Matrix Factorization and Contrast Analysis Techniques for Recommendation

Marharyta Aleksandrova

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Factorisation de matrices et analyse de contraste pour la recommandation

Matrix Factorization and Contrast Analysis Techniques for Recommendation

THÈSE

présentée et soutenue publiquement le 07 juillet 2017

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Marharyta ALEKSANDROVA

Composition du jury

Président :	Elisabeth METAIS	Professeur, CNAM Paris
Rapporteurs :	Stéphane CANU	Professeur, INSA de Rouen
	Patrick GALLINARI	Professeur, Université Pierre et Marie Curie
Examinateurs :	Thomas LARGILLIER	Ingénieur de recherche, Laboratoire privé *ix-labs*
	Anton POPOV	Maître de conférences, NTUU 'Igor Sikorsky KPI'
Directrice :	Anne BOYER	Professeur, Université de Lorraine
Co-directeurs :	Armelle BRUN	Maître de conférences, Université de Lorraine
	Oleg CHERTOV	Professeur, NTUU 'Igor Sikorsky KPI'
Invité :	Yann GUERMEUR	Directeur de recherche, CNRS affecté au LORIA

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thesul.

Acknowledgments

First of all, I would like to express my gratitude to my supervisors: to Anne BOYER for such an accurate guidance in today's boundless scientific field and for so warm welcome in the team; to Armelle BRUN for countless hours spent working with me, many times during the vacations, weekends and even at nights and for helping me in all ways to survive in a foreign country; to Oleg CHERTOV for being my constant source of inspiration in both personal and professional aspects for already 10 years. My dear supervisors thank you for having belief in me some years back and accepting to guide me in the world of research. The years of my PhD program gave me diverse experiences in research and life in general. And of course, I would not have been able to pass all the difficulties and adequately evaluate all the successes on my own. In all senses, this work would have been impossible without you.

The PhD program is the final stage in guided education. And of course, 10 years of school and 6 years of university were indispensable to bring me that far. Thereby I cannot help but be grateful to all my school and university teachers and lecturers who more or less tried to give me some knowledge and in such way shaped my future life. A special thanks to my school teacher of mathematics Ludmyla Stanuslavovna DANYLCHENKO. She is the one who during 7 years was inflaming in me the fire for the love towards sciences. I would also like to express my gratitude to my teachers of English (Tatiana Valentunovna) and French (Svetlana HORLOVA and Alexandre KOUROVSKY). The time that they devoted for me opened boundaries of many countries and, in particular, allowed me to study under the international double supervision program.

My thanks to the members of my PhD committee who despite all obstacles made the defense being possible: to the reviewers Stéphane CANU and Patrick GALLINARI, to the examiners Thomas LARGILLIER and Anton POPOV, and to the president of the committee Elisabeth METAIS. I also express my gratitude to Yann GUERMEUR who apart of being a member of the jury was also my referent and guided me through all the years of my PhD studies.

I want to say many thanks to all ex- and current members of the KIWI research team and LORIA research laboratory. Many of them guided me in my work in different ways, have become my friends, and will remain in my memory forever. Special thanks to Lina FAHED, Hayat NASSER, Laura INFANTE BLANCO, Charif HAYDAR, Ngoc-Chan NGUEN, Florian MARCHAL, Julie BUDAHER, Yacine ABBOUD, Benjamin GRAS, Amaury L'HUILLIER, Pierre-Edouard OSCHE, Geoffray BONNIN, Sylvain CASTAGNOS, Azim ROUSSANALY and many others.

Finally, I want to thank Campus France and the government of Ukraine for providing me with the scholarships, and the University of Lorraine and National Technical University of Ukraine 'Igor Sikorsky Kyiv Polytechnic Institute' for accepting me as a PhD student.

To all teachers in my life,

Contents

List of Figures

List of Tables

Introduction en Français

Cette thèse a été préparée en cotutelle entre l'Université de Lorraine et l'Université Nationale Technique d'Ukraine 'Igor Sikorsky KPI'.

1 Contexte Général

Nous vivons à l'époque de la société de l'information, la société dans laquelle la manipulation de l'information est extrêmement importante dans toutes les sphères de la vie : la politique, l'économique, l'éducation, la culture, etc. [Webster 2014].

Aujourd'hui des données numériques sont devenues la première source d'information et le volume de ces données augmente rapidement. Selon l'étude de la Corporation Internationale de Données (International Data Corporation, IDC), la quantité d'octets produits par l'humanité double tous les 2 ans et vers 2020 elle atteindra 44 zettaoctets ($44 * 10^{21}$ octets) [IDC 2014]. Ces données viennent de sources différentes et sont de natures variées. Nous pouvons par exemple mentionner les données produites par différents capteurs en industrie, des photos/vidéos personnelles et commerciales, des journaux de sites web, des données collectées avec des buts différents (comme des résultats de sondages ou des données de recensement), des messages textes etc. De même, la complexité des données varie grandement. Certaines de ces sources fournissent des données brutes sous la forme de signaux physiques (des données produites par des capteurs), les autres sources fournissent une information codée (des résultats de sondages), d'autres encore peuvent donner accès à des données sémantiquement riches mais difficiles à analyser (comme des messages texte).

Les données, le nouvel or noir...

Le volume de données ne cesse d'augmenter, de même que leur accessibilité par chercheurs du grand public. Par exemple, les initiatives gouvernementales dites *open data* [Ubaldi 2013] lancées par de nombreux pays permettent désormais à quiconque d'accéder aux bases de données gouvernementales.

L'utilisation de données devient maintenant une condition nécessaire de compétitivité dans le marché actuel. Beaucoup d'entreprises telles que Lufthansa [Lufthansa 2016] confirment que l'utilisation des données est indispensable pour leurs affaires. Toutefois, les données peuvent également être utiles pour des moyennes ou petites entreprises [Hazel 2015].

Toutes ces tendances et ce potentiel font que les données sont désormais vues comme le nouvel or noir. Cependant, à la différence d'autres équivalents d'or noir (comme le café ou le pétrole), les données ont la caractéristique de pouvoir être utilisées plusieurs fois et pour résoudre des questions différentes. De plus, la valeur de données ne diminue pas avec le temps,

mais augmente avec l'apparition de nouvelles données. Mais, comme les autres équivalents d'or noir, les données brutes ne sont pas très utiles. Ce qui est utile, c'est l'information extraite de ces données [Singh 2013].

De données aux connaissances...

L'analyse des données est un processus de transformation des données brutes ou partiellement traitées sous la forme d'information [Judd *et al.* 2011]. Non seulement les techniques d'analyse se développent sans cesse, mais les concepts de l'analyse des données évoluent également. Par exemple, selon Gartner [Davis and Herschel 2016], nous pouvons définir 4 types d'analyse des données : l'analyse descriptive, l'analyse diagnostique, l'analyse prédictive et l'analyse prescriptive. L'analyse descriptive vise à décrire les tendances générales dans le jeu de données. L'analyse diagnostique tente d'expliquer la nature des motifs trouvés, et l'analyse prédictive a pour but prédire les évènements futurs. Enfin, l'analyse prescriptive vise à identifier les facteurs qui peuvent mener le développement d'un système dans une direction souhaitée.

Ainsi, nous pouvons remarquer la disponibilité d'outils et de données, le potentiel des données à être une source riche d'information, et la demande de l'information par la société d'aujourd'hui. Toutes ces tendances font de l'analyse des données un des plus importants domaines de recherche actuels.

2 Problématiques

2.1 Problématiques Scientifiques

Dans de nombreux domaines, les données peuvent être de grande dimension (c'est-à-dire avoir de nombreuses caractéristiques). Certaines de ces caractéristiques peuvent être fortement corrélées avec d'autres ou être redondantes. Etant donné que la grande dimensionnalité des données peut restreindre la performance des méthodes de traitement des données [Bingham and Mannila 2001], le problème de la réduction de dimension se pose naturellement.

La réduction de dimension est le processus de réduction le nombre de dimensions (caractéristiques) par la sélection de caractéristiques ou la création des caractéristiques capables à présenter les données le mieux possible [Roweis and Saul 2000]. Selon [Pudil and Novovičová 1998], les techniques de réduction de dimension peuvent être classées en fonction de leur *but* en

- techniques pour la représentation optimale,
- techniques pour la classification;

et/ou en fonction de leur *stratégie* en

- la sélection des caractéristiques,
- l'extraction des caractéristiques (ou la construction des caractéristiques [Guyon and Elisseeff 2003]).

Considérons tout d'abord la stratégie de méthodes de réduction de dimension puis leur but. Les méthodes de sélection de caractéristique identifient un sous-ensemble des caractéristiques originales suffisant pour résoudre une tâche considérée [Guyon and Elisseeff 2003]. Au contraire, les méthodes d'extraction créent un ensemble de nouvelles caractéristiques (dites latentes [Momma and Bennett 2006]) qui ne font pas forcement partie de l'ensemble des caractéristiques

d'origine [Liu and Motoda 1998]. Bien que l'ensemble des caractéristiques résultant des méthodes d'extraction a une puissance descriptive et discriminative élevée, il est généralement non interprétable [Pudil and Novovičová 1998]. Cela peut être un obstacle dans les domaines où la compréhension d'un modèle est importante. Plusieurs travaux de recherche sont consacrés à l'interprétation des modèles avec les caractéristiques latentes [Bylesjö *et al.* 2008, Cruz-Roa *et al.* 2012, Kvalheim and Karstang 1989]. Toutefois, les solutions proposées varient selon la technique d'extraction des caractéristiques adoptée et selon le domaine d'application. Par conséquent, la première problématique scientifique de la thèse est **PS1: comment extraire des caractéristiques latentes interprétables?**

Maintenant nous poursuivons avec l'étude de techniques de réduction de dimension en fonction de leur but. Alors que la réduction de dimension pour la représentation cherche à préserver la structure topologique des données, la réduction de dimension pour la classification vise à améliorer la puissance discriminatoire (ou la puissance de classification) du sous-ensemble sélectionné [Pudil and Novovičová 1998].

Les motifs de classification, c'est-à-dire les motifs obtenus par des algorithmes de classification, peuvent être directement utilisés pour effectuer l'analyse descriptive, diagnostique ou bien prédictive. Cependant, suivons la tendance générale d'évolution de concepts d'analyse des données. Le quatrième type d'analyse des données est l'analyse prescriptive. Dans ce cadre, la tâche de classification peut être considérée comme la tâche d'identification des facteurs déclencheurs, c'est-à-dire des facteurs qui peuvent influencer le transfert d'éléments de données d'une classe à l'autre. Dans la littérature nous pouvons trouver des tentatives d'identification des facteurs capables d'influencer la direction de développement d'un système [Baker *et al.* 2001, Choongo *et al.* 2016]. Cependant, toutes ces approches sont basées sur l'analyse humaine et, à notre connaissance, il n'existe aucune technique d'identification automatique de facteurs déclencheurs dans le cas général. Ainsi, la deuxième problématique scientifique de cette thèse est **PS2: comment identifier automatiquement les facteurs déclencheurs?**

2.2 Domaine d'Applications : Systèmes de Recommandations

Nous visons à résoudre les deux problématiques scientifiques **PS1** et **PS2** dans le domaine d'application des systèmes de recommandation. Notre choix est dicté par deux faits. Premièrement, les systèmes de recommandation sont extensivement utilisées dans des applications diverses (e-commerce [Huang 2011], tourisme [Zanker *et al.* 2008], e-learning [Verbert *et al.* 2012, Klašnja-Milićević *et al.* 2015]). Deuxièmement, le processus de construction de systèmes de recommandation fait face aux deux problématiques scientifiques identifiées ci-dessus.

Les systèmes de recommandation visent à aider un utilisateur à choisir un produit qui correspond à ses besoins. Le filtrage collaboratif [Schafer *et al.* 2007] est une technique très connue et extensivement utilisée. Elle s'appuie sur les préférences des utilisateurs, généralement présentées sous la forme des notes attribuées aux produits par des utilisateurs. Deux approches principales sont utilisées dans filtrage collaboratif : l'approche basée sur le voisinage et l'approche basée sur la factorisation de matrices [Koren 2008].

L'approche basée sur la factorisation de matrices a devenue récemment plus populaire que l'approche basée sur le voisinage [Adomavicius and Tuzhilin 2005], puisqu'elle fonctionne bien avec les données creuses et les données de grande échelle [Takacs *et al.* 2009]. Aussi cette approche permet de construire des modèles qui sont à la fois fidèles et peu complexes [Koren *et al.* 2009]. Cette approche est basée sur l'idée que les préférences des utilisateurs sur les produits peuvent être expliquées par un nombre réduit de facteurs latents [Koren *et al.* 2009]. Compte

tenu que l'approche basée sur la factorisation de matrices est une approche d'extraction des caractéristiques [Guyon and Elisseeff 2003], les facteurs latents du modèle sont construits dans façon à ce qu'ils prédisent au mieux les notes connues. Par conséquent, les facteurs latents n'ont pas de sens « physique » et l'interprétation de ces facteurs devient une tâche difficile. Ainsi, l'un des inconvénients de cette approche est la difficulté à expliquer les recommandations fournies (car les éléments du modèle ne sont pas interprétables).

Ainsi, la première problématique applicative de cette thèse est **PA1: proposer une interprétation automatique de facteurs latents pour les systèmes de recommandation basés sur la factorisation de matrices.**

Les produits recommandés par un système de recommandation peuvent être de nature très variée. Nous pouvons mentionner la recommandation traditionnelle de films [Golbeck *et al.* 2006], de musique [Koenigstein *et al.* 2011], de recettes [Berkovsky and Freyne 2010] et même la recommandation d'activités physiques [Shibata *et al.* 2009]. Dans le cadre de l'analyse prescriptive un système de recommandation peut être utilisé pour former des recommandations sur la façon de stimuler le développement d'un système dans la direction souhaitée. Par exemple, au lieu de prédire la possibilité d'un achat nous pouvons essayer d'identifier les facteurs qui peuvent stimuler l'achat d'un produit particulier (les facteurs qui peuvent déclencher un achat).

Comme indiqué plus haut, nous ne connaissons aucune technique conçue pour l'identification *automatique* des facteurs déclencheurs. Ainsi la deuxième problématique applicative de cette thèse est **PA2: proposer une technique pour l'identification automatique des facteurs déclencheurs et pour la génération des recommandations sur la façon d'atteindre les objectifs souhaités**.

3 Contributions

La **première contribution** de cette thèse (qui correspond à **PA1** et **PS1**), est la proposition d'une interprétation de facteurs latents de systèmes de recommandation basés sur la factorisation de matrices. Nous associons les facteurs latents à des éléments réels du système : des utilisateurs (désignés sous le terme 'utilisateurs représentatifs'). Cette association complète le modèle et les recommandations fournies avec l'interprétation. Au contraire de la plupart d'autres approches conçues pour dériver le sens des facteurs latents dans les modèles basés sur factorisation de matrices [Zhang *et al.* 2006, McAuley and Leskovec 2013], la méthode proposée ici ne demande ni une analyse humaine ni des sources d'information externes. L'interprétation proposée donne aussi une solution élégante de problème de démarrage à froid pour des nouveaux produits [Bobadilla *et al.* 2013].

Cette contribution a pour résultats deux publications en revues (internationale et nationale) et trois publications en conférences internationales :

- Revue internationale : 'Identifying representative users in matrix factorization-based recommender systems: application to solving the content-less new item cold-start problem', Marharyta Aleksandrova, Armelle Brun, Anne Boyer, Oleg Chertov, In *Journal of Intelligent Information Systems*, 2016 [Aleksandrova *et al.* 2016a].

- Revue nationale (Ukraine) : 'Comparative analysis of neighbourhood-based approach and matrix factorization in recommender systems', Oleg Chertov, Armelle Brun, Anne Boyer, Marharyta Aleksandrova, In *Eastern-European Journal of Enterprise Technologies*, 2015 [Chertov *et al.* 2015].

- Article de conférence : 'Can latent features be interpreted as users in matrix factorization-based recommender systems?', Armelle Brun, Marharyta Aleksandrova, Anne Boyer, In *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, Warsaw, Poland, 2014 [Brun *et al.* 2014].

- Article courte d'atelier : 'What about interpreting features in matrix factorization-based recommender systems as users?', Marharyta Aleksandrova, Armelle Brun, Anne Boyer, Oleg Chertov, In *ACM Conference on Hypertext and Social Media (HT)*, International Workshop on Social Personalisation (SP 2014), Santiago, Chile, 2014 [Aleksandrova *et al.* 2014c].

- Article de session doctorale : 'Search for user-related features in matrix factorization-based recommender systems', Marharyta Aleksandrova, Armelle Brun, Anne Boyer, Oleg Chertov, In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2014)*, PhD Session Proceedings, 2014 [Aleksandrova *et al.* 2014b].

La **deuxième contribution** de cette thèse (qui correspond à **PA2** et **PS2**) consiste en l'introduction d'un nouveau type de motifs désigné sous le terme 'ensemble de règles de contraste'. Ce pattern s'appuie sur des règles d'association. Le pattern proposé est conçu pour l'identification *automatique* de facteurs déclencheurs. A traves des expérimentations, nous montrons que le pattern proposé peut en réalité identifier les facteurs déclencheurs. Il peut aussi être utilisé pour concevoir des recommandations avec pour but *d'influencer la direction de développement d'un système*. Nous prouvons aussi que notre pattern appartient au domaine de l'induction supervisée des règles descriptives [Petra 2009]. Il peut être ainsi considéré comme une solution pour le problème de redondance des règles [Zaki 2000]. En plus, nous supposons que l'application du motif proposé pour l'analyse des données démographiques permet d'identifier les éléments de *habitus* [Hillier and Rooksby 2005] et les opportunités dans la théorie de la découvertes des opportunités [Ohsawa 2006].

Les résultats obtenus dans le cadre de cette contribution ont étés publiés en revue nationale et publiés dans des conférences internationales et nationales :

- Revue nationale (Ukraine) : 'Two-step recommendations: contrast analysis and matrix factorization techniques', Marharyta Aleksandrova, Armelle Brun, Anne Boyer, Oleg Chertov, In *Mathematical machines and systems*, 2014 [Aleksandrova *et al.* 2014a].

- Article courte de conférence : 'Sets of contrasting rules to identify trigger factors', Marharyta Aleksandrova, Armelle Brun, Oleg Chertov, Anne Boyer, In *22nd European Conference on Artificial Intelligence (ECAI-2016)*, short paper, 2016 [Aleksandrova *et al.* 2016d].

- Article courte de conférence: 'Sets of contrasting rules: a supervised descriptive rule induction pattern for identification of trigger factors', Marharyta Aleksandrova, Armelle Brun, Oleg Chertov, Anne Boyer, In *28-th IEEE International Conference on Tools with Artificial Intelligence (ICTAI-2016)*, short paper, 2016 [Aleksandrova *et al.* 2016c].

- Article d'telier : 'Automatic identification of trigger factors: a possibility for chance discovery', Marharyta Aleksandrova, Armelle Brun, Oleg Chertov, Anne Boyer, In *22nd European Conference on Artificial Intelligence (ECAI-2016)*, 2nd European Workshop on Chance Discovery and Data Synthesis, 2016 [Aleksandrova *et al.* 2016b].

- Article de conférence nationale : 'Data mining for habitus elements identification', Oleg Chertov, Marharyta Aleksandrova, In *International scientific conference 'State and global social changes: historical sociology of planning and resistance in modern era'*, Kyiv, Ukraine, 2015 [Chertov and Aleksandrova 2015].

Introduction in English

This thesis was prepared under the double-supervision program of the University of Lorraine and the National Technical University of Ukraine 'Igor Sikorsky Kyiv Polytechnic Institute'.

4 General Context

We live in the era of the information society, the society where the manipulation of information is extremely important in all spheres of life: politics, economics, education, culture etc. [Webster 2014].

Today the digital data has become a primary source of information and its amount grows rapidly. According to the research of the International Data Corporation (IDC), the number of digital bytes produced by humanity doubles every 2 years and by 2020 it will reach 44 zettabytes ($44 * 10^{21}$ bytes) [IDC 2014]. This data comes from various sources and is diverse in its nature. We can mention data from different sensors in industry, commercial and personal photos/videos, site logs, specially collected data (like poll results or census data), text messages, etc. The complexity of the data also varies significantly. Some of these sources provide 'raw' data in a form of physical signals (like those coming from sensors), other sources furnish coded information (like poll results), while yet another can provide rich in information but difficult to automatically analyse data with semantic meaning (like text messages).

Data new black gold...

Along with the increase in the amount of digital data, it has become more available for public research. For example, so-called *open data* government initiatives [Ubaldi 2013] launched in many countries provide access to governmental data (see for instance *opendatafrance.net* or *data.gov.ua* which, as for September 2016, provide free access to 9,500 and 6,756 government datasets of France and Ukraine respectively).

Exploiting data becomes a necessary condition for competitiveness in modern business. Many huge well-known firms and corporations such as Lufthansa [Lufthansa 2016] confirm that the usage of data is important for their business. However, not only big firms but also medium and small ones can make use of it [Hazel 2015]. Digital data also invades our everyday life. Modern smart household appliances not only can analyse the data they collect but also exchange it through the internet and optimise their operation. This novel phenomenon is known as the Internet of Things (IoT) [Xia *et al.* 2012] or Industrial Internet [Daugherty *et al.* 2014].

All these tendencies make data being the new 'black gold' which, contrary to other black gold equivalents (like coffee or petrol) can be reused multiple times and for various purposes [Singh 2013]. Moreover, the value of many datasets is not exhausted with time but only grows when

new data is added, which makes it being comparable with such renewable resources as wind or solar energy. The importance of data is so significant, that we are starting to face a 'data-driven' economy, where 'data ... is playing an increasingly pivotal role in the creation and evolution of innovative new services' [Facebook and CtrlShift 2016]. We can conclude that digital data creates a whole new world with corresponding interaction rules. However, like other black gold equivalents raw data is not very useful, but what is useful is information that is extracted from it [Singh 2013].

From data to knowledge...

Data analysis is the process of converting raw or partially processed data into useful information [Judd *et al.* 2011]. This process can include many steps like cleaning, filtering, transforming, and modelling [Schutt and O'Neil 2013]. Even before the computers were invented, people gathered and analysed the data. However, with modern computing machinery, qualitatively new results can be obtained [Kelle and Bird 1995]. For example, now it is possible not only to calculate some statistical characteristics of a dataset (like mean value), but with the help of data mining techniques, a search for hidden patterns in the data can be done [Han *et al.* 2011]. Nowadays, neural networks algorithms attempt to simulate the processes taking place in the human brain while analysing the data [Hagan *et al.* 1996]. We can also note the development of techniques specialised for the analysis of so large and complex datasets, where traditional data processing becomes inadequate [Oracle and FSN 2012], referred to as big data techniques [Gandomi and Haider 2015].

Not only the analysis techniques are being constantly developed, but also the concepts of data analysis are evolving. For instance, according to Gartner [Davis and Herschel 2016], we can define 4 type in data analytics: descriptive analytics, diagnostic analytics, predictive analytics and prescriptive analytics. The descriptive analytics aims to describe the general tendencies in the dataset. The diagnostic analytics tries to understand the nature of the found patterns and dependencies, and the predictive analytics aims to predict what can happen in the future. Finally, the prescriptive analytics identifies the factors that can actually lead the development of the system in the desired direction.

Thereby, we can see that the availability of technical tools and that of data, as well as its natural potential in hidden information, and the existence of the high demand for information from the today's information society, make data analysis being one of the most important spheres of research.

5 Problematics

5.1 Scientific Problematics

In many application areas, data elements can be high-dimensional. That is, they can have a large number of characteristics or features. For example, when using the *bag-of-words* model [McCallum *et al.* 1998, Zhang *et al.* 2010] for text documents analysis, the number of dimensions is equal to the size of the dictionary [Segal and Kephart 2000], that is equal to the total possible number of words in the language. For census [U.S.CensusBureau 2000] and DNA [Avogadri and Valentini 2009] microarray files the number of dimensions can be made up of hundreds and thousands of elements respectively. Some of these features can be highly correlated with the others or just redundant. Considering the fact that high dimensionality of a dataset can

restrict the performance of data processing methods [Bingham and Mannila 2001], the problem of dimensionality reduction arises.

Dimensionality reduction is the process of reducing the number of dimensions under consideration by selecting or creating features that can represent the data in the best possible way [Roweis and Saul 2000]. According to [Pudil and Novovičová 1998], based on their *aim*, the dimensionality reduction techniques can be divided into

- dimensionality reduction for optimal data representation,
- dimensionality reduction for classification;

based on the adopted *strategy* into

- feature selection,
- feature extraction (or feature construction [Guyon and Elisseeff 2003]).

Let us follow the bottom-up analysis and move from adopted strategy of dimensionality reduction to its aim. Feature selection methods are those that create a subset of original features that is sufficient to solve a given task [Guyon and Elisseeff 2003]. Contrarily, the feature extraction methods create a set of new features (often referred to as *latent features*, see for example [Momma and Bennett 2006]) which may not be a part of the set of original features [Liu and Motoda 1998]. Though the set of features resulting from feature extraction methods has higher descriptive and discriminative power, it can become uninterpretable [Pudil and Novovičová 1998]. This can be an obstacle in certain domains, where the understanding of a model is important. For example, when trying to understand the cause of a certain disease (in medicine [Sundgot-Borgen 1994]) or the market behaviour (in finance [Baker *et al.* 2001]). Many research works deal with the interpretation of latent features models [Bylesjö *et al.* 2008, Cruz-Roa *et al.* 2012, Kvalheim and Karstang 1989]. However, the proposed solutions vary based on the used feature extraction method and the application problem. Thereby, the first scientific problematic of this thesis is **SP1: how to extract interpretable latent features?**

Now we proceed to have a look at the dimensionality reduction methods from the perspective of their aim. While the dimensionality reduction for representation seeks to preserve the topological structure of data in a lower-dimensional subspace, the dimensionality reduction for classification aims to enhance the discriminatory (or classification) power of the selected subset [Pudil and Novovičová 1998]. Classification is used in numerous applications like optical character recognition [Impedovo *et al.* 1991], natural language processing [Manning and Schütze 1999], medical image analysis [Chen *et al.* 1989] and corresponds to the natural ability of a human brain to divide objects into a set of classes and then referring to a class while interpreting a previously unseen object. The task of the classification is defined as a task of constructing an algorithm capable of identifying the class of a new data element [Alpaydin 2014].

Classification patterns, those resulting from classification algorithms, can be naturally used for the purposes of descriptive, diagnostic and predictive analytics (to understand the structure of the datasets, characteristics of its elements as well as to predict the class of the elements). There also exist a number of techniques designed for the identification of such patterns. Among them, we can mention *contrast analysis* that aims to discover those patterns, that can highlight the differences between classes of data [Bay and Pazzani 2001]. However, let us follow the general tendency in the evolution of data analysis concepts and move to its fourth type: prescriptive analytics. Within the scope of prescriptive analytics, the task of classification can be viewed as

the task of forcing the transfer of elements from one class to another with the aim to stimulate the system development direction. There are attempts in the literature to identify factors that can affect the system development [Baker *et al.* 2001, Choongo *et al.* 2016] (they will be referred to as *trigger factors*). However, they are based on human analysis and we are not aware of any techniques that are designed for the identification of such trigger factors. Thus, the second scientific problematic of this thesis is **SP2: how to identify automatically factors that can cause the movement of elements from one class of the dataset to another (trigger factors)?**

5.2 Application Domain and Problematics: Recommender systems

We aim to investigate and solve the scientific questions put in the previous subsection within the recommender systems application domain, as these systems are extensively used in diverse applications (e-commerce [Huang 2011], tourism [Zanker *et al.* 2008], e-learning [Verbert *et al.* 2012, Klašnja-Milićević *et al.* 2015]) and the process of their construction faces both **SP1** and **SP2**.

Recommender systems (RS) aim to assist users in their selection or purchase of items by suggesting the items that fit their needs. Collaborative filtering (CF) [Schafer *et al.* 2007] is a very popular and widely used recommendation technique, which relies on users' preferences, generally the ratings they assign to items. This information is usually presented in a form of a rating matrix with rows and columns corresponding to users and items, and values corresponding to actual ratings. One of the tasks of a recommender system is thus to predict the values of unknown ratings of the previously unseen items by a user and recommend those, that have the highest predicted ratings and, thereby, are predicted to be more interesting. There are two major approaches in CF: neighbourhood-based (NB) and matrix factorization (MF) [Koren 2008]. The NB approach [Desrosiers and Karypis 2011a] identifies for each user, his/her similar-minded users (neighbours), using the rating matrix. It estimates the missing ratings of this user by exploiting the ratings of his/her neighbours. NB is quite popular due to its simplicity, efficiency, accuracy and its ability to explain the provided recommendations (through the users' neighbours) [Koren 2008]. However, NB has limitations on large and/or sparse datasets and it is time-consuming.

