calvin Sheu, 'An ontology-based business intelligence application in a financial knowledge management system', *Expert Systems with Applications*, **36**(2, Part 2), 3614–3622, (mar 2009).
- [14] Vincent Cho, 'MISMIS: A comprehensive decision support system for stock market investment', *Knowledge-Based Systems*, **23**(6), 626–633, (aug 2010).
- [15] Jaegul Choo, Daniel Lee, Bistra Dilkina, Hongyuan Zha, and Haesun Park, 'To Gather Together for a Better World: Understanding and Leveraging Communities in Micro-lending Recommendation', in *Proceedings of the 23rd International Conference on World Wide Web, WWW '14*, pp. 249–260, New York, NY, USA, (2014). ACM.
- [16] Seng-Cho Timothy Chou, Chau-Chen Yang, Chi-Huang Chan, and Feipei Lai, 'A rule-based neural stock trading decision support system, 1996'.
- [17] Lorrie Faith Cranor, 'I didn't buy it for myself', in *Designing personalized user experiences in eCommerce*, 57–73, Springer, (2004).
- [18] Elizabeth M Daly, Adi Botea, Akihiro Kishimoto, and Radu Marinescu, 'Multi-criteria Journey Aware Housing Recommender System', in *Proceedings of the 8th ACM Conference on Recommender Systems, RecSys '14*, pp. 325–328, New York, NY, USA, (2014). ACM.
- [19] L Dymova, P Sevastianov, and K Kaczmarek, 'A stock trading expert system based on the rule-base evidential reasoning using Level 2 Quotes', *Expert Systems with Applications*, **39**(8), 7150–7157, (jun 2012).
- [20] Matthias Eickhoff and Jan Muntermann, 'Stock Analysts Vs. The Crowd: A Study on Mutual Prediction', (2015).
- [21] Edwin J Elton and Martin J Gruber, 'The Rationality of Asset Allocation Recommendations', *The Journal of Financial and Quantitative Analysis*, **35**(1), 27–41, (2000).
- [22] Andrew Fano and Scott W Kurth, 'Personal Choice Point: Helping users visualize what it means to buy a BMW', *Control*, 46–52, (2003).
- [23] Mehdi Fasaghari and Gholam Ali Montazer, 'Design and implementation of fuzzy expert system for Tehran Stock Exchange portfolio recommendation', *Expert Systems with Applications*, **37**(9), 6138–6147, (sep 2010).
- [24] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker, 'Constraint-Based Recommender Systems', chapter Constraint, 161–190, Springer US, Boston, MA, (2015).
- [25] Alexander Felfernig, K Isak, K Szabo, and P Zachar, 'The VITA financial services sales support environment', *Proceedings of the National Conference on Artificial Intelligence*, **2**(Felfernig), 1692–1699, (2007).
- [26] Alexander Felfernig, Michael Jeran, Martin Stettinger, Thomas Absenger, Thomas Gruber, Sarah Haas, Emanuel Kirchengast, Michael Schwarz, Lukas Skofitsch, and Thomas Ulz, 'Human Computation Based Acquisition Of Financial Service Advisory Practices', *Organizational Support*, **27**, (2015).
- [27] Alexander Felfernig and A Kiener, 'Knowledge-based interactive selling of financial services with FSAdvisor', *Proceedings of the National ...*, **100**, 1475–1482, (2005).
- [28] Alexander Felfernig and Martin Stettinger, 'Conflict Management in Interactive Financial Service Selection', *Organizational Support*, **3**, (2015).
- [29] José Figueira, Salvatore Greco, and Matthias Ehrgott, *Multiple criteria decision analysis: state of the art surveys*, volume 78, Springer Science & Business Media, 2005.
- [30] Daniel Gallego and Gabriel Huecas, 'An empirical case of a context-aware mobile recommender system in a banking environment', *Proceedings - 2012 3rd FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing, MUSIC 2012*, 13–20, (2012).
- [31] Ángel García-Crespo, José Luis López-Cuadrado, Israel González-Carrasco, Ricardo Colomo-Palacios, and Belén Ruiz-Mezcua, 'SIN-VLIO: Using semantics and fuzzy logic to provide individual investment portfolio recommendations', *Knowledge-Based Systems*, **27**, 103–118, (mar 2012).
- [32] Tomer Geva and Jacob Zahavi, 'Empirical evaluation of an automated intraday stock recommendation system incorporating both market data and textual news', *Decision Support Systems*, **57**, 212–223, (jan 2014).
- [33] Tomas Ginevičius, Jurgita Alchimovien, Paulius Kazokaitis, and Artras Kaklauskas, 'Recommender System for Real Estate Management', *Verstas: teorija ir praktika*, (3), 258–267, (2011).
- [34] Israel Gonzalez-Carrasco, Ricardo Colomo-Palacios, Jose Luis Lopez-Cuadrado, Ángel Garca-Crespo, and Belén Ruiz-Mezcua, 'PB-ADVISOR: A private banking multi-investment portfolio advisor', *Information Sciences*, **206**, 63–82, (nov 2012).
- [35] M Hassani Goodarzi, 'A web-based Implementation of a Portfolio Advisor System based on Fuzzy Expert Systems', in *Conf. Rec. IEEE 6th Int. Conf. Information & Communication Technology and System ICTS, Surabaya, Indonesia*, pp. 15–22, (2010).
- [36] Jörg Gottschlich and Oliver Hinz, 'A decision support system for stock investment recommendations using collective wisdom', *Decision Support Systems*, **59**, 52–62, (mar 2014).
- [37] Yanhong Guo, Wenjun Zhou, Chunyu Luo, Chuanren Liu, and Hui Xiong, 'Instance-based credit risk assessment for investment decisions in P2P lending', *European Journal of Operational Research*, **249**(2), 417–426, (mar 2016).
- [38] Abdhesh Gupta and Anwiti Jain, 'Life Insurance Recommender System Based on Association Rule Mining and Dual Clustering Method for Solving Cold-Start Problem', *International Journal of Advanced Research in Computer Science and Software Engineering*, **3**(7), 1356–1360, (2013).
- [39] Shawndra Hill and Noah Ready-Campbell, 'Expert Stock Picker: The Wisdom of (Experts in) Crowds', *International Journal of Electronic Commerce*, **15**(3), 73–102, (2011).
- [40] Yong Hu, Kang Liu, Xiangzhou Zhang, Lijun Su, E W T Ngai, and Mei Liu, 'Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review', *Applied Soft Computing*, **36**, 534–551, (nov 2015).
- [41] Dietmar Jannach and Uffe Bundgaard-Joergensen, 'SAT: A Web-Based Interactive Advisor for Investor-Ready Business Plans.', in *ICE-B*, eds., Joaquim Filipe, David A Marca, Boris Shishkov, and Marten van Sinderen, pp. 99–106. INSTICC Press, (2007).
- [42] Amir Hossein Kafi, Kazemipour Hamid, and Mohammad Ali Afshar Kazem, 'Design and Implementation of Fuzzy Expert System for Real Estate Recommendation', *International Journal of Information, Security and Systems Management*, **2**(1), 142–147, (jan 2013).
- [43] Eric L Lee, Jing-kai Lou, Wei-ming Chen, Yen-chi Chen, Shou-de Lin, Yen-sheng Chiang, and Kuan-ta Chen, 'Fairness-Aware Loan Recommendation for Microfinance Services', 1–5, (2014).
- [44] Weiyang Lin, Sergio A Alvarez, and Carolina Ruiz, 'Efficient adaptive-support association rule mining for recommender systems', *Data min-*

- ing and knowledge discovery, **6**(1), 83–105, (2002).
- [45] N K Liu and K K Lee, 'An intelligent business advisor system for stock investment', *Expert Systems*, **14**(3), 129–139, (1997).
 - [46] Fabiana Lorenzi and Francesco Ricci, 'Case-based Recommender Systems: A Unifying View', in *Proceedings of the 2003 International Conference on Intelligent Techniques for Web Personalization*, ITWP'03, pp. 89–113, Berlin, Heidelberg, (2005). Springer-Verlag.
 - [47] Chi-Jie Lu, Tian-Shyug Lee, and Chih-Chou Chiu, 'Financial time series forecasting using independent component analysis and support vector regression', *Decision Support Systems*, **47**(2), 115–125, (may 2009).
 - [48] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang, 'Recommender system application developments: A survey', *Decision Support Systems*, **74**, 12–32, (jun 2015).
 - [49] Chao Ma and Xun Liang, 'Online mining in unstructured financial information: An empirical study in bulletin news, 2015.
 - [50] Harry Markowitz, 'Portfolio selection', *The journal of finance*, **7**(1), 77–91, (1952).
 - [51] Nikolaos F Matsatsinis and Eleftherios A Manarolis, 'New Hybrid Recommender Approaches: An Application to Equity Funds Selection', in *Algorithmic Decision Theory*, 156–167, Springer, (2009).
 - [52] Sanghamitra Mitra, Nilendra Chaudhari, and Bipin Patwardhan, 'Leveraging Hybrid Recommendation System In Insurance Domain', **3**(10), (2014).
 - [53] Cataldo Musto, Giovanni Semeraro, Pasquale Lops, Marco de Gemmis, and Georgios Lekkas, 'Personalized finance advisory through case-based recommender systems and diversification strategies', *Decision Support Systems*, **77**, 100–111, (sep 2015).
 - [54] Cataldo Musto, Giovanni Semeraro, Pasquale Lops, and Marco De Gemmis, 'Financial Product Recommendation through Case-based Reasoning and Diversification Techniques', (2014).
 - [55] S R Nanda, B Mahanty, and M K Tiwari, 'Clustering Indian stock market data for portfolio management', *Expert Systems with Applications*, **37**(12), 8793–8798, (dec 2010).
 - [56] P Paranjape-Voditel and U Deshpande, 'An Association Rule Mining Based Stock Market Recommender System, 2011.
 - [57] Preeti Paranjape-Voditel and Umesh Deshpande, 'A stock market portfolio recommender system based on association rule mining', *Applied Soft Computing*, **13**(2), 1055–1063, (feb 2013).
 - [58] Michael J Pazzani and Daniel Billsus, 'Content-based recommendation systems', in *The adaptive web*, 325–341, Springer, (2007).
 - [59] Reza Raei and M Jahromi, 'Portfolio optimization using a hybrid of fuzzy ANP, VIKOR and TOPSIS', *Management Science Letters*, **2**(7), 2473–2484, (2012).
 - [60] S S A Rahman, A A Norman, and K J Soon, 'MyINS: A CBR e-Commerce Application for Insurance Policies', *Electronic Commerce Research*, **5**(1), 373–380, (2006).
 - [61] Naren Ramakrishnan, Benjamin J Keller, Batul J Mirza, Ananth Y Grama, and George Karypis, 'Privacy risks in recommender systems', *IEEE Internet Computing*, **5**(6), 54, (2001).
 - [62] Kumar Ravi and Vadlamani Ravi, 'A survey on opinion mining and sentiment analysis: Tasks, approaches and applications', *Knowledge-Based Systems*, **89**, 14–46, (nov 2015).
 - [63] Francesco Ricci, Lior Rokach, and Bracha Shapira, *Introduction to recommender systems handbook*, Springer, 2011.
 - [64] Marten Risius, Fabian Aklolk, and Roman Beck, 'Differential Emotions and the Stock Market-The Case of Company-specific Trading', (2015).
 - [65] Lior Rokach, Guy Shani, Bracha Shapira, Eyal Chapnik, and Gali Siboni, 'Recommending insurance riders', *Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13*, 253, (2013).
 - [66] Beatriz San Miguel, Jose M del Alamo, and Juan C Yelmo, 'A Personal Data Framework for Exchanging Knowledge about Users in New Financial Services', *Organizational Support*, 19, (2015).
 - [67] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, 'Item-based collaborative filtering recommendation algorithms', in *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295. ACM, (2001).
 - [68] Fr Sayyed, Rv Argiddi, and Ss Apte, 'Generating Recommendations for Stock Market Using Collaborative Filtering', *Ijces.Org*, **3**(1), 46–49, (2013).
 - [69] Robert P Schumaker and Hsinchun Chen, 'Textual Analysis of Stock Market Prediction Using Breaking Financial News: The AZFin Text System', *ACM Transactions on Information System*, **27**(2), 12:1–12:19, (2009).
 - [70] Vivek Sehgal and Charles Song, 'SOPS: Stock prediction using web sentiment', *Proceedings - IEEE International Conference on Data Mining, ICDM*, 21–26, (2007).
 - [71] Giovanni Semeraro and Cataldo Musto, 'Personalized Wealth Management through Case-Based Recommender Systems', 2–4, (2014).
 - [72] Young-Woo Seo, Joseph A Giampapa, and Katia Sycara, 'Financial news analysis for intelligent portfolio management', (2004).
 - [73] William F Sharpe, 'Asset allocation: Management style and performance measurement', *The Journal of Portfolio Management*, **18**(2), 7–19, (1992).
 - [74] Barry Smyth, 'Case-based recommendation', in *The adaptive web*, 342–376, Springer, (2007).
 - [75] Philipp Stephan and Rüdiger von Nitzsch, 'Do individual investors' stock recommendations in online communities contain investment value?', *Financial Markets and Portfolio Management*, **27**(2), 149–186, (2013).
 - [76] Thomas Stone, Weinan Zhang, and Xiaoxue Zhao, 'An empirical study of top-n recommendation for venture finance', *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13*, 1865–1868, (2013).
 - [77] K P Sycara, D Zeng, and K Decker, 'Intelligent Agents in Portfolio Management', in *Agent Technology SE - 14*, eds., NicholasR. Jennings and MichaelJ. Wooldridge, 267–281, Springer Berlin Heidelberg, (1998).
 - [78] Mona Taghavi, Kaveh Bakhtiyari, and Edgar Scavino, 'Agent-based computational investing recommender system', *Proceedings of the 7th ACM conference on Recommender systems - RecSys '13*, 455–458, (2013).
 - [79] Shari Trewin, 'Knowledge-based recommender systems', *Encyclopedia of library and information science*, **69**(Supplement 32), 180, (2000).
 - [80] M Velic, T Grzanic, and I Padavic, 'Wisdom of crowds algorithm for stock market predictions, 2013.
 - [81] Daniel Gallego Vico and Gabriel Huecas, 'Generating Context-aware Recommendations using Banking Data in a Mobile Recommender System', *ICDS 2012 : The Sixth International Conference on Digital Society Generating*, 73–78, (2012).
 - [82] Lei Wang, Xiaowei Hu, Jingjing Wei, and Xingyu Cui, 'A Collaborative Filtering Based Personalized TOP-K Recommender System for Housing', in *Proceedings of the 2012 International Conference of Modern Computer Science and Applications SE - 74*, ed., Zhenyu Du, volume 191 of *Advances in Intelligent Systems and Computing*, 461–466, Springer Berlin Heidelberg, (2013).
 - [83] Qinghua Wen, Zehong Yang, Yixu Song, and Peifa Jia, 'Automatic stock decision support system based on box theory and SVM algorithm', *Expert Systems with Applications*, **37**(2), 1015–1022, (mar 2010).
 - [84] Nima Yahyapour, 'Determining Factors Affecting Intention to Adopt Banking Recommender System', *Technology*, (2008).
 - [85] Jungsoon Yoo, Melinda Gervasio, and Pat Langley, 'An adaptive stock tracker for personalized trading advice', *IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces*, 197–203, (2003).
 - [86] Chien-Chih Yu, 'Designing A Consumer-Oriented Intelligent Decision Support System For Personalized Financial Planning Services', (2004).
 - [87] Xiaofang Yuan, Ji-Hyun Lee, Sun-Joong Kim, and Yoon-Hyun Kim, 'Toward a user-oriented recommendation system for real estate web-sites', *Information Systems*, **38**(2), 231–243, (apr 2013).
 - [88] Yang Yujun, Li Jianping, and Yang Yimei, 'An Efficient Stock Recommendation Model Based on Big Order Net Inflow', *Mathematical Problems in Engineering*, **2016**, (2016).
 - [89] Zhi-yong Zhang, Chuan Shi, Su-lan Zhang, and Zhong-zhi Shi, 'Stock Time Series Forecasting Using Support Vector Machines Employing Analyst Recommendations', in *Advances in Neural Networks - ISNN 2006 SE - 66*, eds., Jun Wang, Zhang Yi, JacekM. Zurada, Bao-Liang Lu, and Hujun Yin, volume 3973 of *Lecture Notes in Computer Science*, 452–457, Springer Berlin Heidelberg, (2006).
 - [90] Zili Zhang and Chengqi Zhang, 'An Agent-Based Hybrid Intelligent System for Financial Investment Planning', in *PRICAI 2002: Trends in Artificial Intelligence SE - 39*, eds., Mitsuru Ishizuka and Abdul Sattar, volume 2417 of *Lecture Notes in Computer Science*, 355–364, Springer Berlin Heidelberg, (2002).
 - [91] Xiaoxue Zhao, Weinan Zhang, and Jun Wang, 'Risk-Hedged Venture Capital Investment Recommendation', in *Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15*, pp. 75–82, New York, NY, USA, (2015). ACM.

Forecasting out-of-the-ordinary financial events¹

Marco Brambilla² and Davide Greco³ and Sara Marchesini² and Luca Marconi²
and Mirjana Mazuran² and Martina Morlacchi Bonfanti² and Alessandro Negrini² and Letizia Tanca²

Abstract. Being able to understand the financial market is very important for investors and, given the width and complexity of the topic, tools to support investor decisions are badly needed. In this paper we present Mercurio, a system that supports the decision-making process of financial investors through the automatic extraction and analysis of financial data coming from the Web. Mercurio formalizes the knowledge and reasoning of an expert in financial journalism and uses it to identify relevant events within financial newspapers. Moreover, it performs automatic analysis of financial indexes to identify relevant events related to the stock market. Then, sequential pattern mining is used to predict exceptional events on the basis of the knowledge of their past occurrences and relationships with other events, in order to warn investors about them.

1 Introduction

Financial data are daily produced and made available on the Web, therefore the possibility to process them allows us to model and study a world that is inherently complex due to the rules governing the financial market and to the internal and external factors influencing it. Investors constantly read financial news and analyze financial indexes, using their knowledge and experience to predict market events and make profitable investments. Our research aims at developing Mercurio, a decision support system to help investors during these activities.

Mercurio identifies relevant financial events, understands how they are related to each other and exploits this knowledge to predict future happenings. It uses: (i) the knowledge of an expert in financial journalism, whose deep understanding of the news does not consist of sole natural language processing and (ii) financial indicators that provide an objective overview of the stock and, more in general, of the companies' performances. On one hand, a domain expert knows "how to" read an article and understand its meaning, especially since its literal inspection might not coincide with the real meaning of what has happened. On the other hand, financial indicators provide an impartial overview of the past and current financial situation of companies. Financial happenings are all about signals and indications that companies leave behind along their life, and that the system must capture and interpret. Investment decisions are still made by human investors, and Mercurio provides them with more knowledge, possibly hidden to human observers, to improve their decision-making process.