The MF approach [Koren *et al.* 2009] relies on the idea that the ratings in the rating matrix can be explained by a small number of latent features (also referred to as factors). It factorizes the rating matrix into two low-rank matrices, which represent the relation between the users and items with the latent features. The MF approach has recently attracted more attention than the traditional neighbourhood-based approach [Adomavicius and Tuzhilin 2005], as it is adequate for large-scale and sparse datasets [Takacs *et al.* 2009] and it has proven to form highly accurate models of low-complexity [Koren *et al.* 2009]. As MF is a feature extraction method [Guyon and Elisseeff 2003], the resulting latent features are formed in such a way, that the model fits the best known ratings. As a consequent, the latent features have no underlying physical meaning and the interpretation of these features is not an easy and obvious task. Thus, one of the main shortcomings of MF is the difficulty to explain the recommendations provided (as elements of the model have no real interpretation). At the same time, several studies [Herlocker *et al.* 2000, Sinha and Swearingen 2002, Ortega *et al.* 2014] show that explanations enhance the user satisfaction and increase user trust (fidelity) in the system. Users feel more comfortable when they understand why a certain item is recommended to them. We can outline some works dedicated to the interpretation of MF-based recommendation models, however, most of them either propose to perform the interpretation manually [Zhang *et al.* 2006] or to align it with other interpretable models [McAuley and Leskovec 2013].

Thereby, the first application problematic of this thesis is **AP1: propose an automatic interpretation of latent features within the matrix factorization-based recommendation models (provide the model explanation without requiring external information) and explore if the resulting interpretation can be used to improve the recommender system performance.**

On a more abstract level when the set of possible items to recommend is predefined the task of recommendation can be viewed as multi-criteria decision making [Adomavicius *et al.* 2011]. The recommended items can also vary significantly in their nature, starting from the traditional recommendation of films [Golbeck *et al.* 2006], music [Koenigstein *et al.* 2011], recipes [Berkovsky and Freyne 2010] to the recommendation of physical activities [Shibata *et al.* 2009]. Within the frame of prescriptive analytics, where the task is to understand how it is possible to make something happen, the recommender systems can be asked to give recommendations on how to stimulate the development of a system in the desired direction. For example, instead of predicting the possibility of a purchase we can try to identify those factors, that can actually stimulate the purchase of a certain item. Another example can be giving recommendations on how to encourage students to finish the started online course or stimulate the birth-rate increase, that is to identify trigger factors within the current application task. In this case, there are alternative states of the elements of the system (an item bought / not bought, a course finished / not finished, a child is born / not born) and it is required to identify factors that can cause the elements to change their state. In our opinion, this can be done through the analysis of the differences between the elements belonging to each of the alternative classes, that is, through the concept of contrast analysis. The idea of using contrast analysis techniques in the frame of recommender systems domain was proposed in [Duan 2014], however with the goal to find differences between users, but not to search for trigger factors. As it was mentioned above, we are not aware of any techniques designed for the *automatic* identification of trigger factors, thus the second application problematic of this thesis is **AP2: propose a technique that can automatically identify trigger factors and generate based on them recommendations to achieve the desired objective.**

6 Contributions

As a **first contribution** of this work (that corresponds to **AP1** and **SP1**), we propose an interpretation of the latent features in MF-based recommender systems. We associate latent features with real elements of the system: with users (referred to as representative users). This association makes the model and the provided recommendations being interpretable as the resulting recommendations are now generated not through abstract features without physical meaning, but through real elements of the system. Unlike most of the other approaches designed for deriving the meaning of latent features in MF-based models [Zhang *et al.* 2006, McAuley and Leskovec 2013], the proposed method neither requires human analysis, nor the external sources of information. Also, the model becomes somewhat similar to the NB models. Indeed, if the features are associated with users, then the preferences of a user are computed through the preferences of representative users. At the same time in the NB approach, the preferences are computed through the neighbours.

The proposed interpretation also results in an elegant solution for the new item cold-start problem, when a previously unseen item enters the system [Bobadilla *et al.* 2013]. Indeed, the MF approach assumes that the relations between features and items can be expressed through the set of latent features. If these features are associated with some real users of the system, then the

relations between users and items can be expressed through a set of these representative users. Thereby, the ratings provided by representative users on new items can be used to estimate the preferences of other users on new items. The proposed solution for the new item cold-start problem also does not require any content information concerning the recommended items, contrary to many state-of-the-art approaches.

This contribution resulted in two journal publications (international and national) and three publications presented at international conferences (regular paper, short workshop paper, and PhD session paper):

- International journal: 'Identifying representative users in matrix factorization-based recommender systems: application to solving the content-less new item cold-start problem', Marharyta Aleksandrova, Armelle Brun, Anne Boyer, Oleg Chertov, In *Journal of Intelligent Information Systems*, 2016 [Aleksandrova *et al.* 2016a].
- National journal (Ukraine): 'Comparative analysis of neighbourhood-based approach and matrix factorization in recommender systems', Oleg Chertov, Armelle Brun, Anne Boyer, Marharyta Aleksandrova, In *Eastern-European Journal of Enterprise Technologies*, 2015 [Chertov *et al.* 2015].
- Regular conference paper: 'Can latent features be interpreted as users in matrix factorization-based recommender systems?', Armelle Brun, Marharyta Aleksandrova, Anne Boyer, In *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, Warsaw, Poland, 2014 [Brun *et al.* 2014].
- Short workshop conference paper: 'What about interpreting features in matrix factorization-based recommender systems as users?', Marharyta Aleksandrova, Armelle Brun, Anne Boyer, Oleg Chertov, In *ACM Conference on Hypertext and Social Media (HT)*, International Workshop on Social Personalisation (SP 2014), Santiago, Chile, 2014 [Aleksandrova *et al.* 2014c].
- PhD session conference paper: 'Search for user-related features in matrix factorization-based recommender systems', Marharyta Aleksandrova, Armelle Brun, Anne Boyer, Oleg Chertov, In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2014)*, PhD Session Proceedings, 2014 [Aleksandrova *et al.* 2014b].

The **second contribution** of this thesis (that corresponds to **AP2** and **SP2**) lies in the introduction of a new type of pattern, referred to as 'set of contrasting rules' (SCR pattern), which is based on association rules. One of the original aspects of the SCR pattern is that it is made up of a set of rules contrary to the state-of-the-art patterns like contrast sets [Bay and Pazzani 1999] or emerging patterns [Dong and Li 1999] that are made up of only one element (such as one rule). The proposed pattern is designed for the *automatic* identification of trigger factors (the factors that can stimulate the system state changes) through the introduction of notions of varying and invariant attributes. To the best of our knowledge, no other techniques in the literature allow to identify automatically such trigger factors, and the manual analysis performed in some works, for example, [Hougaard *et al.* 2013], cannot be efficiently used for solving real-life tasks.

Through the experiments, we show that the proposed pattern can actually identify trigger factors and can be used to form recommendations on how to affect the development of a system.

We also show that the SCR pattern falls within the supervised descriptive rules induction framework [Petra 2009] and that it can be a solution to the problem of rules redundancy [Zaki 2000]. In addition, we assume that when the proposed pattern is used for the analysis of demographic data it can identify the elements of habitus [Hillier and Rooksby 2005] and chances in chance discovery theory [Ohsawa 2006].

The results obtained in the scope of this contribution were published in a national journal, presented at international conferences (as two short papers and a workshop paper) and presented as a thesis in a national conference:

- National journal (Ukraine): 'Two-step recommendations: contrast analysis and matrix factorization techniques', Marharyta Aleksandrova, Armelle Brun, Anne Boyer, Oleg Chertov, In *Mathematical machines and systems*, 2014 [Aleksandrova *et al.* 2014a].

- Short conference paper: 'Sets of contrasting rules to identify trigger factors', Marharyta Aleksandrova, Armelle Brun, Oleg Chertov, Anne Boyer, In *22nd European Conference on Artificial Intelligence (ECAI-2016)*, short paper, 2016 [Aleksandrova *et al.* 2016d].

- Short conference paper: 'Sets of contrasting rules: a supervised descriptive rule induction pattern for identification of trigger factors', Marharyta Aleksandrova, Armelle Brun, Oleg Chertov, Anne Boyer, In *28-th IEEE International Conference on Tools with Artificial Intelligence (ICTAI-2016)*, short paper, 2016 [Aleksandrova *et al.* 2016c].

- Workshop conference paper: 'Automatic identification of trigger factors: a possibility for chance discovery', Marharyta Aleksandrova, Armelle Brun, Oleg Chertov, Anne Boyer, In *22nd European Conference on Artificial Intelligence (ECAI-2016)*, 2nd European Workshop on Chance Discovery and Data Synthesis, 2016 [Aleksandrova *et al.* 2016b].

- National conference paper: 'Data mining for habitus elements identification', Oleg Chertov, Marharyta Aleksandrova, In *International scientific conference 'State and global social changes: historical sociology of planning and resistance in modern era'*, Kyiv, Ukraine, 2015 [Chertov and Aleksandrova 2015].

Chapter 1

Overview of Recommender Systems

Contents

As it was mentioned in the introduction, we choose recommender systems as an application field for solving the scientific questions $SP1$ and $SP2$. This choice is based on two reasons: 1) the field of recommender systems is very popular both in industry and in academia as these systems were designed with the aim to overcome information overload, which is one of the main problems of information society; and 2) within this field we can define application problematics $AP1$ and $AP2$, which correspond to $SP1$ and $SP2$.

The goal of this chapter is to provide a general overview of recommender systems as a research field and to point out the positions of application problematics $AP1$ and $AP2$. The research trends specific for each application problematic will be discussed in the following chapters.

1.1 Current State-of-the-Art

1.1.1 Origins of Recommender Systems

As it was discussed in the introduction, we live a society where information becomes one of the key driving forces of its development. It was also discussed that the amount of data and the amount of information that can be extracted from it grows extremely fast. As a result, it becomes more and more difficult to navigate in such a vast amount of information and to choose something relevant or useful. Thus the problem of overcoming the information overload has arisen, which led to the appearance of the recommender systems (RS) research area in the mid-1990 [Park *et al.* 2012, Ricci *et al.* 2011, Adomavicius and Tuzhilin 2005].

The way we choose something in real life depends on many factors: on some predefined demographic behavioural patterns (for example, it is considered that women usually prefer romantic

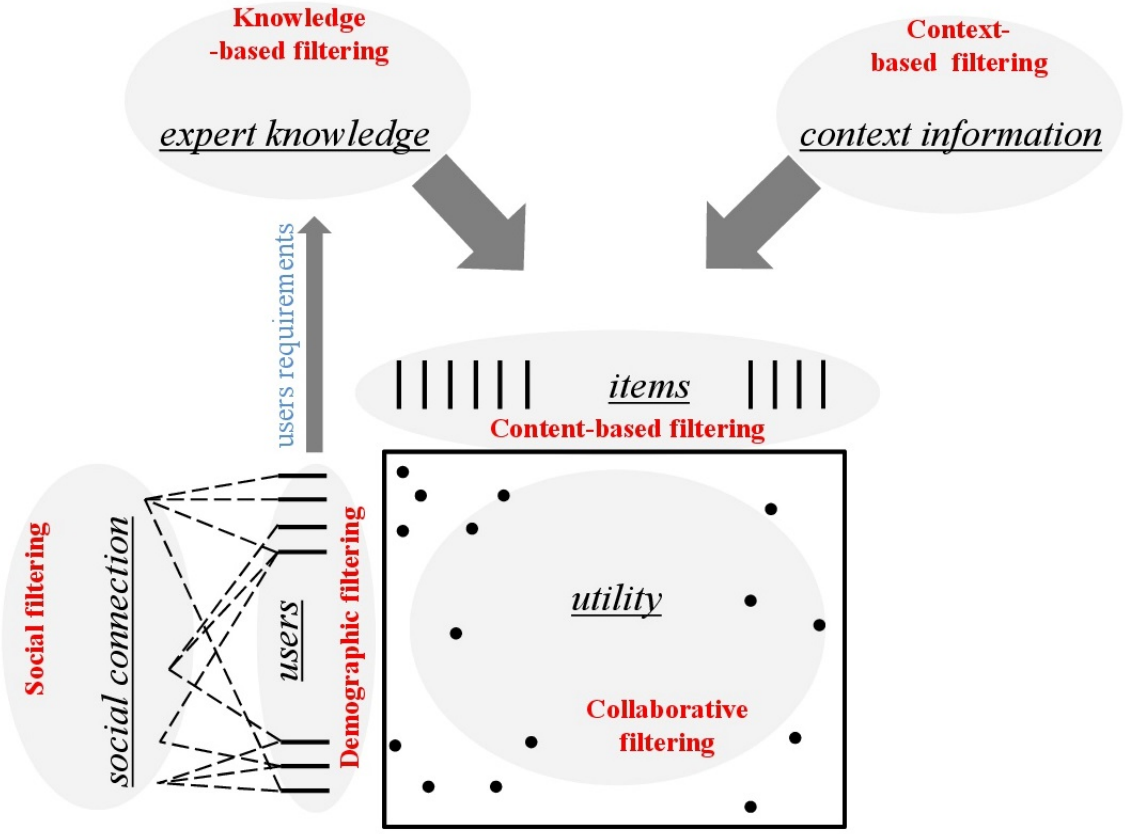

Figure 1.1: Filtering Techniques of Recommender Systems

films and men tend to choose action movies [Fischoff *et al.* 1998]); the characteristics of an item; the opinions of our friends, experts or other like-minded people; circumstances. All these sources of information are now reflected in the 'numerical world' and thus find their application in the recommender systems.

The process of choosing an item from the set of items can be viewed as a process of filtering all the items basing on some criteria. That is why very often recommendation techniques are also referred to as *filtering techniques* [Rao 2008, Bobadilla *et al.* 2013]. In the next subsection, we will describe different filtering techniques based on the source of information being used.

1.1.2 Filtering Techniques

There are different approaches in the literature to classify filtering techniques of recommender systems. We distinguish 6 of them basing on the source of information used for filtering (see Figure 1.1).

Content-based filtering [Lops *et al.* 2011, Aggarwal 2016b] relies on the idea that *users will like items that are similar to those, that they liked in the past.* Thereby, the content-based RS propose to the user those items, that have similar characteristics to the items which were highly appreciated by this user before. In this case, the features of the items are used to perform the filtering process. If the features of the items are numerically annotated, then content-based filtering techniques are straightforward and can easily provide recommendations for new users or

on new items (by matching the preferences the users specified in their profile with the features of the items). However, many times the formed recommendations are obvious and may be not interesting for the users [Aggarwal 2016e]. Indeed, usually users want the recommender system to suggest something new, something that they cannot find on their own. However, recommending only science fiction movies for a user who specified preferences towards this genre in his profile will not help him to discover cosmos documentaries, which may also interest this user. Also, it becomes very difficult to follow the changes in user preferences, as the recommender system always proposes items with a set of features, that were appreciated by the user in the past. Finally, content-based filtering requires the features of items to be numerically encoded either by people or through different analytical algorithms. This may be either computationally expensive or in some cases impossible (for example, when we want to recommend perfumes [Das *et al.* 2007] or pages in social network site [Xie *et al.* 2013]).

Demographic filtering [Aggarwal 2016e, Rao 2008] techniques are based on the assumption that *users with the same demographic characteristics have close preferences.* Demographic filtering is based on stereotypes and their usage for user modelling [Rich 1979, Rich 1989]. It is considered that the demographic filtering does not provide best results on the stand-alone basis [Aggarwal 2016e], however, it can give good insights when no other type of information is available.

Contrary to both mentioned above filtering techniques, **collaborative filtering** [Breese *et al.* 1998] neither uses the information about items (content information), nor the information about users (demographic information). Instead, it utilises the results of user-item interactions, more precisely, the known preferences of the users on those items, which they already consumed. The basis of collaborative filtering relies on the assumption that *users who had similar preferences in the past, will have similar preferences in the future.* Collaborative filtering techniques allow following the trends and the evolution of the users preferences: if users change their preferences concerning a certain type of items, it will be reflected in the resulting recommendations. However, for achieving good results the collaborative filtering system requires each user to provide his preferences on a sufficient number of items (each item to be evaluated by a sufficient number of users). For example, in the benchmark MovieLens dataset[1] each user is required to provide ratings on at least 20 items before recommendations can be generated for him.

With the appearance of social websites, recommender systems started facing new sources of information [Aggarwal 2016e, Godoy and Corbellini 2016]: social connections, folksonomies, review posts etc., as well as new tasks [Guy 2015]: recommendation of friends, tags, social content etc. Despite the fact that these new types of information require specialised processing algorithms (see, for example, [Carmagnola *et al.* 2009]), after processing, most of them can be reduced to information about items, users or their interaction. That is, to those types of information that are used in content-based, demographic or collaborative filtering respectively. However, social networks provide one new type of information, that is not used in filtering techniques mentioned above: information about social connections between people. That is why following [Groh *et al.* 2012] we define **social filtering** as filtering performed on the basis of connections between people. We can formulate the basic assumption of social filtering as follows: *users who have social connections with the active user can be used to predict his preferences.* Indeed, the network of friends or trusted people can be used to form the set of reliable users. Afterwards, the preferences of these reliable users can be used for estimating preferences of an active user (see, for instance, trust-based recommender systems [Aggarwal 2016i, O'Donovan and

[1] http://grouplens.org/datasets/movielens/

Smyth 2005]). As it is mentioned in [Groh *et al.* 2012], social filtering in this formulation can be considered as a special case of collaborative filtering. The main difference here is in the way we choose the set of users used to predict preferences of an active user. In the case of collaborative filtering, it is formed of those users who have similar rating behaviour with an active user. At the same time, in the case of social filtering, these users are defined through social connections such as friendship or the relation of trust.

In some domains, it is difficult to gather sufficient amount of information about user preferences towards some features of the items (for content-based filtering) or known opinions of other users (for collaborative and social filtering). This case may concern rarely consumed items with a complex description like expensive luxury goods or financial services. In such kind of situations the knowledge of experts is used, which form the basis of the **knowledge-based filtering** techniques [Trewin 2000]. One of the key-points of knowledge-based RS is the increase of the user control in the recommendation process [Aggarwal 2016f]. Depending on the user interaction methodology, such RS are divided into constraint-based and case-based [Aggarwal 2016f]. In constraint-based RS users specify the requirements in the form of the constraints on the item features. In the case-based systems, users provide an example of the item they would prefer to consume (a case), while the system tries to find the most similar ones to recommend to the user. As the underlying model of the knowledge-based filtering is formed basing on expert-provided knowledge, no assumptions are used while constructing such models.

Considering the situation of a user who chooses different films to watch depending on with whom he will do it, we can say that very often *the choice made by the user will also depend on external circumstances*. This statement forms the basic assumption of the **context-based filtering** [Aggarwal 2016c, Adomavicius and Tuzhilin 2011]. This type of filtering takes into account information that is external to both the user and the item, but can still affect the consumption of an item. Time of recommendation and location of the user are good examples of the contextual dimensions.

As we can see, there are multiple filtering techniques, which are based on different types of information. Each of them has advantages and disadvantages. For example, it can be annoying for a user to fill in his profile in content-based filtering or providing ratings on items in collaborative filtering. However, this is the price that has to be paid if only one source of information is available for filtering. But when it is possible to get information from multiple sources, different filtering techniques discussed above can be successfully used simultaneously within so-called **hybrid filtering** approaches [Burke 2002, Aggarwal 2016d]. Moreover, many research works prove that the combination of basic filtering techniques can help to alleviate the disadvantages of each of them and achieve higher accuracy of recommendations [Chung *et al.* 2016, Panigrahi *et al.* 2016]. This also supports the naive reasoning that *the more information is available, the better predictions can be done*.

1.1.3 Collaborative Filtering

The two filtering techniques, namely collaborative and social filtering, are of particular interest as they allow following global changes in users preferences (the dynamics of the system) with minimum participation of the active user. Indeed, for example, in content-based filtering the direction of recommendations can change only if the active user updates the information about his preferences in the profile or starts expressing his preferences towards the items with different characteristics (without them being recommended before by a system). Contrarily, collaborative filtering techniques can provide qualitatively new recommendations for an active user basing only on the fact that other users (those having similar preferences or social

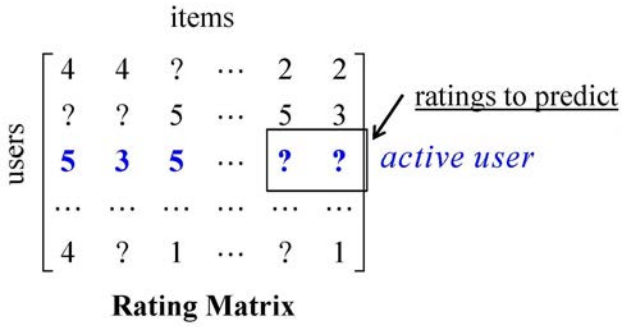

Figure 1.2: Illustration of a Rating Matrix

connections with the active user) expressed their interest towards items with different characteristics. That is why these two types of techniques will always play an important role in recommender systems, specially in the case of dynamic domains such as entertainment industry. Within the scope of formulations we choose to follow, the social filtering is a special case of collaborative filtering where for each active user the set of other users who will be used to predict the interests of an active user on new items is already defined via social connections. So further we consider only collaborative filtering, as a more general problem.

The only source of information for the pure collaborative filtering methods are the preferences of users on certain items. These preferences are traditionally referred to as ratings [Adomavicius and Tuzhilin 2005]. According to their nature, ratings can be explicit or implicit. Explicit ratings are specified directly by users, whereas implicit ratings correspond to the automatically assigned values basing on the user interaction with an item [Oard *et al.* 1998, Claypool *et al.* 2001]. For example, the value of an implicit rating for a film can be estimated basing on the fact that the user watched it completely or not and how many times he watched it (see [Chan 1999, Lee *et al.* 2008, Castagnos 2008] for more details).

The preferences of users (ratings) can be presented in a form of a rating matrix with rows corresponding, for example, to users and columns corresponding to items (see Figure 1.2). The value situated at the intersection of a row and a column corresponds to the value of the rating assigned by the corresponding user to the item.

Some values of the rating matrix are unknown as the users did not consult associated items yet. In this case, the task of collaborative filtering can be considered as the task of filling the unknown values of the rating matrix also called the task of *matrix completion* [Claypool *et al.* 2001]. After completing the rating matrix, the recommendation for an *active user* is formed of those items, that have the highest predicted values or ratings. In real applications, however, it is more useful not to predict the exact values of ratings, but rather to correctly order the items [Shani and Gunawardana 2011]. Indeed, assume we want to recommend one of the items a_1 or a_2 with the corresponding rating values 5 and 2. If the recommendation model estimates the predicted ratings as 3.5 and 3.4 respectively, we are still able to form a valuable recommendation despite the fact that the predicted ratings are far from the values of real ratings. However, the task of filling a rating matrix is more general and includes the task of items ordering [Aggarwal 2016e].

In [Aggarwal 2016g] it was shown that the problem of completing the rating matrix can be also viewed as a classification problem. Indeed, the unknown rating values can be considered as

class labels. For example, when working with binary ratings all items are divided into 2 classes: those items that are of interest for the target user and those that are not. Thereby the task of predicting the value of the ratings, in this case, will be the task of predicting the class label of the corresponding items. The relation between classification and recommendation problems explains the successful implementation of the various classification algorithms in recommender systems (for example, support vector machines [Xia *et al.* 2006] or neural networks [Salakhutdinov *et al.* 2007][2]).

Traditionally collaborative filtering techniques are divided into two classes: memory-based and model-based [Breese *et al.* 1998, Desrosiers and Karypis 2011b, Aggarwal 2016e]. Memory-based techniques do not have a clear border between learning and recommendation phases. Let us consider one of the prominent techniques, the user-based neighbourhood model [Aggarwal 2016h]. The learning phase, in this case, consists of computing the values of similarities between users basing on the known rating values. However, we cannot say that the model is built during the learning phase, as it is impossible to give recommendations basing only on similarities between users. Thus, the final computations are performed during the recommendation phase itself. The latter phase consists in estimating the rating values for an active user basing on the ratings of other users and their similarities with an active user. The memory-based techniques belong to the class of lazy-learning techniques [Aggarwal 2016h] when the system delays the generalization of the model until a query is made [Chatterjee 2011].

In the case of model-based techniques, the phases of training and recommendation are clearly distinguished [Aggarwal 2016g]: the model is built during the training phase. After that during the recommendation phase the model is used to generate appropriate recommendations. As a result, these techniques belong to the class of eager learning techniques, when the system performs generalisation during the learning phase [Aggarwal 2016h]. Among the representatives of the model-based techniques we can mention rule-based and Naive Bayes collaborative filtering, latent factor models etc. [Aggarwal 2016g].

One of the most popular model-based techniques of collaborative filtering is matrix factorization (MF), which is a representative of latent models [Koren *et al.* 2009] (as it was mentioned in the introduction, matrix factorization is essentially a feature construction method). MF-based models share the common problem of latent factors models (or feature construction techniques), namely the *lack of interpretability*. **We address this problem as a first application problematic of this thesis**.

Cold-Start Problem in Collaborative Filtering

As collaborative filtering (whether NB or MF) relies completely on the ratings, a problem occurs when there are either no known ratings for a specific item/user, or the number of known ratings for a specific item/user is very small. In this case, it is impossible to make reliable recommendations [Bobadilla *et al.* 2013]. The first problem (absence of ratings) is known as a cold-start or out-of-matrix prediction, while the second (very small number of known ratings) – as warm-start (in-matrix prediction) [Agarwal and Chen 2010, Lam *et al.* 2008]. State of the art [Bobadilla *et al.* 2013, Park and Chu 2009] distinguishes three kinds of cold-start problems: new community, new item and new user. The new community problem refers to the start-up of a new recommender system. New item and new user problems correspond to the cases when a new item/user enters an already existing system.

A very popular solution to the new item cold-start problem relies on the content of the items. The recommender either switches to a content-based techniques or mixes content with

[2]See Section 4.2 for a more detailed description of these techniques

collaborative filtering. Recommendations are provided by comparing the properties or content of the items to the content of those items that are known to be of interest for an active user [Melville *et al.* 2002, Lam *et al.* 2008]. The main limit of such a solution is the availability of the content, which depends on the type of items. Indeed, in some domains, it is hard to automatically analyse the underlying content. In addition, users' interests cannot always be characterised by content properties contained in an item, for example when perfumes [Das *et al.* 2007] or Facebook pages [Xie *et al.* 2013] are considered.

In the absence of content, a content-less new item cold-start problem is faced. A solution adopted to solve the content-less new item cold-start problem is to form a set of users, who will be asked to rate each new item in the system. The obtained ratings are used to estimate the preferences of other users on these new items. These users should represent the interests of the whole population as fully as possible and/or be capable of affecting the preferences of others. Different autors use different terms to reffere to such a set of users maynly depending on the underlying algorithm used for their identification: seed users or seeds [Liu *et al.* 2011], representative users [Liu *et al.* 2011], influential users [Rashid 2007], power users [Seminario and Wilson 2014] or leaders [Esslimani *et al.* 2013].

Seed users can be chosen randomly, but there is no guarantee that their ratings will represent correctly or affect preferences of other users. They can also be chosen within a set of experts [Amatriain *et al.* 2009], which guarantees the quality of their ratings, but this solution is expensive and in some cases experts may not be available. The problem of the automatic identification of seed users can be considered as a task of active learning, where the system automatically identifies those input elements, which will result in a better model construction [Houlsby *et al.* 2014]. Some approaches in this direction have been proposed. For example, in the frame of neighbourhood-based models, [Esslimani *et al.* 2013] proposes to discover seed users based on their connectivity and average similarity. [Rashid 2007] proposes to evaluate the importance of users as the negative influence rendered on the quality of recommendations, when these users are removed from the system. In [Houlsby *et al.* 2014] a Bayesian Active Learning approach is used. The main idea here is to select those elements, that minimise uncertainty over the parameters of the model.

1.2 Evolution of Recommender Systems

1.2.1 Recommender Systems and World Wide Web: Related Evolution Path

The evolution of recommender systems filtering techniques is related to the evolution of World-Wide-Web (WWW) and related technologies [Bobadilla *et al.* 2013] (see Figure 1.3), as the latter one represents the global information digital space for navigating in which recommender systems were developed [Deitel *et al.* 2002].

The first version of the World Wide Web *Web 1.0* was suggested by Tim Berners-Lee in 1989 [Berners-Lee 1998] as a set of static information pages. It is also known as a *read-only web* [Patel 2013] or *web of cognition* [Aghaei *et al.* 2012] due to the very limited user interaction and the static nature of represented information. With this stage of the evolution of WWW, we can associate such filtering techniques as **content filtering**, **demographic filtering** and **collaborative filtering**. The information associated with each type of filtering could be transmitted via limited, but still available interaction tools like HTML forms.