Among the many financial data available on the web, Mercurio looks for those that convey "important" happenings, i.e., happenings

that influence and possibly shake the market: we call them *events*. Some of them are more relevant because they represent considerable changes of the financial market: we call them *catastrophes*, and they coincide with extraordinary financial moves (not necessarily negative, though), e.g. merger and acquisition, or other significant moves of the company management, or stockprice variations. The occurrence of a catastrophe is usually anticipated by "symptoms" that we call *signals*. For example, an investor might observe that often, before a crash, a company gives an interview stating that profits are increasing; from now on, whenever such an interview is published the expert will expect the related stock to fall in the stock market. Thus, an article containing an interview about increasing profit is a signal, while a stock crash is a catastrophe.

The paper is organized as follows: Section 2 briefly describes some proposals with aims similar to ours, Section 3 gives the details of the Mercurio system, Section 4 provides the current implementation state and, finally, Section 5 draws the conclusions we have currently reached and future research directions.

2 Related work

Market prediction always receives high interest in the financial literature: mostly, only numerical data are used, but some approaches exploit also textual information to increase the quality of input data and improve predictions.

Works in [3, 4, 5, 6, 7] use Automated Text Categorization techniques to predict short-term market reactions to news. Articles are categorized depending on the influence their publication has on financial indexes, and then correlated with financial trends and different approaches use different types of classifiers. Our approach differs from these as we use expert knowledge to determine the relevance of articles. Among the examined works, [8] has a similar goal as Mercurio, to find sequences of articles that anticipate a changing trend. Once again the focus is on numerical data, while we are interested in predicting strategically extraordinary financial moves.

Existing works are primarily data driven, however some proposals use a-priori knowledge about the application domain. Works in [9, 10] analyze financial articles and create a handcrafted thesaurus containing words that drive the stock prices and that are later used to predict stock prices. Similarly, [11] uses a-priori domain knowledge to predict interest rates: a cognitive map represents cause-effect relationships among the events in the domain and is used as the basis to retrieve the relevant news; these are then classified as either positive or negative according to the way they influence the rates. A work similar to ours is [12], where the objective is to predict the Tokyo stock exchange price using a-priori knowledge in the form of rules. Domain rules are defined eliciting non-numerical factors that influence the stock price, however these rules differ from ours as they

¹ This research is partially supported by the IBM Faculty Award "SOFIA: Semi-autOMatic Financial Information Analytics"

² Politecnico di Milano

³ Accentio

convey general knowledge about political and international events. On the contrary, we focus on financial and economic events typical of a company's life. The latter approaches differ from ours either in the way knowledge is represented or in the kind of knowledge adopted as background; we are currently trying to find a basis for an effective comparison, since the systems are not available and thus an experimental comparison on the same corpus is for the moment impossible.

To the best of our knowledge, a comprehensive system that makes use of both textual and numerical information to predict strategically extraordinary financial moves is still missing.

3 The Mercurio system

We envision an integrated and modular system that draws information from various sources and uses them appropriately with the final aim of predicting the happening of extraordinary financial events, that is, catastrophes. Finance is a kind of domain in which the key to successful data analysis is the integrated analysis of heterogeneous data, where time-dependent and highly frequent numerical data (e.g., price and volume) and textual data (e.g., news articles) should be considered jointly [13]. Both categories might encompass various data sources that can be easily added to the system (as shown in Figure 1). Each of the textual data sources is managed by an Event Recognizer that is able to extract events from the data and feed them into Mercurio. Events can be *catastrophes* (i.e. they convey considerable changes of the financial market) or *signals* (i.e. symptoms anticipating a catastrophe). Event recognition strategies vary depending on the type and nature of the managed data, for instance, each financial market (Italian, British, etc.) has its own language and dynamics, and there are differences also among financial newspapers of the same country.

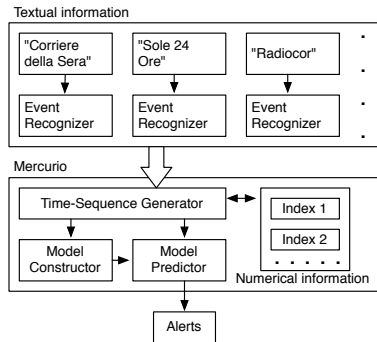


Figure 1. Mercurio architecture

In Mercurio, the events extracted from the financial news are received by the Time-Sequence Generator that arranges them on one or more timelines depending on the use the system has to make of them. If the aim is to construct a model from them all, then the Time-Sequence Generator creates a single timeline where all the received events are placed and provides this timeline as input to the Model Constructor. On the other hand, if the aim is to predict the future happenings related to specific companies, each created timeline contains only events related to a specific company, and inputs these data to the Model Predictor.

The Model Constructor module takes a sequence of events and uses Sequential Pattern Mining techniques to find frequent subsequences of events and thus creates a model of the data represented in terms of a set of sequential patterns. These patterns, together with

the timeline of a company are taken as input by the Model Predictor module that uses them to forecast the happening of a certain catastrophe with respect to a certain company. The output provided by the Model Predictor is composed by a set of alerts such as “there is a P% probability that company A will encounter catastrophe C within X timeslots”.

The most challenging and crucial aspect of the project is thus the process of event recognition and sequencing; however, as a side analysis, the time series generated by the Time-Sequence Generator can be compared with numerical data (indexes), arranged on their own timeline, in order to understand correlations between them.

3.1 Textual information

Events can be recognized inside textual information through text analysis; in Mercurio we propose the use of three different approaches:

- *Semantic approach*: events are recognized by means of semantic rules that formalize the knowledge and experience of our domain expert.
- *Automatic approach*: events are identified by applying clustering algorithms to financial news.
- *Hybrid approach*: a combination of the previous approaches where catastrophes are recognized with semantic rules and signals by means of clustering.

In the semantic approach, in particular, rules define a relationship between sentence structures and corresponding events. This is one of the innovative features of Mercurio and can be further improved by introducing different formalization strategies.

Some rules are independent of each other in the sense that they represent events that do not interact in any way. Other rules instead might represent events that are somehow related, e.g., one event might be a composition of two different events. Moreover, some rules are related to events that involve only one company while others might represent an interaction among different financial players. These considerations generate a rule categorization that also introduces the need for rule ordering. Such ordering is needed during the phase when rules are applied to the financial news in order to ensure the correct event recognition.

An interesting idea is to organize and formalize the semantic rules into an ontology. The concepts in the ontology would represent events, and relationships among concepts would describe how events are related to each other and how they interact and depend on each other. Each concept should be related to a set of words (or sentence structures): those that express the corresponding rule. These words could be defined ad-hoc according to the semantic rules in Mercurio, but can also originate from external ontologies describing the financial scenario or others. This addition helps to enrich the semantic formalization by taking into account both synonyms and new terms.

The use of an ontology would also allow us, through the use of inference, to discover novel information about the formalized data, possibly stimulating the discovery of new events.

3.2 Numerical information

Time-dependent series such as financial indexes are represented as values on a timeline. Each timeslot (e.g. hour or day) is associated – according to the index – with a value, e.g. an opening value, price, closing value, average and so on. The timeline containing these values can be used, in addition to the timeline containing events coming

from textual data, to enrich our data representation for the user. This is possible not only by taking into consideration single values but also by looking at some patterns inside the index.

A first technique is based on Bollinger Bands⁴ that, given a numerical series, provide an upper and lower band such that the observed values usually oscillate within them. Whenever a value goes beyond these bands, it means that an unusual oscillation is happening. Thus, a trend that goes below the lower band is an unexpected price fall while a trend that goes above the upper band is an unexpected price rise.

A second technique that has been applied in the financial context is the detection of specific patterns, in terms of curve shape, inside financial time series (rather than single interesting points). The financial domain comprises some well known and meaningful trend patterns [14] such as “double top”, “spike bottom”, “wedge” and so on.

Another interesting approach is to approximate financial time series through the use of segments, for example by using piecewise segmentation [8]. In such way each segment represents a trend in the series, thus, we might have segments representing increasing, stable or decreasing volumes or prices.

Yet another segmentation technique specifically adopted in the financial scenario is based on Turning Points (TP) [15]. TPs are local minimum and maximum points from the historical data and are widely used in technical analysis for predicting the movement of a stock. In fact, they represent the trend of the stock change and can be used to identify the beginning or end of a transaction period.

4 Current implementation

Currently, our system predicts catastrophes by taking into consideration the information coming from financial news, while the part allowing the comparison with financial indexes is not implemented yet. The system comprises three main phases:

1. *Data acquisition and management*: financial news are extracted from web sources, structured and stored into a relational database; their contents are then cleaned and pre-processed;
2. *Event recognition*: articles are analyzed to identify both catastrophes and signals. Mercurio adopts the three different approaches introduced in Section 3: (i) semantic approach, (ii) automatic approach and (iii) hybrid approach.
3. *Model construction*: the events found in the previous step are used in combination with sequential pattern mining to learn a model, represented by means of temporal patterns, to predict the arrival of catastrophes.

4.1 Data acquisition and management

Mercurio currently monitors 250 Italian mid-cap companies and the information about them is gathered from important Italian financial and economic web sources such as “Il Sole 24 Ore”, “Radiocor”, “La Repubblica” and “Il Corriere della Sera”. Articles about companies are extracted directly from the newspaper websites and stored into a MySQL database (our initial data contains about 14,000 articles, from year 2010 to 2015) keeping only those that: (i) are part of financial and economic sections and (ii) refer to one of the chosen companies. After this phase the article texts are cleaned by tokenization, stopword elimination and word stemming.

⁴ <http://www.investopedia.com/terms/b/bollingerbands.asp>

Two different text pre-processing strategies are adopted, one used during the semantic event recognition and the other for the automatic event recognition. In the first strategy we kept all special characters, symbols, punctuation marks, numbers, words, company names and persons details because they are needed by the expert’s rules. In the second strategy these data are not significant, sometimes even misleading when applying clustering algorithms, thus they are eliminated from the texts.

4.2 Event recognition

Events are detected through text analysis of the financial news. Mercurio implements three event recognition approaches; all of them output a temporal sequence containing the recognized events.

4.2.1 Semantic event recognition.

Mercurio uses a set of rules that formalize the recognition of relevant events inside financial news. Rules define a relationship between some keywords, regular expressions (in general, sentence structures), and corresponding events (e.g. “take” is a keyword related to an acquisition event). An article that contains the expressions defined in a rule is assigned a label corresponding to the event formalized by the rule. Each article is assigned zero, one or more labels depending on the rules it triggers.

Rules capture meanings that go beyond the sole natural language processing. For example, financial newspapers, usually, publish interviews when requested by a company. The question is: why would a company want to be interviewed? When this breaks a trend of non-communication it must be a signal. Also, an article that mentions the gross profit of a company is not a good sign because this indicator does not provide the amount of real revenue of the company, thus it could hide a negative trend of the company, whereas the net profit is not ambiguous, so this is a positive financial communication.

Currently, Mercurio encompasses 30 semantic rules, 7 of which identify catastrophes while the rest formalize signals.

4.2.2 Automatic event recognition.

This approach does not use any a-priori knowledge but relies on the detection of events by only applying clustering algorithms to the pre-processed financial news. Articles are represented in the Vector Space Model [1] where the weight of each term is the TF-IDF frequency of its occurrences in the article. Then, articles are clustered using the K-means algorithm and each article is assigned one label, corresponding to the cluster it belongs to.

The process of article clustering has proven to be quite challenging because at the end of the clustering phase we tried to interpret the results and found it impossible to distinguish between clusters representing signals and those representing catastrophes. This was a big drawback from our point of view since we were not able to understand how to predict catastrophes.

4.2.3 Hybrid event recognition.

To overcome the problem exposed above, we “added some semantics” to the automatic approach, obtaining what we called the hybrid one. In this approach, catastrophes are found by using the semantic rules that formalize catastrophic events, while the other signals are obtained by clustering all those articles that were not isolated by the rules defining catastrophic events.

4.3 Model construction

The output of the event recognition phase is a sequence of events, each associated with a timestamp that corresponds to the date and time of publishing of the article in which the event was found. Based on this sequence, Mercurio uses Sequential Pattern Mining to find “recurring” temporal patterns in the input data which are then used to predict future catastrophes.

This step is performed by using AIDA [2], a tool that encompasses both the model creation and prediction features. The tool is applied in two phases: (i) given as input a temporal sequence of events, a specific event e from the sequence and a minimum support threshold, it finds all temporal patterns that end with e and whose support is above the threshold; (ii) given the found model and a real-time flow of previously unseen articles, it predicts the happening of the learned events within a certain time span. In particular, during the prediction phase, each incoming new article is processed and labeled according to the events it triggers. Then, the system tries to match each event to the ones in the patterns of the model. If this happens, it waits for another event that would match the next event in the pattern. This process is repeated until a pattern expires because of time constraints or its last but one event is reached. When this happens, we can predict the happening of the next event, which is the one corresponding to the last node of the pattern, which, by construction, is always a catastrophe.

4.4 Experiments

Let us briefly discuss on the performance of our prototype and compare the semantic approach (SA) and hybrid approach (HA). First of all, let us recall the differences between the two approaches, in terms of article-event relationships: (i) in SA an article might contain both catastrophes and signals, while in HA this is not possible because clustering is computed only on those articles that do not trigger any catastrophe; (ii) in SA an article might not trigger any rules thus not generate any event; in HA all the articles are associated with exactly one event, either a catastrophe or a cluster label; (iii) in SA an article might trigger more than one signal, while in HA each article belongs to only one cluster, thus, it is related to only one signal. These differences make it difficult to qualitatively compare the results of the two approaches, articles that trigger the same events in SA often belong to different clusters in HA.

In the semantic approach we considered 2549 instances of events (556 of catastrophes, 1993 of signals) and, for each catastrophe, built a model to predict it. The constructed models contain an average of 9 patterns whose lengths vary between 2 and 7. In the hybrid approach we consider 3283 articles (438 catastrophes, 2845 are clustered). The constructed models contain an average of 13 patterns whose lengths vary between 2 and 6. The hybrid approach allows us to obtain a greater number of patterns w.r.t. the semantic approach and results in an increase of the average number of patterns for each catastrophe. All the constructed models were tested on previously unseen data to determine the precision and recall of the predictions. We recall that low precision means that there are many wrong predictions, i.e. many times the system predicts a catastrophe which does not actually happen, and a low recall means that there are many missed predictions, i.e. many times the system does not predict a catastrophe and the catastrophe actually happens.

The results obtained by applying the two methods vary depending on the catastrophe: (i) some catastrophes cannot be predicted because their model has only one pattern which does not appear in the testing

set; (ii) some catastrophes have maximum precision and maximum recall thus they are perfectly predicted, i.e., there are only right predictions and not wrong or missed ones; (iii) other catastrophes have always maximum precision because the system makes only right predictions about them, however (iv) some have low recall which means that many times the catastrophe happens and the system was not able to predict it.

These results strongly depend on the minimum support threshold: the higher the support threshold, the higher the precision and the lower the recall; conversely, the lower the support threshold, the lower the precision and the higher the recall. In general, we noticed that both approaches offer satisfactory performances, however we are working at making the models more accurate, so that the final prototype will be based on more training data and on an integration of the two techniques.

5 Conclusion

In this paper we discussed Mercurio, a system that supports the decision-making process of investors through the automatic extraction and analysis of financial data, with the aim of predicting extraordinary financial moves. Current results are encouraging but leave space for many improvements, especially related to enrichments of the current model, such as introducing weights and polarity to each event and the use of statistical information about the whole financial market, its different sectors and each monitored company.

REFERENCES

- [1] G. Salton, A. Wong, C. S. Yang. A vector space model for automatic indexing. *Commun. ACM* 18, 11 (November 1975), 613-620.
- [2] M. Mazuran, M. Simoni, L. Tanca. AIDA: Automatic Indexing based on Data mining. *SEBD* 2015. pp.176-183.
- [3] S. Bacher. Mining Unstructured Financial News to Forecast Intraday Stock Price Movements. PhD Thesis. University Mannheim. 2012.
- [4] G.P.C. Fung, J. Xu Yu, W. Lam. News Sensitive Stock Trend Prediction. *PAKDD* 2002 pp.481-493.
- [5] G. Gidofalvi. Using News Articles to Predict Stock Price Movements. Department of Computer Science and Engineering, University of California, San Diego. 2001.
- [6] M.A. Mittermayer. Forecasting Intraday Stock Price Trends with Text Mining Techniques. *HICSS* 2004.
- [7] D. Peramunetilleke, R.K. Wong. Currency exchange rate forecasting from news headlines. *ADC* 2002. pp.131-139.
- [8] V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, J. Allan. Language models for financial news recommendation. *CIKM* 2000. pp.389-396.
- [9] M.A. Mittermayer, G. F. Knolmayer. NewsCATS: A News Categorization and Trading System. *ICDM* 2006. pp.1002-1007.
- [10] B. Wuthrich, V. Cho, S. W. Leung, D. Permunetilleke, K. Sankaran, J. Zhang. Daily stock market forecast from textual web data. *ICSMC* 1998. pp.2720-2725.
- [11] T. Hong, I. Han, Knowledge-based data mining of news information on the Internet using cognitive maps and neural networks. *Expert Systems with Applications* 2002, 23(1):1-8.
- [12] K. Kohara, T. Ishikawa, Y. Fukuhara, Y. Nakamura. Stock Price Prediction Using Prior Knowledge and Neural Networks. *Intelligent Systems in Accounting, Finance and Management* 1997, 6(1):11-22.
- [13] F. Wanner, T. Shreck, W. Jentner, L. Sharaliev, D. A. Keim, Relating Interesting Quantitative Time Series Pattern with Text Events and Text Features. *SPIE* 2013.
- [14] T. Fu, F. Chung, V. Ng, R. Luk. Evolutionary Segmentation of financial time series into subsequences. *Evolutionary Computation* 2001.
- [15] J. Yin, Y. Si, Z. Gong. Financial Time Series Segmentation Based On Turning Points. *ICSSE* 2011.

Online User Behavioural Modeling with Applications to Price Steering

Van Tien Hoang¹ and Vittoria Cozza² and Marinella Petrocchi³ and Rocco De Nicola¹

Abstract. Price steering is the practice of “changing the order of search results to highlight specific products” and products prices. In this paper, we show an initial investigation to quantify the price steering level in search results shown to different kind of users on Google Shopping. We mimic the category of *affluent* users. Affluent users visit websites offering expensive services, search for luxury goods and always click on the most costly items results at Google Shopping. The goal is checking if users trained in specific ways get different search results, based on the price of the products in the results. Evaluation is based on well known metrics to measure page results differences and similarities. Experiments are automatised, rendering large-scale investigations feasible. Results of our experiments, based on a preliminary experimental setting, show that users trained on some particular topics are not always influenced by previous search and click activities. However, different trained users actually achieve different search results, thus paving the way for further investigation.

1 Introduction

Popular e-commerce websites, such as the Amazon Marketplace, offer a window to thousands of merchants, able to advertise their goods and services to millions of potential buyers.

Recently, the traditional advertising approach has moved towards a targeted one: the ad is shown only to online users with a specific profile - location, gender, age, e-shopping history are among the monitored aspects. This way, the merchant pays only for ads shown to users matching the ideal buyer for the merchant products. Targeted advertising is possible since the ads system is able to build a user profile tracking her online behaviour, e.g., on the e-commerce website, plus considering the data inserted by the user on the platform, at registration time.

Although personalised ads have the significant advantage to guide the customer mostly towards products she likes, concerns were born since the ads system could 1) hide to the user other potential interesting products [20]; and 2) expose user private information [3].

Price steering refers to the practice of changing the order of search results to highlight specific products prices [18]. In this work, we aim at studying if e-commerce websites rely on user past online behaviour to show her different product prices. In particular, we focus on Google Shopping, to discover if Google shows products of different price based on the *user willingness to pay*.

Google Shopping⁴ is a promising platform to study the effect of

price personalization. It allows vendors to reach a large number of customers, really interested in specific products, thus showing the right product to the right customer. Google Shopping creates a selling campaign, placing specific products “in front of millions of online shoppers searching on Google.com”⁵. This is possible since Google can access several information on the user search activity, not only including that on Google Shopping. Actually, Google monitors the circle of websites known as Google Display Network (GDN), a large set of websites publishing Google ads⁶.

As shown in [13], Google builds *ad user profiles*, monitoring and learning behaviors when the users navigate on the GDN websites. Among the elements considered to build the ad profiles, there are the list of the visited websites, their topics, the time spent on each website, the number of times the user went to the website, the device the user is using for accessing at the platform, geo-localization of the user IP address.

Past work showed that price steering is affected by the user location, see [10, 19] and by the user device, as for the case of the online travel agency Orbitz [17]. Orbitz realised that Mac users were more interested in costly hotels and traveling services than Windows users. Consequently, the agency showed the most costly results as the first results for Mac users.

Our work focuses on how the user behaviour, e.g., visiting a website of luxury goods, clicking on expensive products, affects the price of the items shown as the result of future queries over Google Shopping. We emulate the on-line behaviour of an *affluent* user and we compare her results list with one of a fresh user, which has not searched before on Google Shopping. Preliminary results show that, overall, affluent profiles have been shown different results with respect to those shown to the fresh control user. However, there is no a fixed rule, leading first to the most expensive products shown to the affluent users. The difference in the results list is however worth to be acknowledged, and calls for further investigation, with a more complex experimental settings and a more extensive evaluation.

The rest of the paper is as follows. Next section briefly presents related work in the area. Section 3 describes our methodology. In Section 4, we describe the experiments and we give the results. Finally, Section 5 concludes the paper.

2 Related Work

This section briefly relates on literature in the area of personalization of web results, particularly focusing on price steering and price

¹ IMT School for Advanced Studies Lucca, Italy, email: {vantien.hoang, rocco.denicola}@imtlucca.it

² Electrical & Information Engineering Department (DEI), Polytechnic University of Bari, Italy, email: vittoria.cozza@poliba.it

³ IIT-CNR, Pisa, Italy, email: m.petrocchi@iit.cnr.it

⁴ <http://www.google.com/shopping>

⁵ https://services.google.com/fh/files/misc/product_listing_ads_intro.pdf

⁶ <https://support.google.com/adwords/answer/2404190?hl=en>

discrimination. While price steering denotes the practice of showing different products with different prices to different users, discrimination is a similar practice, but related to the same product. Work in [18] gives an alike definition of price steering: a scenario in which e-commerce websites show to the unaware user different search results, based on the user willingness to pay (defined as the maximum amount of money a customer is willing to spend for a product).

Mikians et. al [19] detected online price discrimination by collecting data from 340 real Internet users over 18 countries. The analysis focused on how the price of the same product, offered from a set of retailers, varies retailer per retailer. Outcome denotes geographic location as the main factor affecting the prices. Work in [10] analysed price discrimination by adopting fictitious users, mimicking a visit to shopping websites, from 6 different locations, for 7 days. Even in this case, results show that user location has an impact on price discrimination.

In [12], the authors extensively measures both price steering and discrimination. With both real data collected through Amazon Mechanical Turks and synthetic data from controlled experiments using a headless web browser⁷, the authors analyse the prices offered by a plethora of online vendors. The work finds evidence of price differences by different merchants and retailers: their websites record the history of clicked products to discriminate prices among customers.

The issue of price steering analysed in this paper has close relationships with the ties between online user behaviours and the search results (and/or advertisements) presented by a search engine (or a website) to the same user. Indeed, price steering and discrimination constitute only one aspect of a wide phenomenon, originally put in the spotlight by Pariser with his Filter Bubbles [20] and investigated by seminal work on web search personalization, like, e.g., the one in [11]. Particularly, online user behaviour has been investigated widely in relation to targeted advertising. As an example, work in [16] reveals how Google ads are selected for specific users, according to their activities over the Internet. Experimental results in the paper claim that 65% of ads categories shown to users have been targeted according to their behaviours. Targeted ads have been studied also in [2], achieving results consistent with [16]. Recently, this targeting phenomenon has been investigated in [15] on ads in Gmail, resulting in an evidence of linking users behaviours and shown ads also on the email service. Overall, tracing online behaviour is commonly adopted - and such information is commonly exchanged among websites - to determine which ads are shown to users.

In this work, we take inspiration from the analysis of online user behaviour to evaluate the effect on that behaviour on product prices shown to the user.

3 Methodology

Our goal is measuring how the user online behaviour affects price steering on Google Shopping. We use the approach which is similar in [9, 12, 21]. At first, we consider users of two categories: *affluent* and synthesized *control* users. Intuitively, the former feature higher willingness to pay than the latter. Thus, aiming at mimicking their behaviour, we assume that affluent users search and click on more expensive products than control users. We have considered two kinds of behaviours: 1) visiting web pages, and 2) searching for keywords on Google Shopping. Visiting a page means staying on that page for a while and also scrolling the page. We define the affluent user behaviour as follows:

- an affluent user visits websites selling luxury goods;
- an affluent user searches for keywords representing luxury goods on Google Shopping and
 - she visits the three results representing the first three products whose price is above the average (among all the obtained results)
 - she visits those result pages that are the same of a previously visited website selling luxury goods.

The behaviour of a control user is different, she is idling while the affluent is in action.

Our expectation is that, when users query Google Shopping after a training phase where they behave as described above, affluent (resp., control) ones will likely see the highest (lowest) price products ranked first in their list of results.

In order to identify websites and keywords for luxury goods, we have exploited a tool originally intended for setting up targeting advertisements, the Display Planner tool of Google AdWords. The tool guides the user to find websites and keywords inherent to specific topics and terms⁸.

It is well known that Google monitors the users' behavioural activities over the Internet through tools such as Google Analytics, Google Plus, and the Google ads system⁹. Thus, we train two affluent user profiles according to the specific online behaviours described above. Each profile has a control user profile associated. A control user has the same configuration as the user she is associated to (same browser, same OS). The difference is that control users are not logged into a Google account. Then, we compare the results obtained searching on Google Shopping the same keywords for both the trained users and the control users. In detail, the training and test phase are as follows:

- Training step 1: The two affluent users visit a list of websites with topics related to their category. Websites have been chosen using the Google Display Planner.
- Training step 2: The two affluent users search on Google Shopping for keywords related to their user category (keywords have been chosen according to the Display Planner, too). Then, they click on the most expensive product results and on those results coming from websites visited at step 1 (when present).
- Test: All users (trained plus control) search for new products on Google Shopping. They do not interact with the results.

We let the two affluent users repeat the training phase eight times. At the end of each training, we run the test. This is for building a longer behavioral history of the user. Indeed, Google itself states that it uses browsing and search histories to personalise the results and enhance the user experience¹⁰. This is why we have repeated the training phase eight times, always on the same profile, to emphasize possible personalization aspects. One evidence of the efficacy of this modality is in [1]: Google infers the interests of users only after a certain amount of websites visiting. Another evidence is in [6], where the authors described that Google took five days of training on a single user to produce personalized news content.

⁸ <https://adwords.google.com/da/DisplayPlanner/Home>

⁹ <https://www.google.com/policies/privacy/>

¹⁰ <https://www.google.com/policies/terms/>

⁷ <http://phantomjs.org>

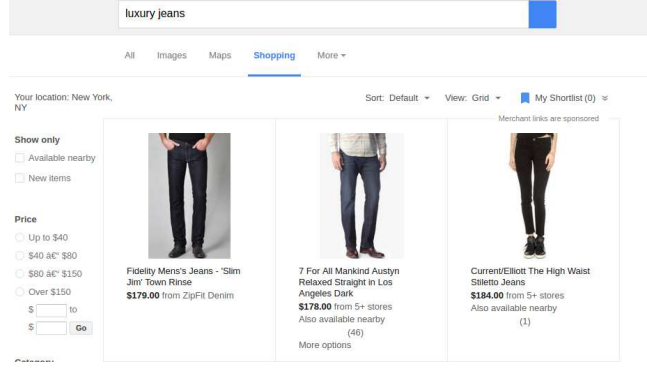


Figure 1. Top 3 results on Google Shopping: control, 3rd test block

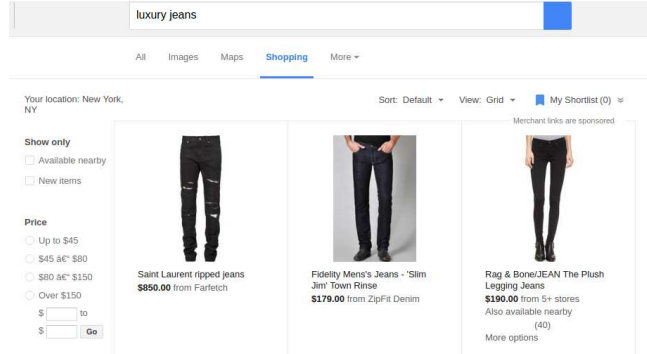


Figure 2. Top 3 results on Google Shopping: affluent, 3rd test block

4 Experiments

For our experiments, we use AdFisher [7], a freely available automated tool¹¹. Natively, the tool functionalities allow to analyse interactions between online user behaviors, advertisements shown to the user, and advertisements settings. Later, AdFisher has been extended for handling Google searches and news searches, to measure personalization in query results, see, e.g., [6] and applied to novel news search experiments [5]. AdFisher has also been used for statistical evaluations, e.g., to measure how users are exposed to Wikipedia results, in return to their web searches, see, e.g., [4]. The interested reader can refer to [8] for a full survey on tools for measuring and analysing users' interactions with online services (including AdFisher).

In our work, AdFisher runs browser-based experiments that emulate search queries and basic interactions with the search results, e.g., interacting with those search results whose price is above or below the price average on the total of the results, or those results belonging to a list of previously visited websites. Github hosts our extended version¹².

AdFisher interacts with Selenium, a web browser automation tool. Selenium allows to run a unique instance of Firefox creating a fresh profile, with new associated cookies, the so called *Firefox profile*. The Firefox profile that is used is stripped down from what is installed on the machine, to only include the Selenium WebDriver.xpi plugin. Further, we take advantage of a plugin to automatically obtain the Python code for recording actions on web pages (e.g., clicking,

typing, etc.), provided by Selenium IDE for Firefox¹³. All the experiments are done on the Firefox web browser version 43.0.4, controlled by Selenium in Python, under XUbuntu 14.04.

To simulate different real users, we browse from different IP addresses, implementing a solution based on SSH tunneling to remote computers. To simulate many computers from one geographic area, we use the VPS services of Digital Ocean¹⁴. It is well known that, when a user enters a query to Google, the query is unpredictably sent to many distributed servers, to retrieve the results. This could produce noise due to inconsistent data among different servers. To avoid the issue, we query only towards specific Google servers IP addresses, as in [11].

Finally, we use the following metrics to evaluate the search results of the test:

- Jaccard Index: given two sets P and Q, Jaccard Index is 1 when the sets are identical and 0 when their intersection is empty.
- NDCG (Normalized Discounted Cumulative Gain), measuring the similarity between a given list of results and the ideal list of results. Originally introduced as the non-normalised version DCG in [14], NDCG has been adopted in [12] for measuring price steering. For each result r, there is one gain score g(r), representing its price. For a result page $R = [r_1, r_2, \dots, r_k]$, we have $DCG(R) = g(r_1) + \sum_{i=2}^k (g(r_i) / \log_2(i))$. NDCG is $DCG(R) / DCG(R')$, where R' is the ideal result page (a list in which the results are shown from the most expensive to the least one). We create R' by unionising the results returned, for the same query, to affluent and control profiles. Then, we sort such results from the most expensive to the least expensive one.

¹¹ <https://github.com/tadatitam/info-flow-experiments>

¹² https://github.com/tienhv/Adfisher_for_GoogleShopping

¹³ <http://www.seleniumhq.org/projects/ide/>

¹⁴ <https://www.digitalocean.com/>

4.1 Settings and results

We have extended the AdFisher functionalities to handle Google Shopping pages and to mimic the behaviours described in Section 3. We automatically implement the whole experiments for the affluent and control users. The extended AdFisher also stores the query results for further analysis, as the calculation of the NCDG metric.

For the training phase, we emulate two user profiles logged into Google. We consider 80 websites for each type of user profile and we let the profiles visit all of them. Such choice has been driven by [1], which proved that visiting 50 websites is enough for Google Ads to infer the user interests. The website visit time for a user is a random value, however less than 30 seconds (such threshold being estimated following the `alexa.com` statistics). Furthermore, each profile has been trained with 15 training keywords, and 3 were the resulting links to be clicked, associated to the top 3 most expensive products. Table 1 shows an excerpt of the visited websites and the searched keywords, selected with the help of the Google Display Planner.

For the test phase, we still consider the two trained profiles, plus two control ones. All feature the same behaviour, which consists of querying Google Shopping with the same test keywords. Then, we collect the results. Figure 1 and Figure 2 show the first results showed to affluent and control users, for a specific test query. We extract links and prices of all the results to calculate the metrics listed in Section 3.

The training and the test phases are repeated eight times per user. Each session lasts around 90 minutes.

Table 2 shows NCDG values for each test session (results are for query “luxury shoes”), for each user.

Figures 3 and 4 plot, resp., the Jaccard index and the Kendall index, for the eight test sessions about “luxury shoes”. The blue lines represent are calculated over the results pages of *Affluent 1* and *Control 2* users, while red lines are calculated over the result pages of *Affluent 2* and *Control 1* users. This is to consider two users trained in same ways and connected from two different machines. Jaccard index shows evidence of results customization, while Kendall index says that, most of the times, the results for affluent and control profiles have a level of agreement (featuring a positive values for that index).

Figure 5 plots the values obtained calculating the average NCDG of the two affluent users and the two control users, over the eight test sessions. The test query is “luxury shoes”. Average NCDG indicates that even if affluent and control users follow the same pattern of pricing ordering, the former is closer to the ideal list results (where the most expensive products are in the first positions).

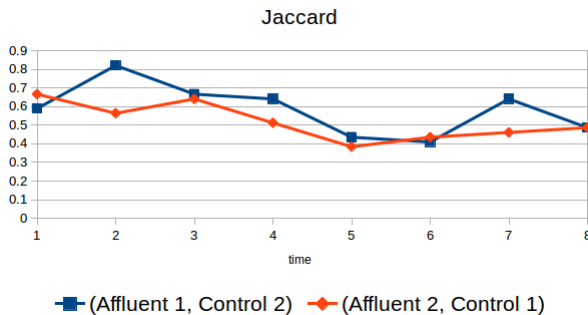


Figure 3. Jaccard Index for “luxury shoes”

Figure 6 shows the average NCDG value for all the test queries (averaged both per profile and over the eight sessions). The Figure

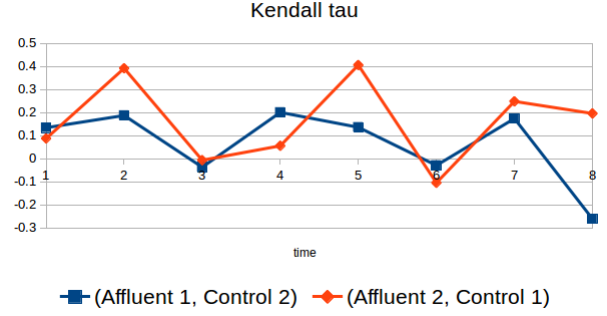


Figure 4. Kendall Index for “luxury shoes”

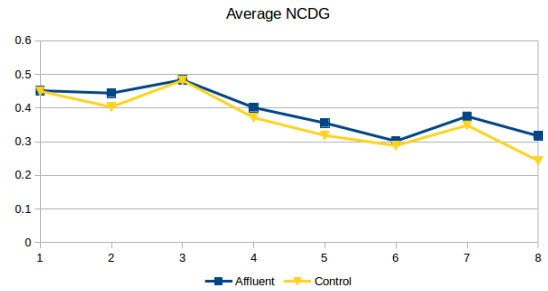


Figure 5. Average NCDG for “luxury shoes”

shows, at a glance, how, in some cases, NCDG values are higher for affluent users than for control. The control users almost always obtain NCDG values lower than the affluent (or comparable in those cases with very similar values for the two kind of profiles). However, we argue that results are also affected by the specific search query over Google Shopping. As an example, Figure 7 shows the NCDG values for the test query “luxury jeans” for affluent 1 and the average of results of the two controls, over the eight sessions. The picture clearly indicates NCDG values that are greater for the affluent user. Instead, “mens dress casual shoes” (not shown in a Figure over the eight sessions) provides a higher value for the control which is an opposite result with respect to our expectations. While these initial results are promising, there is the need of further evaluation, where more, and more generic keywords, should be tested, at a larger scale.