The next generation of the WWW, Web 2.0, was defined in 2004 [Aghaei *et al.* 2012] with the main feature that users from now are not passive viewers of content, but are active participants in the process of the new content creation. Web 2.0, or *read-write* web, is also related to the

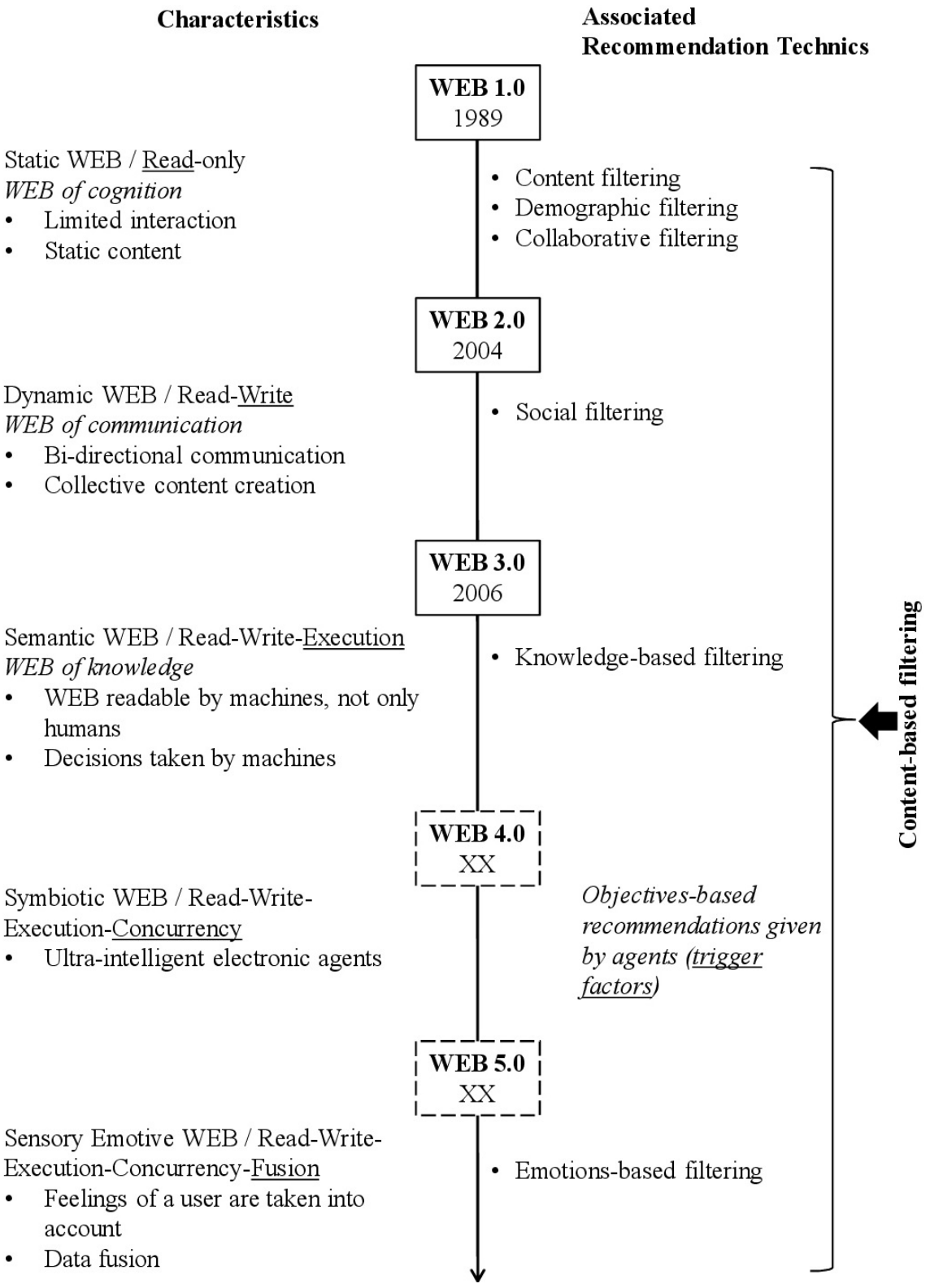

Figure 1.3: Evolution of World Wide Web and Recommender Systems

appearance of such services as social networks, blogs, wikis, which provide means not only to share information but also to create networks of friends, colleagues or like-minded users. Thereby, Web 2.0 also opened new opportunities for the recommender systems, which from now can also perform **social filtering**.

The third generation of the web, Web 3.0 was announced in 2006 [Spivack 2006] as *web of knowledge* or *semantic web*. Its aim is to make the content being understandable for machines (through meta-data tagging) and, as a consequent, to make the web being more intelligent. The intelligent web can take over some decision tasks, which were previously done by humans, that is why it is called *read-write-execution* web. With the third phase of web development, we can associate the **knowledge-based filtering**, as both of them are based on manipulation of knowledge.

The new generations of the Web 4.0 and Web 5.0 were also announced. The Web 4.0 is considered to be a *symbiotic web*, which through the use of artificial intelligence will become an ultra-intelligent electronic agent [Algosaibi *et al.* 2015] capable to take important decisions in collaboration with the user (read-write-execution-*concurrence* web). No new sources of information are predicted to become available in Web 4.0, that is why we cannot foresee the appearance of new filtering techniques. However, we can predict the need for recommendation algorithms capable to identify **trigger factors** from the side of the new electronic agents, **what corresponds to the second application problematic of this thesis**. For example, consider the case of e-learning. Assume that a user does not have good results in the course he is following. In this case, an electronic agent can take a decision that it is required to take those actions (or rather recommend the user to take those actions), which can help him succeed in the course. That is, the agent will require identifying the factors that can stimulate the move of the user from the class of backward students to the class of successful students (to identify trigger factors). Thereby, we consider that the techniques of trigger factors identification will be actively used in the frame of recommender systems associated with the fourth generation of the web.

Finally, the Web 5.0 or *sensory-emotive web* is predicted to be able to take into account feelings and emotions of the user and perform data fusion (read-write-execution-concurrency-*fusion* web). Thus we can suggest the appearance of **emotions-based filtering** techniques, which will generate the recommendations basing on the emotional state of a user.

Through the whole history of the web development, we can also see the **context-based filtering**. Indeed, with the development of technologies recommender systems have access to more and more diverse contextual information. For example, time tracking was possible even in the era of Web 1.0 and location-aware recommendations busted with the appearance of mobile devices capable of identifying geographical position of users. We should note that the development of the web and filtering techniques does not exclude previous functionalities, but only adds new possibilities. Also, the relation between the evolution of the web and RS filtering techniques is not rigid. In fact, the evolution of the web represents the evolution of ideology, which is based on the progress in the available technologies. For example, knowledge-based filtering techniques were used even before 2006 when Web 3.0 was announced (see, for instance, [Trewin 2000], which was published in 2000). However, the proposed relation between the evolution of the web and the evolution of recommender filtering techniques in our opinion reveals the stages, on which each filtering technique receives a new impetus for its global usage. For example, only with the appearance of the social networking technologies, it became possible to use the social relations between users for the recommendation, and semantic web should make knowledge-based filtering being less dependent on expert knowledge, as the web is considered to become more intelligent itself.

1.2.2 Actual Trends and Research Directions in Recommender Systems

Although the domain of recommender systems is evolving constantly, there are many open research questions. It was mentioned before that now we are having more possibilities to gather information about user preferences in both direct and indirect ways and use it simultaneously in different applications. However, this poses a question: **how can this diverse information be efficiently used in recommender systems?** One of the proposed solutions suggests gathering information about users from different domains. Like, for example, the preferences of a user on films can be used to estimate his preferences on music. This is the aim of cross-domain RS [Cremonesi *et al.* 2011, Fernández-Tobías *et al.* 2012]. The technological developments also affect recommender systems. For instance, now there is a **requirement on recommendation algorithms capable of operating in distributed environments** (distributed RS [Ricci *et al.* 2011]).

The increase of user participation also necessitates recommender systems to be **stable towards diverse human factors**. As examples, we can give malicious usage of personal information (privacy preserving recommender systems [Erkin *et al.* 2013, Zhan *et al.* 2010]) and deliberate provision of incorrect information with the aim to influence the recommendation algorithms performance (attack-resistant [Aggarwal 2016a] or robust [Burke *et al.* 2011] recommender systems).

As information technologies invade more and more parts of our life, recommender systems face **more diverse tasks**. For example, many items can be rated on a multidimensional scale. Indeed, when evaluating a restaurant a user may pay attention to the quality of the food and the serving. In this case, we have to deal with multi-criteria recommender systems [Adomavicius *et al.* 2011]. Group recommender systems focus not on personal recommendations, but on the recommendations for the groups of users [Masthoff 2011]. Finally reciprocal recommender systems deal with the cases when an 'item' can also have its preferences, like, for example, in job recommendation (the chosen job should not only fit user preferences, but also the user should feet job requirements) or in online dating (in this case both users and items are represented as people who should like each other) [Pizzato *et al.* 2010, Li and Li 2012].

Finally, there is also a need for **improving the existing solutions**. For example, according to [Tintarev and Masthoff 2011, Tintarev and Masthoff 2012] incorporating explanations into existing recommender systems is an important task which has multiple aims: to explain the user how the system works (*transparency*), to increase users' confidence in the system (*trust*), to convince users to try or buy an item (*persuasiveness*) etc. **This highlights once again the importance of the first application problematic of this thesis.**

1.3 Resume

As we see, the domain of recommender systems is an active research field with a great variety of research directions and open problems, on some of which we aim to work in this thesis. We consider recommender systems being a very important and topical domain as the goal of RS is *to help people overcome information overload*, the problem which is crucial in information society, but was not 'an issue of the day' before. This, together with the fact that both scientific problematics posed in the introduction of this thesis are reflected in this domain, determined our choice of recommender systems as an application domain for our research.

The provided overview of recommender system research field allows us to position within this domain the two application (and consequently two scientific) problematics which make the core of current research.

The process of recommendations generation can be viewed as filtering of available items basing on certain criteria. There exist different filtering techniques which use different sources of information for performing the filtering process. One of the very popular filtering techniques is collaborative filtering, which is based on exploiting rating values for predicting users liking or disliking of previously non-seen items. Contrarily to others, collaborative filtering allows following the global changes of liking patterns with minimum requirements on the active user interactions (see Section 1.1.3). Matrix factorization, a widely used collaborative filtering technique, gained its popularity due to the accuracy of the provided recommendations, as well as its scalability. Being essentially a feature extraction method, MF lacks interpretation and thereby explanation of generated recommendations. However, as it was mentioned in Section 1.2.2 providing explanations to recommender systems is an important task in recommender systems. Thereby the **first application problematic** arises.

Aligning the evolution of recommender systems with the evolution of the world wide web allows us to foresee the future development of recommender systems and predict new important research questions that may arise. The next generation of the world wide web Web 4.0 is considered to become an ultra-intelligent electronic agent capable of taking decisions in symbioses with a human. We suggest that this type of web will require generating recommendations which will not only help the person overcome information overload, but will also lead him to the desired objective. In this way we foresee the requirements on the techniques for trigger factors identification, which leads to the **second application problematic**.

In the rest of this thesis, we concentrate on the description of the proposed solutions for *AP1* (Part I) and *AP2* (Part II). Next, we conclude our work in Part III and show to which extent the proposed solutions of application problematics can be used to solve scientific questions *SP1* and *SP2* posed in this thesis.

Part I

Interpretation of Latent Features in Matrix Factorization-based Recommendation Models

Chapter 2

State-of-the-Art: Recommendations via Matrix Factorization

Contents

2.1 Matrix Factorization for Predicting Unknown Ratings

We start with the introduction of some notations. Let U be the set of users and I be the set of items, of size N and M respectively. Let R, $dim(R) = N \times M$, be the rating matrix and r_{lj} be the rating value of user $u_l \in U$, with $1 \leq l \leq N$, on item $i_j \in I$, with $1 \leq j \leq M$. Let G be a matrix. We denote by $\mathbf{g}_{(\mathbf{l},*)}$ the l^{th} row-vector of a matrix G and by $\mathbf{g}_{(*,\mathbf{l})}$ – the l^{th} column-vector of G. By $\vec{e}$ we denote some general vector, that is not necessarily associated with a matrix.

In recommender systems, the matrix factorization approach is based on the assumption that a small number of latent features (we denote this number by K and the set of features by F) influences the ratings of users on items [Sarwar *et al.* 2000b, Koren *et al.* 2009]. Thereby, MF aims to form two low-rank matrices W and V of dimensionality $K \times N$ and $K \times M$ respectively whose product will approximate the rating matrix (see equation (2.1)).

$$R \approx W^T V \tag{2.1}$$

Matrices W and V represent the extent to which users and items are related to the latent features. When multiplying these two matrices the complete original rating matrix is reconstructed,

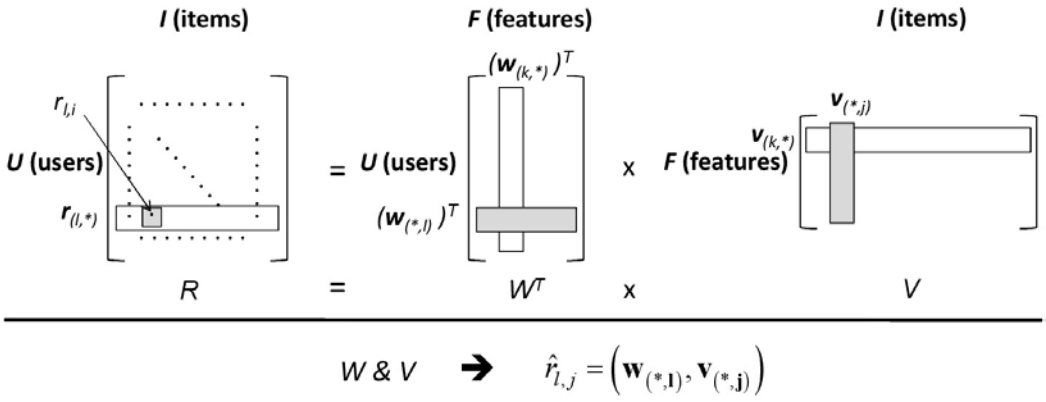

Figure 2.1: Matrices and Notations for MF

thus allowing to estimate the missing ratings. The values in the matrix V represent the relation between the items and the latent features. The vector $\mathbf{v}_{(\mathbf{k},*)}$ $(1 \leq k \leq K)$ is the M-dimensional representation of feature f_k and $\mathbf{v}_{(*,\mathbf{j})}$ $(1 \leq j \leq M)$ the K-dimensional representation of the item i_j. The values in W represent the relation between the users and the latent features. Similarly, the vector $\mathbf{w}_{(\mathbf{k},*)}$ is the N-dimensional representation of the feature f_k and the vector $\mathbf{w}_{(*,\mathbf{l})}$ $(1 \leq l \leq N)$ is the K-dimensional representation of user u_l. Figure 2.1 shows these notations on the corresponding matrices. To calculate the unknown rating $r_{(l,i)}$, equation (2.2) is used, with vector inner product operation denoted by $(*,*)$.

$$r_{l,i} = (\mathbf{w}_{(*,\mathbf{l})}, \mathbf{v}_{(*,\mathbf{j})}) \tag{2.2}$$

Equations (2.1) and (2.2) represent a basic MF model, however, in recent years, many derivative models were proposed in the frame of recommender systems. For example, the Probabilistic Matrix Factorization (PMF) models the predictive error of traditional MF as a Gaussian distribution, adding in such a way the aspect of uncertainty [Mnih and Salakhutdinov 2007, Salakhutdinov and Mnih 2008]. PMF was proven to be efficient and accurate on the Netflix dataset [Mnih and Salakhutdinov 2007]. When the number of dimentions of the original rating matrix is above two (for example, the third dimension may correspond to the content information), the generalisation of matrix factorization, called Tensor Factorization, is used [Karatzoglou *et al.* 2010]. In some situations, it can be useful to perform simultaneous factorization of multiple matrices, for example for transferring knowledge from different domains [Huang *et al.* 2012]. For such cases Collective Matrix Factorization [Singh and Gordon 2008] was proposed. If we want to obtain different latent spaces for users and items (note that in traditional MF the same set of features is shared by both users and items), Matrix Tri-Factorization should be used. This model decomposes the original rating matrix on the product of three matrices [Yoo and Choi 2009]. Two of them correspond to user- and item-related features and the additional third matrix represents the relation between these two sets of features. *In this thesis we concentrate on the basic MF model and aim to extend our results to other models in the future.*

2.2 Factorization Techniques

The task of forming factor matrices W and V is usually formulated as an optimization problem in equation (2.3), where operation $||*||_F^2$ denotes the Frobenius norm. In this section, we focus on the methods that allow to calculate the entries of both W and V.

$$\min\left(\frac{1}{2}\left|\left|R - W^T V\right|\right|_F^2\right) \tag{2.3}$$

2.2.1 Analytical Approach (Singular Value Decomposition)

When the number of features K is known, the task of matrix factorization from equation (2.3) comes down to the task of Low-Rank Matrix Approximation which, in the case of a full matrix, has an analytical solution obtained via the Singular Value Decomposition Method [Simon and Zha 2000]. For a rectangular matrix R defined on *real numbers*, there exists a factorization called Singular Value Decomposition of the form (2.4)

$$R = U\Sigma Q^T, \tag{2.4}$$

where U and Q are orthogonal matrices of dimensionality $N \times N$ and $M \times M$ respectively, and Σ is a diagonal $N \times M$ matrix with diagonal being formed of singular values of the matrix R that are normally listed in descending order. The columns of matrices U and Q are formed of left and right singular vectors of R. That is matrix U is formed of singular vectors of RR^T and matrix Q is formed of singular vectors of $R^T R$. It is know [Eckart and Young 1936] that the best K-low-rank approximation of the matrix R can be obtained in a form (2.5)

$$R = U\Sigma_K Q^T, \tag{2.5}$$

where matrix Σ_K is matrix Σ with only K largest singular values (the others are replaced by zeros). From the model given in equation (2.5) we can move to 2 factor models using square roots [Sarwar *et al.* 2000b, Sarwar *et al.* 2002] (see equations (2.6) and (2.7)).

$$W^T = U\sqrt{\Sigma_K} \tag{2.6}$$

$$V = \sqrt{\Sigma_K}Q^T \tag{2.7}$$

One of the major restrictions of the SVD method is the requirement of having a dense matrix. However, this is usually not the case in the domain of recommender systems; see, for example, popular 100K, 1M or 10M MovieLens datasets[3], which have only about 6%, 4% and 1% of known ratings respectively. The missing ratings can be filled with some values like user- or item-mean [Kim and Yum 2005, Sarwar *et al.* 2000a]. But this solution is known to distort the data [Koren 2008, Adams *et al.* 2002]. That is why techniques that use only known ratings were proposed recently.

[3] http://grouplens.org/datasets/movielens/

2.2.2 Numerical Methods

When not all entries of the matrix R are known, we can add a binary matrix of weights B of size $dim\,(B) = N \times M$ with the values $b_{l,j}$ equal to 1 if the corresponding rating is known and 0 otherwise. The new optimisation problem given in equation (2.8) uses only the known entries of the rating matrix R for finding optimal W and V.

$$\min \frac{1}{2} \left|\left|B \otimes \left(R - W^T V\right)\right|\right|_F^2, \tag{2.8}$$

where operation $\otimes$ stands for element-wise matrix multiplication. The new optimisation problem cannot be solved analytically, however, numerous numerical approaches can be used. In order to avoid the problem of over-fitting [Hawkins 2004], a regularization parameter λ is usually added [Ma *et al.* 2011, Zhou *et al.* 2008] (see equation (2.9)). As a result, the optimisation problem in equation (2.8) transforms into the optimisation problem in equation (2.9) (or in equation (2.10) in vector form). This parameter is used to penalise too large values in factor matrices and helps to avoid the case when the model memorises the training data and looses its generalisation abilities on unseen examples.

$$\min \left(\frac{1}{2} \left|\left|B \otimes \left(R - W^T V\right)\right|\right|_F^2 + \frac{\lambda}{2} \left(||W||_F^2 + ||V||_F^2 \right) \right). \tag{2.9}$$

$$\min \left(\frac{1}{2} \sum_{l=1}^{N} \sum_{j=1}^{M} b_{l,j} \left((\mathbf{w}_{*,\mathbf{l}}, \mathbf{v}_{*,\mathbf{j}}) - r_{l,j} \right)^2 + \frac{\lambda}{2} \sum_{l=1}^{N} ||\mathbf{w}_{*,\mathbf{l}}||_F^2 + \frac{\lambda}{2} \sum_{j=1}^{M} ||\mathbf{v}_{*,\mathbf{j}}||_F^2 \right) \tag{2.10}$$

Different procedures can be used for solving the formulated optimization problems, among them Stochastic Gradient Descent (SGD) [Koren *et al.* 2009], Alternating Least Squares (ALS) [Zhou *et al.* 2008], Multiplicative Update Rules [Lee and Seung 2001].

MF via Stochastic Gradient Descent

Steepest descent method [Snyman 2005] is one of the most popular numerical optimisation techniques. For a given objective function $f(\vec{x})$, the value of the parameter vector is updated in the direction inverse to the direction of the gradient (the latter one shows the direction of the steepest ascent of the function). Thereby, the update rule is given by the formula (2.11).

$$\vec{x}^{p+1} \leftarrow \vec{x}^{p} - \gamma \frac{\partial f}{\partial \vec{x}} (\vec{x}^{p}) \tag{2.11}$$

On each iteration, the steepest gradient descent method performs optimisation of the objective function using all the information available in the dataset. In contrast to the stochastic gradient descent, which randomly picks one datapoint of the dataset and performs the optimisation procedure for this datapoint only [Bousquet and Bottou 2008]. This allows to speed up the process. The stochastic gradient descent method has proven to perform well for large scale datasets while also preserving good convergence properties [Bottou 2010].

In the case of matrix factorization, the stochastic gradient descent optimisation problem (that is optimisation problem for every known rating $r_{l,j}$) is reduced to the optimisation problem given in equation (2.12).

$$\min \left(\left((\mathbf{w}_{*,\mathbf{l}}, \mathbf{v}_{*,\mathbf{j}}) - r_{l,j} \right)^2 + \lambda \left|\left|\mathbf{w}_{*,\mathbf{l}}\right|\right|_F^2 + \lambda \left|\left|\mathbf{v}_{*,\mathbf{j}}\right|\right|_F^2 \right) \tag{2.12}$$

The values of partial gradients of the objective function from equation (2.12) can be easily computed. Then, using the update rule given in equation (2.11) we can formulate update rules for the vectors of factor matrices W and V as given in equation (2.13).

$$\begin{aligned} \mathbf{w}_{(*,\mathbf{l})} &\leftarrow \mathbf{w}_{(*,\mathbf{l})} - \gamma \left[\left(\left(\mathbf{w}_{(*,\mathbf{l})}, \mathbf{v}_{(*,\mathbf{j})} \right) - r_{l,j} \right) \mathbf{v}_{*,\mathbf{j}} + \lambda \mathbf{w}_{(*,\mathbf{l})} \right] \\ \mathbf{v}_{(*,\mathbf{j})} &\leftarrow \mathbf{v}_{(*,\mathbf{j})} - \gamma \left[\left(\left(\mathbf{w}_{(*,\mathbf{l})}, \mathbf{v}_{(*,\mathbf{j})} \right) - r_{l,j} \right) \mathbf{w}_{*,\mathbf{l}} + \lambda \mathbf{v}_{(*,\mathbf{j})} \right] \end{aligned} \tag{2.13}$$

Note that for each given rating $r_{l,j}$, not all vectors of factor matrices are updated, but only those, that participate in the approximation of the given rating, that is vectors $w_{(*,l)}$ and $v_{(*,j)}$. Thereby, one iteration of the optimisation procedure consists in optimising the vectors of factor matrices for all known ratings in the training set.

The proposed procedure was used for the NetFlix prize challenge by Simon Funk[4]. Since then some modifications of SGD for matrix factorization in recommender systems were proposed, in particular, distributed SGD[Gemulla *et al.* 2011] and Parallel SGD [Zhuang *et al.* 2013].

MF via Alternating Least Squares

The objective function (2.9) can be considered as a function of 2 matrix arguments W and V, and this function can be optimised using the alternating least squares method [Zhou *et al.* 2008]. This method consists in fixing matrix V and then equating to 0 the derivative of the objective function with respect to W. Solving the obtained equation with respect to W we obtain the optimal value of W for the given value of V. After that, the similar procedure is performed for updating matrix V when the matrix W is fixed. From the described above procedure we can obtain the update rules for column vectors of both matrices (see equation (2.14)).

$$\begin{aligned} \mathbf{w}_{(*,\mathbf{l})} &\leftarrow \left(V \tilde{B}^{(l)} V^T - \lambda I \right)^{-1} V \tilde{B}^{(l)} \mathbf{r}^{\mathbf{T}}_{(\mathbf{l},*)} \\ \mathbf{v}_{(*,\mathbf{j})} &\leftarrow \left(W \tilde{B}^{(j)} W^T - \lambda I \right)^{-1} W \tilde{B}^{(j)} \mathbf{r}_{(*,\mathbf{j})}, \end{aligned} \tag{2.14}$$

where $\tilde{B}^{(l)}$ and $\tilde{B}^{(j)}$ are diagonal matrices such that $B_{l,j} = \tilde{B}^{(l)}_{j,j} = \tilde{B}^{(j)}_{l,l}$.

We can see that contrary to SGD, ALS performs a global optimisation (that is all known ratings are used when updating either W or V). Analysing performance of ALS and SGD, authors of [Yu *et al.* 2014] show that ALS converges faster and is less sensitive to parameters than SGD. However, it is also less scalable in the case of large datasets.

MF via Multiplicative Update Rules

In [Lee and Seung 2001] authors proposed and proved the convergence of multiplicative update rules for the components of matrices W and V given in equation (2.15).

$$\begin{aligned} W &\rightarrow W \otimes \frac{VR^T}{VV^TW + \lambda W} \\ V &\rightarrow V \otimes \frac{WR}{WW^TV + \lambda V}, \end{aligned} \tag{2.15}$$

[4] http://sifter.org/ simon/journal/20061211.html

where operation $\frac{*}{*}$ stands for element-wise matrix division. Note that these update rules require matrix R to be full. In the case of non-full matrix R the update rules are formulated as given in the equation (2.16). These modified update rules also possess a convergence property [Mao and Saul 2004].

$$\begin{aligned} W &\to W \otimes \frac{V(B \otimes R)^T}{V\left(B \otimes W^T V\right)^T + \lambda W} \\ V &\to V \otimes \frac{W\left(BR\right)}{W\left(B \otimes W^T V\right) + \lambda V}, \end{aligned} \tag{2.16}$$

The update rules from equations (2.15) and (2.16) are of particular interest as they allow to preserve the positive sign of elements of W and V. Indeed, if all elements of matrices W, V and R are positive, then the proposed update procedure will always result in new approximations of factor matrices with guaranteed positive elements. In this way, the non-negative matrix factorization (NMF) can be performed, which requires all entries of both factor matrices to be non-negative [Devarajan 2008].

NMF was introduced as a method that allows learning parts of objects, for example discovering parts of faces on an image [Lee and Seung 1999]. Later this property of NMF was used in many domains: speech processing [Behnke 2003], text mining [Chagoyen *et al.* 2006], computational biology [Devarajan 2008], etc.

2.3 Features Interpretation in Matrix Factorization-based Recommender Systems

2.3.1 General Overview

The values in both matrices W and V that result from factorization are those that optimally describe the known ratings in the original rating matrix: they are designed to be those that minimise the loss function in equations (2.3), (2.8) or (2.9). As a consequence, the features are not directly interpretable. So they are generally only used to predict ratings. However, interpreting these features could be an important added value. It could help to understand the underlying relation between users and items and to explain the recommendations presented to users. As it was mentioned in introduction, providing explanations in RS is important to increase the user satisfaction and trust in the system [Herlocker *et al.* 2000, Sinha and Swearingen 2002, Ortega *et al.* 2014].

Some authors were interested in providing interpretation for MF features. For example, [McAuley and Leskovec 2013] proposes to align features of the matrix factorization models with review topics learned through Latent Dirichlet Allocation (LDA), by introducing functional dependence between features and topics and simultaneously learning both models. This idea was further extended in a number of works. For example, contrary to the baseline [McAuley and Leskovec 2013], [Rossetti 2014] merges all the reviews into a single document; [Hu *et al.* 2015] and [Zhao *et al.* 2015] add social-based information; [Xin *et al.* 2015] uses heterogeneous topics instead of homogenous. Using close ideas [Zhang *et al.* 2014] extracts features from the reviews through the Phrase-level Sentiment Analysis, and then incorporates them into an MF-based framework. Finally [Donkers *et al.* 2015b, Donkers *et al.* 2015a] incorporates into MF model information about user-provided item tags.