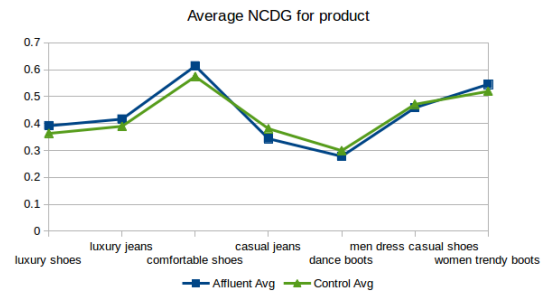


Figure 6. NCDG values averaged over profiles and sessions, per product

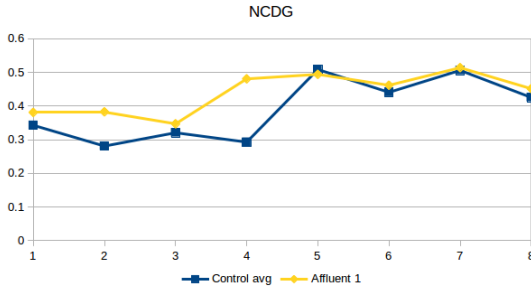
It is worth noting that, as introduced and motivated at the end of Section 3, the two affluent users (as well as the two control ones) are identical in terms of behaviour. Further, browser and OS settings are the same, while the IP address from which they browse is different

Table 1. Training phase: An excerpt of websites and keywords

training websites	training keywords	test keywords
outfitideashq.com seriousrunning.com lululemonmen. com storelocate.us, skateboardingmagazine.com haircutinspiration.com	mens fashion shoes, trendy boots, formal shoes for men, jogging shoes, cheap designer shoes, ath- letic shoes	mens dress casual shoes, luxury shoes, dance boots, comfortable shoes, women trendy boots, luxury jeans, casual jeans

Table 2. NCDG of “luxury shoes” for 8 test sessions

luxury shoes	slot 1	slot 2	slot 3	slot 4	slot 5	slot 6	slot 7	slot 8
Affluent 1	0.45	0.45	0.49	0.40	0.38	0.32	0.37	0.33
Control 1	0.45	0.42	0.48	0.38	0.32	0.30	0.35	0.24
Affluent 2	0.45	0.44	0.47	0.40	0.33	0.28	0.38	0.30
Control 2	0.45	0.39	0.48	0.36	0.32	0.28	0.34	0.25

**Figure 7.** NCDG per session, for “luxury jeans”

(different devices, from the same geographical area). We have not compared directly the two affluents, since their results pages could be different for uncontrollable effects, such as timeouts or network delays (we are indeed emulating different IP addresses). The existence of sources of noise is also the reason why we have chosen to show, for some experiments, the average of the results of the two users.

5 Conclusions

We have designed and implemented a methodology to train and test user behaviours on Google Shopping, for evaluating a potential price steering, based on the *willingness to pay* attitude of the users. We have analysed the results list of affluent and control users. Affluent users were trained over eight training sessions. The results lists were obtained over eight test sessions, one at the end of each training session. The outcome of the experimentation is that, for most of the test queries, the result list of the affluent user is biased towards more expensive products than the one of the control user. However, the experiments results pave the way for further investigation. Indeed, we can imagine to 1) mimic queries from different geographical areas (not considered here, but recognised by past work as an impact factor for price manipulation); 2) use location via IP address as the major measurement, instead of artificial user profiles, because in some countries (like USA), the location/postcode is a strong indicator for economic situation, religion and race; 3) augment the number of training and test queries; 4) expand the duration of each training and test experiment; 5) mimic queries by different kind of users, e.g., mimic *budget* users, which always search for cheap products and services. Our experimental approach is general enough to be applicable to other e-commerce websites, like, e.g., *Amazon.com* and *eBay.com*.

6 Acknowledgment

This research has been partially funded by the Registro.it project *MIB* (My Information Bubble).

REFERENCES

- [1] Barford, P., Canadi, I., Krushevskaia, D., Ma, Q., Muthukrishnan, S.: Adscape: Harvesting and analyzing online display ads. In: Proceedings of the 23rd International Conference on World Wide Web. pp. 597–608. WWW ’14, ACM (2014)
- [2] Carrascosa, J.M., et al.: I Always Feel Like Somebody’s Watching Me Measuring Online Behavioural Advertising. In: 11th Emerging Networking EXperiments and Technologies. ACM (2015)
- [3] Conti, M., Cozza, V., Petrocchi, M., Spognardi, A.: TRAP: using targeted ads to unveil google personal profiles. In: IEEE International Workshop on Information Forensics and Security. pp. 1–6 (2015)
- [4] Cozza, V., Hoang, V.T., Petrocchi, M.: Google web searches and Wikipedia results: a measurement study. In: Proc. of 7th Italian Workshop on Information Retrieval (IIR) 2016. CEUR Workshop Proceedings (2016)
- [5] Cozza, V., Hoang, V.T., Petrocchi, M., Spognardi, A.: Experimental measures of news personalization in Google News. In: SoWeMine 2016: 2nd International Workshop on Mining the Social Web. Springer (2016)
- [6] Datta, A., Datta, A., Jana, S., Tschantz, M.C.: Poster: Information flow experiments to study news personalization. In: Computer Security Foundations Symposium (CSF), 2015 IEEE 28th. IEEE (2015)
- [7] Datta, A., Tschantz, M.C., Datta, A.: Automated experiments on ad privacy settings. Proceedings on Privacy Enhancing Technologies 2015(1), 92–112 (2015)
- [8] Englehardt, S., Narayanan, A.: Online tracking: A 1-million-site measurement and analysis (May 2016), [Technical Report]
- [9] Englehardt, S., et al.: Web privacy measurement: Scientific principles, engineering platform, and new results. Manuscript posted at <http://randomwalker.info/publications/WebPrivacyMeasurement.pdf> (2014)
- [10] Guha, S., Cheng, B., Francis, P.: Challenges in measuring online advertising systems. Internet Measurement Conference pp. 81–87 (2010)
- [11] Hannak, A., al.: Measuring personalization of web search. In: 22nd World Wide Web. pp. 527–538 (2013)
- [12] Hannak, A., al.: Measuring price discrimination and steering on e-commerce web sites. In: IMC. pp. 305–318. ACM (2014)
- [13] Haveliwala, T., Jeh, G., Kamvar, S.: Targeted advertisements based on user profiles and page profile (Nov 27 2012), US Patent 8,321,278
- [14] Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Trans. Inf. Syst. 20(4), 422–446 (Oct 2002)
- [15] Lecuyer, M., Spahn, R., Spiliopoulos, Y., Chaintreau, A., Geambasu, R., Hsu, D.: Sunlight: Fine-grained targeting detection at scale with statistical confidence. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. pp. 554–566. CCS ’15, ACM (2015)

- [16] Liu, B., Sheth, A., Weinsberg, U., Chandrashekar, J., Govindan, R.: AdReveal. In: HotNets-XII. pp. 1–7. ACM (2013)
- [17] Mattioli, D.: On Orbitz, Mac users steered to pricier hotels. Wall Street Journal (2012), <http://on.wsj.com/LwTnPH>
- [18] Mikians, J., Gyarmati, L., Erramilli, V., Laoutaris, N.: Detecting price and search discrimination on the Internet. In: Proceedings of the 11th ACM Workshop on Hot Topics in Networks. pp. 79–84. HotNets-XI, ACM (2012)
- [19] Mikians, J., et al.: Crowd-assisted search for price discrimination in e-commerce: First results. CoNEXT pp. 1–6 (2013)
- [20] Pariser, E.: The Filter Bubble: What the Internet is hiding from you. Penguin UK (2011)
- [21] Tschantz, M.C., Datta, A., Datta, A., Wing, J.M.: A methodology for information flow experiments. CoRR abs/1405.2376 (2014)

Application of Constraint-based Technologies in Financial Services Recommendation

Alexander Felfernig¹

Abstract. Constraint-based recommender systems rely on an explicitly defined set of constraints that are used to take into account product domain properties, customer requirements, and legal requirements. This paper focuses on different aspects of the application of constraint-based technologies in financial service related scenarios. We show how to support the process of defining and maintaining recommendation knowledge and how to efficiently support users when interacting with constraint-based recommender systems. Finally, we discuss psychological issues that have to be taken into account when implementing such types of recommender systems.

1 Introduction

Recommender systems can be regarded as one of the most successful applications of Artificial Intelligence technologies [12]. The basic approaches of *collaborative* and *content-based filtering* (and variants thereof) are primarily used for recommending simple products such as books, movies, and songs. Complex products such as financial services, cars, and apartments in many cases require a different recommendation approach. For example, cars are not purchased very frequently, therefore collaborative filtering and content-based recommendation are not the first choice. Furthermore, such products are related to constraints, for example, not every financial service can be offered to every customer and a car recommendation should take into account the preferences defined by the customer.

Recommendation functionalities for complex products and services are provided on the basis of *knowledge-based recommendation technologies* [1]. Knowledge-based recommenders are either *constraint-based* [4] or *case-based* [2]. Case-based approaches are often implemented as critiquing-based recommender systems [2, 14] where an item (product) is identified on the basis of similarity metrics [16]. Identified items are shown to the user and the user can provide feedback in terms of critiques. For example, if a user perceives the return on investment of a financial service as too low, he or she can articulate a corresponding critique *higher return on investment*.

In the context of this paper we focus on *constraint-based recommendation* where the recommendation knowledge is represented in terms of a set of constraints that primarily relate user requirements with corresponding item properties. Constraint-based recommenders can determine recommendations on the basis of *constraint solving* [20] or on the basis of *conjunctive queries* [7]. The result of solution search (of a query) is a set of items that fulfill a given set of requirements. These *candidate items* can be ranked on the basis of utility-based methods such as the multi-attribute utility theory (MAUT) [21].

The remainder of this paper is organized as follows. In Section 2 we sketch approaches to make constraint acquisition efficient. In Section 3 we provide an impression of how conversational scenarios are supported by constraint-based recommendation. In Section 4 we show how knowledge bases can be applied for generating learning content. Aspects of human decision making in constraint-based recommendation are discussed in Section 5. The paper is concluded with a short discussion of issues for future work (Section 6).

2 Recommender Development

Constraint-based recommenders are based on a recommendation knowledge base that includes a definition of questions to be posed to the user (e.g., *what is the expected return rate?*), items to be recommended (e.g., *bankbooks* and *funds*), and a set of constraints that relate answers to questions with the corresponding items (e.g., *a low willingness to take risks excludes the recommendation of equity funds*). Such constraints are also denoted as *filter constraints*. Furthermore *incompatibility constraints* define in which way different user requirements can be combined with each other [7].

Especially in financial services recommendation scenarios, the correctness of the underlying knowledge base is crucial. Items recommended to the user (customer) have to be consistent with the user requirements. Furthermore, the knowledge base has to reflect product- and sales-related rules defined by the company and also corresponding legal requirements. In order to assure the correctness of a knowledge base, different test methods are applied where examples (test cases) are exploited in a regression testing process [7].

If regression testing fails (some test cases were not accepted by the knowledge base), those constraints in the knowledge base have to be identified that are responsible for the inconsistency. Since recommender knowledge bases can become quite large (in an order of magnitude of a few hundred constraints), knowledge engineers are in the need of support to identify faulty constraints as soon as possible. The efficiency of this process is crucial since, for example, with the introduction of a new product, the corresponding recommendation knowledge base has to be available (e.g., for supporting sales representatives in their sales dialogues [8]).

An approach to support the automated identification of faulty constraints is to apply the concepts of model-based diagnosis [17] where faulty constraints are identified on the basis of conflict set detection (see, e.g., [13]) combined with the determination of corresponding hitting sets (diagnoses) [6, 17]. Since many different diagnosis candidates potentially exist, diagnosis discrimination can be supported in an interactive fashion [18] or automatically [9].

¹ TU Graz, Austria, email:alexander.felfernig@ist.tugraz.at.

3 Conversational Recommendation

When interacting with a constraint-based recommender, users typically specify their requirements (preferences) by answering corresponding questions [7]. If a set of candidate solutions can be identified for the given set of requirements, these are ranked, for example, on the basis of MAUT [21]. If no solution can be identified for the given set of requirements, a diagnosis component can indicate those requirements that have to be adapted such that at least one solution can be identified.

A diagnosis in this context does not indicate faulty constraints in the knowledge base but user requirements that induce an inconsistency with the knowledge base (e.g., *if you change your preference "return rate = high" and keep "willingness to take risks = low", a corresponding solution can be identified*). Diagnosis determination can be implemented on the basis of the traditional approach documented in [6, 17] or on the basis of direct diagnosis algorithms such as FASTDIAG [10] that are able to determine personalized diagnoses without the need of predetermining conflicts. An alternative to the presentation of diagnoses is the direct presentation of conflicts that have to be resolved by the user in an interactive fashion [7].

4 Operationalizing Recommendation Knowledge

When a financial service sales representative interacts with a customer, he or she should not solely rely on the recommendations determined by the recommender system but should also be able to explain a recommendation in his/her own words. Knowledge bases can be exploited for the automated generation of question/answer combinations which can be imported into a corresponding e-learning environment. This way, time-intensive learning content development tasks can be at least partially replaced by automated mechanisms using, for example, constraint technologies [20]. Technologies that support such a kind of e-learning content generation have been implemented in the STUDYBATTLES environment.² This system is based on the idea of a quiz-based acquisition of (sales) knowledge.

5 Issues of Human Decision Making

Recommender systems can be regarded as decision support components that support a user when trying to identify a product that fits his/her wishes and needs. An important aspect to be taken into account in this context is that user preferences are not known beforehand and are not stable but rather frequently change within the scope of a recommendation process [3]. The ordering of items in a result set (recommendation) can have an impact on the item selection probability. *Decoy effects* influence the selection behavior of users by the inclusion of inferior items that in many cases are not even selected [19]. Such effects could be shown on the basis of a real-world financial service dataset [11]. Furthermore, *primacy/recency effects* are a cognitive phenomenon where list items are memorized significantly more often if these were placed at the beginning and the end of a list. In the recommendation context it has been shown that the probability of recalling item properties increases if the properties are presented at the beginning or the end of a property list [5]. On overview of different types of decision biases in the context of recommender systems can be found in [15].

² www.studybattles.com.

6 Conclusions and Future Work

In this paper we provide a short overview of different aspects of constraint-based recommendation technologies in the context of financial service recommendation. Future work will include the provision of end user knowledge acquisition environments, intelligent methods of test case generation and selection, and further user studies on the role of human decision making in recommender systems.

REFERENCES

- [1] R. Burke, 'Knowledge-based recommender systems', *Encyclopedia of Library and Information Systems*, **69**(32), 180–200, (2000).
- [2] R. Burke, K. Hammond, and B. Young, 'The FindMe approach to assisted browsing', *IEEE Expert-Intelligent Systems and their Applications*, **12**(4), 32–40, (1997).
- [3] L. Chen, M. deGemmis, A. Felfernig, P. Lops, F. Ricci, and G. Semeraro, 'Human Decision Making and Recommender Systems', *ACM Transactions on Interactive Intelligent Systems*, **3**(3), 17, (2013).
- [4] A. Felfernig and R. Burke, 'Constraint-based recommender systems: Technologies and research issues', in *IEEE ICEC'08*, pp. 17–26, Innsbruck, Austria, (2008).
- [5] A. Felfernig, G. Friedrich, B. Gula, M. Hitz, T. Kruggel, R. Melcher, D. Riepan, S. Strauss, E. Teppan, and O. Vitouch, 'Persuasive Recommendation: Exploring Serial Position Effects in Knowledge-based Recommender Systems', in *Persuasive 2007*, LNCS, pp. 283–294, Stanford, CA, (2007). Springer.
- [6] A. Felfernig, G. Friedrich, D. Jannach, and M. Stumptner, 'Consistency-based diagnosis of configuration knowledge bases', *Artificial Intelligence*, **152**(2), 213–234, (2004).
- [7] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker, 'An Integrated Environment for the Development of Knowledge-based Recommender Applications', *International Journal of Electronic Commerce (IJEC)*, **11**(2), 11–34, (2006).
- [8] A. Felfernig, K. Isak, K. Szabo, and P. Zachar, 'The VITA Financial Services Sales Support Environment', pp. 1692–1699, Vancouver, Canada, (2007).
- [9] A. Felfernig, S. Reiterer, M. Stettinger, and J. Tiihonen, 'Intelligent Techniques for Configuration Knowledge Evolution', in *VAMOS Workshop 2015*, pp. 51–60, Hildesheim, Germany, (2015).
- [10] A. Felfernig, M. Schubert, and C. Zehentner, 'An Efficient Diagnosis Algorithm for Inconsistent Constraint Sets', *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing (AIEDAM)*, **25**(2), 175–184, (2011).
- [11] A. Felfernig, E. Teppan, and K. Isak, 'Decoy Effects in Financial Service E-Sales Systems', in *Decisions@RecSys'11*, pp. 1–8, Chicago, IL, (2011).
- [12] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems – An Introduction*, Cambridge University Press, 2010.
- [13] U. Junker, 'Quickxplain: Preferred explanations and relaxations for over-constrained problems', in *19th National Conference on AI (AAAI04)*, pp. 167–172, San Jose, CA, (2004).
- [14] M. Mandl and A. Felfernig, 'Improving the performance of unit critiquing', in *20th International Conference on User Modeling, Adaptation, and Personalization (UMAP 2012)*, pp. 176–187, Montreal, Canada, (2012).
- [15] M. Mandl, A. Felfernig, E. Teppan, and M. Schubert, 'Consumer Decision Making in Knowledge-based Recommendation', *Journal of Intelligent Information Systems (JIIS)*, **37**(1), 1–22, (2010).
- [16] D. McSherry, 'Similarity and Compromise', in *5th International Conference on Case-based Reasoning*, pp. 291–305. Springer, (2003).
- [17] R. Reiter, 'A theory of diagnosis from first principles', *AI Journal*, **23**(1), 57–95, (1987).
- [18] K. Shchekotykhin, G. Friedrich, P. Rodler, and P. Fleiss, 'Sequential Diagnosis of high cardinality faults in knowledge bases by direct diagnosis generation', in *ECAI 2014*, pp. 813–819. IOS Press, (2014).
- [19] E. Teppan and A. Felfernig, 'Minimization of Decoy Effects in Recommender Result Sets', *Web Intelligence and Agent Systems*, **10**(4), 385–395, (2012).
- [20] E. Tsang, *Foundations of Constraint Satisfaction*, Academic Press, London, 1993.
- [21] D. Winterfeldt and W. Edwards, *Decision Analysis and Behavioral Research*, Cambridge University Press, 1986.

The Art of Spending and Recommendations in Personal Finance

Leonid Ivonin¹, Mark Perry², and Sriram Subramanian³

Abstract Happiness is one of the most important aspects of human lives, yet the literature on emotional well-being indicates that people often fail to correctly anticipate the hedonic consequences of future events. As a result, individuals end up being not as happy as they thought they would be. This phenomenon also applies to the domain of personal finance where people make bad decisions about purchases. In this paper, we identified a new opportunity for the research on recommender systems in personal finance and through analysis demonstrated that intelligent recommenders can help to minimize errors in affective forecasts and enhance happiness of people in the domain of consumption. Furthermore, we reviewed problems associated with design of such recommenders and proposed approaches to overcome them.