In the literature, we can also find an interpretation for specialised models. For example, in [Hu *et al.* 2008] authors propose an MF-based model for implicit feedback datasets. This model differs from the basic matrix factorization model in two aspects. First, as no explicit ratings are provided, the elements of matrix R can take only two values 1 and 0. The value 1 shows that an item was consumed by a user and thus serves as an indicator of liking the item. The value 0 corresponds to the case when the item was not consumed and thereby the preferences are unknown. Second, a new weight matrix is introduced in the model which reveals the confidence in each particular value of the matrix R (for example, a user may buy an item as a gift without liking it). Authors show that recommendations in such a model can be viewed as a linear combination of past preferences of the users and features, thereby, are used to calculate the coefficients of this linear combination.

There is also a group of methods that try to interpret the matrix factorization model without changing its structure and incorporating external knowledge. We consider these methods to be of particular interest, as they correspond to the more general case, and we proceed to the discussion of these methods in the next subsection.

2.3.2 Interpretation of Basic Matrix Factorization Model

Here we discuss 4 approaches for the interpretation of MF recommendation models: interpretation based on the definition of MF, interpretation via non-negative matrix factorization, associating of features with groups of users via probabilistic model, and Representative-based Matrix Factorization.

Interpretation Based on Definition

To this group of approaches, we attribute those that simply follow the general definition of an MF model to provide its interpretation. Recall that matrix factorization is based on the assumption that a small number of latent features explains the user-item interaction. Let us consider [Koren *et al.* 2009], where a movie dataset is used. The authors say that features can represent obvious dimensions such as comedy/drama, amount of action, orientation to children, less well-defined dimensions such as quirkiness, or represent completely uninterpretable dimensions (what corresponds to the basic assumption of MF models). The procedure for features interpretation presented in this paper is reduced to plotting items in the (sub)space of latent features and deriving the possible meaning of features from the positions of items and their characteristics. It may be not always possible to derive the meaning of features in such a way. As a result, the provided explanations can be incomplete.

Interpretation via Non-Negative Matrix Factorization

Recall that non-negative matrix factorization was introduced as a factorization method capable of representing the original data as a sum of components. This property of NMF was used in [Pessiot *et al.* 2006] and [Zhang *et al.* 2006] to interpret recommendations. [Pessiot *et al.* 2006] interpret features as *imaginary* users, who represent a certain behavioural type (they are prototype users). [Zhang *et al.* 2006] interpret features as communities of users grouped basing on similar interests. These two interpretation approaches can be considered identical. Indeed, from the theoretical point of view, both interpretations associate features with specific preference patterns, which can be visualised either as a community of users or as an imaginary user, whose interests represent a certain community. From the experimental point of view, in both papers the same technique was used to illustrate the links between behavioural patterns and features:

the authors analysed the content of items having strong association with different features (recall that association of features and items is reflected in the matrix V) and show that these items usually have something in common, like the production period [Pessiot *et al.* 2006] or an actor [Zhang *et al.* 2006].

Associating Features with Groups of Users via Probabilistic Model

The model proposed in [Hernando *et al.* 2016] assumes that latent features should represent different groups of users and builds a probabilistic factor model with respect to this assumption. That is the relation between features and groups of users is incorporated during the model construction. This model depends on two additional parameters: α standing for the possibility of obtaining the overlapping groups of users, and β standing for the amount of evidence required for the algorithm to deduce that a group of users likes or dislikes a particular item. The matrix W is formed in such a way, that each column-vector $\mathbf{w}_{(*,\mathbf{l})}$ represents the probability that the user l belongs to each of K groups, thereby equality (2.17) holds.

$$\sum_{k=1}^{K} w_{k,l} = 1 \tag{2.17}$$

The rows of the matrix V in their turn reveal the probability of the fact that users from each particular group like different items. That is $v_{k,j}$ shows the probability of users from the group k liking the item j. Because liking of a particular item i_j does not restrict liking of another item $i_{j'}$ there are no restrictions on the sum of the elements in matrix V. However, as entries of both W and V stand for some probabilities, the factor matrices are forced to be non-negative.

Representative-based Matrix Factorization

The Representative-based Matrix Factorization (RBMF) [Liu *et al.* 2011] is based on the idea of searching for a set of representative elements (either users or items), through which the general user-item relation can be modelled, and associating these elements with latent features. Authors propose two models depending on the nature of representative elements: user-RBMF and item-RBMF.

User-RBMF models the rating matrix R as $R \approx XA$. The matrix A is formed of ratings of K users that were chosen as representatives. And the matrix X is formed as the solution of the optimisation problem (2.18). We can see that the optimisation problem of user-RBMF (2.18) is almost the same as the optimisation problem for the standard regularised MF with the difference that one of the factor matrices is fixed and the entries of the other one are calculated through an optimisation procedure. The item-RBMF model is formulated in a similar way.

$$\min \left(\frac{1}{2} ||R - XA||_F^2 + \frac{\lambda}{2} ||X||_F^2 \right). \tag{2.18}$$

The representative elements are found through the matrix maximum volume concept (see [Liu *et al.* 2011] and [Goreinov *et al.* 2010]), which is used to identify those columns in the matrix that are large in magnitude and are linearly independent. First, the rank-k SVD decomposition of the rating matrix R for user-RBMF or of the transposed rating matrix R^T for item-RBMF is performed. After that the *maxvol* algorithm [Goreinov *et al.* 2010] is used to find a $k \times k$ maximal-volume submatrix of the first matrix in the SVD decomposition. The users/items, that correspond to the rows of the chosen submatrix are considered as representative elements.

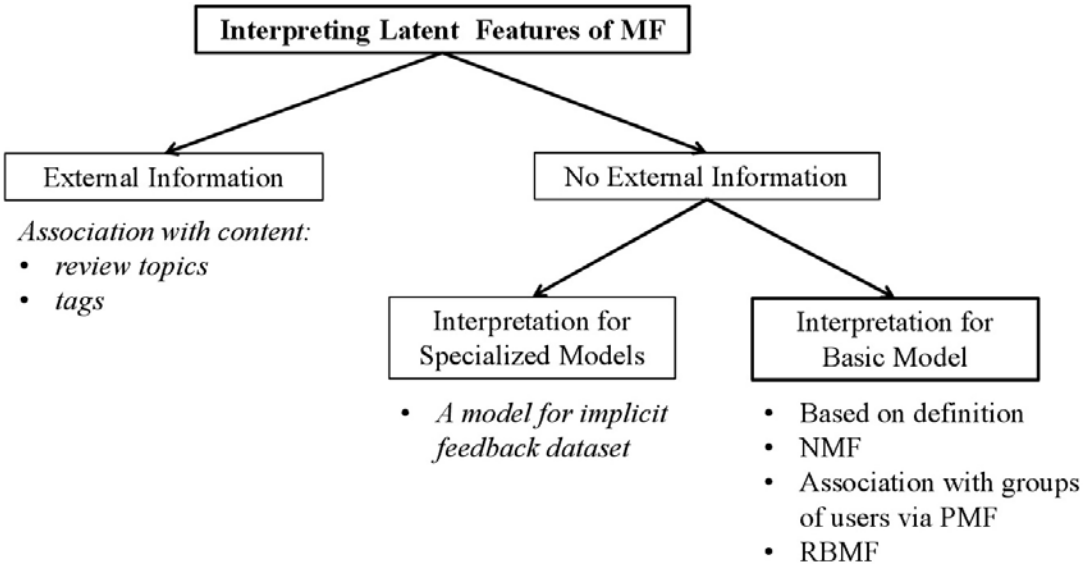

Figure 2.2: Research Directions in Interpretation of Latent Features Resulting from Matrix Factorization

2.3.3 Discussion

Although the importance of model interpretation in recommender systems was stressed many times (see [Herlocker *et al.* 2000, Sinha and Swearingen 2002, Ortega *et al.* 2014] and Section 1.2.2), to the best of our knowledge, not many research works are dedicated to the interpretation of MF recommendation models. However, from the works presented above, we can derive some structure and general tendencies in this research domain. Our view of research directions in MF interpretation is schematically presented in Figure 2.2.

On the top level, we divide all research approaches on those that do or do not incorporate external information into MF model (as external information here we understand all information apart of user-provided ratings). All the approaches mentioned in the second paragraph of Section 2.3 incorporate into matrix factorization model external information: primary content information about the items.

Further, we divide the methods that do not incorporate external information into those providing interpretation for specialised matrix factorization models and those working over the interpretation of the basic MF model. Among known to us approaches, we can attribute to the first group only one model which was presented in the third paragraph of the Section 2.3.1. This model was developed for implicit feedback datasets and it uses additional weight matrix to represent the confidence in each particular implicit rating.

The methods dedicated to the interpretation of the basic MF-model were detailed in the previous subsection. From the description of these methods, we can see that it is possible to try to interpret latent features as more or less obvious dimensions (comedy/drama, quirkiness, etc.), as it is done in interpretation based on definition. Alternative approach consists of exploiting the non-negative factorization model (NMF-based interpretation techniques) what allows to suggest *a priori* the link between features and behavioural patterns of users. In both these approaches (definition-based and NMF-based interpretation), the analysis of the content of items having high associations with particular features is afterwards used to derive the characteristics of each

feature. It may seem that these methods can be attributed to the group of those, which use external information. However, we distinguish them as the methods described in this paragraph do not incorporate external information into MF model, but rather use content to show that features indeed can be interpreted in the way authors propose (as obvious/non-obvious dimensions and behavioural patterns). Note, however, that the procedure of deriving the actual meaning of features in these models is complex, as it is done manually by exploiting human knowledge.

We also mentioned two other models dedicated to the interpretation of the basic MF-model: association of features with groups of users via probabilistic factorization and Representative-based Matrix Factorization. The first approach builds factorization model in such a way, that features are associated with groups of users. It requires additional parameters such as the possibility of obtaining overlapping groups and amount of evidences required to deduce that a group of users likes a certain item. However, it can be difficult to identify the values of these parameters as they are application-dependent. Finally, the RBMF model initialises one of the factor matrices with ratings associated with representative elements (users or items) and calculates the entries of the other factor matrix through an optimisation procedure. In this way, it is ensured that features are associated with the chosen representative elements. The identification of these elements, however, requires additional computations.

As we can see the task of interpretation of latent features is not trivial. Authors either try to incorporate in MF model external information or perform some modifications of the factorization model (like RBMF or implicit association with groups of users). The approaches that try to interpret the original model directly without modifying it (interpretation based on the definition and NMF-based interpretation) do not provide an automated procedure for deriving actual characteristics of features and require human analysis.

Inspired by definition- and NMF-based interpretation models we suppose that latent features should represent some behavioural pattern. However, in order to make the optimisation procedure automatic, we suggest to associate features not with imaginary users or groups of users, but with *real users of the system that can represent the corresponding pattern in the best possible way.*

Note that many techniques intended to interpret latent features rely on at least partially non-negative models. Indeed, apart from NMF-based interpretation techniques, the model proposing association of users groups with features is also composed of non-negative matrices, as their elements represent probability values. Even in RBMF model, in the case of non-negative ratings, one of the factor matrices is guaranteed to be non-negative, as it is composed of ratings of representative elements. Thereby, we also decide to rely on non-negative matrix factorization in our work and choose Multiplicative Update Rules as an optimisation technique.

As we do not incorporate either external information into the model nor modify it in such a way that it becomes interpretable on its own, we have to validate the proposed interpretation. In the literature we can see two possible validation schemes for such situations: to analyse the information about elements associated with features (what is done in definition- and NMF-based interpretation approaches) or to evaluate the ability of the chosen elements to correctly represent user-item relation via the cold-start problem (used for validating the RBMF). In the case of RBMF, the new item cold-start problem was used for the evaluation of chosen users and the new user cold-start problem was used for the evaluation of chosen items. We choose the approach based on cold-start problem, new item cold start problem in particular. This allows us not only to be content-independent but also to validate our hypothesis of the association between latent features and users. As it was mentioned before, *if latent features correspond to real users then, considering the fact that latent features represent the relation between all users and items, the chosen feature-associated users should represent the basic behavioural patterns of the whole*

population of the users. Thereby, next we proceed to brief discussion of the cold-start problem solutions in the frame of MF recommendation models.

2.4 Cold-Start Problem in Matrix Factorization

Some approaches for the cold-start problem solution have been proposed in the frame of MF. Following the general tendency in CF (see Section 1.1.3), they can be divided into two groups: 1) those that require additional external information (like content, social structure or both), and 2) those that use seed users/items (for the content-less problem).

Among the first group of approaches, we can mention [Gantner *et al.* 2010] that proposes to solve the new item cold-start problem by learning a mapping function between latent features and item attributes. Given a new item and its corresponding attributes, the latent feature vector for the new item is computed. Then the rating matrix is filled for this new item. [Saveski 2013] proposes a joint factorization of the rating matrix and the content matrix. [Enrich *et al.* 2013] introduce the MF-based model, which provides interpretation via transfering the knowledge from one domain to another through shared tags. In [Jamali and Ester 2011] the structure of the social network is used as a source of additional information. The feature vector of a user in this case is constructed as being dependent on the feature vectors of his/her neighbours. Finally, authors of [Salakhutdinov and Mnih 2008] incorporate both social and content information into Probabilistic Matrix Factorization model (through the parameters of the model).

Moving to the second group of approaches, [Zhou *et al.* 2011] and [Sun *et al.* 2013] propose a solution to the new user problem (which is symmetric to the new item problem), where the feature vectors of MF are learned through the new user answers in the initial interview process. The seed items (those used in the interview) in this case are chosen through passing a decision tree which being a component of the model is constructed during the learning phase. Note that in this case, the set of seeds is not fixed and changes from user to user. In [Liu *et al.* 2011] seed users/items are chosen as those who can be associated with the features through the matrix maximal-volume concept (see the description of RBMF in the previous section). For solving the new item cold-start problem, the matrix A (see equation (2.18)) is filled with ratings of seed users on new items and the missing values are predicted through the multiplication of matrices X and A. Within RBMF it is also possible to formulate a similar solution for the new user cold-start problem.

2.5 Resume

There is a grate variety of MF-based recommendation models, what proves the popularity of this approach among researches. Diverse MF-based models allow to calculate predictions based on probability inference (PMF), incorporate content information or information from other domains (collective MF), learn different sets of features for users and items (Matrix Tri-Factorization) etc. However, despite the proved prediction efficiency MF models possess one significant drawback: lack of interpretation.

Some approaches were proposed to overcome this problem. However, most of them either 'import' interpretation from external sources usually via aligning features with content information, or change the structure of MF model in such a way, that it becomes interpretable. There also exist attempts to perform the direct interpretation of latent features of the model, however, they are based on human analysis and thereby cannot be automated.

Basing on the idea expressed in the state-of-the-art that *features represent behavioural patterns*, we suggest making a link between latent features and some real users from the system, whose behaviour will approximate the corresponding patterns. We describe our proposed approach in the next chapter. Such an interpretation allows us to be independent of content information and perform automatic interpretation. However, the proposed interpretation has to be justified. Following the state-of-the-art and our hypothesis that features can be associated with users, we choose to validate the proposed interpretation via the new item cold-start problem. From the brief discussion of the existing new item cold-start problem solutions presented (see Section 2.4), we define two global directions. Methods belonging to the first direction consist in using item-related content information. At the same time, methods belonging to the second direction form a set of users, whose opinions on new items can be used to derive the preference of all users on the new items (seed users). Thereby, if our proposed interpretation is correct, it should be possible to use the feature-associated users as seed users. However, the procedure of using ratings of feature-associated users to solve the new item cold-start problem depends on the way the user-feature association is done. This procedure is also detailed in the next chapter.

Finally, following the majority of the works related to the interpretation of latent features we choose to rely on non-negative matrix factorization model, which we obtain using the multiplicative update rules optimisation procedure.

Chapter 3

Proposed Solution: A Technique for Automatic Interpretation of Latent Features

Contents

3.1 Preliminaries

Let us start with an example. Let L_1 and L_2 be two linear spaces of dimensionality respectively 6 and 3, with basic vectors in canonical form $\{\vec{w}_n\}$, $n \in \overline{1,6}$ and $\{\vec{f}_k\}$, $k \in \overline{1,3}$. Let the transfer matrix from L_1 to L_2 be specified by matrix P (equation (3.1)).

$$P = \begin{pmatrix} 0 & 0 & p_{13} & p_{14} & 1 & p_{16} \\ 1 & 0 & p_{23} & p_{24} & 0 & p_{26} \\ 0 & 1 & p_{33} & p_{34} & 0 & p_{36} \end{pmatrix} \tag{3.1}$$

We can say that $\vec{w}_5$, $\vec{w}_1$ and $\vec{w}_2$ are direct pre-images of $\vec{f}_1$, $\vec{f}_2$ and $\vec{f}_3$ respectively. Indeed, $P\vec{w}_5 = P\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}^T = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T = \vec{f}_1$. By analogy, $P\vec{w}_1 = \vec{f}_2$, $P\vec{w}_2 = \vec{f}_3$. At the same time vectors $\vec{w}_3$, $\vec{w}_4$ and $\vec{w}_6$ will be mapped into linear combinations of basic vectors $\vec{f}_1$, $\vec{f}_2$ and $\vec{f}_3$. For example, $P\vec{w}_3 = p_{13}\vec{f}_1 + p_{23}\vec{f}_2 + p_{33}\vec{f}_3$ corresponds to the linear combination for $\vec{w}_3$.

The matrix W, resulting from the factorization of R, can be considered as a transfer matrix from the space of users to the space of features [Koren *et al.* 2009]. So, analysing the example considered above, we can say that if matrix W has a form similar to P (in equation (3.1)), i.e. *W has exactly K unitary columns with one non-0 and equal to 1 element in different positions, then the users corresponding to these columns are direct pre-images of the K features.* Following [Guermeur *et al.* 2004], we say they represent the canonical coding of the features. These feature-related users will be referred to as *representative users* (*RUs*, abbreviation *RU* will be used to refer to one representative user). As a consequence, we consider that the features can thus be directly interpreted as users (representative users). This idea is very simple and can be related to the task of searching for the basis of the vector space in linear algebra. However, in the case of recommender systems, one important constraint should be taken into account, that is the sparseness of the rating matrix. Still, as we are not aiming to find the perfect solution but the one that can ensure the good-enough performance quality, we find this idea to be promising. Also, to the best of our knowledge, it was not exploited previously.

Obviously, in the general case, one cannot guarantee that the matrix W will be in a form similar to matrix P. Worse, none of the column-vectors of matrix W may directly represent the canonical form of a feature. However, we choose not to modify the values of this matrix, but to consider as RUs those users, whose vectors in W are the closest to the required canonical form. In this way, we can provide the interpretation of an already existing factorization model (during the post-processing step). The procedure, which we propose for the identification of representative users is described in details in the next section.

3.2 Our Approach

3.2.1 Identification of Representative Users

Here we describe the approach we propose for the identification of real users, which is based on the ideas described in the previous section. Our approach was presented at two conferences [Aleksandrova *et al.* 2014c] and [Aleksandrova *et al.* 2014b]. It consists of 3 steps further detailed below.

Step 1: Normalize Matrix W. Once the matrices W and V are obtained, the normalization of each of the N column vectors of the matrix W is performed, that results in unitary columns. The resulting normalized matrix will be denoted by W^{norm}. The normalization is performed in order to obtain the matrix W in the form closest to P. After such a transformation, the new matrix W^{norm} still represents the same relations between users and features, but with certain scaling coefficients. Next, the column-vectors of the W^{norm} matrix are analysed with the aim to identify the best candidates for representing latent features (those that are close to the canonical form).

Step 2: Form Groups of Pre-image Candidates. In this step the groups of pre-image candidates are formed.

We consider a user u_l to be the best pre-image candidate for a feature f_k if the vector in matrix W^{norm} that corresponds to u_l (column-vector $\mathbf{w}^{norm}_{(*,l)}$) is the closest to the corresponding

canonical vector (a vector with the only one non-0 and equal to 1 value on the position k, denoted by $\vec{c}_k$). The notion of closeness between vectors is expressed through the Euclidean distance. That is the task of identification of the representative user u_l is reduced to solving the optimization problem and his/her position (l) is defined by equation (3.2).

$$l = \arg\min_{l' \in U} \left[dist(\vec{c}_k, \mathbf{w}^{norm}_{(*,l')}) \right] \tag{3.2}$$

Let us consider the following example. Assume that we have a vector $\vec{\alpha}$ of the form $\begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_K \end{pmatrix}^T$ with the value of the norm equal to 1 ($\sqrt{\alpha_1^2 + \alpha_2^2 + \ldots + \alpha_K^2} = 1$). Then the distance between $\vec{\alpha}$ and the first canonical vector $\vec{c}_1 = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}^T$ is expressed by $dist^2(\vec{c}_1, \vec{\alpha}) = (1-\alpha_1)^2 + \alpha_2^2 + \ldots + \alpha_K^2$. Simple mathematical transformations result in equation (3.3).

$$dist^2(\vec{c}_1, \vec{\alpha}) = 2(1 - \alpha_1) \tag{3.3}$$

This means that the minimum of the distance is obtained under the condition $\alpha_1 \to \max$. Taking into account this reasoning, we consider a user u_l as a pre-image candidate for the feature f_k if the maximum value of the appropriate vector $\mathbf{w}^{norm}_{(*,l)}$ is situated on the position k.

Therefore, all users are divided into groups of pre-image candidates according to the position of the maximal value in the associated column vectors from the matrix W^{norm}. The corresponding formal procedure is presented in Algorithm 1. The algorithm has one input parameter – the matrix W^{norm}. Based on the dimensions of the input matrix W^{norm}, the number of groups K and the total number of users N is defined on the first step of the algorithm. Next, we initialise K groups of pre-image candidates $GC\_1, \ldots, GC\_K$ as empty sets. Finally, making a loop through all N users of the system a user u_l is put in the group which is associated with the position of the maximal value in the corresponding vector $\mathbf{w}^{norm}_{(*,l)}$. The position of a value e' in a vector $\vec{e}$ is calculated using the function $pos\,(\vec{e}, e')$ (see line 7 of Algorithm 1).

Algorithm 1 Form Groups Of Pre-image Candidates

```
1:  procedure FormGroupsOfPrCand(W^norm)
2:      [K, N] = size(W^norm)
3:      for k = 1 : K do
4:          GC_k = {}
5:      end for
6:      for l = 1 : N do
7:          k = pos(w^norm_(*,l), max(w^norm_(*,l)))
8:          GC_k = GC_k ⋃ {u_l}
9:      end for
10:     return GC_1, ..., GC_K
11: end procedure
```

Step 3: Identify RUs. Once all users are divided into subgroups of pre-image candidates for each feature, we can identify RUs using Algorithm 2. In each group of pre-image candidates GC_k, the representative user is defined as a user $u_{l''}$ whose vector $\mathbf{w}_{(*,l'')}$ is the closest to the canonical vector $\vec{c}_k$ (see line 3 of the algorithm). If for some reasons the chosen representative user cannot be used for solving the assigned task, the next best candidate within the current pre-image candidates group can be considered as RU. Additionally, if a certain group of pre-image

candidates, say GC_k, is empty, that is in step 2 there were no columns in the matrix W^{norm} with the maximum value being situated on the position k, the user from another group with the smallest value of distance to the canonical vector $\vec{c}_k$ can be chosen as RU for the feature k. In this way, we ensure that all features will be associated with representative users.

Algorithm 2 Find RUs

1: **procedure** FINDRUS($GC\_1, \ldots, GC\_K$)
2: **for** $k = 1 : K$ **do**
3: $l'' = \underset{u_{l'} \in GC\_k}{\arg\min} \left[dist(\vec{c}_k, \mathbf{w}^{norm}_{(*,l')}) \right]$
4: $RU\_k = u_{l''}$
5: **end for**
6: **return** $RU\_1, \ldots, RU\_K$
7: **end procedure**

Note that the original MF model remains unchanged. The normalization of the matrix W in our approach is performed only for the identification of representative users. However, when computing recommendations, the original W and V matrices are used.

The presented procedure of RUs Identification (see Algorithm 3) results in a set of users (RUs) that are associated by bijective mapping with the latent features of MF. As latent features are considered to represent the relations between users and items, the obtained feature-related users should be capable of representing the same interconnections. It means that the set of these users should correctly reflect the interests of the whole population of users. Therefore, representative users can be used as a set of seed users to solve the new item cold-start problem. Indeed, asking their opinion on new items we can infer the opinion of the entire population.

Algorithm 3 RUs Identification

1: **procedure** RUSIDENTIFICATION(W)
2: $W \rightarrow W^{norm}$
3: $[GC\_1, \ldots, GC\_K] = FormGroupsOfPrCand\,(W^{norm})$
4: $[RU\_1, \ldots, RU\_K] = FindRUs\,(GC\_1, \ldots, GC\_K)$
5: **return** $RU\_1, \ldots, RU\_K$
6: **end procedure**

3.2.2 Interpretation of Recommendations

Theoretical Statements

The way we propose to interpret features also has the advantage to help explaining the recommendations provided by MF. Let us rewrite equation (2.2) in the form of equation (3.4).

$$r_{l,i} = \sum_{k=1}^{k=K} w_{k,l} v_{k,j} \tag{3.4}$$

As it was discussed in Section 2.1 the vector $\mathbf{w}_{(*,\mathbf{l})}$ is a K-dimensional representation of a user u_l (that is the representation of a user u_l in the space of latent features), and the vector $\mathbf{v}_{(*,\mathbf{j})}$ is a K-dimensional representation of the item i_j (that is the representation of an item i_j in the same space of latent features), see Figure 2.1. Thus, if the set of features is interpreted

as a set of representative users, then both vectors $\mathbf{w}_{(*,\mathbf{l})}$ and $\mathbf{v}_{(*,\mathbf{j})}$ represent user u_l and item i_j in the space of representative users. Therefore, value $v_{k,j}$ may express the preferences of a representative user k on the item j and $w_{k,l}$ – closeness of the user u_l to the representative user k. This makes the rating estimation process of MF being close to the one of NB. Indeed, the rating estimation process in NB is done using equation (3.5).

$$\hat{r}_{l,j} = \sum_{k'=1}^{k'=K'} sim(u_l, u_{n_{k'}}) r_{n_{k'},j}, \tag{3.5}$$

where K' – is the number of neighbours, $sim(u_l, u_{n_{k'}})$ – the similarity between an active user u_l and his/her k'-th neighbour $u_{n_{k'}}$, and $r_{n_{k'},j}$ – is the rating assigned by the neighbour $u_{n_{k'}}$ to the item j. The notion of similarity in NB is usually expressed through the correlation [Desrosiers and Karypis 2011a].

Let us compare equations (3.4) and (3.5) with $w_{k,l}$ corresponding to $sim(u_l, u_{n_{k'}})$ (closeness of the user u_l to the k-th representative user or similarity between user u_l and his neighbour $u_{n_{k'}}$) and $v_{k,j}$ corresponding to $r_{n_{k'},j}$ (preferences of the k-th representative user on the item j or rating value of the neighbour $u_{n_{k'}}$ on this item). Note, however, that though the relation between latent features and representative users is bijective, it is not identical (recall from the previous section that representative users correspond to latent features with certain scaling coefficients). Thus vectors $v_{k,j}$ and $w_{k,l}$ may not directly correspond to ratings and similarity values, but reflect these dependencies in an indirect way.

The link between MF and NB makes the basis of our publication in a national journal [Chertov *et al.* 2015].

Toy Example

In order to show that values $v_{k,j}$ and $w_{k,l}$ resulting from MF can be interpreted as ratings of representative users and as similarity values between representative users and the rest of the users from the data set, we provide here an analysis on a small toy example. In this way, we show how recommendations can be interpreted. Let us consider a rating matrix given in Table 3.1, which represents the ratings of 12 users on 12 movies (items). Each movie is annotated with genre (with possible values *comedy* or *drama*) and the release decade (with possible values *70*, *80* or *90*). Analysing the ratings provided in the table we can draw some conclusions concerning the preferences of each user. For example, the first user likes comedy films and dislikes dramas, the second one has inverse preferences. The fifth user prefers the films released in 80's regardless of the genre and the sixth user likes comedies released in 70's, has middle preferences towards comedies released in 80's and dislikes comedies from 90's as well as drama films. The short description of the preferences for each user is provided in the last column of Table 3.1.

After performing a non-negative matrix factorization[5] of the rating matrix in Table 3.1 with number of features $K = 5$, we can identify representative users, following the approach presented in Section 3.2.1. The following users were identified as representative: u_3, u_8, u_1, u_7 and u_9 corresponding to features f_1, f_2, f_3, f_4 and f_5 respectively (see the first column of Table 3.1 with RUs shadowed in grey). Using the values in the matrix W and the provided association of latent features with the representative users, the preferences of the rest of the users can be decomposed into the linear combination of the interests of the representative users. For example, the linear combination for the second user is given in equation (3.6). The coefficients in the second part

[5]Recall that in Section 2.3.3 following the state-of-the-art approaches, we chose to use non-negative matrix factorization model with multiplicative update rules as an optimisation procedure.