1 INTRODUCTION

One of the most fundamental instincts that people have is to be happy and to live a good life. There are many criteria for defining a good life but an important point is that an evaluation of one's life is a subjective process. Positive psychology defines a happy life more formally using the notion of "subjective well-being" (SWB). SWB refers to how people evaluate their lives in terms of both affective and cognitive aspects. There are several components of SWB such as a general life satisfaction, satisfaction with important domains (e.g., relationships with loving others), and positive affect (experiencing pleasant moods and emotions) [1]. Improvements in any component of SWB can help to increase a person's happiness. It seems that nowadays ordinary people tend to grant increasing importance to SWB. This is especially true in developed countries where basic material needs of people are satisfied and they are progressing towards the post-materialistic phase of self-fulfillment [2].

Often people are looking for earning more money in the quest for having a happy life. There is a common belief that more income has a positive impact on well-being and can make people feel happier. Therefore, a desire for higher income is a common motive among many people at all income levels [3]. On the other hand, research on income and SWB showed that among the non-poor the relationship between money and happiness is surprisingly weak. Although money seems to be able to buy happiness, it buys much less than what most people think. Data showing a weak correlation between SWB and income presents a puzzle [3]. Absence of a strong relationship is intriguing because as Dunn, et al. [4] argued "money allows people to live longer and healthier lives, to buffer themselves against worry and harm, to have leisure time to spend with friends and family, and to control the nature of their daily activities – all of which are sources of happiness". Moreover, people with high income have better nutrition, more free time, and more meaningful labor. The contradiction between potential possibilities for

improving well-being offered by money and the lack of a strong link between SWB and income seems to be partly explained by the fact that people often are not particularly happy with their purchases.

When individuals make a decision to buy something, they usually try to make predictions about the hedonic value or consequences of this purchase in the future. The process of foreseeing the future with respect to affective states is called affective forecasting and, according to the review provided by Wilson and Gilbert [5], people are often wrong in their forecasts. They discovered several sources of biases that cause errors in affective forecasting. Any of them could lead to inaccurate predictions and the situation where a wealthy person is not much happier than anyone else. Overall, it seems that in most of the cases people are neither good in affective forecasting nor are aware of characteristics indicating purchases that will potentially make them happier, and for this reason, do not use the opportunities for better SWB provided by wealth.

We suggest that people can potentially benefit from a recommender system with abilities to improve their affective forecasts and to offer intelligent guidance about spending. Many psychological biases that disturb affective forecasts of individuals are known to behavioral scientists and documented in the literature. For this reason, we argue that design of such a recommender system should be feasible taking into account excellent progress in the area of recommender systems that we saw from the early 1990s.

To the best of our knowledge, current research in recommender systems has not yet approached the problem of forecasting enjoyment and satisfaction in the domain of personal finance. We are still to see if technology can help people become happier with their purchasing decisions and improve SWB by recommending clever choices. It is however not clear how to approach design of such technology. What are the challenges and possible solutions?

The novel contribution of this paper is related to meta-analysis of the literature in behavioral sciences related to SWB and demonstration of how developments in this area enable design of new recommender systems for personal finance. We aim not just to identify new opportunities but also foresee and analyze major difficulties associated with designing a recommender system for application in personal finance that helps to optimize spending in terms of savings and SWB. Our analysis will be complemented with discussion of approaches towards overcoming these difficulties and further implications. We hope that it will help to initiate discussion and provoke thoughts on new research directions in the areas of personal finance and recommender systems.

2 SUBJECTIVE WELL-BEING

A brief review of the literature on consumption and happiness is necessary to demonstrate the current state of affairs in this area. The

¹ Department of Computer Science, University of Bristol, Bristol, UK, email: leonid.ivonin@bristol.ac.uk

² Department of Computer Science, Brunel University, London, UK

³ Department of Computer Science, University of Sussex, Brighton, UK

review will help to understand what are the current developments in social sciences and what input they can provide in design of recommender systems for personal finance.

We first propose to look at how psychologists approach measurement of SWB. Data on SWB usually consists of self-reports that reflect what people say about themselves when asked a particular set of questions [6]. There are a number of well-known surveys on happiness that are regularly conducted in several countries. They include British Household Panel Survey, European Social Survey, German Socio-Economic Panel, and the World Values Survey. Many researchers apply data from these panels in their work. An alternative to using data or question formulations from the widely recognized surveys is independent collection of data. This alternative often needs to be exploited when new hypotheses cannot be confirmed or rejected using existing data sets. It is not surprising that research on happiness is almost exclusively based on data collected with questionnaires. It seems that currently there is no better way of finding out how much individuals enjoy their lives than asking them questions.

Is it possible to sustainably increase SWB or this pursuit is futile? This is a point of debate between psychologists. Historically, it was considered that every person has a genetically determined set point for happiness and people tend to fluctuate around their baselines during lives [7]. Also, there is a concept of hedonic treadmill [8] that implies temporality of any gains in SWB. The argument behind this concept is that individuals always adapt to new situations or circumstances and their effect quickly diminishes. However, there is some recent evidence that SWB can be sustainably enhanced by practicing *intentional activities* [9]. Intentional activities are any actions in which people choose to engage. Not every activity suits to every individual. People have different psychological profiles and different strategies of intentional activities need to be applied. Examples of intentional activities include committing acts of kindness and practicing grateful thinking. These findings are important in the context of recommender systems because they indicate that cognitive or behavior interventions suggested by intelligent technological systems may lead to sustainable changes in well-being.

One of the findings from consumer psychology that we have already mentioned earlier is that high income is not always a recipe for a happy life [4]. More does not mean better and an individual need to be able to make right choices in the quest for happiness [10]. There are advocates of low-consumption lifestyles whose points of view are supported by outcomes of this research [3]. They argue that after a certain threshold increase in consumption does not make much sense and people ought to focus on different goals or values. However, it is not clear if low-consumption lifestyles will become mainstream.

Research in behavioral science demonstrated that people often make mistakes in forecasts about their own emotional states in the future [11]. There is a number of known prediction biases such as durability bias or impact bias [12]. Due to the biases, individuals tend to anticipate different duration and intensity of emotional feelings. As a result of such forecasting mistakes people sometimes put too much effort in pursuing goals that will not make them happy. From our point of view, biases in affective forecasting seem to be particularly suitable for being corrected by recommendations from intelligent systems for managing personal finance.

The last piece of research from behavior sciences that we are going to consider in this brief review is related to types of purchases and hedonic return. It was demonstrated that material and

experiential purchases lead to different profiles of satisfactions [13]. Consumers seem to consistently derive greater happiness from buying experiences than from tangible or material goods. This is another example of knowledge about SWB that is widespread between academics but not commonly applied in the real life.

From our analysis it is evident that science has accumulated some interesting findings about SWB generally and specifically with application to the consumption domain. We argue that now may be a good time to start exploiting this knowledge and attempt to design recommender systems that help to identify gratifying purchases.

3 RECOMMENDER SYSTEMS

Next, let us have a look at the state of the art recommendation systems in personal finance. The number of services providing intelligent recommendations has significantly increased during the last decade. The research on improvement of recommendation algorithms is being actively expanded [14]. Also, academics began inquiry into user experience with recommenders [15] by addressing the issues related to transparency of recommendations and trust of users to the system.

The majority of recommender systems reported in the literature work for a specific category of goods or services. For instance, they can support users in choosing a movie or a book. There are also cross-domain recommenders that enable support of personal decision making across different categories of goods [16], [17]. The research in the area of cross-domain recommenders seems to be the most relevant for the task of building a recommender for shaping spending in terms of SWB because decisions about the best purchases in terms of satisfaction and enjoyment usually require comparing alternatives from different domains. Recommender systems are usually deployed on the side of a company that is offering goods or services (e.g., on an e-commerce website) with the main motivation to increase sales. However, in the case of recommendations with regard to happiness and satisfaction, it seems to be more appropriate if a recommender system is run on devices belonging to an individual who receives the recommendations. Since we talk about a recommender system for personal finance, it will be best if a personal finance manager and a recommender engine are integrated in a single application. The main advantage of the integration is that the recommender system will receive data regarding consumption in real-time. So, it is necessary to review what are the latest advances in the area of personal finance.

Nowadays, we witness how the modern technology is changing the way people manage their personal finance. Ubiquitous computing has triggered an appearance of personal informatics systems that support people in collecting and reflecting data on their finance [18]. Such tools enable individuals to aggregate financial information, track transactions, create budgets, and set up goals [19]. One of the examples of a digital system for managing personal finance is Mint.com. Users of modern tools for managing personal finance benefit from precise information about their money and convenient interfaces for collecting this data [20]. These instruments help to optimize spending in the dimension of wealth. This approach for managing money is clear and well-established. However, it does not take into consideration the dimension of pleasure or happiness with regard to how the money is or should be spent.

4 PROBLEMS

Based on the review of modern systems for managing personal finance and a variety of recommender systems for different domains it is evident that there is no solution that would enable people to budget their spending in accordance with overall enjoyment of consumption. However, our analysis of the literature about SWB indicates that the latest finding enable design of such recommender systems. Now, let us analyze what are the challenges in development of recommender systems for personal finance that take into account enjoyment of consumption and identify potential opportunities to overcome them. We will not attempt to present an exhaustive list of problems but rather mention the most challenging and interesting ones.

4.1 Happiness and Consumption

All recommender systems operate based on underlying models that enable them to forecast what items a person is likely to enjoy or be interested in. If one is to approach the problem of designing an intelligent recommender system that guides users towards smarter and more enjoyable purchasing decisions, it is necessary to know how individual purchases contribute to the overall happiness in the domain of consumption. In other words, one needs a model describing relationships between spending and happiness. The problem of obtaining such a model probably needs to be tackled by academics from behavioral sciences or human-computer interaction because it requires conduction of user studies expanding our knowledge about SWB and consumption.

4.2 Measurement of Enjoyment

Another challenge in building recommender systems for allocation of personal finance is understanding how happy a user is with a particular purchase. An ability to quickly receive this information is crucial for performance of the recommender because it enables to identify inaccurate forecasts and build a knowledge graph for generation of next forecasts. The most straightforward approach towards measuring how happy a user is with a particular purchase is asking them questions. It is very similar to what researchers of happiness have been doing so far. However, when it comes to recommender systems used in a real life, asking questions about purchases is associated with certain difficulties. First, they are likely going to be intrusive and users may feel annoyed by the questions. Second, it is necessary to understand the context and know when is the best time to ask a question. For example, consumption of certain categories of goods (e.g., tickets for holidays) is delayed until some time in the future or can be continuous over a period of time. In such cases, the system will need to forecast when is the optimal time for measuring enjoyment of a purchase.

4.3 Meaningful Advice

The importance of capability to provide meaningful and persuasive feedback cannot be overestimated in the domain of recommender systems. Even a very accurate recommendation generated by a system can be of low value for a user if it is not communicated or presented in a way that encourages the user to trust the recommender. This also applies to recommender systems that attempt to understand emotional experiences associated with purchases and provide an advice about spending in terms of its affective value. Perhaps, the aspect of designing a trustworthy user

interface is even more significant when it comes to emotional experiences because people will not believe that a machine is able to understand their feelings and recommend purchases that will make them feel better. For this reason, a major challenge for a recommender system is not just forecast items that are likely to enhance SWB of users but also to intelligently present the recommendation. Since the recommendations are related to the area of personal finance, it is interesting to explore possibilities of integrating feedback into modern payment interfaces. For instance, a recommender system might communicate a warning that a potential purchase is going to be a waste of money by providing subtle feedback when a user is considering committing a transaction. The users might not trust it from the first time but, if the warning turned out to be correct, they are likely to pay more attention in the future.

5 METHODS AND APPROACHES

It is proposed to approach the *first* problem outlined above (4.2) through a number of quantitative experiments with individuals. The goal of these experiments will be to see how their happiness in the domain of consumption is related to past emotional experience with certain things and services that they purchased. The experiments will require collection of data about psychological backgrounds of participants that will help to see how personality traits influence consumption and SWB. Next, it will be necessary to record experiences of the individuals using either self-reports or techniques of affective computing [21]. The latter approach can potentially enable researchers to obtain objective data about emotional states as opposed to subjective data from questionnaires.

The techniques of affective computing [22] can also be valuable for approaching the *second* problem that we outlined. Indeed, if a recommender system can receive real-time feedback about enjoyment of consumption using recordings of physiological signals that indicate specific emotional states, it will be an efficient solution to the measurement problem. In this case, there is no need to bother users with questions and the system can receive continuous feedback on enjoyment of a particular purchase. Although automatic analysis of affective data eliminates the necessity of using questionnaires, the problem of understanding the context knowing when to measure remains. One possible solution is to use additional environmental data such as location and agenda if users authorize the recommender system to access them.

The *third* problem that we considered in this paper is related to presentation of feedback from the recommender system. As we wrote earlier, it is likely that users will not have confidence in the recommendations provided by the system because they will concern very sensitive aspects such as feelings and SWB. People strongly prefer basing affective predictions on their own mental simulations of future events or purchases rather than relying on previous experiences of other people. Even worse if forecasts of enjoyment need to be based on feedback from a machine. However, since information about how much complete strangers enjoyed an experience could help significantly improve forecasts, it is necessary to use and present it in a persuasive way. The research on recommender systems has already identified some clever ways of making recommendations look more trustworthy. For example, by presenting users how a system came to a particular conclusion or mentioning interests that two individuals have in common. The best way to approach this problem is to evaluate different ideas of

communicating recommendations in qualitative user studies that will shed light on possibilities for increasing credibility of the feedback.

6 IMPLICATIONS

Not just technology but also people themselves do not understand very well how things work in the realms of happiness and SWB. The implication from the analysis presented in the paper is that there are many areas for further investigation in the field of recommender systems for personal finance. Moreover, this work can hardly be done by people from one discipline. Ideally, a joint effort is required from researchers with backgrounds in behavioral science, computer science, recommender systems, and human computer interaction. Another implication is that research in engineering disciplines can potentially drive and contribute to the inquiry in behavioral sciences by developing systems for collection of data that will help to advance knowledge about SWB.

7 CONCLUSION

In this paper, we identified new opportunities for the research related to recommender systems in personal finance and analyzed the latest developments in the areas of SWB and recommender systems that are relevant to these opportunities. We argued that it is a good time to attempt design of recommenders that aim to optimize happiness in the domain of consumption. Following the analysis, important problems concerning development of such recommender systems were discussed. They included understanding of relationship between purchases and SWB, measurement of enjoyment, and credible presentation of recommendations. Then, possible solutions were suggested, and finally, we briefly outlined implications of our analysis. Being happy is one of the most important goals of people but unfortunately they often make inaccurate forecasts about hedonic value of events in the future and spend their money on things that do not make them happy. We demonstrated that advances in research on recommender systems have a potential to enhance SWB of individuals.

ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 655723.

REFERENCES

- [1] E. Diener, "Subjective well-being. The science of happiness and a proposal for a national index.," *Am. Psychol.*, vol. 55, no. 1, pp. 34–43, 2000.
- [2] R. Inglehart, *Culture Shift in Advanced Industrial Society*. Princeton University Press, 1989.
- [3] A. Ahuvia, "If money doesn't make us happy, why do we act as if it does?," *J. Econ. Psychol.*, vol. 29, no. 4, pp. 491–507, 2008.
- [4] E. W. Dunn, D. T. Gilbert, and T. D. Wilson, "If money doesn't make you happy, then you probably aren't spending it right," *J. Consum. Psychol.*, vol. 21, no. 2, pp. 115–125, 2011.
- [5] T. D. Wilson and D. T. Gilbert, "Affective Forecasting," *Adv. Exp. Soc. Psychol.*, vol. 35, pp. 345–411, 2003.
- [6] G. MacKerron, "Happiness economics from 35000 feet," *J. Econ. Surv.*, vol. 26, no. 4, pp. 705–735, Sep. 2012.
- [7] D. Lykken and A. Tellegen, "Happiness is a Stochastic Phenomenon," *Psychol. Sci.*, vol. 7, no. 3, pp. 186–189, 1996.
- [8] P. Brickman and D. Cambell, "Hedonic Relativism and Planning the Good Society," *Adaptation-level theory: A symposium*. pp. 287–302, 1971.
- [9] S. Lyubomirsky, K. M. Sheldon, and D. Schkade, "Pursuing happiness: The architecture of sustainable change.," *Rev. Gen. Psychol.*, vol. 9, no. 2, pp. 111–131, 2005.
- [10] C. K. Hsee, R. Hastie, and J. Chen, "Hedonomics: Bridging Decision Research With Happiness Research.," *Perspect. Psychol. Sci.*, vol. 3, no. 3, pp. 224–43, May 2008.
- [11] D. Kahneman and J. Snell, "Predicting a changing taste: Do people know what they will like?," *J. Behav. Decis. Mak.*, vol. 5, no. 3, pp. 187–200, Jul. 1992.
- [12] T. D. Wilson and D. T. Gilbert, "Affective Forecasting," *Adv. Exp. Soc. Psychol.*, vol. 35, pp. 345–411, 2003.
- [13] L. Van Boven and T. Gilovich, "To Do or to Have? That Is the Question.," *J. Pers. Soc. Psychol.*, vol. 85, no. 6, pp. 1193–1202, 2003.
- [14] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative Filtering beyond the User-Item Matrix: A Survey of the State of the Art and Future Challenges," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 1–45, May 2014.
- [15] J. A. Konstan and J. Riedl, "Recommender systems: from algorithms to user experience," *User Model. User-adapt. Interact.*, vol. 22, no. 1–2, pp. 101–123, Mar. 2012.
- [16] R. Chung, D. Sundaram, and A. Srinivasan, "Integrated personal recommender systems," in *Proceedings of the ninth international conference on Electronic commerce - ICEC '07*, 2007, pp. 65–74.
- [17] A. Fano and S. W. Kurth, "Personal choice point: helping users visualize what it means to buy a BMW," in *Proceedings of the 8th international conference on Intelligent user interfaces - IUI '03*, 2003, pp. 46–52.
- [18] I. Li, A. Dey, and J. Forlizzi, "Understanding My Data, Myself: Supporting Self-Reflection with Ubicomp Technologies," *Proc. 13th Int. Conf. Ubiquitous Comput.*, pp. 405–414, 2011.
- [19] I. Li, A. Dey, and J. Forlizzi, "A stage-based model of personal informatics systems," *Proc. 28th Int. Conf. Hum. factors Comput. Syst. CHI 10*, p. 557, 2010.
- [20] C. Lu, "Enhanced user experience in managing personal finance," in *Proceedings of the 14th international conference on Human-computer interaction: users and applications*, 2011, pp. 375–383.
- [21] R. W. Picard, *Affective computing*. Boston, MA: MIT Press, 2000.
- [22] L. Ivonin, H.-M. Chang, W. Chen, and M. Rauterberg, "Automatic recognition of the unconscious reactions from physiological signals," in *SouthCHI 2013, LNCS 7946*, A. Holzinger, M. Ziefle, M. Hitz, and M. Debevc, Eds. Berlin Heidelberg: Springer-Verlag, 2013, pp. 16–35.