Table 3.1: Model Example of a Rating Matrix for Interpretation

item characteristics													descript
genre	comedy			comedy			drama			drama			
year	70	80	90	70	80	90	70	80	90	70	80	90	
IDs/F	1	2	3	4	5	6	7	8	9	10	11	12	
$1/f_3$	*5*	*5*	*5*	*5*	*5*	*5*	*1*	*1*	*1*	*1*	*1*	*1*	comedy
2	*1*	*1*	*1*	*1*	*1*	*1*	*5*	*5*	*5*	*5*	*5*	*5*	drama
$3/f_1$	*5*	*3*	*1*	*5*	*3*	*1*	*5*	*3*	*1*	*5*	*3*	*1*	70_1 or $80_{0.5}$
4	*1*	*3*	*5*	*1*	*3*	*5*	*1*	*3*	*5*	*1*	*3*	*5*	$80_{0.5}$ or 90_1
5	*1*	*5*	*1*	*1*	*5*	*1*	*1*	*5*	*1*	*1*	*5*	*1*	80
6	*5*	*3*	*1*	*5*	*2*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	comedy & (70_1or$80_{0.5}$)
$7/f_4$	*2*	*5*	*2*	*1*	*5*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	comedy&80
$8/f_2$	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*5*	*1*	*1*	*5*	drama & 90
$9/f_5$	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*5*	*1*	*1*	*5*	*1*	drama & 80
10	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*4*	*1*	*1*	*5*	*1*	drama & 80
11	*5*	*5*	*1*	*5*	*5*	*1*	*5*	*5*	*1*	*5*	*5*	*1*	70 or 80
12	*1*	*1*	*5*	*1*	*1*	*5*	*1*	*1*	*5*	*1*	*1*	*5*	90

of the equation correspond to those in the matrix W^{norm}, that is the second linear combination corresponds to a vector with a unique norm.

$$\begin{aligned} u_2 &= \mathbf{1.4411}u_3 + \mathbf{1.5208}u_8 + 0u_1 + 0.0273u_7 + \mathbf{1.5027}u_9 \\ &= 2.5785\,(\mathbf{0.5589}u_3 + \mathbf{0.5898}u_8 + 0u_1 + 0.0106u_7 + \mathbf{0.5828}u_9) \end{aligned} \tag{3.6}$$

Recall that the second user likes drama regardless of the release decade. From the equality provided above, we can see that representative users u_8, u_9 and u_3 have the main impact on the preferences of the considered user u_2. The user u_8 likes drama films released in 90's, user u_9 - dramas from 80's and user u_3 adds to this linear combination preferences of the films released in 70's, as it is his/her major interest. User u_7, who likes the comedies released in 80's has a small impact in the linear combination. At the same time, the linear combination coefficient for the user with opposite interests (user u_1, who prefers the comedy films) is equal to 0.

Let us consider one more example: the linear combination for user u_{10}, provided in equation (3.7). The major coefficient in the linear combination corresponds to user u_9, who has exactly the same preferences as the analysed user u_{10} (both of them prefer drama films released in 80's). The rest of the coefficients in the linear combination (3.7) are relatively minor.

$$\begin{aligned} u_{10} &= 0u_3 + 0.3594u_8 + 0.0284u_1 + 0.4621u_7 + \mathbf{1.6522}u_9 \\ &= 1.7530\,(0u_3 + 0.2050u_8 + 0.0162u_1 + 0.2636u_7 + \mathbf{0.9425}u_9) \end{aligned} \tag{3.7}$$

In this way, we have shown that the values $w_{k,l}$ can be interpreted as similarity values between representative users and the rest of the users from the data set. Now we proceed to show that the values $v_{k,j}$ represent preferences of representative users on items.

Table 3.2: Correlation of Rows from Matrix V with Ratings of Users

RUs	u_3	u_8	u_1	u_7	u_9
$\mathbf{v}_{(*,\mathbf{k})}$	$\mathbf{v}_{(\mathbf{1},*)}$	$\mathbf{v}_{(\mathbf{2},*)}$	$\mathbf{v}_{(\mathbf{3},*)}$	$\mathbf{v}_{(\mathbf{4},*)}$	$\mathbf{v}_{(\mathbf{5},*)}$
$corr\left(\mathbf{v}_{(*,\mathbf{k})}, \mathbf{r}_{(*,\mathbf{k})}\right)$	0.9187	0.8620	0.9894	0.9325	0.9790
$\underset{1\leq l\leq N}{\text{mean}}\left(corr\left(\mathbf{v}_{(*,\mathbf{k})}, \mathbf{r}_{(*,\mathbf{l})}\right)\right)$	-0.0521	-0.0657	-0.0023	0.1476	0.1411
$\underset{1\leq l\leq N}{\text{mean}}\left(abs\left(corr\left(\mathbf{v}_{(*,\mathbf{k})}, \mathbf{r}_{(*,\mathbf{l})}\right)\right)\right)$	0.4192	0.5554	0.4247	0.3825	0.4526

In order to show that vectors $\boldsymbol{v_{(k,*)}}$, resulting from MF, can be interpreted as ratings of the representative users, we calculated the value of the Pearson correlation between the ratings of representative users and the lines of matrix V corresponding to the associated features (see Table 3.2). In the provided table the first row lists the representative users, the second row lists the rows of matrix V corresponding to the features associated with representative users, and the third row provides correlation values between ratings of representative users and corresponding rows in V. In the fourth row, the mean correlation between a certain row in the matrix V and all 12 users is given and the last row presents the mean of the absolute correlation between a certain row in matrix V and all 12 users.

From Table 3.2 we can see that ratings of representative users are highly correlated with corresponding rows of the matrix V (compared with the mean correlation). This shows that the vectors $\boldsymbol{v_{(k,*)}}$ can be interpreted are ratings of representative users on items. Note that this example has shown that features of the MF approach can be associated with representative users when multiplicative update rules are used as an optimization procedure, that is the algorithm that results in non-negative factor matrices V and W. The validity of this statement should be tested for other optimization procedures, like ALS or SGD.

We also would like to stress that the usage of content information in our example has only demonstrative purposes. Indeed, we propose to interpret features as users (not to associate them with content-related information) and use content only to show that the users from the system can be associated with chosen representative users basing on similar or dissimilar interests.

3.2.3 Seed Users for Alleviating Cold-Start Problem in MF-Based Models

Here we present our approach for solving the cold-start problem. As it was shown in the literature review section, the idea of asking some predefined users (seed users) to provide their rating on new items and then using these ratings to solve the cold-start problem is widely exploited. We thus propose to exploit the ratings provided by the seed users, either the set of RUs or any other set of users suitable according to certain criteria. The novelty of our approach relies upon the way these ratings are used in the frame of MF models and the way the seed users are chosen. Indeed, we propose to choose as seeds those users whose corresponding vectors in the matrix W are the closest to the canonical form. After that, the proposed solution of the cold-start problem is based on using the ratings of the seed users and factor matrices. To the best of our knowledge, no other approaches from the literature propose the solution of the cold-start problem for the original factorization model, that is without imposing restrictions on the form of matrices W and V.

Let I_{new} be the set of new items (those that make the cold-start situation to happen) and S be the set of indexes of users, who are considered as seeds. The way the new item cold-start

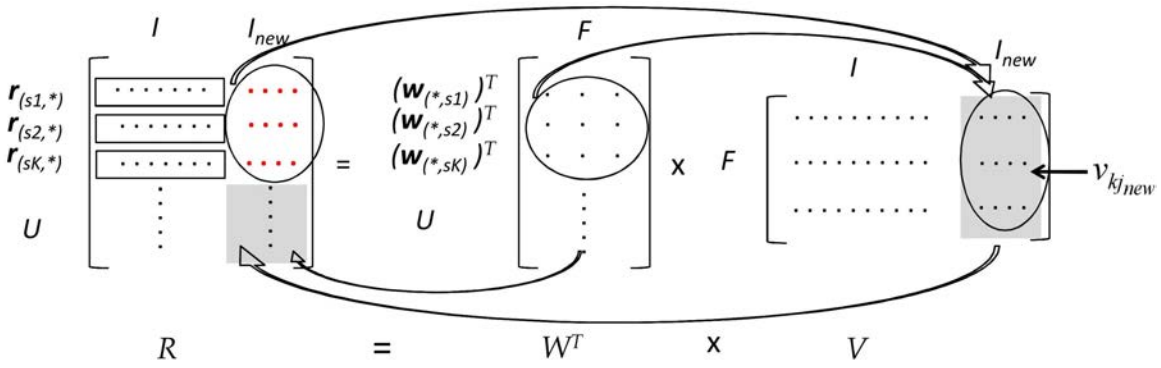

Figure 3.1: Solving Cold-Start Problem with Seed Users for Matrix Factorization

problem can be solved is explained below and is schematically presented in Figure 3.1. The information in red represents additional data ratings provided by seed users and the information in grey areas represents values computed automatically.

By asking seed users u_{s_1}, u_{s_2}, ... ($s_k \in S$) to rate new items $i_{j_{new}} \in I_{new}$, the matrix R can be filled with these new ratings (represented as red points in matrix R in Figure 3.1). For simplicity sakes, seed users are presented in the upper part of the matrix R, and the new items – in its right part. If the values in V that correspond to the new items (grey part of matrix V, Figure 3.1) can be automatically computed from these new ratings, then the estimated ratings of other users (not seed users) on the new items (the grey part of the matrix R in Figure 3.1) can also be computed by multiplying matrices W and V (arrows in the lower part of Figure 3.1).

The challenge here is thus to define a way to compute the new values $v_{kj_{new}}$ in matrix V, for each $k \in 1..K$ and $i_{j_{new}} \in I_{new}$. The matrix V can be completed by exploiting both new rating values from matrix R and the vectors $\mathbf{w}_{(*,s_k)}$ in W (see arrows on the upper part of Figure 3.1). For each new item $i_{j_{new}} \in I_{new}$ this task simply comes down to solve a system of linear equations (3.8). Note that, in order to obtain the unique solution of the system (3.8), the number of seed users should be equal to the number of latent features. That is we should have exactly K seed users.

$$\begin{cases} r_{s_1,j_{new}} = \sum_{p=1}^{K} w_{p,s_1} \cdot v_{p,j_{new}} \\ \vdots \qquad\qquad \vdots \\ r_{s_k,j_{new}} = \sum_{p=1}^{K} w_{p,s_k} \cdot v_{p,j_{new}} \end{cases} \tag{3.8}$$

For solving the system of linear equations (3.8) all the seed users have to provide their ratings on the new item j_{new}. However, it is not always the case in reality. Users, who have been chosen as seeds, may not be familiar with the item that they are asked to provide the rating on, and/or may have no desire to rate some items. In this situation, filling procedures can be used. That is missing ratings from seed users can be replaced by either some mean values (global mean rating or the mean values by certain item/user) or the ratings of the other closest candidates (for example, the next best RU candidates, see Section 3.2.1). In the latter case, not only the rating of the next best seed user should be taken, but also vector from the matrix W, that corresponds to this new seed user, should be used while solving the system (3.8).

The proposed solution of the cold-start problem was presented at a conference [Brun *et al.* 2014] and further investigated in a journal paper [Aleksandrova *et al.* 2016a].

3.2.4 Resume

The two main points of our approach lie not only in the way we propose to interpret MF latent features but also how we use the obtained set of users as seeds for solving the new item cold-start problem. Our interpretation is based on a simple idea: finding canonical column-vectors (or those whose form is the closest to the canonical) in the matrix W and then associating the users corresponding to these vectors with the features depending on the position of the maximum values in the vectors. The proposed interpretation procedure is simple and neither requires any human interaction (compared to the state-of-the-art approaches that associate features with groups of interest [Zhang *et al.* 2006] or certain behavioural patterns [Pessiot *et al.* 2006]) nor any additional content information like the review topics [McAuley and Leskovec 2013].

Among works from the state of the art, RBMF model [Liu *et al.* 2011] is the closest to ours in both ways: the way features of MF are interpreted (associated with users) and the resulting solution to the cold-start problem (the ratings on new items are predicted using the ratings of feature-associated users on these new items). However, to start with, authors of [Liu *et al.* 2011] form a rigid dependence of the features on the chosen users. This does not allow to solve the cold-start problem when one of the chosen users does not provide his/her ratings for some reasons. Contrarily, our approach allows using ratings of the next best candidates in this case. Second, our method allows simultaneous identification of both representative users and representative items. In this case, the matrix V should be analysed in the same way as W was analysed for the identification of representative users. At the same time, the referred RBMF model is either user- or item-oriented. Third, to obtain an interpretable model, authors first fill one of the factor matrices with ratings of the representative elements (users or items), then the second matrix is formed through an optimization procedure, while our approach proposes interpretation within the original MF model (without altering it).

3.3 Data Description and Experimental Protocol

3.3.1 Data description

To perform experimental evaluations of the proposed ideas we use 2 benchmark datasets: 100K MovieLens[6] and Jester[7].

MovieLens provides 100K discrete ratings on films ranging from 1 to 5 for 943 users on 1682 items. 6.3% of user/item pairs have a rating value, the rest of ratings are unknown. Jester dataset (more precisely, its first most dense part) has 72.5% of known ratings, that are real values ranging from -10.00 to +10.00 and are given by 24,983 users on 100 jokes. As in our experiments non-negative matrix factorization is used, which requires non-negativity of the input rating matrix, the ratings of the Jester dataset were offset by 11, thus resulting in the $[+1; +21]$ values range. Table 3.3 summarizes information about both datasets used with percent of available ratings in the dataset, denoted by θ.

We choose MovieLens dataset as it is very popular among researchers (see, for example, [Park and Tuzhilin 2008, Lam *et al.* 2008, Tinghuai *et al.* 2015, Kim and Kim 2003] and many other papers) and is a benchmark in RS. However, as we can see, MovieLens dataset is rather

[6] https://movielens.org/
[7] http://www.ieor.berkeley.edu/ goldberg/jester-data/

Table 3.3: Information about used Data Sets (MovieLens and Jester); θ – % of known ratings

charateristic	MovieLens	Jester
recommendation domain	movies	jokes
# users	943	24,983
# items	1,682	100
ratings characteristic	dicrete	real
ratings range	1 – 5	-10.00 – +10.00
ratings range after offset	1 – 5	+1.00 – +21.00
θ,%	6.3	72.5

sparse. While simulating the procedure of asking seed users to provide their ratings on new items (extracting appropriate ratings from the test set), usually we can get no more than 40% of answers. Thus we are forced to use a filling procedure (see Section 3.2.3). Using a dataset with so small number of given ratings makes it impossible to study some aspects of the proposed approach. Contrary to MovieLens, using Jester dataset we can obtain a considerable number of new items, for which ratings of all seed users are known. Consequently, we can study in details the proposed approach. Therefore, Jester dataset is used as a basic dataset in our experiments. The MovieLens dataset is used to confirm some results (those, which do not require the presence of the ratings of all seed users) and to study their data-independence. It may seem that using Jester dataset reduces our approach to the case of non-sparse rating matrices, which is not the case in many real applications. However, we study the performance of our models depending on the sparseness of the input rating matrix (learning set) as well. For this, some ratings are discarded from the learning subset to obtain, for example, a learning matrix with 10% or 5% of known ratings. At the same time, all ratings in the test set are preserved in order to provide the high rate of answers of seed users concerning new items.

3.3.2 Alternative Methods for Seeds Identification

In this subsection, we describe several strategies proposed in the literature for finding sets of seed users (ensembles of seed users, or seeds), which will be further used as alternatives for comparison with representative users.

Inspired by works of [Rashid *et al.* 2002] and [Liu *et al.* 2011], we have chosen the following strategies for seeds identification:

1. Top raters (*topK*) – the set of users, who provide the largest number of ratings in the system.

2. Most diverse users (*mDiv*) – the set of users, who have different rating behaviour; the diversity is measured in terms of pairwise correlation, that is this set is formed of those users, who have the lowest pairwise correlation within their set.

3. Most dispersive (*mDisp*) – users of this set are chosen in such a way, that every selected user rates items differently; that is, he/she has dispersion in the values of provided ratings (contrary to the case, when a user rates all items equally, for example).

4. Most neighbours (*mNeigh*) – the set of users, who occur in the largest number of neighbourhoods in the NB approach.

In all following experiments, the number of seed users in the set is equal to the number of features of the MF model. The performance of the proposed model is evaluated similarly to [Liu *et al.* 2011], where different strategies of seed users identification were compared in the frame of the same method for the cold-start problem solution. Additionally, we compare our approach with the baseline RBMF model presented in Section 2.3.2.

3.3.3 Experimental Protocol

Following the general tendency in the recommender systems community, in order to obtain more reliable experimental results, a 5-fold cross validation is performed. In every case, both for cold-start and non-cold-start evaluations, the original rating matrix is divided into test and learning sets, containing 20% and 80% of the original information respectively. Using this proportion of ratings in test and learning sets, 5 independent folds (pairs of test and learning sets) are formed. After that values of required characteristics (evaluation measures) are calculated on each fold. The final result is a mean value of 5 values obtained for each fold.

For the non-cold-start case, we use a classical evaluation protocol, that is 20% of randomly chosen ratings form the test set and the rest 80% are used as a learning set for the model training. In the case of cold-start experiments, 20% of items are randomly chosen as the new items (I_{new}) with their ratings forming the test set. The ratings of the remaining 80% of items are used to train the model. Note that in this case the learning and the test sets do not necessarily contain exactly 80% and 20% of ratings respectively. Once the model is trained and the set of appropriate seed users is formed, their ratings are extracted from the test set (this procedure simulates the process of asking seed users to provide their ratings on new items). After this, the remaining ratings in the test set are used for the model evaluation. The procedure of forming learning and test sets for both cold-start and non-cold-start cases is schematically presented in Figure 3.2. For the simplicity of visualization, the test ratings and test items are grouped in the right part of the rating matrix R. In the experiments, when all seed users are required to provide a certain percent of ratings on new items, the test set is formed in another way. Only those new items, on which seed users of the chosen ensemble have the required percent of ratings are used. We refer to this set of items as *actual test set* and schematically present it in a red bold rectangle in the left part of Figure 3.2. In such cases, the test set contains less than 20% of items and it may vary for different ensembles of seed users.

3.3.4 Evaluation metrics

For the experimental evaluation, we use four metrics classically studied in the literature: Normalized RMSE (NRMSE), Normalized Distance-based Performance Measure (NDPM), Relative Deterioration (DET) and Test Coverage (COV). NRMSE is the normalized version of the widely used evaluation measure RMSE (Root Mean Square Error), which is computed through the difference between real and estimated ratings [Shani and Gunawardana 2011]. NRMSE is a fraction of the RMSE value and the difference between maximum ($maxR$) and minimum ($minR$) possible rating values in the dataset (see equation (3.9)). Possessing the normalization property, NRMSE does not depend on the ranges of ratings in the input data.

$$NRMSE = \frac{RMSE}{maxR - minR} = \frac{\sqrt{\sum_{l=1}^{L}(r_l - \hat{r}_l)^2 / L}}{maxR - minR} \tag{3.9}$$

where L corresponds to the number of ratings in the test set, r_l represents a rating value from the test set, $\hat{r}_l$ – the corresponding estimated value. The $NRMSE$ measure evaluates how

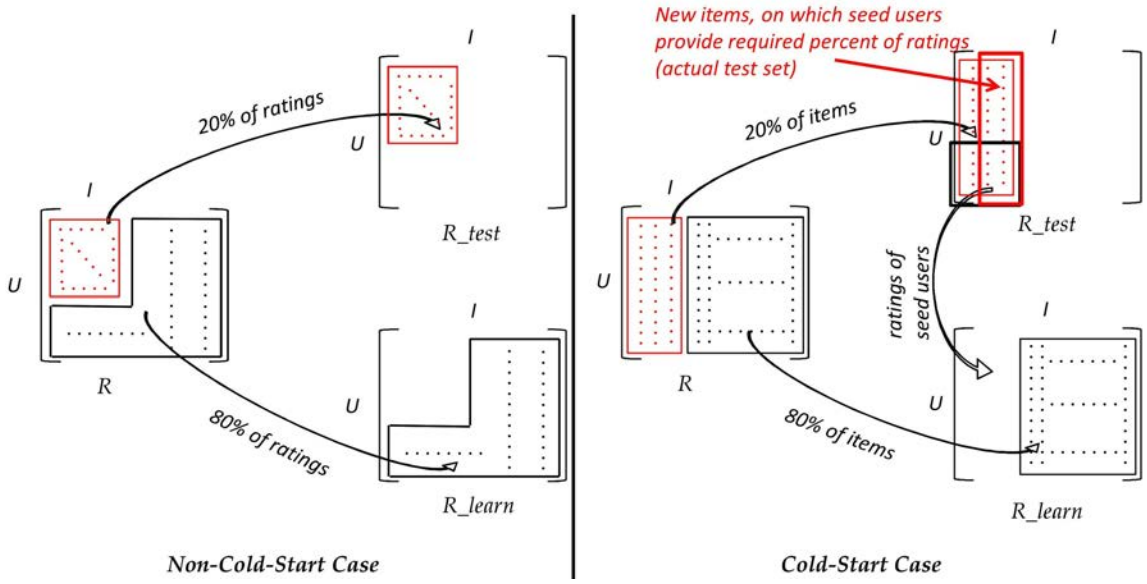

Figure 3.2: Forming Learning and Test Sets for Non-Cold-Start and Cold-Start Experiments

close is the predicted rating to the real one.

However, in practice, it is more important to predict the order of the items corresponding to the preferences of a certain user rather than to estimate the exact value of the rating. In reality, the recommendation engine is usually requested to provide an ordered list of items, that the current user will like the most. Therefore, in the current work, we will use a ranking measure as a primary evaluation measure. Still, we present the results in terms of NRMSE as well for the comparison purposes, as this metric is often used for RS evaluation.

Ranking-based evaluations can be done through the NDPM measure [Shani and Gunawardana 2011]. Assume that we have real and predicted ratings of items for a user u. Define C_u as the number of pairs of items for which the real ranking asserts an ordering (*i.e.* not tied), that is the number of pairs with different values of the ratings. Define as C^+ and C^- the number of pairs for which the model ranking asserts the correct order and incorrect order respectively. Finally, denote by C_u^0 the number of pairs where real ranking does not tie the elements (they have different ranking positions) but the model ranking ties. Thus the following equality holds: $C_u^0 = C_u - (C_u^+ + C_u^-)$. The NDPM is given by equation (3.10) according to [Shani and Gunawardana 2011].

$$NDPM = \frac{C_u^- + 0.5\ C_u^0}{C_u} \tag{3.10}$$

The maximum NDPM value, namely 1, is reached when the model places all non-tied by real ranking pairs of elements in inverse order, that is $C_u = C^-$, $C^+ = 0$ and $C_u^0 = 0$. Thus $NDPM = 1$ is the worst obtained value. The mean value (0.5) can be obtained for the case when the model ties all the elements, that is all the elements are predicted to have the same value of ratings. Note, however, that other configurations (not only tying all elements) can also result in $NDPM = 0.5$. Finally, the lowest NDPM (0) indicates that all non-tied by real ranking pairs of elements (those, having different ranking positions in reality) were correctly ordered by

the model (the best ranking), that is $C_u = C^+$, $C^- = 0$ and $C_u^0 = 0$.

For comparison purposes, in some cases we will use not the exact values of the error metrics, but the relative deterioration of the error of analysed model (AM) compared to the error of the base model (BM) (see equation (3.11)).

$$DET\,(AM, BM) = \frac{err\,(AM) - err\,(BM)}{err\,(BM)} 100\% \tag{3.11}$$

As it was mentioned in the previous section, the size of the actual test set can change in our experiments. Thereby, in order to evaluate the number of ratings predicted by a certain model, we use additionally Test Coverage (COV) metric. As Test Coverage, we understand the percentage of ratings from the test set, for which the model can estimate rating values (see equation (3.12)).

$$COV = \frac{|T_{predicted}|}{|T|} 100\%, \tag{3.12}$$

where $|T_{predicted}|$ is the number of predicted ratings in the test set (the *actual test set* in Figure 3.2), and $|T|$ is total number of ratings in test set.

3.4 Experimental Results

In this section, we present the experimental evaluation of the proposed approach. First, we focus on the identification of the optimal number of features for MF. Second, we analyse the characteristics of different sets of seed users. Finally, we analyse the performance of the MF-based solution for the new item cold-start problem.

3.4.1 Matrix Factorization: Performance Analysis

In this subsection, we search for the optimal value of the number of features K for the MF-based models.

We conduct a series of experiments with different number of features and different values of the regularization parameter λ, for both Jester and Movielens datasets. For Jester, λ changes from 0 to 300 with an increment of 5, for Movielens – from 0 to 30 with an increment of 1. Note that the number of features K and the value of the regularization parameter λ are not the parameters of our model (as the proposed interpretation is made for the existing MF model), but the parameters of MF itself.

Table 3.4 and Table 3.5 present the values of optimal configurations, with respect to different error measures for Jester and Movielens datasets, as well as errors (NRMSE and NDPM) for the boundary values of λ (0 and 300/30). Minimum and maximum values of NRMSE and NDPM through different numbers of features K are presented in the tables as shadowed. The last row of the tables contains the difference between these maximum and minimum error values. As it is seen from the tables, when the optimal value of λ is used the difference between error values for different number of features is insignificant and does not exceed for NRMSE and NDPM respectively 0.0007 and 0.0064 (for Jester), 0.0109 and 0.0135 (for MovieLens).

Analysing the values given in Table 3.4 and Table 3.5, we can note that when the number of features K increases, the value of optimal λ also has a tendency to increase. This fact supports the existence of the overfitting problem, that is with the growth of its size the MF-model becomes more precise on the learning set, but less accurate on the test set. It is obvious that the quality

Table 3.4: Jester: Optimal Parameter Values; maxDif – difference between maximum and minimum error values (presented as shadowed) through different number of features K

K	config	value/λ	
		NRMSE	NDPM
5	min	0.2120 / 0	0.3645 / 0
	opt	**0.2061 / 60**	**0.3533 / 55**
	max	0.2297 / 300	0.3837 / 300
10	min	0.2147 / 0	0.3678 / 0
	opt	**0.2054 / 75**	**0.3482 / 70**
	max	0.2296 / 300	0.3836 / 300
15	min	0.2190 / 0	0.3700 / 0
	opt	**0.2057 / 90**	**0.3493 / 110**
	max	0.2295 / 300	0.3836 / 300
20	min	0.2238 / 0	0.3721 / 0
	opt	**0.2056 / 90**	**0.3482 / 100**
	max	0.2296 / 300	0.3836 / 300
25	min	0.2290 / 0	0.3834 / 0
	opt	**0.2059 / 95**	**0.3492 / 100**
	max	0.2294 / 300	0.3835 / 300
50	min	0.2539 / 0	0.4073 / 0
	opt	**0.2055 / 100**	**0.3469 / 100**
	max	0.2294 / 300	0.3835 / 300
75	min	0.2675 / 0	0.4195 / 0
	opt	**0.2057 / 100**	**0.3484 / 100**
	max	0.2295 / 300	0.3836 / 300
maxDif		0.0007	0.0064

Table 3.5: MovieLens: Optimal Parameter Values; maxDif – difference between maximum and minimum error values (presented as shadowed) through different number of features K

K	config	value/λ	
		NRMSE	NDPM
5	min	0.2584 / 0	0.3423 / 0
	opt	**0.2364 / 3**	**0.3050/ 4**
	max	0.2769 / 30	0.3237 / 30
10	min	0.2457 / 0	0.3330 / 0
	opt	**0.2421 / 6**	**0.3058 / 8**
	max	0.2755 / 30	0.3262 / 30
15	min	0.2711 / 0	0.3622 / 0
	opt	**0.2451 / 8**	**0.3100 / 10**
	max	0.2759 / 30	0.3247 / 30
20	min	0.2759 / 0	0.3734 / 0
	opt	**0.2452 / 9**	**0.3075 / 11**
	max	0.2763 / 30	0.3258 / 30
25	min	0.2852 / 0	0.3731 / 0
	opt	**0.2455 / 10**	**0.3051 / 12**
	max	0.2763 / 30	0.3261 / 30
50	min	0.2969 / 0	0.3868 / 0
	opt	**0.2473 / 10**	**0.3056 / 13**
	max	0.2763 / 30	0.3248 / 30
100	min	0.2979 / 0	0.3883 / 0
	opt	**0.2461 / 10**	**0.3047 / 14**
	max	0.2765 / 30	0.3246 / 30
500	min	0.2692 / 0	0.3586 / 0
	opt	**0.2435 / 8**	**0.2982 / 9**
	max	0.2754 / 30	0.3234 / 30
1000	min	0.2566 / 0	0.3374 / 0
	opt	**0.2424 / 7**	**0.2965 / 5**
	max	0.2751 / 30	0.3231 / 30
maxDif		0.0109	0.0135

Table 3.6: Jester: Characteristics of seed users; χ – ratio of the mean number of ratings provided by seeds to the mean number of ratings per user in the whole dataset

seeds	χ	innerC	outerC	outerC/innerC
RUs	0.7111	0.0107	0.0617	5.77
topK	1.1306	0.0650	0.0981	1.51
mDiv	0.8598	0.0137	0.0891	6.50
mDisp	0.9446	0.1091	0.1224	1.12
mNeigh	0.4367	0.7136	0.3290	0.46

of prediction on the learning set increases with the number of features K, thus the higher penalty (value of the regularization parameter λ) should be used to smooth this effect. As we can see, using the optimal value of the regularization parameter λ lets us obtain a precise model when the number of features K is not very large as well. Thus, in order to have an optimal model in terms of its size and representativeness, we used $K = 10$ as the number of features in all further experiments on both datasets.

3.4.2 Analysis of Different Sets of Seed Users

This subsection is dedicated to the analysis of the main characteristics of different sets of seed users. This is done in order to understand the ability of seeds to represent the interests of the entire population of users. We suppose that seed users should not have a too small number of ratings (otherwise, they can be unable to represent the preferences of other users on most of the items) and they should have different behavioural patterns. Thereby, we focus on the following characteristics of the set of seed users: the mean number of ratings per seed user, the average correlation within the set of seed users ($innerC$) and the average correlation of seed users with other users ($outerC$), as well as the ratio of these two correlation values ($outerC/innerC$).

The characteristics of different studied sets of seed users: representative users (RUs), top raters (topK), most diverse users (mDiv), most dispersive users (mDisp) and most neighbours (mNeigh), are depicted in Table 3.6 and Table 3.7 (Jester and MovieLens datasets respectively). By χ (the first column of the tables) we denote the ratio of the mean number of ratings provided by seed users to the mean number of ratings per user in the whole dataset (not seed users). As χ is a relative characteristic, it is data-independent and lets us compare results for different datasets. We can see from the Table 3.6 and Table 3.7 that for both datasets, RUs tend to rate less than in the other sets of seed users. However, RUs of the Jester dataset has a higher ratio χ then in MovieLens. By definition, top-raters have the highest number of ratings.

Now we proceed to the analysis of the correlations. We assume that a set of users is more suitable for representing the interests of the entire population of users if it is composed of users with different behavioural patterns. That is the users of this set should be less correlated between them than with the users outside the set (the value of the ratio $outerC/innerC$ should be high).

For both datasets, RUs are almost 6 times less correlated within their set than with other users of the dataset. The set of most diverse users (mDiv) has a considerably lower inner correlation, compared to the value of the correlation with not seed users ($outerC/innerC > 1$). This is logical, as this set was formed as a set of those users, that are not correlated with each other. But as a random factor was used when forming this set (the first user of the most diverse set is chosen randomly), it can be not optimal. For example, for the Jester the value of inner

Table 3.7: MovieLens: Characteristics of seed users; χ – ratio of the mean number of ratings provided by seeds to the mean number of ratings per user in the whole dataset

seeds	χ	innerC	outerC	outerC/innerC
RUs	0.4838	0.0134	0.0766	5.72
topK	4.1616	0.2344	0.1840	0.79
mDiv	1.4892	0.0381	0.1143	2.70
mDisp	1.3750	0.1072	0.1520	1.42
mNeigh	1.0849	0.4468	0.2757	0.62

correlation for the RUs and mDiv sets is very close (both sets are composed of highly diverse users), however for the MovieLens dataset we can see that the set of representative users has lower inner correlation. It means that both sets RUs and mDiv represent different behavioural patterns, but the set of representative users is more optimal.

The ratio of outer and inner correlation ($outerC/innerC$) of the top raters (topK) and most dispersive (mDisp) sets is close to 1 for both datasets. The value of the inner correlation of users from these sets is close to the mean pairwise correlation of the users from the whole datasets. The ensemble of most neighbours is composed of users that have higher correlation within the set than outside it. Therefore, these three sets are less suitable for representing the interests of the whole population of users.

Considering statements above, we can conclude that the set of representative users tend to be composed of users with different behavioral patterns (as it has the lowest inner correlation and a high value of the ratio $outerC/innerC$) and thus it can be used for representing interests of the entire population of the users.

3.4.3 Cold-start for Jester

In the following experiments, we analyse the performance of the proposed solution of the new item cold-start problem, i.e. exploiting seed users. We start with the detailed analysis of our approach. For this, we first need that all seed users provide ratings on the new items. Due to the high sparsity of the MovieLens dataset, none of the new items in our simulation can get ratings from all seeds, but this is not the case for the Jester dataset. Therefore, in this subsection we focus on different aspects of the cold-start problem analysis performed on the Jester dataset. We consider the case when all seed users provide ratings and analyse different filling procedures when not all seed users can give their opinion on a specific new item. Also, we compare performance for different levels of learning dataset sparseness. Some results for the MovieLens dataset are presented in the next subsection.

Let us introduce some notations. By *MF-RUs* model we denote an MF-based algorithm for solving cold-start problem (Section 3.2.3) with representative users (RUs) used as seed users. *MF-topK* will correspond to the same algorithm with the set of top raters used as seed users. Therefore, in such abbreviations the second part will correspond to the set of seed users used (*RUs*, *topK*, *mDiv*, *mDisp*, *mNeigh*).

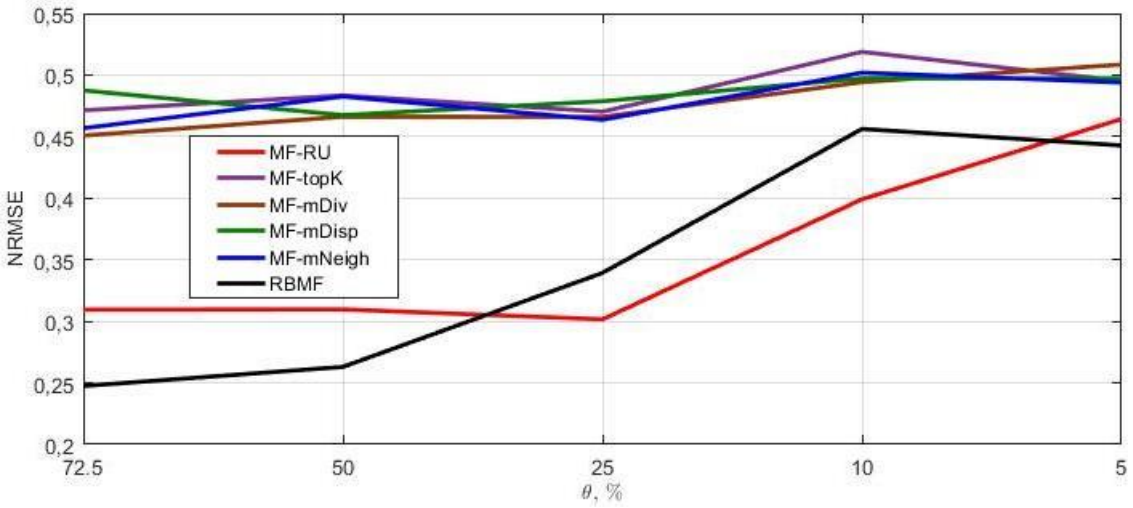

Figure 3.3: Jester: Dependence of NRMSE on Percent of Known Ratings in the Input Rating Matrix (θ)

Comparison of Different MF-based Models and RBMF

We start with the analysis of the performance of different MF-based models on Jester dataset. In this set of experiments, seed users are required to provide all ratings on new items. Thus, the test set is formed only of those new items, on which all seed users can give their ratings. We also compare the performance of the proposed approach with the baseline approach from the Section 2.3.2 (RBMF). Additionally, we study the effect of the sparsity of the learning set, that is we randomly discard some portion of ratings from the learning set in order to obtain the learning matrix with the required percentage of known ratings. At the same time, all ratings in the test set are preserved, what ensures the presence of ratings provided by seed users on new items.

Figures 3.3 and 3.4 present the evolution of NRMSE and NDPM respectively for different models (MF-RUs, MF-topK, MF-mDiv, MF-mDisp, MF-mNeigh and RBMF) with respect to different percents of known ratings in the learning set, denoted by θ. The figures show that among all MF-* models, MF-RUs performs the best regardless the value of θ. The results of the MF-topK, MF-mDiv, MF-mDisp, MF-mNeigh models tend to be close between themselves. The RBMF model performs better than MF-RUs in terms of RMSE, however, it provides consistently worse results in terms of NDPM. Still, the results provided by RBMF are the closest to those of MF-RUs model. Therefore we can conclude that the set of RUs can represent the interests of the entire set of users better than other sets of seed users and can better predict elements ranking than the benchmark model (RBMF). The MF-RUs model performance for small values of θ ($\theta = 10\%$ or $\theta = 5\%$) proves that the proposed method can be used for sparse datasets as well. Now we move to the more detailed analysis of MF-RUs model.

MF-RUs: Comparison of Filling Procedures (No Coverage Growth)

Till now, we were studying the case when seed users in MF-based models provide all the ratings on new items. However, in reality not all users from the set of seed users may be able to provide

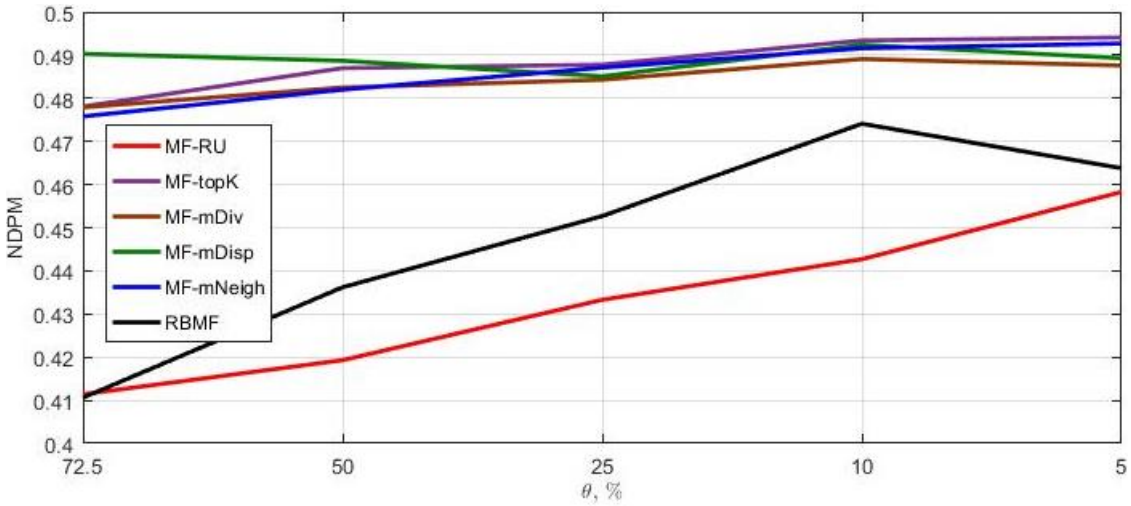

Figure 3.4: Jester: Dependence of NDPM on Percent of Known Ratings in the Input Rating Matrix (θ)

ratings on new items for different reasons, like the absence of knowledge about the item or simply non-willingness to answer. As it was proposed in the Section 3.2.3, unknown ratings can be filled with either mean values (we will use global mean *Gmean*, user mean *Umean* and user-item mean *UImean*) or ratings of other users *User* (in the case when RUs are chosen as seed users – the ratings of the next closest representative user, see Section 3.2.1).

We focus on studying different filling procedures. We compute error values of MF-RUs model depending on the percent of ratings provided by representative users for different mean-fillings and for filling with the ratings of the next closest representative user.

In order to analyse the influence of the missing ratings, as in the previous experiments, we evaluate error measures only on those new items, that have ratings from all representative users. Thus, *the coverage (COV) on the test set does not change when the threshold of required number of ratings from the representative users (γ) decreases.* Indeed, when the threshold of required number of ratings from representative users γ decreases, usually more new items can be used in the actual test set (see Figure 3.2, right part). However, in this case the actual test set is fixed and is composed of those new items, on which representative users provide 100% of ratings. The coverage growth when γ decreases will be studied in the next series of experiments. In order to simulate the absence of some ratings of representative users, we randomly delete the required number of votes for each considered new item.

The resulting evolutions of NRMSE and NDPM on different values of γ and for different filling procedures are presented in Figures 3.5 and 3.6. The lowest value of NRMSE is obtained when the Umean-filling (filling with the per-user mean rating) is used. Also, contrary to what is expected, NRMSE has a tendency to decrease for Umean and UImean filling procedures. When Gmean-filling is used, the value of NRMSE increases following the case for the User-filling, but usually stays lower than for the latter one. The decrease of the NRMSE when γ decreases shows that the rating value of the new item can be predicted as a per-user mean rating. Indeed, performing non-cold-start test for predicting new ratings with the per-user mean model (unknown ratings are estimated as the mean rating of an active user) we obtain RMSE equal to 0.2283 that is only 11% higher than RMSE of the MF model for $K = 10$ (0.2054, see the Table 3.4) and even

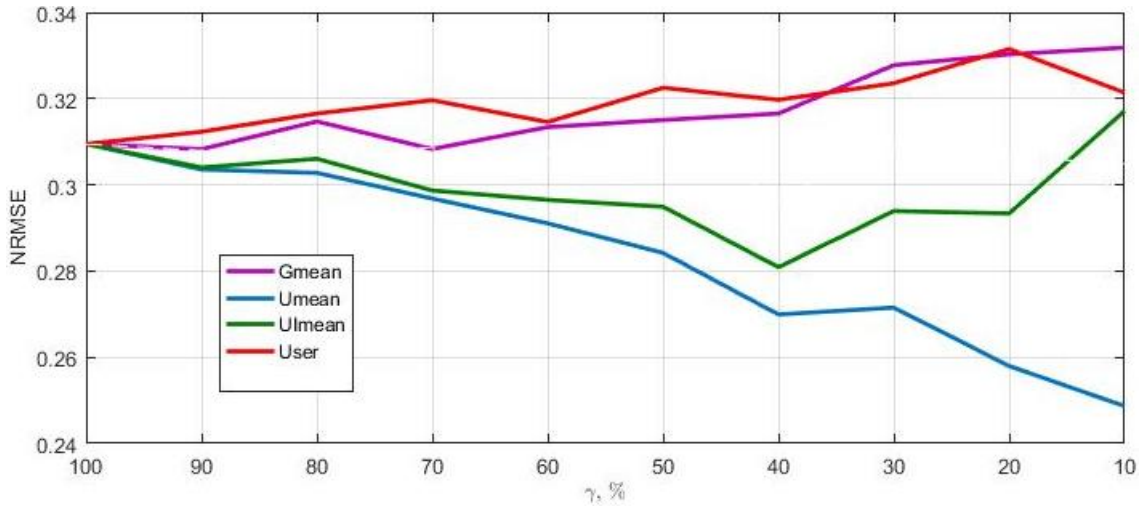

Figure 3.5: Jester: Dependence of NRMSE on the Required Number of Ratings from RUs (γ) for different filling procedures

less than RMSE of the NB model with 10 neighbours equal to 0.2382. However, despite the fact that the values of the ratings can be predicted with the mean model, this model cannot estimate the ranking of the items, as all of them are predicted to have the same rank. This shows that the NRMSE measure has a bias when mean-fillings are used. Also, as it was discussed earlier (see Section 3.3.4), the NDPM error measure has more practical meaning, so we consider it as the main evaluation criteria.

Analysing the dependence of NDPM (see Figure 3.6) we can see that, as expected, the error value grows with the decrease of γ. Also, using the ratings of real users (filling with the ratings of the next closest candidate, User-filling procedure) can significantly improve the performance, especially for the case when the value of γ is small (30%–10%). It shows that the ratings of real users have practical value and are more suitable for ranking estimation than mean-fillings.

MF-RUs: Comparison of Filling Procedures (Coverage Growth)

Finally, we now analyse the performance of the models in the case of coverage growth. In this series of experiments, the actual test set is not fixed for different values of γ. For example, if γ is set to 40%, this means that all new items that were rated by at least 40% of the representative users are analysed in the actual test set (in the previous case, the actual test set was composed of only those new items that have 100% of ratings from representative users). This naturally results in a coverage growth while γ decreases.

We compute NRMSE and NDPM, as well as the test coverage COV, for different values of γ and different filling procedures. The results are presented in Figures 3.7 and 3.8.

As in the previous case, where no coverage growth was considered, we can observe an unexpected behaviour of NRMSE: the error tends to decrease with γ, either on a certain interval (for Gmean, UImean and User-filling) or for all values of γ (Umean-filling). Also, mean-filling procedures result in a lower NRMSE than User-filling (due to the bias of the NRMSE measure discussed before). On the other hand, the analysis of the NDPM shows that the usage of ratings of real users instead of mean-fillings results in better elements ranking.

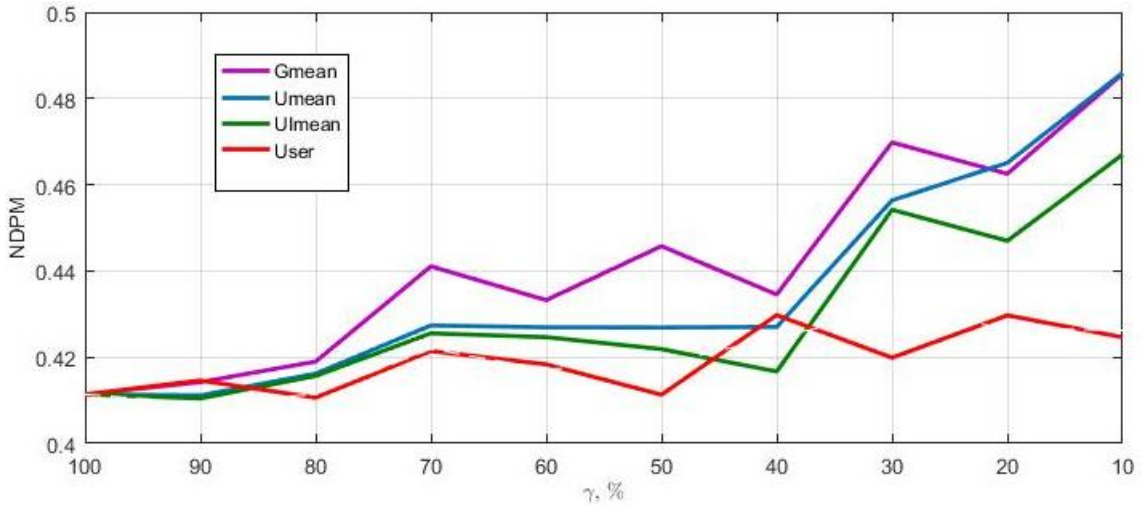

Figure 3.6: Jester: Dependence of NDPM on the Required Number of Ratings from RUs (γ) for different filling procedures

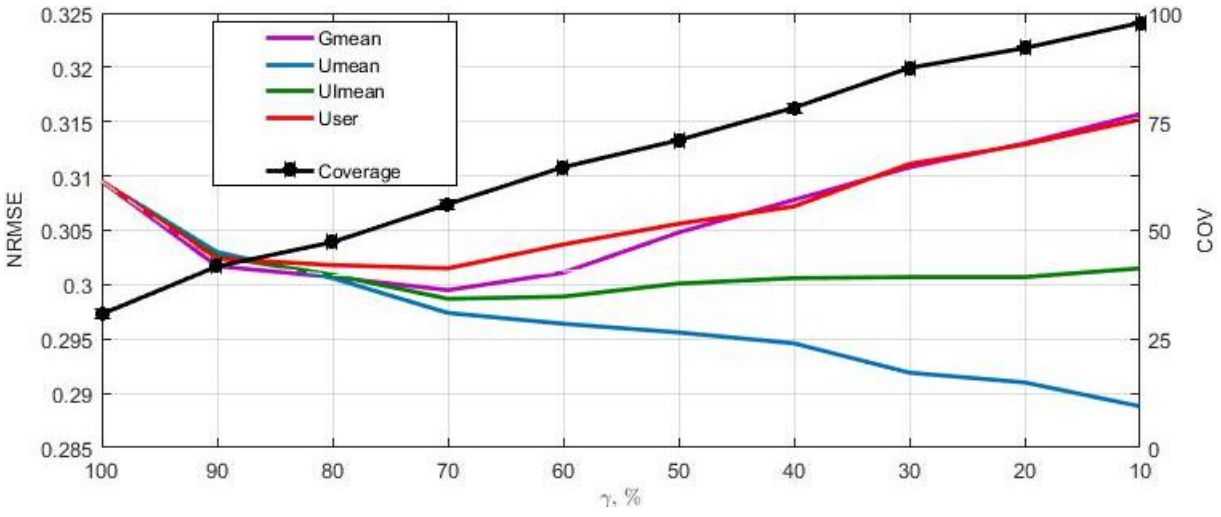

Figure 3.7: Jester: Dependence of NRMSE for Different Filling Procedures and Test Coverage (COV) on the Required Number of Ratings from RUs (γ)

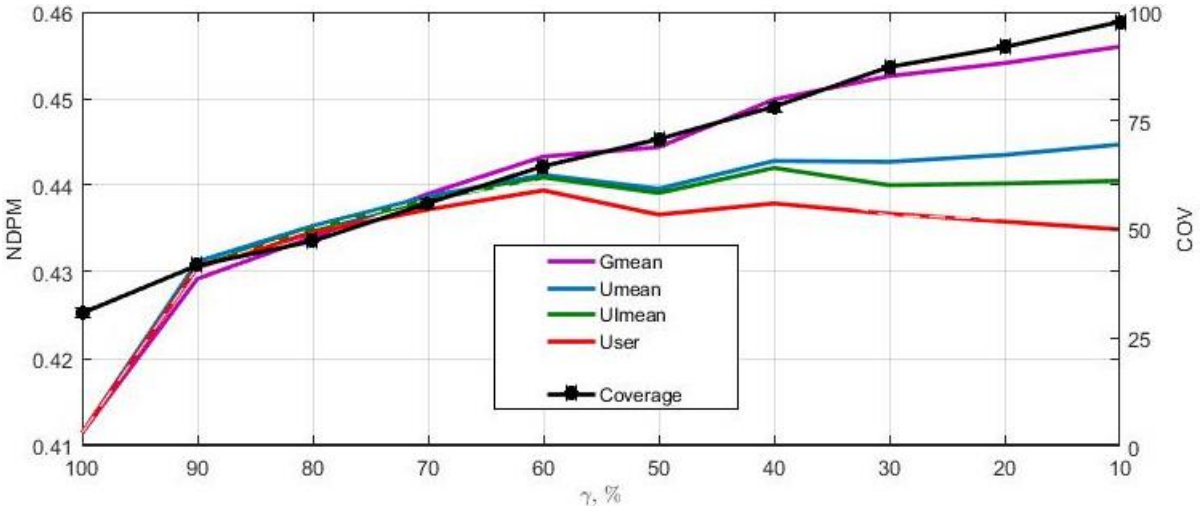

Figure 3.8: Jester: Dependence of NDPM for Different Filling Procedures and Test Coverage (COV) on the Required Number of Ratings from RUs (γ)

Moreover, in the case where ratings of other RU candidates are used instead of unknown values, we obtain noticeable improvements in terms of NDPM (compare $NDPM(\gamma = 10\%) = 0.4357$ for User-filling and $NDPM(\gamma = 10\%) = 0.4560$ for Gmean-filling) and in terms of coverage (compare $COV(\gamma = 100\%) = 31\%$ and $COV(\gamma = 10\%) = 98\%$). Thus we showed that using ratings of representative users or other closest candidates can not only improve the quality of the ranking, but also allows accurately predict ratings for more new items.

To conclude, we can say that our approach (MF-RUs) outperforms in terms of NDPM alternative methods of seed users identification as well as the benchmark RBMF approach. Also, it is not computationally complex and does not require performing additional calculations, like calculation of the correlation matrix for the identification of mDiv and mDisp sets or prior SVD decomposition, as RBMF. Additionally, when using representative users as seeds the unknown ratings from seed users can be replaced with the ratings of other closest candidates for being representative users (User-filling procedure). This also allows calculating predictions for more new items.

3.4.4 Cold-start for MovieLens dataset

In the previous subsection, we studied different aspects of the proposed approach on the Jester dataset. To confirm some of the results, we now move to the analysis of our approach on the MovieLens dataset. As this dataset contains only 6% of known ratings, none of the new items has more than 40% of ratings provided by seed users ($\gamma \leq 40\%$). Note that the previous phrase does not mean that the proposed approach cannot identify the required number of representative users, but that due to the sparsity of the dataset there is not enough ratings in the test set provided by RUs. In a real situation, as it was mentioned in [Liu *et al.* 2011], the chosen representative users will be encouraged to provide all ratings on the new items through, for example, the proposition of additional services from the side of RS. So, we can expect that most of their ratings will be known.

Thereby, in every experiment we have to use one of the filling procedures. So it is impossible

Table 3.8: MovieLens: Relative deterioration (*DET*) of NDPM in % of different models compared to MF-RUs model through different values of γ (UImean-filling is used)

γ,%	SEEDS for MF-models				RBMF
	topK	mDiv	mDisp	mNeigh	
40	8.5	6.4	8.2	8.6	4.9
30	11.0	9.6	11.0	11.3	7.3
20	7.1	6.0	7.1	8.2	3.7
10	5.5	5.7	5.0	6.6	2.3

to perform the same series of experiments for MovieLens, as it was done for Jester. Therefore, in this subsection we search an answer for only two questions: 1) does MF-RUs model remain the best among other MF-based models as well as RBMF? and 2) which filling procedure will result in better performance of MF-RUs model? As both questions concern the comparison of the models, we will use relative deterioration (*DET*, see equation (3.11)) as an evaluation metric. Experiments performed on the Jester Dataset support the statement from Section 3.3.4, that NDPM measure has more practical meaning than NRMSE. Thus in this subsection, we will focus on the evaluation of NDPM only.

Searching for the answer to the first question, we calculate the relative deterioration in terms of NDPM for different models (analysed models), compared to the MF-RUs model (base model), for different values of γ. UImean-filling procedure (filling with the mean of the user and item mean values) was used, as it resulted in the best NDPM values for the Jester Dataset among other mean-filling procedures (see Figures 3.6 and 3.8). Corresponding results are presented in Table 3.8.

A positive value of *DET* means that the MF-RUs model gives better results (lower value of NDPM). An absolute value in Table 3.8 indicates the relative deterioration in percents of the analysed model comparing to the base one (MF-RUs). For example, the value 11.0 in the line $\gamma = 30\%$ and the column *mDisp* shows that for the specified γ MF-RUs model performs 11.0% better than MF-mDisp (that is MF-RUs results in NDPM that is 11.0% less than NDPM for MF-mDisp). As all the values in the Table 3.8 are positive, we can conclude that for all values of γ the MF-RUs model results in better ranking, compared to other models. Therefore we obtained the answer on the first question. Also we can see that when γ decreases from 30% to 10% the value of *DET* also decreases. It means that when seed users provide fewer ratings, the difference between the performance of models diminishes. Indeed, for $\gamma = 10\%$, 90% of ratings are filled with UImean values, but not with the ratings of seed users, thus the models with different ensembles of seed users become more similar. Also, supporting the same conclusion for the Jester dataset, results of the RBMF are the closest to MF-RUs model results.

The next question that we raise is: which of four filling procedures (Gmean, Umean, UImean, and User) will be the best for the *MF-RUs* model. Similar to the previous case, we calculate the relative deterioration *DET* in terms of NDPM of the analysed models (MF-RUs with the mean-filling procedures) compared to the base model (MF-RUs with the User-filling). The positive value of *DET* indicates that MF-RUs with User-filling provides a lower NDPM value. Results are presented in the Table 3.9.

From the Table 3.9 we see that using User-filling procedure we can obtain better ranking (as all the values in the table are positive). Also, we can see that with the decrease of γ (percentage

Table 3.9: MovieLens, MF-RUs model: Relative deterioration of NDPM in % of mean-filling procedures compared to User-filling for different values of γ

γ,%	DET filling		
	Gmean	Umean	UImean
40	7.8	9.1	3.4
30	11.7	12.7	6.5
20	19.6	19.2	17.4
10	25.2	24.4	21.6

of known ratings provided by RUs) the gain of using User-filling increases. Indeed, the lower the value of γ, the more real information is obtained through the User-filling compared to the mean-filling procedures. This was also the case for the Jester dataset (see Figures 3.6 and 3.8: when γ decreases the gain of using User-filling increases).

To conclude, we can say that the results obtained for MovieLens dataset support the conclusions drawn from Jester: 1) among all models, MF-RUs gives the best values of NDPM and 2) using ratings of other closest users instead of mean-filling procedures improves the quality of items ranking. However, due to the sparseness of the MovieLens dataset we were not able to perform exactly the same evaluation as for the Jester dataset. In particular, we could not study the cases when either all or the majority of seed users provide ratings on new items.

3.5 Conclusions

We presented in this chapter a new approach for interpreting features of a matrix factorization model. As announced in Chapter 2, we associate features with real users of the system – *representative users*. The proposed interpretation is done completely automatically and, contrary to many state of the art approaches, requires neither human experience or interaction, like in [Zhang *et al.* 2006, Pessiot *et al.* 2006], nor external sources of information, like item reviews in [McAuley and Leskovec 2013]. Our approach works with an already existing matrix factorization model and does not alter it for making the model being interpretable. Thereby, we do not propose a new factorization technique. The proposed interpretation allows us not only interpreting the features but also explaining recommendations provided by MF models in a way similar to the NB approach (see Section 3.2.2).