NORMASEARCH: a Big Data application for financial services.

Ylenia Maruccia¹ and Giovanni Pansini¹ and Gloria Polimeno¹ and Felice Vitulano¹

Abstract. In the recent years banking and financial markets are trying to learn how Big Data can help to transform their processes and organizations, improving customer intelligence, reducing risks, and meeting regulatory objectives. The collection and the analysis of new legislations, understanding if they are introducing new aspects with potential impacts on different fields, could be the basis of a system able to give support in the strategic decision making process and to evaluate the potential impacts on both management and strategic activities. Here we want to present *NormaSearch*, a Big Data application developed by Exprivia, an international leading company in Italy in the process consulting, technology services and information technology solutions. *NormaSearch* is able to analyse specific information taken from the web, both in a structured and unstructured form, and its application in the financial fields.

1 Introduction

The last years have seen a continuous increase of data generated in many fields, from science to social life, passing through industries, which we refer to as “Big Data”. Each actor, e.g. individual, administration, organization or business, is a producer of new forms of data, both in structured or unstructured form: they can be personal data, conversations on social network, medical data, meteorological information, shared photos, and so on.

The challenge of both public and private companies is to easily manage these huge amount of data with improved technologies, different from the traditional ones, and extract knowledge from all these hidden information.

The main characteristics of Big Data are that they are too big, move too fast and do not fit the structures of traditional database architectures, so new technologies are necessary to manage them. Moreover, one of the main difficulties, as aforementioned, is the format in which all the information are generated: in particular, they can be in a structured, unstructured or semi-structured form. So new forms of databases, programming languages and hardware architectures are used to either store Big Data or to transform it from unstructured or semi-structured format into a well-structured one, with consequences in many fields of application.

According to [1], Big Data help to better listen to customers, understand their ways of using services and hence the offer, simplifying also the decision making process. To this aim, an important role is played by those applications that tailor the information based on the needs of the customers. Technologies such as Recommender System are now used by many brands with the aim of suggesting products or services which a user may be interested in.

In these years banking and financial markets firms are continuing to learn how Big Data can help to transform their processes and organizations. In particular, for banks, Big Data initiatives predominately still revolve around improving customer intelligence, reducing risk, and meeting regulatory objectives.

Machine learning techniques, for example, can be applied within the fraud and risk sectors, improving models and allowing acceleration towards more real-time analysis and alerting. Finding new legislations, understanding what are the differences with the existing ones and/or if new aspects have been introduced, can be a very important challenge in this field of application, with the purpose of evaluating the potential impacts on both management and strategic activities and of giving support in taking those strategic decisions that could minimise potential costs.

Here we want to present our solution, called *NormaSearch*, developed in Exprivia to manage the data generated from different sources and coming mainly in an unstructured format, at the aim of adapting it in the banking system. In Section 2 it will be described the scenarios in which *NormaSearch* could be applied, while in Sections 3 and 4 it will be presented the application with its component.

2 Scenarios

The financial crisis and the speculative use of the derivative instruments has placed the reform of the derivative markets “Over The Counter” among the priorities of the legislature in terms of standard negotiation procedures, as well as more stringent rules pertaining to the capitalization of financial intermediaries:

- In terms of rules designed to standardise the trading of OTC derivatives, it has been promulgated different regulations, such as *EMIR/DOIT Frank Act* (European Market Infrastructure Regulation), that revived the role of the Central Counter-Parties (CCPs), with the aim of increasing transparency and reduce both the counterpart risk and the operational one (see Fig. 1).
- To ensure the soundness of the banking system, the Basel agreements require the banks of the leading world countries some limits about their operational activities, especially regarding the amount of assets which they have to equip themselves for their clients’ protection, thus allowing the capitalization of banks (and, consequently, liquidity guarantees), to guarantee the operations - collection, financing an investment - put in place with customers.

Therefore, it can be deduced as today a Financial Intermediary is called to observe the dictates imposed by the regulations in the area of interest, involving adjustments to the operational processes and/or IT architectures, in compliance with regulations.

¹ Exprivia Spa, Bari - Italy, email:{ylenia.maruccia, giovanni.pansini, gloria.polimeno, felice.vitulano}@exprivia.it

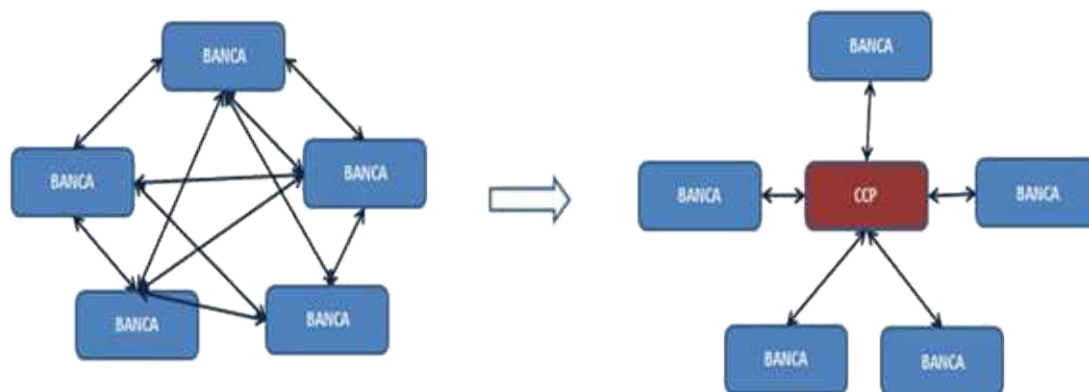


Figure 1. The role of the Central Counter-Parties.

The granularity and, at the same time, the complexity of these regulations, necessitate a constant attention and monitoring of them, in order to anticipate future changes, or integrations, or evolutions.

In this context it engages the idea of providing a machine learning tool which, through the analysis of the newly introduced legislation (or in the approval process) and/or the changes in the requirements previously promulgated (detectable by special certificates internet sites), may provide guidance on the bank process involved and, therefore, indicate with almost predictive function the impacts on the IT application, in terms of changes or new implementations, in support of the above processes.

To this aim, we developed a Big Data application that is able to analyse specific information, both in a structured and unstructured form, taken from the web. This application, called *NormaSearch*, is described in the next section.

3 NormaSearch Functionalities

As aforementioned, *NormaSearch* is a Big Data application that allows to browse the web, search and analyse specific information on the bases of given rules that are defined by the user. The application has two distinct sections, one of Administration and one of Fruition: the first allows to train to recognise and classify the information of its own interest through a series of examples (weakly supervising training), while the second allows users the analysis of the sites collected independently by the system. Once trained, the application allows to analyse web pages and documents in the sites, news group, blogs, forums and so on, according to a process specifically designed for linguistic and conceptual analysis of online content.

This process allows to achieve an optimum precision to coverage ratio in the search, as well as to limit in an important way the amount of downloaded web pages and, consequently, the hardware resource consumption.

The application operates their own research in an incremental way: by doing so, the pages are presented to the user only in the case in which they have never previously been recognised or, in the case where the content is changed. In details, the application is therefore able to:

- Refer autonomously a set of sites, blogs, forums and so on, looking for info about a set of concepts of interests identified by the

machine training activity by examples (weakly supervised training); the system is able to consult a set of predetermined sites (authoritative sites) or even the whole www.

- Identify, in every web page retrieved, the individual portions of text (HTML page section) in which are expressed the sophisticated concepts, by associating a percentage indicator of relevance to such concepts with each section identified.
- Automatically classify and organize web sites and pages that belong to them according to a predetermined conceptual taxonomy or derivable during the training phase machine.
- Filter, as needed, specific types of web sites that tend to generate noise, such as for example search engines based on search engine spamming techniques.
- Identify only new content found on each new consultation.
- Present the results through a simple web interface or as a report directly downloadable from the interface; reports can also be sent from the application via e-mail.
- Independently identify potentially authoritative sites and recognise inactivity of authoritative sites.

4 NormaSearch Architecture

In Fig. 2 it is shown the architecture of this application.

It is made up of two main components, a client and a server ones, both described below.

NormaSearch Client. It is specialised on the interaction with the user and the transmission of user requests to the server component. It is structured into two main parts:

- A fruition console (in Fig. 2, Retr.UI). Here the user can manage documents and decide which of them have to be processed, or could be useful for the launch of new experimental projects on specific themes, or dismissed.
- An administration console (in Fig. 2, Admin. Ui). Here the user can define the security rules, the loader dedicated to the web monitoring and to the retrieval of documents of interest, the definition of categories and subcategories of the safety rules through which the conceptual framework of the rule itself is defined in terms of topics and the organization of them, the training of the security rules and the related conceptual categories. Moreover, here it can be also specified new projects where where the user can insert additional or parallel categories to the security rules established

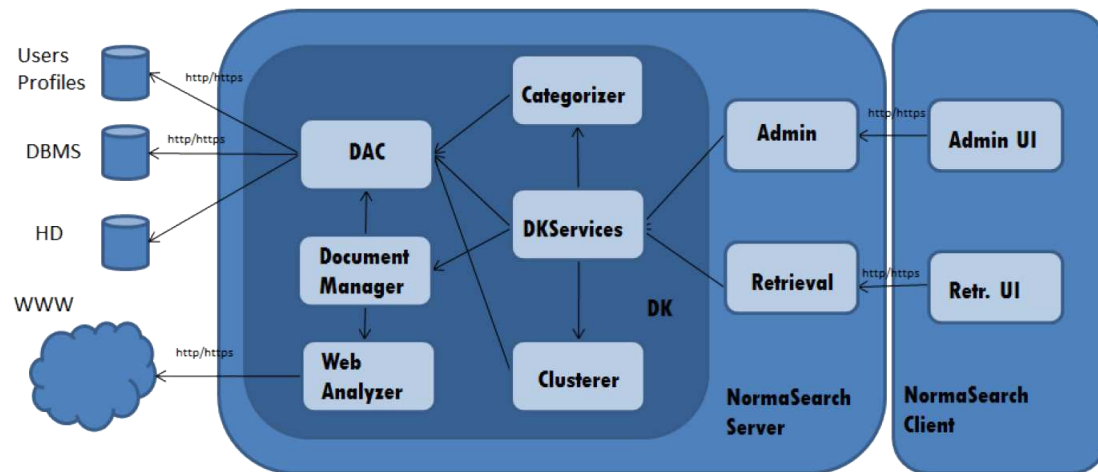


Figure 2. Norma Search Architecture.

above, in order to look “cool stuff” that can be used as a starting point to expand the research in the field of banking regulation or to add/modify existing safety rules.

NormaSearch Server. It is specialized on the receipt of the users’ requests and the sorting of the same to the server components, these ones suitable of taking charge of specific requests. As NormaSearch Client, it has two components: the administration component and the fruition one, designed to manage requests from the administration and fruition consoles, respectively, and send them to the specified server components.

At the moment NormaSearch is in productive use by an Italian bank. Moreover, in NormaSearch Server there is an important software component, called Big Knowledge [4], developed by Exprivia.

Big Knowledge is able to manage both structured and unstructured data and, as it can be seen in Fig. 2, it is made up by six main components:

- **DAC** The Data Access Component is the centralized component to access data, either in a DBMS or, in the BigData cases Solr.
- **Document Manager** It is the component through which BK is able to convert the document provided in textual form, and clean it from useless portion of text (e.g. html banners).
- **Information Extraction Manager** It annotates the document, extracting relevant information like Named Entity Recognition by the usage of Finite State Automata, and elements inside custom gazeteers.
- **Clusterer** It is dedicated to the extraction of conceptual groups (clusters). These ones are automatically extracted using advanced techniques of NLP (Natural Language Processing) based on Latent Semantic Analysis[2] (LSA) and a Markov clustering algorithm[3]. The generated clusters are crucial for the training tuning of the system.
- **Categorizer** It is intended for the automatic classification and organization of the document according to a conceptual taxonomy expressed as a set of clusters and constructed manually or semi-automatically by describing the category with a small text.
- **Geo Recognizer** Using a gazeteers of places kept from Geon-

ames², it annotates the documents gathering: nations, regions, cities, airport, port and generically geographic points of interest. A kml export of single of multiple document, is provided when a WMS³ compliant system is integrated.

5 Conclusion

Big Data are changing the industrial world and, for this reason, all kind of companies need to be capable of managing this huge amount of data and to extract useful information from them. This great interest in Big Data is present also in the financial services, which can obtain important information from the analysis of both structured and unstructured data. It is also important to have these information in a useful time, in order to prevent losses and to predict important event before they happen. In this paper we discussed a Big Data solution in financial service field developed by Exprivia. It is called NormaSearch and it aims at predicting the impact of a legislation change, or the introducing of a new one. After a brief introduction about Big data and machine learning technologies, we described NormaSearch and its components and how it works. It can be analysed the new introduced legislation and also provided a guidance on the bank process involved. It can be predicted, in particular, the impacts on the IT application in support of that bank process. Moreover, in this paper it has been described an important Big Data solution that is present in NormaSearch. It is called Big Knowledge and it is composed by six components that speak together in order to manage all the documents in input and extract important information that can be then classified. This paper showed how a Big Data solution can be useful in financial field and can predict important information in an useful time in the strategic decision making process.

REFERENCES

- [1] C.L.P. Chen and C.Y. Zhang, ‘Data-intensive applications, challenges, techniques and technologies: A survey on Big Data’, *Informatics and Computer Science Intelligent Systems Applications*, 275.

² <http://www.geonames.org/>

³ <http://www.opengeospatial.org/standards/wms>

- [2] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, 'Indexing by latent semantic analysis', *Journal of the American Society for Information Science*.
- [3] Stijn van Dongen, *Graph Clustering by Flow Simulation*, PhD thesis, University of Utrecht, 2010.
- [4] F. Vitulano, M. Cammisa, and Y. Maruccia, 'Unleashing big data power for sea emergency control', in *Proceedings Tethys 2015*.

A Smart Financial Advisory System exploiting Case-Based Reasoning.

Giorgio Leonardi¹ and Luigi Portinale¹
and Paolo Artusio² and Marco Valsania²

Abstract. In the financial advisory context, knowledge-based recommendations based on Case-Based Reasoning are an emerging trend. They usually exploit knowledge about past experiences and about the characterization of both customers and financial products. In the present paper, we report the experience related to the development of a case-based recommendation module in a project called SMARTFASI. We present a solution aimed at personalizing the asset picking phase, by taking into consideration choices made by customers who have a financial and personal data profile “similar” to the current one. We discuss the notion of distance-based similarity adopted in our system and how to actually implement an asset recommendation strategy integrated with the other software modules of SMARTFASI. We finally discuss the impact such a strategy may have both from the point of view of private investors and professional users.

1 Introduction

The evolution of the international financial context (often dictated by the worldwide economic and financial crisis) has progressively changed, often in a radical way, the attitude of investors. One direct consequence is that single investors are no longer simply classifiable into as *private*, *retail* and *affluent* in the traditional way; on the contrary, a common aspect among all the different types of investors is the need to have more clarity on the financial products and the possible benefits from tailor-made services. Likewise, there is a change in commercial strategies, switching from different approaches to each market segment, toward the adoption of common strategies covering multiple segments [6]. This can lead to a global standardization of banking services through the identification of common needs among different market segments [13].)

A partial answer to the first issue (i.e., the difficulty in exploiting a traditional investor’s classification scheme) has been provided by the introduction of specific norms (as for example the EU MiFID guideline [1]). On the other hand, concerning the needs of the users (i.e., the investors), a rapid evolution of the financial advisory process is taking place; the goal is to provide the user with a financial proposal that is most suitable for the users needs and profile, and goes beyond the consideration of legal issues as the only guideline. For this reason, recommendation strategies are becoming quite popular in the financial advisory context, with particular attention to the Case-Based Reasoning (CBR) paradigm [10, 16, 14, 15]. In general, we can use three main approaches in recommendation operations [12]:

- Collaborative Filtering: assuming that human preferences are correlated, we can collect preferences of a large set of customers in order to define a recommendation based on preferences of people with similar interests.
- Content-based filtering: use of preferences of a specific customer to infer recommendations, based on specific categories (keywords) connected to a profile.
- Knowledge-based: recommendations are based on different levels of knowledge about the product domain.

In finance, knowledge-based method (among which CBR) is mostly applied, as investment recommendations must primarily conform to legal regulations (i.e. MiFID) in order to ensure investors against mismatching and/or fraudulent financial proposals [20]. Moreover, historical data are available, making possible, as predicated by the CBR paradigm, to exploit knowledge about past experiences and about the characterization of both customers and financial products.

In addition, thanks to the IT advances, an emerging trend is to base financial services on web and mobile technologies, with strict collaboration between the end-user and the consultant, in such a way as to get the users more and more involved in the final definition of their stock portfolio. In this context, a phase of basic importance is that of *asset picking*; in this phase, advanced data analytic tools are adopted, in order to compare the risk and performance of the considered financial products, perhaps prior filtering of the assets by means of specific features, either identity-based (as asset class, country, region, currency) or measured (as duration, historical volatility, time to maturity, historical performance, etc...). In this paper, we present the solution adopted in the SMARTFASI project, which has the goal of designing and implementing a web-based architecture for a financial decision support system able to supply a set of advanced consultancy services for the management of financial assets, whilst taking into account the risk/performance trade-off. The advisory system prototype has been designed with different goals in mind:

- the exploitation of Cloud and High-Performance Computing (HPC) paradigms at the *infrastructure level*;
- the exploitation of stochastic modeling and Montecarlo simulation, together with Case-Based Reasoning (CBR) [2] at the *methodological level*.

Cloud and HPC infrastructure have been introduced to support stochastic simulation which is a computationally intensive activity. The aim is to provide the user with a set of simulation tools, in such a way that he/she can simulate the assets behaviour in a specific time horizon, by computing for instance the expected yield and indices like the CVaR (Conditional Value at Risk) with a given confidence

¹ Università del Piemonte Orientale, Italy, email: {giorgio.leonardi, luigi.portinale}@uniupo.it

² ORS srl, Italy, email: {paolo.artusio, marco.valsania}@ors.it

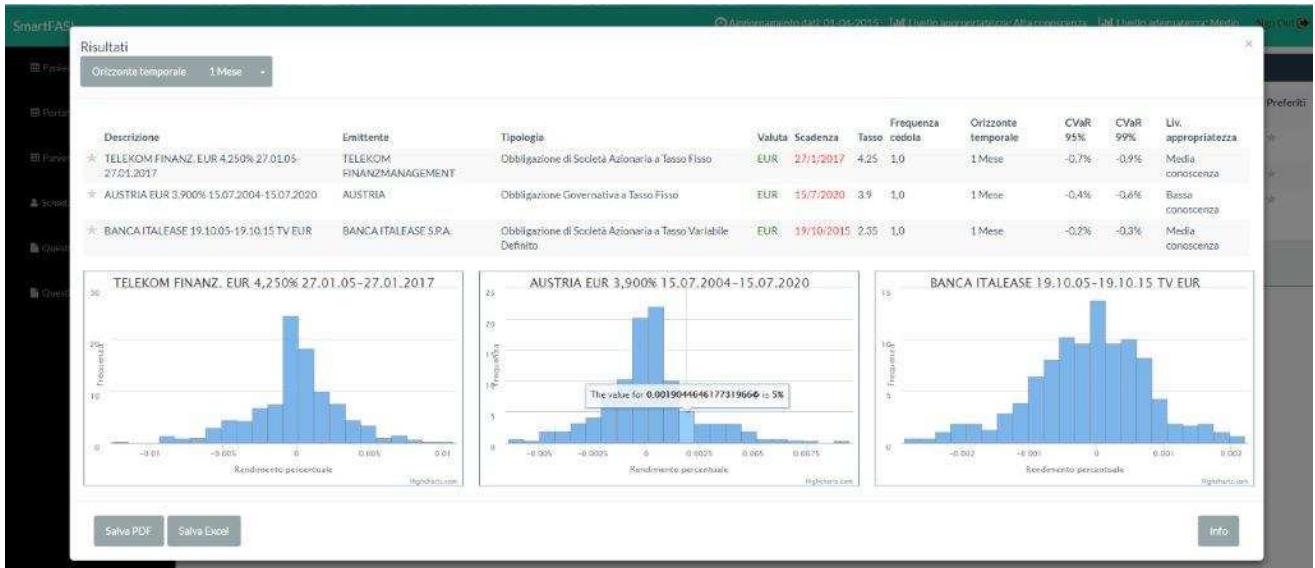


Figure 1. Comparison of different products

level (e.g., 95% and 99%). This can be done by considering either a single product or by comparing several options. Figure 1 shows an example of products comparison exploiting Monte Carlo simulation [8, 5].

However, the use of simulation tools leave the user alone in the choice of the financial products; for this reason the system has been enriched with a *case-based recommendation engine*, implementing a knowledge-based recommendation strategy, and able to suggest to the users a set of options tailored to their needs. The asset picking phase has then been expanded by taking into consideration, among the others, the frequency of use of products selected by customers who have a financial and personal data profile “similar” to the current one. The underlying assumption is that individuals who share several features (in terms of financial needs), will act on the market in a similar way.

The focus of the present paper is on such a case-based recommendation engine; in the following sections we will details both the methodological issues as well as the architecture on which this part of the SMARTFASI system is based. The exploitation of CBR techniques allows us to address the following targets with potential different end-users:

- **Private Investors:**

- to improve the vision of the global investment scenario, by putting more emphasis and focus on the individual user features (e.g., financial attitude), producing more informed choices for the users;

- **Professional Users** (e.g., consulting agents or firms):

- to propose to the customers some investment scenarios which are no more generically based on the financial features of the products only, but also more tailored to the specific customer profile, by personalizing in this way the service (for example by comparing benchmarks more suitable to the customers);
- to exploit new analytical tools to evaluate the value of the set of potential investments, or alternatively to suitably modify this set, in order to fulfil the customers needs, preferences and requirements.

- to perform historical analyses on clusters of clients, discovering potential trends of investments that may be consequently supported or contrasted, by evaluating the commercial offer in a more informed way
- to improve customers acquisition process, tying business targets to the interests of the consumers, so boosting the value of the company’s clients portfolio

The remainder of the paper will be organized as follows: Section 2 introduces the basics of the CBR paradigm exploited in the recommendation engine, Section 3 discusses the case-based recommendation methodology introduced in the SMARTFASI project, while in Section 4 the basic architecture of the advisory system is outlined. Final considerations are then reported in Section 5.

2 The CBR paradigm

Case-Based Reasoning (CBR) [18] is a problem solving methodology that addresses the task of solving a new problem (the *target case*), by retrieving, and possibly adapting, the solutions of past problems similar to the one to be solved. The basic idea is to store a set of solved cases in a *case library*, and then to re-use such cases when a new problem has to be solved. The main assumption underlying the CBR process is that similar problems have similar solutions; in this way the solution of a past case can be used to address the solution of a new similar case.

CBR is also considered as a *lazy learning* technique [3], in contrast with *eager learning* where a suitable model is constructed from training cases, which are then no longer needed for problem solving. In CBR, training instances are kept in memory and are directly used when a new case is presented as a target. Following the classical framework described in [2], there are four main step in a CBR problem solving session, the so-called 4R’s (see figure 2):

- **Retrieve.** It determines the cases that are most similar to the new problem. The notion of similarity is implemented by defining a notion of distance among the case features, and by finally combining such local distances (at the feature level) into a global measure (at

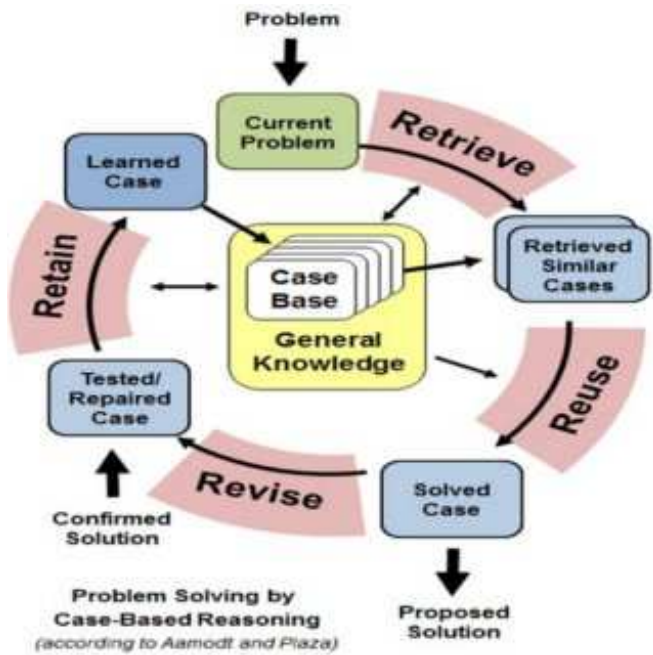


Figure 2. The 4R's CBR cycle

the case level). The retrieve step is usually implemented through k-Nearest Neighbour (kNN) search [19].

- **Retrieve.** The solution of a retrieved case is selected and proposed as a candidate solution to the new problem. If it can be suitably applied to the target case it becomes a solution for the latter as well. Otherwise, it is passed to the next CBR step.
- **Reuse.** This step adapts the candidate solution to the target case, in such a way that it can be applied to it. Knowledge intensive methods can be necessary in this step to perform such an adaptation. If revision is not possible, the system fails in finding a suitable solution to the target.
- **Revise.** This is the actual learning step: it evaluates the obtained solution and it decides whether to retain the new solved case in memory. Because of the well-known *utility problem* [7], not every solution should be stored in the case library, and the case library should be properly maintained (see [17]).

The step that has received most attention is definitely the *retrieve* step; indeed, case retrieval is essential to every application of case-based systems, and in particular to case-based recommendation [12, 4]. Case-based recommendation is usually considered a particular instance of content-based recommendation [11], where cases are typically used to model items, through a classical feature-based description. However, case-based recommenders are more suitably considered as knowledge-based recommenders [12], since they exploit both similarity-based retrieval and general knowledge about users and items (e.g., user's preferences). In fact, one can regard case-based recommenders as collaborative filtering recommenders as well, since the suggestion of similar items to similar users is in principle possible. Instead of directly manipulating matrices of rankings as in standard collaborative filtering approaches [9], they can adopt content-based similarity measures to compare users and their preferences with respect to the items of interest. Next section will discuss the CBR methodology we have introduced in the SMARTFASI advisory system, by presenting the details of an asset retrieval strategy

based on customer's similarity.

3 Case-Based Recommendation

The SMARTFASI recommendation module uses the information available about the customer currently under study (the new or current or target case following the scheme of Section 2), to provide a recommendation of financial products (i.e., the solution in the CBR framework), based on the investments made by similar customers. In this project, the customers are defined by the features presented in Table 1, in Section 3.1. According to the CBR paradigm, similarity is implemented as a proper dual notion of a distance measure. In our case, the distance functions involve features concerning the personal information of the customers, their spending power, their knowledge of the financial domain and the composition of the portfolios they may manage at the moment.

The recommendation strategy takes place as a multi-step procedure. The first step (*Step 1*) performs a selection of the most similar customers with respect to the target one, on the basis of personal data and of the overall composition of their portfolios, as described in detail in Section 3.2. While the above step focuses on the general characteristics of the target customer to retrieve the most similar ones, the next step (*Step 2*, see Section 3.3) concentrates on the investment strategies of these customers, in order to perform a further selection which identifies the subset of the most similar portfolios owned by the previously selected customers, with respect to the portfolios owned by the target one. This means that the recommender module will specifically focus on the financial features only after the first filtering step, thus working on a restricted set of customers who share the same personal data, lifestyle and investment capabilities with the target customer. Moreover, *Step 2* is optional, since its execution depends on whether the target customer already has an active portfolio at the current time. If no active portfolio is available for the target customer, then *Step 2* is not performed, since no portfolio comparison can be made. The third step (*Step 3*, described in Section 3.4), finally extracts the K products to be returned as recommended to the users for their evaluation.

In the rest of this section, we will detail each step described so far, together with the characterization of the features defining a customer and with the distance metrics introduced for the similarity evaluation.

3.1 Case Definition

In the approach we propose, a case describes the characteristics of a customer (the investor) in the SMARTFASI system. The customer's features describe their personal characteristics, their investment capabilities, their financial adequacy (knowledge of the financial domain) and the composition of any portfolio they hold. As usual in the CBR setting, each of these features is associated with a weight that defines its *importance* (we assume three possible levels of importance: 3 = high, 2 = medium, 1 = low). The features defining a customer and their relative weights are determined by the domain experts involved in the SMARTFASI project, and are listed in Table 1.

These features are a mix of heterogeneous information, such as numeric values (Age, Available capital, Adequacy and N. of children), coded information (Marital status, Education, Sex and Type of employment) and arrays (Asset allocation for each portfolio). Among such features, it is worth noting that the Adequacy is a pre-computed value, identifying the ability of the customer to understand the implications of buying financial products having different risk levels. The Adequacy is directly linked with the MiFID profile (see [1]) assigned

Feature #	Feature	Weight
1	Age	3
2	Available capital	3
3	Risk tolerance / Adequacy	3
4	Asset allocation for each portfolio	3
5	N. of children	2
6	Marital status	2
7	Education	2
8	Sex	1
9	Type of employment	1

Table 1. Features defining the personal data of a case

to the customer by the financial organisation which manages his/her interests.

Furthermore, the arrays describing the asset allocation in a case are defined at two different levels, as shown in Figure 3.

In this representation, each single asset can be classified as *C* (Corporate) or *G* (Government). Moreover, each asset can be associated with a *Fixed* (F), *Variable* (V) or *Floating with Cap* (C) rate. The combination of these two classifications generates six different groups of assets, in such a way that each asset stored in the reference data base belongs to one of these groups. For each portfolio, the first-level representation is an array containing the percentage of assets in each of the above 6 classes. Considering all the portfolios of a customer at the first level, it is possible to characterise the general investment preferences of a customer's investment. Since this level of abstraction is useful for characterising the overall investment behaviour, it is exploited together with the other personal data to compose, in the *Step 1* of the recommendation module, the ranking of the customers who are globally more similar to the target one.

The second level representation of the portfolios is an array as well, where each location identifies a specific asset. The contents of the array indicates, for each title, its share in the composition of the portfolio. The description of the portfolios at this level of detail shows which investments have been made by a customer at the maximum granularity available. This information describes exactly the financial behaviour of a customer, therefore it is used, in *Step 2*, in order to select the most similar portfolios, by taking exclusively into account the financial aspects of customers sharing their anagraphical and life-style information with the target one.

In the next subsections, we will detail each specific step on which the recommendation strategy is based.

3.2 Step 1

The first step is devoted to the selection of the most similar customers with respect to the query one, using the personal information shown in Table 1; this focus on the general characteristics of the target customer, without taking into account the financial preferences yet. This selection is performed through a Nearest Neighbour search [22], comparing the query with the cases stored in the case library and cutting the results to the first *N* best matches. The value of *N* can be set by the system as a default value (for example, a given percentage of the number of cases in the case base), or provided by the user while defining the query. Since the cases are composed by features of different types, the Heterogeneous Euclidean-Overlap Metric (HEOM) is a natural choice for distance definition [21]. Consider a given feature *f* with possible values *x, y* \in $\text{range}(f)$, the HEOM metric is defined as follows:

$$D_{HEOM}^f(x, y) = \begin{cases} 1 & \text{if } x \text{ or } y \text{ is unknown} \\ \text{overlap}(x, y) & \text{if } f \text{ is nominal} \\ \text{rn_diff}(x, y) & \text{otherwise} \end{cases} \quad (1)$$

The first possibility of Eq. 1 refers to the situation where the feature *f* has no value either in the target or in the retrieved case (or in both). In case of a nominal feature, *overlap* is an $n \times n$ square matrix ($n = |\text{range}(f)|$), where $\text{overlap}(x, y) \in [0, 1]$ measures the distance between values *x* and *y* of *f* (in the extreme case $\text{overlap}(x, y) = 0$ if $x = y$ and $\text{overlap}(x, y) = 1$ if $x \neq y$).

Finally, $\text{rn_diff}(x, y) = \frac{|x-y|}{\text{range}(f)}$ is the range normalized absolute difference of the feature values, in case of a linear (e.g. numeric) feature. The range of each linear feature *f* is updated every time a new case is added to the case base, in order to keep the *rn_diff* in the $[0, 1]$ range for each linear feature, preserving the retrieval order of the customers. The definition in Eq 1, has the advantage of returning a distance value in the range $[0, 1]$; similarity can then be expressed as $S^f(x, y) = 1 - D_{HEOM}^f(x, y)$ where $S^f(x, y) = 1$ means perfect similarity and $S^f(x, y) = 0$ means total dissimilarity.

By considering Table 1, features 1, 2, 3, 5 and 8 are treated as linear features. On the other hand features 6, 7, 8 and 9 are considered nominal and an appropriate distance matrix is adopted for each feature. What cannot be dealt with by the standard HEOM metric is the portfolio representation in a case. However, in *Step 1* we need to compare also first-level portfolios among cases. Since this information is stored as an array, a natural choice is to consider a local metric based on *cosine distance*; this choice is well justified in the financial domain where it has been adopted in several advisory systems [10, 15].

Given two arrays $\mathbf{a} = (a_1, a_2, \dots, a_n)$ and $\mathbf{b} = (b_1, b_2, \dots, b_n)$, the cosine distance between **a** and **b** is defined as:

$$D_{cos}(\mathbf{a}, \mathbf{b}) = 1 - \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2} \sqrt{\sum_{i=1}^n b_i^2}} \quad (2)$$

Since in our application every component of the array is non-negative, the above definition returns a value in the range $[0, 1]$. In particular, the asset allocation contained in a case is composed by a set of different portfolios, each one represented as a two-level array (as shown in Figure 3). The goal in comparing asset allocations is to determine the best match between the portfolios associated with the retrieved case and the portfolios owned by the target customer.

The strategy implemented in SMARTFAS1 is the following. Let $P_t = (p_t^1, p_t^2, \dots, p_t^n)$ and $P_c = (p_c^1, p_c^2, \dots, p_c^m)$ be the set of first-level portfolio arrays owned by the target customer *t* and a given customer *c* respectively (customer *c* is the one we are comparing the target to); each p_t^i and p_c^j are then arrays corresponding to the first-level representation of a specific portfolio for user *t* and *c* respectively.

Let $\text{Perm}(P)$ be a permutation of a set of portfolios *P*; the best match is the pair of permutations $\langle P'_t, P'_c \rangle \in \text{Perm}(P_t) \times \text{Perm}(P_c)$ resulting in the minimum overall distance D_p between the portfolios as defined in Eq. 3.

$$D_p(P_c, P_t) = \min_{\text{Perm}(P_c) \times \text{Perm}(P_t)} \frac{\sum_{i=1}^{\min(n, m)} D_{cos}(p_c^i, p_t^i)}{\min(n, m)} \quad (3)$$

The best matching portfolios of user *t* and *c* are then extracted as shown in Eq. 4.

$$(P'_t, P'_c) = \arg \min_{\text{Perm}(P_c) \times \text{Perm}(P_t)} D_p(P_c, P_t) \quad (4)$$

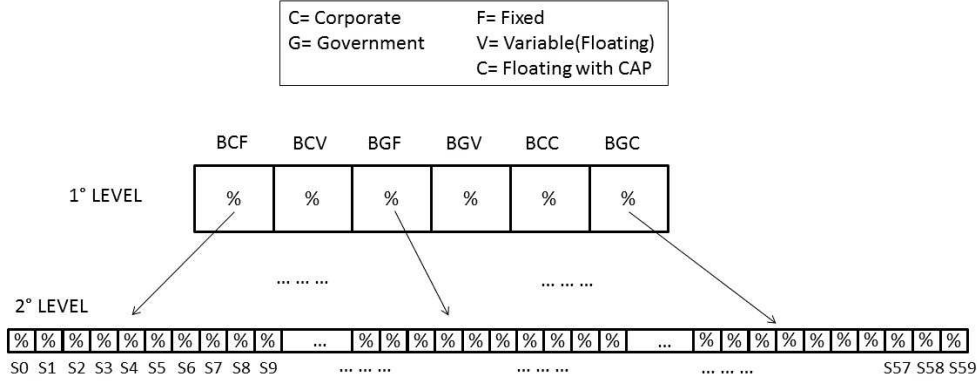


Figure 3. Two-level description of a portfolio

In particular, if one customer (either t or c) has more portfolios than the other, then the portfolios in excess in any given permutation are discarded. Since we consider any possible permutation, they are taken into account when a different permutation is considered. The best matching portfolios for each customer c (i.e., P'_c in Eq. 4) are finally stored in order to be re-used in *Step 2*. In case the target customer has no available portfolio yet, then we consider the asset allocation as a missing feature and we set $D_p(P_c, P_t) = 1$ as in the HEOM metric.