As latent features represent the relations between users and items, the resulting feature-related representative users should be capable to correctly represent the interests of the whole population of users. Thereby, following [Liu *et al.* 2011] we choose to use the new item cold-start problem for the validation of the proposed interpretation. Indeed, the fact that the preferences of representative users on new items can be used to correctly estimate the preferences of other users on these items (or the fact that representative users can be used as seed users) is a reliable proof of the proposed interpretation. However, this statement has to be verified experimentally.

In Section 3.2.3 we formulate our solution for the new item cold-start problem in MF-models based on the usage of ratings of seed users. This solution, contrary to many state-of-the-art approaches, does not require content information and, thereby, is of particular interest when such information is not available.

Using two datasets (MovieLens and Jester), the evaluation of the proposed approach was performed. First of all, analysing characteristics of the set of representative users, it was shown that they tend to be composed of users with different behavioural patterns and can thus be used for representing the interests of the entire population of users.

Considering the performance on the cold-start, it was shown that using representative users as seeds results in better ratings predictions than alternative sets of seed users (such as top raters or the set of most diverse users) and the MF-RUs model provides better ranking than the baseline approach (RBMF). Also, if for some reasons the chosen representative users do not provide their ratings on new items, the next best candidates for being representative users can be successfully used. This allows not only to increase the accuracy of prediction (compared to the filling unknown ratings with some mean values, like global mean rating) but also to predict ratings for more new items. In our opinion, this ability of representative users to solve the new item cold-start problem can be considered as a proof of the validity of the proposed interpretation.

The two used datasets represent the rating behaviour of users in two different domains: ratings on jokes for Jester and ratings on movies for MovieLens. Also, ratings are given as real values in the first dataset and as discrete values in the second. Recall that neither domain-specific information nor the information about the nature of the ratings was used in our approach. Also, the results obtained for the MovieLens dataset support the corresponding results obtained for Jester. Thereby, we can conclude that the proposed solution is domain independent and will provide the results of the same quality when predicting ratings on items of different nature.

Part II

Identification of Trigger Factors

Chapter 4

State-of-the-Art: Theoretical Foundations for Trigger Factors

Contents

4.1 Identification of Trigger Factors: Next Step for Classification

As it was mentioned in the introduction, according to Gartner [Davis and Herschel 2016] it is possible to define 4 types of data analytics:

- *descriptive analytics* - describes general tendencies in the dataset (*what happened?*);
- *diagnostic analytics* - attempts to understand the nature of the found patterns and dependencies (*why did it happen?*);
- *predictive analytics* - predicts the future development (*what will happen?*);
- *prescriptive analytics* - identifies the factors that can lead the development of the system in the desired direction (*how to make it happen?*).

All four types of data analytics are important, however, nowadays prescriptive analytics becomes more and more popular due to the high demand from the business community [Evans and Lindner 2012, Basu 2013]. Research in prescriptive analytics is an emerging field [Song *et al.* 2014] and the existing papers are mostly dedicated to the development of analytical systems that support the generation of prescriptions for different case studies. As examples, we can mention advising systems aimed to help a researcher building a successful career [Song *et al.* 2014, Weber *et al.* 2014, Lee and Cho 2015], analytical systems for manufacturing [Krumeich *et al.* 2015, Gröger *et al.* 2014] and learning analytics systems aiming to help students and/or teacher [Aguilar *et al.* 2014, Miller *et al.* 2015, Sharma *et al.* 2016].

We consider that it is possible to define four phases of analytical tasks that will correspond to four types of data analytics. For example, let us consider the task of *class analysis*. In many practical cases, the analysed dataset is organised into non-intersecting classes. We can give the following examples: men and women in demographic data, defective and faultless articles in manufacturing data, successful and backward students in e-learning. A researcher can be interested in understanding the characteristics of each class: what values of attributes are more common in each of the classes; if there are similar and/or different tendencies; which attributes can be used to predict the class of a datapoint, etc. In this way, the task of *class analysis* arises.

Our view of the four phases of the *class analysis* task is presented in Figure 4.1. We consider that on the level of descriptive analytics the task of class analysis takes form of the task of *class description*. We can formulate this task as follows:

Class description: Define the patterns specific to each data class.

Essentially the task of *class description* is reduced to the separate analysis of different classes of the dataset. However, what is more interesting, is to compare the patterns found in each class and to identify those, that are similar and different. Thereby we come to the task of *class comparison*, which corresponds to diagnostic analytics and which we formulate as follows:

Class comparison: Define the patterns that are similar and different for different classes of the dataset.

We can note that the task of *class comparison* rests upon the task of *class description*. Indeed, once the patterns for each class are identified (*class description*) what is left to do is to identify those of them, that are similar and different between classes (*class comparison*). However, data analysis techniques are usually sharpened to find interesting/non-trivial patterns [Fayyad *et al.* 1996]. The patterns that are non-interesting within the scope of one data class can be useful when compared with corresponding patterns of the other class. Thereby, although *class comparison* relies directly on *class description*, it cannot be reduced to the latter one.

The next phase of *class analysis* (within the scope of predictive analytics) can be viewed as the task of *class prediction*:

Class prediction: Define the pattern for predicting the class label for a previously unseen element.

The latter task is known in the literature as the task of *classification* [Aggarwal 2014b, Duda *et al.* 2012]. It is well-studied and due to the high number of applications has many solutions [Duda *et al.* 2012, John Lu 2010], which are mainly based on the identification of interconnections between the values of the attributes of datapoints and the values of the class label. Essentially,

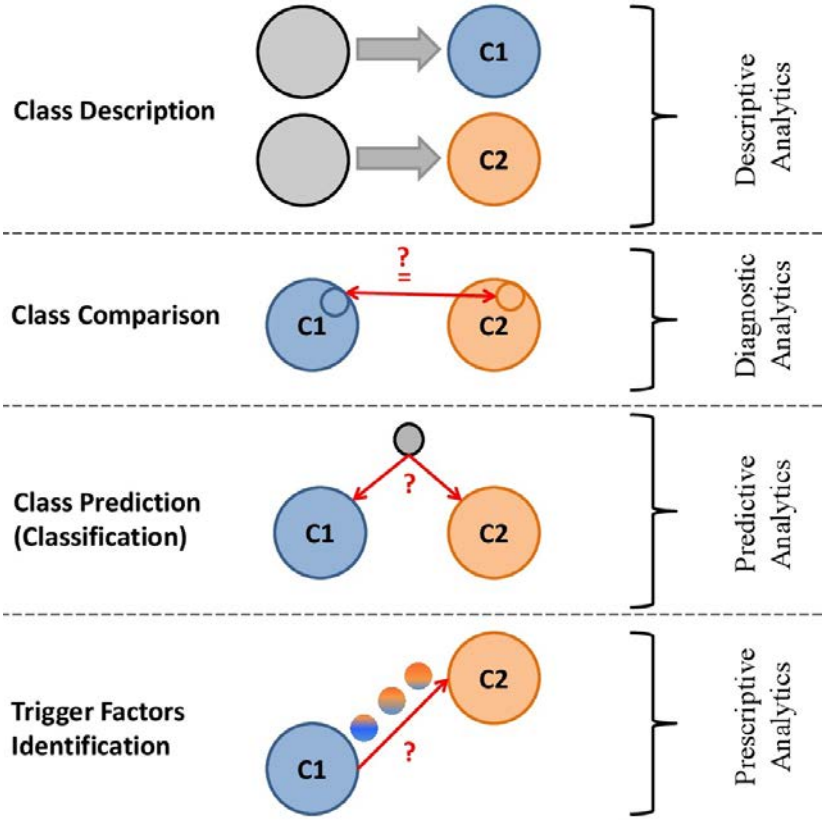

Figure 4.1: Four Phases of the Class Analysis Task

the task of *classification* can be viewed as the task of prediction in the case of discrete number of possible future states.

As in the previous case, the solution of the *class prediction* or *classification* task can be based on the solution of the *class comparison* task. Indeed, selecting those patterns that are dissimilar for different data classes, we can afterwards use them to predict the class label for the new data element. This is done basing on the fact to which of the patterns the data element corresponds. However, the data element can correspond to multiple patterns and thus many class labels can be predicted. Thereby within the *classification* task not all possible patterns are used, but usually the set of those patterns that can predict the class label with the highest accuracy.

According to the structure provided in Figure 4.1, the task of *trigger factors identification* can be considered as a direct descendant of the *classification* task. However, to the best of our knowledge, the task of *class analysis* in the frame of prescriptive analytics was not discussed before in the literature. We formulate this task as follows:

Trigger factors identification: Define the factors, that can stimulate the transition of data elements from one class to another.

It was mentioned in the introduction section that we are not aware of any technique that allows to identify automatically such trigger factors in the general case. However, as the task of *trigger factors identification* is the direct descendant of the *classification* task, we suppose that classification techniques can be used to solve it (following the mentioned above regularity that

the solution of one task can be based on the solution of the task standing one level before).

In real life, if we want to transfer elements from one class to another we often follow a simple intuition: it is required forcing the data elements to correspond to the patterns that are associated with the target class (the class to which we want to move the data elements). That is *to change the values of the attributes which determine the value of the class label from those that are typical for non-target class (classes) to those that are typical for the target class.* This idea forms the heuristics that we use in this thesis to identify trigger factors. However, to do so we need to find the patterns that reveal the dependence between the feature attributes and the target attribute for each class. This actually can be done by classification techniques.

There exists a great variety of classification techniques. Some of them are known to be interpetable, while others not [Letham *et al.* 2012]. However, trigger factors are defined as those that can stimulate the transition of data elements from one class to another. That is they should correspond to some real-life characteristics of the elements from the dataset. Thereby, we believe that interpretable classification techniques will suit the best our task. Hence, the next section is dedicated to the analysis of classification techniques with the aim to identify those that produce patterns suitable for the task of trigger factors identification (interpretable patterns).

4.2 Classification Approaches

It is possible to define two general approaches for classification: eager classification and instance-based classification [Aggarwal 2014b]. Eager classifiers [Chatterjee 2011] have two clearly defined phases: a learning phase and a prediction phase. On the learning phase, the available data elements are used to build the classification model, which is afterwards used to predict the class labels for new data elements. Instance-based classifiers [Aha *et al.* 1991] do not have a clearly defined learning phase. In this case, the predictions are calculated for a specific new instance that needs to be classified [Aggarwal 2014a]. The instance-based classification techniques belong to lazy learning techniques, as opposite to eager learning classification methods that try to build a general model before the new instance will appear [Hendrickx and Van Den Bosch 2005]. When the instance-based classifier is used, no time is required for preprocessing (as no model is built). At the same time, more computational time will be spent during the classification phase itself, as the calculations should be performed for every new instance. Also, because instance-based classifiers optimise the prediction for each instance, they can provide better results. However, eager classifiers are less sensitive to noisy data [Aggarwal 2014a]. Recall that our aim is to form classification patterns (that is a classification model) that afterwards will be used for trigger factors identification. Thus, the instance-based classifiers are of no interest to us. So we focus on eager classifiers.

Following [Aggarwal 2014b] we can outline 4 commonly used groups of eager classification techniques: probabilistic classification, support-vector machines (SVM), artificial neural networks (ANN) and rule-based classifiers. The following subsections will be dedicated to the description of the structure of the models generated by these different eager classification techniques. We start this description with the introduction of notations. Let D be a dataset defined on a set of N attributes $\{A^1, A^2, \ldots, A^N\}$. Assume that for each attribute A^j there is a set of possible values. We will refer to this set as the domain of the attribute A^j, denoted by $domain\left(A^j\right)$. Assume that K mutually exclusive classes $G_1, G_2, \ldots, G_K$ are defined on the dataset D with $D = G_1 \cup G_2 \cup \ldots \cup G_K$ and $G_i \cap G_j = \emptyset$, $\forall i \neq j$. Let us also assume that the class of a particular element is defined by the value of an attribute A^G with K possible values ($domain\left(A^G\right) = \{G_1, G_2, \ldots, G_K\}$). This attribute will be referred to as the *target attribute*,

contrary to other attributes referred to as *feature attributes* (or features). We also use the term *class label* to refer to the value of A^G for a certain element.

4.2.1 Probabilistic Classification

The main characteristic of probabilistic classifiers is the fact that they use statistical inference while building the model [Deng *et al.* 2014]. Also, they do not predict the class label of the new instance, but rather estimate the probability of its belonging to a certain class. Let us consider the Naive Bayes classifier [Murphy 2006], which is a prominent example of probabilistic classification techniques and is widely used in many applications (see, for example, [Moore and Zuev 2005, Wang *et al.* 2007]).

Assume that all feature attributes are organised into a feature vector $\vec{A_F}$ with coordinates A_*^1, A_*^2, ..., A_*^N standing for the values of corresponding feature attributes in each particular case. Using the theorem of Bayes, for a given feature vector the conditional probability $p\left(A^G = G_k|\vec{A_F}\right)$ can be rewritten in the form 4.1.

$$\begin{aligned} p\left(A^G = G_k|\vec{A_F}\right) &= p\left(A^G = G_k|A_*^1, A_*^2, \ldots, A_*^N\right) = \\ &= \frac{p\left(A^G = G_k\right) p\left(A_*^1, A_*^2, \ldots, A_*^N|A^G = G_k\right)}{p\left(A_*^1, A_*^2, \ldots, A_*^N\right)} \end{aligned} \tag{4.1}$$

In practice one is interested only in the numerator of (4.1) as the denominator does not depend on A^G, so (4.1) can be rewritten as (4.2).

$$p\left(A^G = G_k|A_*^1, A_*^2, \ldots, A_*^N\right) \sim p\left(A^G = G_k\right) p\left(A_*^1, A_*^2, \ldots, A_*^N|A^G = G_k\right) \tag{4.2}$$

Applying Bayes theorem sequentially, we can get relation (4.3).

$$\begin{aligned} &p\left(A^G = G_k|A_*^1, A_*^2, \ldots, A_*^N\right) \sim \\ &\sim p\left(A^G = G_k\right) p\left(A_*^1, A_*^2, \ldots, A_*^N|A^G = G_k\right) \sim \\ &\sim p\left(A^G = G_k\right) p\left(A_*^1|A^G = G_k\right) p\left(A_*^2, \ldots, A_*^N|A^G = G_k, A_*^1\right) \sim \\ &\sim p\left(A^G = G_k\right) p\left(A_*^1|A^G = G_k\right) p\left(A_*^2|A^G = G_k, A_*^1\right) p\left(A_*^3, \ldots, A_*^N|A^G = G_k, A_*^1, A_*^2\right) \sim \\ &\sim p\left(A^G = G_k\right) p\left(A_*^1|A^G = G_k\right) p\left(A_*^2|A^G = G_k, A_*^1\right) \cdots p\left(A_*^N|A^G = G_k, A_*^1, A_*^2, \ldots A_*^{N-1}\right) \end{aligned} \tag{4.3}$$

Now the 'naive' conditional independence assumptions can be used: assume that each feature A^n is conditionally independent of every other feature given the class $A^G = G_k$. This means, for example, that $p\left(A_*^N|A^G = G_k, A_*^1, A_*^2, \ldots A_*^{N-1}\right) = p\left(A_*^N|A^G = G_k\right)$, what leads us to the equation (4.4).

$$\begin{aligned} &p\left(A^G = G_k|A_*^1, A_*^2, \ldots, A_*^N\right) \\ &\quad\sim p\left(A^G = G_k\right) p\left(A_*^1, A_*^2, \ldots, A_*^N|A^G = G_k\right) \\ &\quad\sim p\left(A^G = G_k\right) p\left(A_*^1|A^G = G_k\right) p\left(A_*^2|A^G = G_k\right) \cdots p\left(A_*^N|A^G = G_k\right) \\ &\quad\sim p\left(A^G = G_k\right) \prod_{i=1}^{N} p\left(A_*^i|A^G = G_k\right) \end{aligned} \tag{4.4}$$

As a result under the above independence assumptions, the conditional distribution over the target attribute A^G is defined by (4.5):

$$p\left(A^G = G_k | A^1_*, A^2_*, \ldots, A^N_*\right) = \frac{1}{Z\left(A^1_*, A^2_*, \ldots A^N_*\right)} p\left(A^G = G_k\right) \prod_{n=1}^{N} p\left(A^n_* | A^G = G_k\right), \quad (4.5)$$

where $Z = \left(A^1_*, A^2_*, \ldots A^N_*\right)$ is a scaling factor dependent only on $A^1_*, A^2_*, \ldots A^N_*$, that is, a constant if the values of the feature attributes are known.

Despite the fact that Naive Bayes classifier relies on the assumption that feature attributes are independent, what is not always true, Naive Bayes performs well in many practical applications. This was proven both experimentally [Rish 2001] and theoretically [Zhang 2004].

As we can see, in the case of Naive Bayes classifier the statistical inference is related to the Bayes Theorem, which shows how to compute the conditional probability of an event dependent on other events. Other reflections can be used as the basis of statistical inference. For example, the logistic regression classifier [Dreiseitl and Ohno-Machado 2002] assumes that the probability of belonging to the class can be modelled by a logistic function.

In general, probabilistic classifiers model the class belonging probability as a probabilistic function dependent on the values of feature attributes. It is possible to derive the class-specific patterns which could be used for the trigger factors identification. However, it requires additional actions to be performed.

4.2.2 Artificial Neural Networks

The artificial neural networks (ANN) are inspired by biological neural networks and are considered by some authors as universal functional approximators [Hornik 1991, Cybenko 1989]. They consist of a set of artificial neurons connected into a network.

The structure of an ANN is characterized by the following elements [Biem 2014]:

- **Mathematical model of a neuron** that describes how that neuron processes input signals and calculates the value of an output. The model of a neuron is comprised of two functions:
  - *net value function*, which uses the parameters of the neuron (the values of the weights associated with each input channel) to summarize the input data and form a *net value*;
  - *activation function* that transforms the calculated net value into the output value of the neuron.
- **Structure** or topology of the network, which specifies the interconnections between neurons.
- **Learning algorithm** that is used to update the values of the weights.

The net value and activation functions can be of different types, but typically they are represented as a weighted sum/distance/kernel and linear/step/sigmoid functions respectively [Biem 2014]. The structure of a network represents the graph of units. Usually, the network consists of multiple layers, among which we can define an input layer responsible for bringing information into the network, the output layer, which provides the network outputs, and hidden layers that perform the transformation of the information (see Figure 4.2). ANN may contain feedbacks. The training of the network consists in updating the values of the weights associated

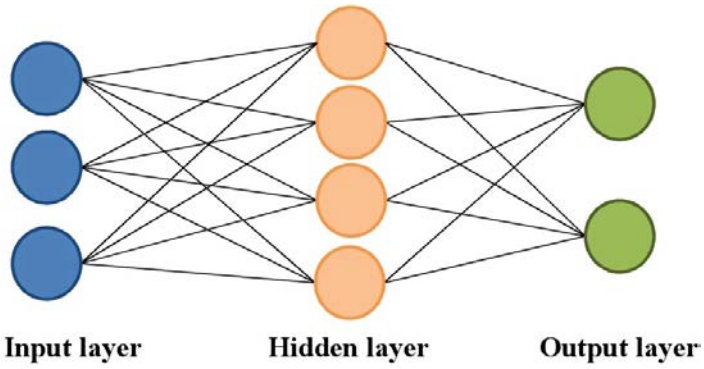

Figure 4.2: Example of an Artificial Neural Network

with each unit with the aim to minimize the classification error. Different classes of networks usually provide different learning schemes that are suited for their typology.

The ANN models are considered to work as black-boxes [Benítez *et al.* 1997], that is the model is usually non-interpretable and is difficult to understand. Nowadays deep neural networks [Arisoy *et al.* 2012] (those having multiple intermediate levels) attract more and more attention [Schmidhuber 2015]. However, the deeper the network is, the more difficult it becomes to follow the learning process and thus to understand the resulting model.

4.2.3 Support Vector Machines

The idea of support vector machines (SVM) is originally related to linear classifiers [Wang and Lin 2014, Shmilovici 2005], that is classifiers defined as linear functions. Assume that a dataset in the N-dimensional feature space can be separated into 2 classes by an $N-1$ dimensional hyperplane (example for $N=2$ is given in Figure 4.3).

It is evident that in this case an infinite number of hyperplanes can be used as classifiers. However, the most reasonable choice corresponds to that hyperplane, which has the largest distance from the closest instances of both classes (maximum margin). In such a way the classifier is less prone to misclassification error (see the line in red in Figure 4.3). These instances (those data points that are the closest to the chosen hyperplane, or those, that are lying on the margin) are called *support vectors*.

The canonical equation of a hyperplane can be written in the form (4.6).

$$\theta' \vec{A_F} + \theta_0 = 0 \tag{4.6}$$

The coefficients θ' and θ_0 can be rescaled in such a way that the support vectors, which are given by their support vectors $\vec{A_{F_1}}$ and $\vec{A_{F_0}}$, will satisfy equalities (4.7).

$$\begin{aligned} \theta' \vec{A_{F_1}} + \theta_0 &= 1, for A^G\left(\vec{A_{F_1}}\right) = G_1 \\ \theta' \vec{A_{F_0}} + \theta_0 &= -1, for A^G\left(\vec{A_{F_0}}\right) = G_0 \end{aligned} \tag{4.7}$$

Then the distance between the both support vectors is expressed by the equation (4.8).

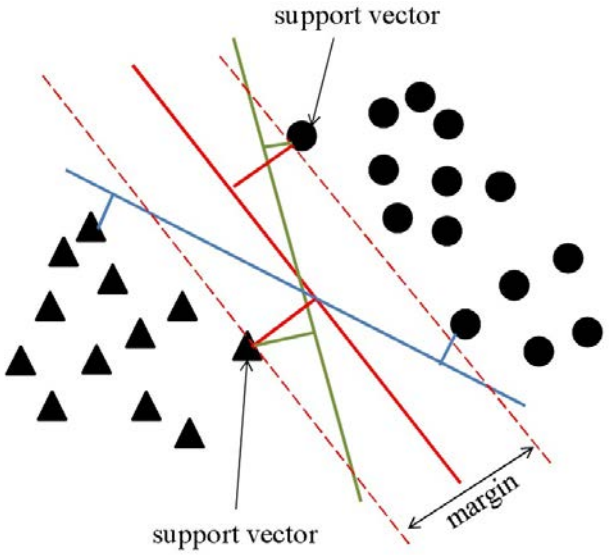

Figure 4.3: SVM: Examples of Linear Classifiers for 2-Dimensional Feature Space (maximum-margin classifier is in red)

$$dist\left(\vec{A_{F_1}} - \vec{A_{F_0}}\right) = \frac{2}{||\theta'||} \tag{4.8}$$

As it was mentioned, among all the hyperplanes, we want to choose the one, that has the maximum margin, that is we want to maximise the value from the equation (4.8) or obtain the optimal solution of the conditioned optimisation problem (4.9) (assume $k = -1$ for G_0 and $k = 1$ for G_1). This optimisation problem can be solved by the method of Lagrange multipliers [Shmilovici 2005].

$$\begin{aligned} &\min_{\{\theta',\theta_0\}} ||\theta'||^2 \\ &k_i\left(\theta' \vec{A_{F_i}} + \theta_0\right) \geq 1 \, (\forall i, 1 \leq i \leq M) \end{aligned} \tag{4.9}$$

The SVM classifier is then represented by a function of the constructed hyperplane (see equation (4.6)). For each new instance that needs to be classified the value of the hyperplane function is calculated with the instance's feature vector being used as a vector of parameters. Depending on the sign of the calculated value the instance is predicted to be on one of two sides of a hyperplane and in this way the class label is predicted. Obviously, not all datasets can be linearly separated. In this case, SVM method can be still used with the introduction of soft margins or making the hyperplane fitting the feature space through using the kernel tricks [Shmilovici 2005, Wang and Lin 2014], which map the space of feature attributes on a set of latent features. The space of latent features may not be always interpretable.

4.2.4 Rule-based Classification

Association rules learning is a popular technique in data mining [Kotsiantis and Kanellopoulos 2006]. Let an *item* stand for any pair $\{attribute, value\}$ and an itemset stand for a set X of

items. An association rule [Agrawal *et al.* 1993] is an induction rule of the form $X \rightarrow Y$, where X and Y are itemsets and $X \cap Y = \emptyset$. X is the left-hand side (LHS) of the rule also called the antecedent, and Y is its right-hand side (RHS), or consequent.

If we restrict the consequent of an association rule to be composed of only the class attribute A^G, then such types of rules are called classification rules [Agrawal *et al.* 1994] or classification association rules [Ma 1998] (we will use the first term). Such types of rules can be naturally used for classification purposes.

A rule-based classifier is defined as a list of ordered or non-ordered classification rules [Li and Liu 2014], which are used to identify the class label of a new element (see our example in Figure 4.4). If an element corresponds to the pattern in the antecedent of an association rule, then it is said that the rule is fired by this instance. In the case of an ordered list, the rules are checked following the order in the list and the first fired rule identifies the class of the analysed instance. When using a non-ordered list of rules, in the case when multiple rules are fired by the same data instance (with possibly different predicted class labels) it is impossible to predict the class label, as none of the rules is considered as more reliable than others. However, in this case, the fired rules can be used to predict the class label via the process of 'voting'. The list of rules can also contain a 'default rule', that assigns an instance to the default class if no other rules are fired.

1: $\{Age \leq 40\} \& \{Education = School\} \Rightarrow \textbf{Non-eligible}$

2: $\{Age \leq 40\} \& \{Education = College\} \& \{OwnVehicle = No\} \Rightarrow \textbf{Non-eligible}$

3: $\{Age \leq 40\} \& \{Education = College\} \& \{OwnVehicle = Yes\} \Rightarrow \textbf{Eligible}$

4: $\{Age \leq 40\} \& \{Education = University\} \Rightarrow \textbf{Eligible}$

5: $\{Age > 40\} \& \{HouseOwned = no\} \& \{Income \leq 2000\} \Rightarrow \textbf{Non-eligible}$

6: $\{Age > 40\} \& \{HouseOwned = no\} \& \{Income > 2000\} \Rightarrow \textbf{Eligible}$

7: $\{Age > 40\} \& \{HouseOwned = yes\} \Rightarrow \textbf{Eligible}$

Figure 4.4: Example of a Rule-based Classifier for the Task of Credit Eligibility Identification

Following the idea formulated in [Aggarwal 2014b], we consider decision trees [Lee *et al.* 2014, Kohavi and Quinlan 2002] as a special case of rule-based classifiers, despite the fact that different reasoning procedures are used to construct both models (see for example C4.5 algorithm for decision trees construction [Quinlan 2014] and the Apriori algorithm for association rules mining [Agrawal *et al.* 1994]). Indeed, let us consider an example of a decision tree classifier given in Figure 4.5, which is equivalent to the rule-based classifier given in Figure 4.4. Decision trees perform hierarchical partition of the input dataset on subsets until the majority of the elements in the resulting subset belongs to the same class. The root of the decision tree typically covers all the input set, the nodes of the tree correspond to the partition criteria, and its leafs correspond to the resulting class labels. The classification is performed by following from the root of the tree to one of the leafs according to the partition criteria presented in the nodes. Each path in the decision tree can be considered as a rule. In a more general case, rule-based classifiers as opposite to decision trees do not assume the presence of the hierarchical structure in the dataset and thus the rules can overlap.

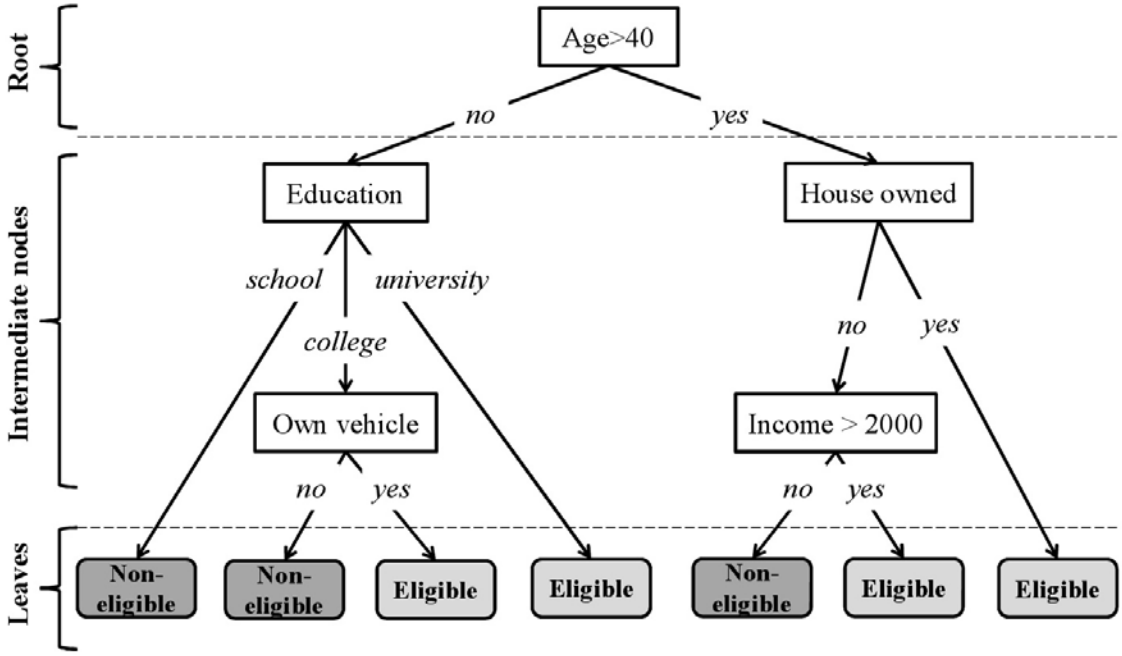

Figure 4.5: Example of a Decision Tree Classifier for the Task of Credit Eligibility Identification

As it is seen from the examples given in Figures 4.4 and 4.5, the rule-based classification models are formed of patterns that are specific to each data class and are highly interpretable. In fact, a classification rule shows which values and of which attributes (or their combination) define the belonging of an instance to the corresponding class.