Finally, once the local distance for each feature has been computed (including the portfolio's distance), the overall distance function between two customers C_1 and C_2 is the normalized weighted average of all the local contributions:

$$D(C_1, C_2) = \frac{\sum_{i=1}^s w_i \cdot D(v_1^i, v_2^i)}{\sum_{i=1}^s w_i} \quad (5)$$

where s is the number of features ($s = 9$ in our application as shown in Table 1), v_1^i, v_2^i are the values of the i -th feature of customer C_1 and C_2 respectively, and w_i the importance weight of the i -th feature (see Table 1 again); furthermore

$$D(v_1^i, v_2^i) = \begin{cases} D_p(v_1^i, v_2^i) & \text{if } i = 4 \text{ in Table 1} \\ D_{HEOM}^i(v_1^i, v_2^i) & \text{otherwise} \end{cases}$$

The global distance defined by Eq. 5 is applied to compare the target customer with all the customers in the case base, in order to obtain the list of the N most similar customers to the target one. This list is then input to *Step 2* if the target customer owns at least one portfolio, in order to further filter these results using the financial information available; otherwise the list is a direct input to *Step 3* since *Step 2* is not applicable.

3.3 Step 2

In *Step 2* the system receives from *Step 1* the list of the N customers globally more similar to the target one, together with the list of the portfolios that best match the portfolios of the target customer. The set of such best matching portfolios is considered and a further filter over the financial information is applied; the goal is to extract the best assets to be recommended, by considering the specific allocations (second-level portfolio information) of such pre-selected similar customers.

Technically, the cosine distance over the arrays representing the second-level description of a portfolio is applied; this level of description details the percentage of investment of each individual asset, while the first-level description (exploited in *Step 1*) details only the percentage of the general classes of investment to which the individual assets belong. In this step, we then concentrate our attention on the actual behaviour of the considered investors, comparing their investment strategies asset by asset.

The output of this phase is a ranked list of portfolios, extracted from the most similar users. An optional system parameter can then be set to cut such a list to the J most similar portfolios, if they are more than J . The aim is to provide the next phase (*Step 3*) with a set of interesting assets, extracted from the most similar portfolios of the most similar customers.

3.4 Step 3

Step 3 receives as input either the ranked list of the J most similar portfolios selected at *Step 2*, or the list of the N most similar customers selected at *Step 1*, if *Step 2* was not applicable. In the latter case, every portfolio belonging to the N most similar customers is extracted, and ranked by user similarity; this means that in both cases this phase consider a ranked list of portfolios (i.e., asset allocations) as input. Starting from this list of asset allocations, the system derives the assets to be returned to the user. This is simply done by looking at the individual assets contained in the list of portfolios, by possibly limiting the set of assets to the first K products found by examining the portfolios in the order provided by their ranking.

In order to provide a more informed decision support, each asset is further associated with some statistics; they can help the user to analyze the provided recommendation, by evaluating a broader spectrum of information. These values are summarized below:

1. *Frequency (F)*: it is the frequency of the asset in the set of retrieved portfolios. For example, if the asset is part of 2 retrieved portfolios out of 5 (i.e. the list input to *Step 3* contains 5 portfolios), then $F = 0.4$.
2. *Average Percentage (AP)*: it represents the average percentage of the considered asset with respect to the retrieved portfolios where it appears. For example, if the asset is part of 2 retrieved portfolios and has a 30% allocation in portfolio p_1 and a 50% allocation in portfolio p_2 , then $AP = 40\%$.

3. *Average Distance of Customers (ADC)*: it summarizes the average (global) distance of the retrieved customers who possess the considered asset, using the distance metric described in Eq 5. For example, if the asset is part of the portfolios of 3 customers C_1, C_2, C_3 who are retrieved as similar to the target one in *Step 1*, then the distance between each pair is computed using Eq 5 and then averaged (i.e., $ADP = \frac{D(C_1, C_2) + D(C_2, C_3) + D(C_1, C_3)}{3}$).
4. *Average Distance of Portfolios (ADP)*: it summarizes the average distance of the retrieved portfolios containing the asset (the computation is clearly similar to that of *ADC*). If the target customer does not have any portfolios, this value is not calculated.

In particular, the F statistic is considered particularly useful, since both most frequently and less frequently used assets (among those recommended) are usually interesting for several reasons. In fact, if the user is a private investor, it could be interesting to him/her to consider which are the financial products that are most popular among users similar to him/her; on the other hand, if the user is a professional one (e.g., a consultant), then it could be important to analyze the set of products that are not yet popular among the ones that can be recommended to the customers, since it could be a way of differentiating the offer. Moreover, differently from other recommendation situations, in the SMARTFASI context, it makes sense to consider, in the recommended list, also products already owned by the customer, since this may be food for thoughts. For example, a private investor can receive confirmation from the fact that an asset present in one of his/her portfolios is pretty popular among similar customers, and he/she may decide to increase the percentage of such an asset; or he/she can discover that one of his/her assets is not very popular among similar customer, and to decide to reduce the percentage in the corresponding portfolio. In any case, finding among the recommended financial products some of their assets can trigger interesting analyses from the customer point of view (either if performed directly by the customer in case of a private investor, or if performed by a consultant for the customer's benefit).

Finally, before presenting the user with the list of recommended assets, the system removes those assets which are not compliant with the level of financial knowledge of the target customer; in this way, the system avoids recommending financial products which are not compatible with the customer's MiFID profile. This is done by comparing the risk level of each product with the level of the user's financial adequacy (feature 3 in Table 1).

The final list of products is then presented to the user who can then inspect each asset, by visualizing together with the associated statistics mentioned above, all the basic characteristics of the financial product, as well as its performances, both historical and simulated. In the current version of SMARTFASI, such a list is also ordered by frequency F .

4 System Architecture

In this section we discuss the implementation of the recommendation subsystem of the SMARTFASI project. The general architecture of the recommendation module and its integration/interaction with the other parts of the SMARTFASI software is illustrated in Figure 4. In fact, the SMARTFASI advisory system is a web-based application following a standard 3-tier architecture as follows:

- a *web/mobile browser* providing the client level and user interface,
- an *application server* organized into several submodules
 - a *middleware* receiving requests from the client and dispatching them to the requested service manager

- a *simulation engine*, providing the Monte Carlo simulation service
- a *recommendation module*, providing the recommendation service which is the focus of the present paper
- a client/server *RDBMS*, providing the data tier where information about customers and financial products are stored.

The recommendation module (Recommender subsystem, in Figure 4) is implemented in JAVA as a standard TCP server; even if part of the whole application server of SMARTFASI, the recommender subsystem can in principle be separated from it, resulting in an independent module that can be remotely queried from multiple installations of the SMARTFASI middleware. Indeed, the middleware acts as a client of the recommendation module through a standard client-server interaction and communication.

Concerning a recommendation session, at the browser level, the software interacts with the user whose requests are sent to the middleware; the latter then builds one or more queries, containing both the target customer(s) identification code(s) and all the requested query parameters. These queries are then sent to the recommendation module through a TCP request message. The recommendation module, on the other side, acts as a server, so it is constantly waiting for requests from the middleware. For each submitted query, the server checks its syntax and, in case of positive response, creates a new instance of the recommendation engine, which performs all the steps described in Section 3. Each instance is encapsulated in a new thread, created by the recommender subsystem to handle each query separately. This mechanism creates a robust and responsive server, able to properly act even if one or more instances of the recommendation module unexpectedly fail. It is also able to effectively distribute the workload when many queries must be satisfied simultaneously. Every time an instance terminates its computation, it communicates the query results to the middleware through a TCP answer. If no answer reaches the middleware within a maximum time limit (due to any unexpected error occurred to the relative server instance), the middleware module closes the TCP connection and reports a timeout error.

Two different types of queries can be sent to the recommendation module from the SMARTFASI middleware:

1. a query for a single target customer;
2. a query to manage a collective recommendation for a group of user-selected homogeneous target customers.

For each query, in addition to the customer's code, the user must provide the values for all the parameters necessary for the execution of the query. For this reason, the format of the message of type *Request* is a TCP string consisting of the following fields:

- ⟨01⟩ Internal code for command: *Request*
- ⟨Querycode⟩ Unique code associated with the query, in order to correctly associate each answer with the related request.
- ⟨CustomerID⟩ Multiple lines containing target customer ID
- ⟨NULL⟩ Null string indicating the end of the customers list
- ⟨A/D⟩ The ranking of the assets should be ascending (to consider the most frequently used assets) or descending (in case the user wants to evaluate the less frequently used assets by similar customers)
- ⟨N⟩ Number of similar customers in the ranking of *Step 1* (nullable, since it is optional)
- ⟨J⟩ Number of similar portfolios in the ranking generated by *Step 2* (nullable, since it is optional)

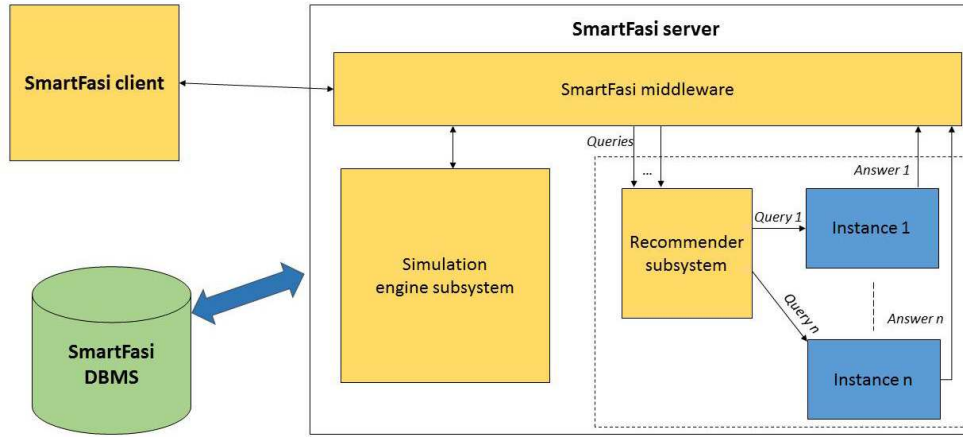


Figure 4. The SMARTFASI system's architecture

- ⟨K⟩ Number of assets to be received in response and to be shown to the user
 ⟨.⟩ End of message

Once the server has received a query, it creates the instance aimed at computing the query result (i.e., the set of recommended financial products). The latter is then packed in a TCP *Answer* message and sent back to the SMARTFASI middleware. The result is a list of assets, each one associated with the corresponding statistics *F*, *AP*, *ADC* and *ADP*. The format of the *Answer* message is composed by the following fields:

- ⟨A1⟩ Internal code for command: *Answer*
 ⟨Querycode⟩ Unique code to correctly associate this answer to the corresponding request
 ⟨Asset; F, AP, ADC, ADP⟩ *K* lines containing the asset code list and their parameters
 ⟨.⟩ End of message

In addition, the message protocol provides answer messages and codes to manage potential server malfunctions and errors (for example, to answer with an error code when a query does not contain a target customer ID).

5 Conclusion and Final Remarks

In the present paper we have described the recommendation module of a smart financial advisory system developed as part of the SMARTFASI project. Following an emerging trend [16, 14, 15], we based the recommendation strategy on Case-Based Reasoning, by defining a suitable notion of similarity among customers and their investment preferences characterized by their portfolios of financial products. The recommended module is complementary to an asset analytical engine, based on Monte Carlo simulation.

Apart from standard recommendation of titles (potentially exploitable by both private as well as professional investors), the proposed methodology can also be exploited by financial companies during the definition of the *Asset Basket* to be proposed to the customers. The standard way of implementing the above process is to cluster customers depending on their (a-priori defined) economic/trading features, and on their adequacy to the financial products; for each cluster the so called *Investment Universe - IU* (the basket of suitable products for the cluster) is then defined and used as

a basis for each proposal (see Fig. 5). This fixed strategy can be im-

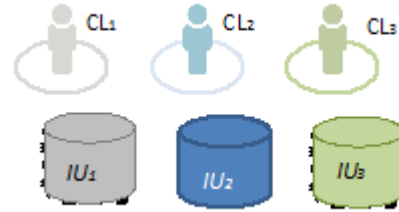


Figure 5. Fixed basket definition and proposal

proved by resorting to similarity-based recommendation as follows: a cluster representative element is identified based on standard features (e.g., A_1, B_1, C_1) and by considering an “average” value for them³. The cluster representative can then play the role of the target user in the SMARTFASI recommendation engine, allowing one to extract the most (or the less) frequently used assets by customers similar to the selected profile. (Fig. 6) In this way, the recommendation engine is exploited to build alternative baskets more tailored to the actual behaviour of the customers in the considered cluster (the so called *Behavioural Investment Universes - BIU*). They may be used to update the asset baskets currently used by the company, as well as to determine the actual effectiveness of such baskets, by considering in the analysis also the appeal of some assets at the cluster level.

More importantly, an historical analysis of such BIUs may discover specific investment trends inside each cluster, by allowing the company to implement better marketing strategies with respect to the given segment of customers. We are planning in the next future to set up an experimental plan to evaluate these kind of strategies.

ACKNOWLEDGEMENTS

The presented work has been conducted in the project SMARTFASI (funded by the ICT Innovation Cluster of the Region of Piedmont, Italy).

³ Alternatively, we can also select several cluster representatives and to use them as a target group of customers (see Section 4.)

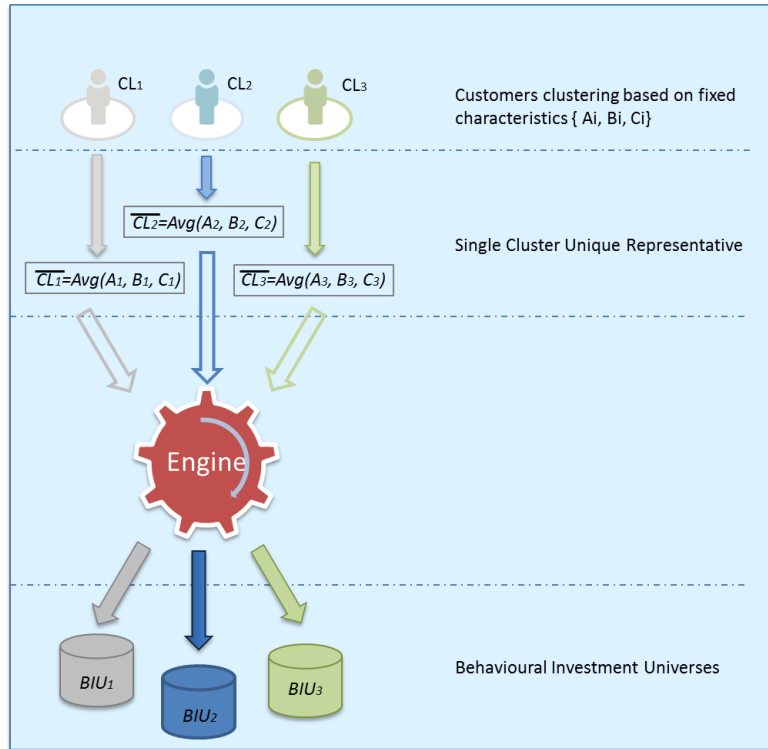


Figure 6. Producing Behavioural Investment Universes

REFERENCES

- [1] Markets In Financial Instruments Directive 2004/39/EC. <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32004L0039:EN:HTML>.
- [2] Agnar Aamodt and Enric Plaza, 'Case-based reasoning: Foundational issues, methodological variations, and system approaches', *AI Communications*, **7**(1), 39–59, (1994).
- [3] *Lazy Learning*, ed., D.W. Aha, Kluwer Academic Publishers, 1997.
- [4] D. Bridge, M.H. Goeker, L. McGinty, and B. Smyth, 'Case-based recommender systems', *Knowledge Engineering Review*, **20**(3), 315–320, (2006).
- [5] D. Brigo and F. Mercurio, *Interest Rate Models - Theory and Practice: With Smile, Inflation and Credit*, Springer Finance, 2006.
- [6] A. Fano and S. Kurth, 'Personal choice point: Helping users visualize what it means to buy a bmw', in *Proceedings of International Conference on Intelligent User Interfaces IUI03*, (2003).
- [7] A.G. Francis and A. Ram, 'Computational models of the utility problem and their application to a utility analysis of case-based reasoning', in *Proceedings of the AAAI Workshop on Knowledge Compilation and Speed-Up Learning*, (1993).
- [8] J. C. Hull, *Options, Futures, and Other Derivatives (6th ed.)*, Upper Saddle River, N.J: Prentice Hall, 2006.
- [9] Y. Koren and R. Bell, 'Advances in collaborative filtering', in *Recommender Systems Handbook*, eds., F. Ricci, L. Rokach, B. Shapira, and P.B. Kantor, 145–186, Springer, (2011).
- [10] Sheng-Tun Li and Hei-Fong Ho, 'Predicting financial activity with evolutionary fuzzy case-based reasoning', *Expert Systems with Applications*, **36**(1), 411 – 422, (2009).
- [11] P. Lops, M. de Gemmis, and G. Semeraro, 'Content-based recommender systems: state of the art and trends', in *Recommender Systems Handbook*, eds., F. Ricci, L. Rokach, B. Shapira, and P.B. Kantor, 73–106, Springer, (2011).
- [12] F. Lorenzi and F. Ricci, 'Case-based recommender systems: a unifying view', in *Intelligent Techniques for Web Personalization*, 89–113, Springer, (2005).
- [13] H. Miyamoto and T. Yonemura, 'New wave of retail asset management business from private banking to sales at bank branches', *NRI Papers*, (129), May 1, 2008.
- [14] C. Musto, G. Semeraro, P. Lops, M. de Gemmis, and G. Lekkas, 'A framework for personalized wealth management exploiting case-based recommender systems', *Intelligenza Artificiale*, **9**(1), 89–103, (2015).
- [15] C. Musto, G. Semeraro, P. Lops, M. de Gemmis, and G. Lekkas, 'Personalized finance advisory through case-based recommender systems and diversification strategies', *Decision Support Systems*, **77**, 100–111, (2015).
- [16] K. J. Oh and T. Y. Kim, 'Financial market monitoring by case-based reasoning', *Expert Systems with Applications*, **32**, 789–800, (2007).
- [17] L. Portinale and P. Torasso, 'Case base maintenance in a multimodal reasoning system', *Computational Intelligence*, **17**(2), 263–279, (2001).
- [18] M.M. Richer and R.O. Weber, *Case-Based Reasoning: a Textbook*, Springer, 2013.
- [19] *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*, eds., G. Shakhnarovich, T. Darrell, , and P. Indyk, MIT Press, 2006.
- [20] *Personalization Techniques and Recommender Systems*, eds., G. Uchyigit and M. Y. Ma, World Scientific Publ., 2008.
- [21] D. Randall Wilson and Tony R. Martinez, 'Improved heterogeneous distance functions', *J. Artif. Int. Res.*, **6**(1), 1–34, (January 1997).
- [22] Pavel Zezula, Giuseppe Amato, Vlastislav Dohnal, and Michal Batko, *Similarity Search: The Metric Space Approach*, volume 32 of *Advances in Database Systems*, Springer, 2006.