4.2.5 Discussion

Now we proceed to the analysis of the eager classification techniques presented above, with the aim to identify those, that can be used for the task of trigger factors identification. Recall that in Section 4.1 we suggested that the algorithms that identify interpretable patterns specific to each data class will suit better.

The first two techniques presented in this section model either probabilistic (probabilistic classifiers) or non-probabilistic (ANN) functional dependencies between the values of the feature attributes and the class label. However, none of them is interpretable in a straightforward way. Support vector machines (SVM) in the general case can be considered as classifiers in the space of latent features. Indeed, when the data cannot be separated into classes with a linear function (a property, which cannot be guaranteed), SVM maps the data on a space of latent features using kernel tricks. The structure of the latent space usually does not correspond to the structure of the original features space. Thus the obtained model cannot be easily interpreted either.

As opposite to the three mentioned above approaches, rule-based classifiers essentially form sets of patterns specific for each data class. These patterns show which combinations of feature attribute values are more common in each class, and thus can be used for predicting the class label of new data elements. Rule-based models are also intuitively understandable. Thereby, we choose this approach as the foundation for identification of trigger factors. Among the rule-based classifiers, we can distinguish subclass of decision trees, which assume the presence of hierarchical

structure in the dataset. We decide not to restrict ourselves by using this subclass of rule-based classifiers as the presence of the hierarchical structure may not always be the case.

In general, there are two strategies in building rule-based classifiers [Li and Liu 2014]. Within the frame of the first strategy, the minimum set of classification rules required to cover all training data instances is built (*rule induction*). The second strategy consists of searching for all possible classification rules in the dataset and then choosing the subset of rules that will be used for building the classifier (*classification based on associations*). Recall, we assume that trigger factors can be identified via the comparison of the class-specific patterns. However, not all patterns describing different classes can be comparable (for example, they can consist of different feature attributes). Thereby, we decide to use the more general second approach, as using it we have more chances to find comparable patterns for different classes. We aim to investigate the possibility of using *rule induction* for trigger factors identification in future.

4.3 Class-specific Association Patterns

This section is dedicated to the description of association patterns that are used for the identification of differences between classes. We start with the presentation of metrics used for evaluating the quality of association rules. After that, we discuss related techniques: contrast mining and treatment learning. Next, we describe the rule mining algorithm which we choose to use in our work. We finish this section with a brief discussion of the rules redundancy problem and its possible solutions.

4.3.1 Evaluating Quality of Rules

Association rules are usually evaluated using measures such as support and confidence [Lenca *et al.* 2008]. Let us denote by $supp_D(X)$ the support of the itemset X in the dataset D. The support of an itemset X is calculated using the formula (4.10).

$$supp_D(X) = \frac{count_D(X)}{|D|}, \tag{4.10}$$

where $count_D(X)$ is the number of elements in D containing X and $|D|$ is the total number of elements in D. The support of the rule $X \to Y$ in D is calculated by formula (4.11) and its confidence by formula (4.12). The support of the rule shows the proportion of the elements which can be covered by the rule and its confidence – how strong is the association between the antecedent X and the consequent Y. The rule with support/confidence equal or greater than the user-specified threshold $minSupp/minConf$ value is said to be *frequent*/*confident*.

$$supp_D(X \to Y) = supp_D(X \cup Y), \tag{4.11}$$

$$conf_D(X \to Y) = \frac{supp_D(X \cup Y)}{supp_D(X)} = \frac{count_D(X \cup Y)}{count_D(X)} \tag{4.12}$$

From the probabilistic point of view the support of an itemset (or rule) corresponds to the probability of its appearance in the dataset, that is $supp_D(X) = p(X)$. The confidence of the rule $X \to Y$ is in fact the conditional probability of the appearance of Y if X appears, that is $conf_D(X \to Y) = p(Y|X)$.

Many of the existing association rules mining algorithms search for large and confident rules [Lenca *et al.* 2008, Kotsiantis and Kanellopoulos 2006]. Indeed, non-confident rules have no

practical meaning and those with the low support value can be too rare. However, the combination of these two evaluation measures is not always enough to ensure the quality of the rules [Brin *et al.* 1997, Tan *et al.* 2004]. For example, assume that the purchase of the bread and fruits is completely unrelated and the bread is bought in 70% of purchase transactions. As the purchase of fruits does not depend on the purchase of bread, the bread is bought as well in 70% of cases when the fruits are bought. Thereby, the confidence of the rule $fruits \rightarrow bread$ is equal to 0.7 despite the uselessness of this rule.

Many alternative quality evaluation measures were proposed in the literature [Geng and Hamilton 2006, Bhargava and Shukla 2016]. Nevertheless, none of them can be considered as universal because each of the measures reveals the specific characteristics of the rules and can be more or less important depending on the application domain and/or on the task being solved [Lenca *et al.* 2008]. As an example, let us now have a look at two alternative evaluation measures: lift and conviction.

The measure of lift defined by formula (4.13) is considered to be the measure of independence of the antecedent and the consequent (or the measure of their random co-occurrence). Indeed, if the appearances of X and Y are two independent events, then $p(X \cup Y) = p(X)\,p(Y)$ and the value of lift is equal to 1. A value greater than 1 shows that Y appears more often under the condition of the appearance of X, as compared to the case when X did not appear. That is the appearance of X has the positive effect on the appearance of Y. Contrary, if the value of lift is below 1, the appearance of X has a negative effect on the appearance of Y. Obviously, reliable rules should have the value of lift greater than 1.

$$lift_D(X \rightarrow Y) = \frac{p(X \cup Y)}{p(X)p(Y)} = \frac{supp_D(X \cup Y)}{supp_D X supp_D Y} = \frac{conf_D(X \rightarrow Y)}{supp_D(Y)} \tag{4.13}$$

Conviction is used to estimate the direction of the rule. If we assume that the presence of X implies Y, then the presence of X should not imply the absence of Y (event $not(Y)$). This reasoning is revealed in the equation (4.14), which defines how the value of conviction is calculated.

$$\begin{aligned} conv_D(X \rightarrow Y) &= \frac{p(X)p(not(Y))}{p(X \cup not(Y))} = \frac{p(X)(1-p(Y))}{p(X) - p(X \cup Y)} = \\ &= \frac{1-p(Y)}{1 - \frac{p(X \cup Y)}{p(X)}} = \frac{1 - supp_D(Y)}{1 - conf_D(X \rightarrow Y)} \end{aligned} \tag{4.14}$$

Using similar reasoning as for lift, we can say that if the value of conviction is equal to 1, then events X and $not(Y)$ are independent. If $conv(X \rightarrow Y) < 1$, then the presence of X has positive effect on the presence of $not(Y)$. Finally if $conv(X \rightarrow Y) > 1$ presence of X has the negative effect. Thereby conviction will be in favour of the discovered rule if its value is above 1.

4.3.2 Supervised Descriptive Rule Induction

Supervised descriptive rule induction (SDRI) is the process of inducing a set of comprehensible rules in the classification rule form from class-labeled data [Novak 2009]. The paradigm of SDRI was introduced with the aim to unify three different research directions: contrast set mining, emerging pattern mining and subgroup discovery which at that time developed independently of one another, had different learning algorithms and were used in practical applications.

Contrast set mining searches for the conjunctions of attribute-value pairs (itemsets) whose support in D differs meaningfully across classes [Bay and Pazzani 1999] (see equation (4.15)).

$$max_{i,j} \left| supp_{G_i}(X) - supp_{G_j}(X) \right| \geq \delta, \tag{4.15}$$

where X is a candidate contrast set and δ is a user-defined parameter. Contrast sets are designed in a way to show the differences between classes of the dataset (see equation (4.15)) such as differences between bachelor and PhD holders or freshman students of different years [Bay and Pazzani 2001]. A specialised algorithm STUCCO [Bay and Pazzani 2001] based on the statistical hypothesis testing was proposed for mining contrast sets. Its ability to search for 'surprising' contrast sets was tested on different datasets: Mushroom dataset[8], Adult Census dataset[9], Integrated public use microdata series (IPUMS)[10] and Admissions Data of the University of California, Irvine.

According to [Novak *et al.* 2009b], emerging patterns mining aims to discover itemsets whose support increases significantly from one class to another. The *GrowthRate* measure (see equation (4.16)) is used to evaluate this increase. Given P (with $P > 1$) a growth rate threshold, if $GrowthRate(X, G_1, G_2) \geq P$, then the itemset X is said to be a P-emerging pattern from G_2 to G_1 [Dong and Li 1999].

$$GrowthRate(X, G_1, G_2) = \frac{supp_{G_1}(X)}{supp_{G_2}(X)} \tag{4.16}$$

Emerging patterns were proposed as a tool for discovering 'emerging trends or useful contrasts' together with a mining algorithm based on border manipulation in [Dong and Li 1999]. This type of pattern was used for building a classification algorithm [Dong *et al.* 1999] and a clustering quality index [Liu and Dong 2009]. Also some variations of emerging patterns can be found in the literature: jumping emerging patters [Bailey *et al.* 2002] (growth rate is infinite $GrowthRate(X, G_1, G_2) = \infty$) and disjunctive emerging patterns [Loekito and Bailey 2006] (allow disjunctions as well as conjunctions in the itemsets).

Finally, subgroups discovery [Wrobel 1997] searches for as large as possible subgroups that have unusual statistical characteristics of a target attribute value distribution. Several heuristics are used for subgroups evaluation, for example *weighted relative accuracy* $WRAcc$ [Lavrač *et al.* 2004, Novak *et al.* 2009b] (see equation (4.17)).

$$WRAcc = \frac{p+n}{P+N} \left(\frac{p}{p+n} - \frac{P}{P+N} \right) \tag{4.17}$$

where p - is the true-positive rate of the classification rule, n - false-positive rate, P - total number of true positive and false negative predictions by the rule, and N - total number of false positive and true negative predictions. Subgroup discovery found its application as a knowledge-discovery techniques in many tasks such as heart diseases group detection [Gamberger *et al.* 2003], spam identification [Atzmueller *et al.* 2009], social bookmarking [Atzmueller *et al.* 2011] etc. Also several approaches were proposed for performing subgroup mining (see [Carmona *et al.* 2014] and [Atzmueller 2015]).

As it was shown in [Novak *et al.* 2009b], these three patterns namely *contrast sets*, *emerging patterns* and *subgroups*, have compatible learning goals and heuristics, and the mining algorithm of one pattern can be used to extract another type of pattern (see for example [Novak *et al.* 2009a], where subgroup discovery technique was used to mine contrast sets). That is they solve similar tasks and use similar evaluation measures. We will refer to the group of patterns

[8] https://archive.ics.uci.edu/ml/datasets/Mushroom
[9] https://archive.ics.uci.edu/ml/datasets/Adult
[10] https://www.ipums.org/

mined within the supervised descriptive rule induction paradigm as *contrast patterns* and to the process of constructing these patterns as *contrast mining* or *contrast data mining* following [Ramamohanarao *et al.* 2005, Liu and Dong 2009] and [Dong and Bailey 2012] respectively.

Although some works suggested that contrast patterns differ meaningfully from itemsets and thus association rules [Ramamohanarao *et al.* 2005, Bay and Pazzani 1999], it was shown in [Webb *et al.* 2003] that the straightforward application of an existing association rules discovery algorithm can be successfully used to mine contrast patterns. That is, the association rules discovery algorithm results in the same set of contrast patterns as the technique specially designed for contrast patterns mining. Moreover, below we provide the formal proof that the two concepts: the itemset forming the antecedent of classification rule and contrast pattern, are equivalent under certain conditions. To the best of our knowledge no analogous proofs were presented in the literature before.

Assume that two classes G_1 and G_2 are defined on the dataset D (we will restrict ourselves to the case of two classes as the notion of contrast is always defined between two parts of the dataset). Suppose we have a classification rule $X \to A^G = G1$ with $conf\left(X \to A^G = G1\right) = \alpha$. As there are only two classes defined on the dataset, the confidence of the rule $X \to A^G = G2$ is $conf\left(X \to A^G = G2\right) = 1 - \alpha$. Also note that the $count_{G_k}(X) = count_D\left(X \to A^G = G_k\right)$. Let us now estimate the value of the growth rate of the itemset X from the class G_2 to the class G_1 (see equation (4.18)).

$$\begin{aligned} GrowthRate(X, G_1, G_2) &= \frac{supp_{G_1}(X)}{supp_{G_2}(X)} = \frac{count_{G_1}(X)/|G_1|}{count_{G_2}(X)/|G_2|} = \\ &= \frac{|G_2|}{|G_1|}\frac{count_{G_1}(X)/count_D(X)}{count_{G_2}(X)/count_D(X)} = \left[\gamma \equiv \frac{|G_2|}{|G_1|}\right] = \\ &= \gamma\frac{conf\left(X \to A^G = G_1\right)}{conf\left(X \to A^G = G_2\right)} = \gamma\frac{\alpha}{1-\alpha}, \end{aligned} \tag{4.18}$$

where γ stands for the ratio of the sizes of the two classes $\frac{|G_2|}{|G_1|}$.

From equation (4.18) we can get the relation between the value of the confidence of the rule α and the value of the growth rate ρ for every value of γ (see equation (4.19)).

$$\rho = \gamma\frac{\alpha}{1-\alpha} \tag{4.19}$$

Equation (4.19) shows that for every given value of γ we can fix the value of the rule confidence threshold $minConf$ in such a way, that the antecedent of any rule will form a contrast pattern from G_2 to G_1 with a desirable value of the growth rate threshold P. For example, if we want the growth rate to be above 1, then the value of $minConf$ should satisfy the inequality given in the equation (4.20). Also for every value of the confidence threshold $minConf$ we can choose the value of P in such a way, that every P-emerging pattern will correspond to the classification rule with the value of confidence superior or equal to $minConf$.

$$\alpha > \frac{1}{1+\gamma} \tag{4.20}$$

In general, the equation (4.19) provides a relation between contrast patterns and classification rules and shows that mining one type of the pattern is equivalent to mining the other type

if the equality in equation (4.19) holds. This explains both the possibility to use classical association rules mining algorithms for mining contrast patterns and the successful implementation of contrast patterns for solving the tasks of classification [Dong *et al.* 1999, Ramamohanarao *et al.* 2005].

4.3.3 Treatment Learning

Suppose that each of the K classes defined on the dataset D is associated with a numeric score $S(G_k)$, and the value of the latter one represents the degree of importance of the class within the specified application task. Assume, we want to change the per-class distribution of the elements in the dataset D in such a way that more elements will belong to important classes (those, having large values of the score $S(G_k)$), *i.e.* to perform 'treatment' of the data elements. The authors of [Hu 2003] propose *treatment learning*: an approach designed to identify factors that can 'treat' the data elements.

Formally, treatment learning is formulated as the task of the identification of those itemsets, that result in a large value of the treatment lift given in equation (4.21).

$$treatment\_lift = \frac{worth(D_X)}{worth(D)}, \tag{4.21}$$

where D_X stands for the projection of the dataset D on the itemset X (the subset of all elements from D that contain X) and $worth(D) = \sum_{k=1}^{K} S(G_k)\, p(G_k)$ or $worth(D) = \sum_{k=1}^{K} S(G_k)\, supp_D\left(A^G = G_k\right)$. Analysing the formula given in equation (4.21), we can see that the value of the treatment lift is superior to 1 if the distribution of elements among the classes in D_X is more favourable towards the classes with highest values of the score $S(G_k)$ then the same distribution in the original dataset D. And the more favourable the value of distribution is, the larger is the value of the treatment lift. The corresponding itemset X is referred to as *treatment* and the subset of elements containing this itemset D_X is composed of elements belonging primary to the desired class (classes).

The main idea of treatment learning relies on the assumption that in the dataset there is a small number of *funnel* feature attributes, that is attributes which actually affect the per-class distribution of the elements [Gunnalan *et al.* 2003, Menzies *et al.* 2003]. This assumption also forms the foundation of the proposed treatment learner algorithm TAR [Hu 2003] and its variations [Gunnalan *et al.* 2003, Menzies *et al.* 2003], which in such a way are designed to identify treatments of a small size. Treatment learning can also be seen as a feature subset selection method, that seeks for the minimal set of features that favour the distribution of the data elements towards the desired classes (those classes, that have higher score values) [Gunnalan *et al.* 2003]. From the data mining perspective, a treatment learner is a contrast set learner with weighted classes [Hu 2003].

Treatments can be considered as trigger factors. Indeed, the itemsets resulting after treatment learning show which values and of which attributes can change the per-class distribution of the elements of the dataset, that is can potentially stimulate the transition of the elements from one class to another. However, the basic assumption of treatment learning is that *there are a small number of funnel features*, though being true in many cases [Holte 1993], cannot be guaranteed in all application tasks. Also, we suppose that the funnel feature attributes can be different for different subgroups of the dataset. This makes them difficult to be found using treatment learning techniques.

4.3.4 Mining Association Rules

The problem of association rules mining was introduced by Agrawal et al. in [Agrawal *et al.* 1993] as *the problem of mining all association rules that satisfy the user-specified minimum support minSupp and minimum confidence minConf values.* This task is usually divided in 2 steps [Kotsiantis and Kanellopoulos 2006]: 1) constructing all frequent itemsets and 2) forming confident rules of these itemsets.

The first task is essentially reduced to checking the frequency property for every possible itemset in the dataset. However, the search space can be pruned basing on the downward-closure property: *all subsets of the frequent itemset are themselves frequent itemsets* [Agrawal *et al.* 1993]. It means that every non-frequent itemset cannot be a part of any larger frequent itemset. The search strategy through the set of items can be done in depth-first or width-first manners [Zaki *et al.* 1997]. The width-first strategy forms the basis of the Apriori algorithm proposed by Agrawal and Srikant [Agrawal *et al.* 1994]. This algorithm is very popular and proved its efficiency in a great variety of applications [Wang *et al.* 2015, Guo *et al.* 2014]. Basing on this, we choose the Apriori algorithm for constructing the classification association patterns in our work.

Let us now have a look on the procedure of constructing frequent itemsets used in the Apriori algorithm (see Algorithm 4). It is assumed that all items in the dataset D are sorted in the lexicographical order with the operator '$<$' defining the order of the items. We call an itemset containing exactly p attributes with corresponding values as a p-itemset and denote by L_p a set of all p-itemsets. The procedure *constructFrequentItemsets* starts with the initialisation of L_1 with the set of all possible combinations of attributes and their values (see line 2). After that, all non-frequent itemsets are excluded from L_1. Next in the cycle, for every p starting from 1 the set of all frequent $(p+1)$-itemsets is constructed. This is done by generating all candidate $(p+1)$-itemsets from the set L_p and then filtering them according to the *minSup* threshold. This procedure is repeated until the newly generated set is empty. The set of all frequent itemsets of different length is returned as the result of the *constructFrequentItemsets* procedure in the variable *AllFrequentItemsets*.

Algorithm 4 Apriori algorithm: constructing frequent itemsets

```
1:  procedure constructFrequentItemsets(D, minSup)
2:      L_1 = 1-itemsets
3:      L_1 = chooseFrequent(L_1, minSup)
4:      p = 1
5:      AllFrequentItemsets = L_1
6:      while L_p ≠ ∅ do
7:          L_{p+1} = genCandidate(L_p)
8:          L_{p+1} = chooseFrequent(L_{p+1}, minSup)
9:          AllFrequentItemsets = AllFrequentItemsets ∪ L_{p+1}
10:         p = p + 1
11:     end while
12:     return AllFrequentItemsets
13: end procedure
```

Set L_{p+1} is constructed through the self-joining of L_p according to the procedure presented in Algorithm 5. We can see that the new $(p+1)$-itemset is constructed through the union of two p-itemsets that have the same first $p-1$ items and different items on the position p. The

constructed $(p+1)$-itemsets are filtered basing on the criteria of frequency of its p-itemsets (see line 10): if any of the p-itemsets does not belong to L_p, then using the downward closure property we can conclude that the new generated itemset c_{p+1} is not frequent either.

Algorithm 5 Apriori: generate candidates

```
1:  procedure genCandidate(L_p)
2:      L_{p+1} = ∅
3:      for i = 1 : (size(L_p) − 1) do
4:          r = L_p[i]
5:          for j = i + 1 : size(L_p) do
6:              q = L_p[j]
7:              if r.item_1 = q.item_1 & ... & r.item_{p−1} = q.item_{p−1} & r.item_p < q.item_p then
8:                  c_{p+1} = r ∪ q.item_p
9:                  if all p-subsets of c_{p+1} are in L_p then
10:                     L_{p+1} = L_{p+1} ∪ c_{p+1}
11:                 end if
12:             end if
13:         end for
14:     end for
15:     return L_{p+1}
16: end procedure
```

After that, all possible confident association rules are constructed by exploiting the set of generated frequent itemsets. Suppose we have a frequent itemset c and its non-empty subset β. Then an association rule is constructed according to the equation (4.22). The confidence of the rule is calculated using formula (4.11).

$$(c - \beta) \rightarrow \beta \tag{4.22}$$

The Apriori algorithm can be used for the identification of classification rules through post-processing of the obtained association rules: we choose only those rules, whose consequent is composed of only one target attribute. However, the computational costs can be reduced if the search of classification rules is incorporated into the Apriori algorithm. Such an algorithm (CAR-Apriori) was proposed in [Ma 1998]. This algorithm is based on the introduction of a new concept: a ruleitem[11]. A ruleitem is the construction of the form (4.23), containing a *condset* which is essentially an itemset, and a specified class label.

$$< condset, A^G = G_k > \tag{4.23}$$

The support of the ruleitem is defined as the ratio of the number of elements containing the specified condset and belonging to the class G_k to the number of all elements in the dataset. Each ruleitem of the form (4.23) essentially represents a rule of the form (4.24) with the support equal to the support of the ruleitem and the confidence calculated as a fraction of the support of the ruleitem to the support of the corresponding condset.

$$condset \rightarrow A^G = G_k \tag{4.24}$$

[11] Ruleitem is one word. This term is introduced like this in the literature.

The procedure of mining CARs is similar to the Apriori algorithm with the difference that instead of frequent itemsets frequent ruleitems are mined and the constructed rules always have the form similar to (4.24). Also the classification rules are formed directly from the ruleitems. Both Apriori and CAR-Apriori result in the same set of classification rules. However, the incorporation of class-information that is done in the CAR-Apriori algorithm allows to speed-up the process.

4.3.5 Redundancy Between Association Rules: Possible Solutions

One of the major shortcomings of the various association rules mining algorithms (and of Apriori and CAR-Apriori as well) is a large amount of produced rules, which are redundant and that have to be, afterwards, analysed by experts to identify the interesting and non-obvious ones [Kotsiantis and Kanellopoulos 2006]. In order to overcome this drawback, a number of approaches have been proposed. We identify 4 research directions in this area, which are schematically presented in Figure 4.6.

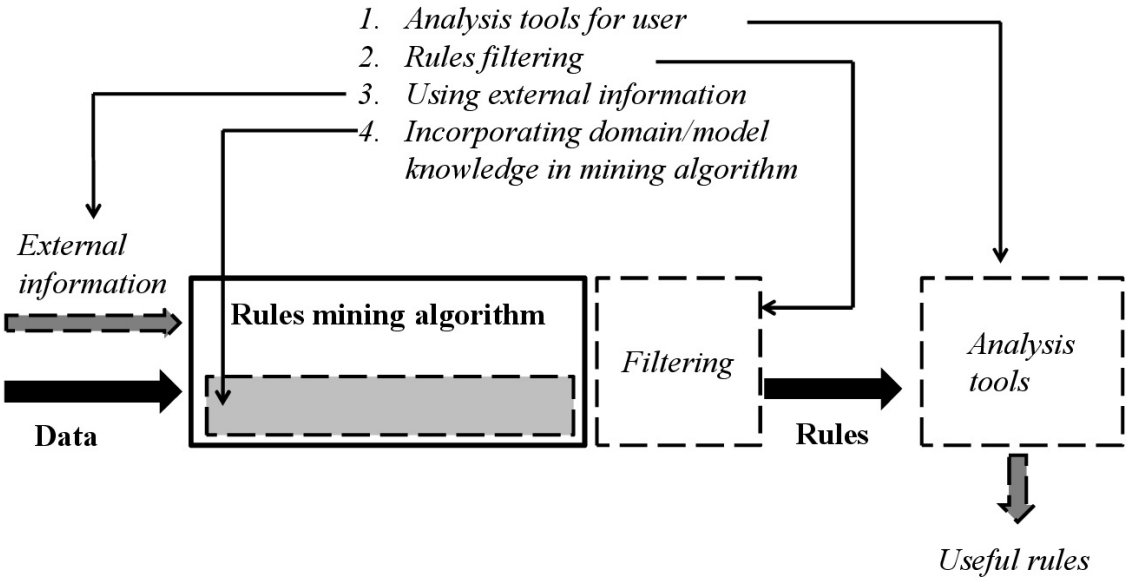

Figure 4.6: Redundancy of Association Rules: Possible Solutions

The **first** group of methods aim to help the final user, or the expert, to filter the rules by providing him with tools, for example a visualization tool [Techapichetvanich and Datta 2004].

The **second** group of methods limits the number of rules mined by the algorithm. This goal is reached, for example, by cutting the set of provided rules (by presenting to the final user only the top-K most interesting rules [Webb and Zhang 2005]). Different rules evaluation measures, like lift or conviction [Tew *et al.* 2014], can also be used as a supplement to [Ordonez *et al.* 2006] or as a substitute for [Brin *et al.* 1997] the traditional support and confidence measures, to choose the most interesting rules.

The methods of the **third** group use external information which is incorporated in the mining process in real time. For instance, the utility-based itemset mining approach [Yao and Hamilton 2006] evaluates the itemsets (which will be further used to form rules) not only through statistical characteristics like support but also through utility values provided by experts. For example, in

the case of the market basket data analysis, this lets to discover not only items frequently bought together, but also those combinations of items, which will increase the profit of the supermarket. The other approach in this group exploits user-provided restrictions on the association rules. For example, how many items are allowed to be in the consequent/antecedent of the rule, which attributes are allowed to be in which part of the rule, etc. [Ordonez *et al.* 2001, Srikant *et al.* 1997].

Finally, in the **fourth** group of methods we relate methods, whose mining process has some specific built-in logic, which usually reflects the domain knowledge or the particularities of the task being solved. In this group of approaches, we can refer to methods that pose some restrictions on the structure of the mined patterns. As an example, we can mention supervised descriptive rule induction techniques that aim to mine contrast patterns. Contrast patterns were initially proposed to solve specific tasks (like searching for the differences among different classes of data or different databases), not with the aim to reduce the number of produced association rules. However, since only a subset of the produced rules corresponds to the imposed conditions, supervised descriptive rule induction techniques actually result in a reduced number of output association rules, those that answer a specific question. Thus, the resulting set of rules contains highly useful information (within the specified task).

4.4 Resume

To our point of view, the four stages of the development of data analysis namely descriptive, diagnostic, predictive and prescriptive analytics correspond to the tasks of class description, class comparison, classification and trigger factors identification, which form the four stages of class analysis task development. These four tasks are related and usually the solution of one task is at least partially based on the solution of the previous task. The second scientific problematic of this thesis consists in the identification of trigger factors, factors that can stimulate the transition of elements between classes. We see this task as the fourth stage of the development of the class analysis task (see Section 4.1). Thereby we consider that the solution of the task of trigger factors identification can be based on classification approaches. More precisely, we assume that the solution of this task can be based on the following heuristic:

In order to stimulate the transition of the data elements from one class to another, we need to change the values of feature attributes from those that are typical for non-target class (classes) to those that are typical for the target class.

To identify the feature attributes whose values should be changed and what these new values should be, we need to find the patterns that reveal the dependencies between the feature attributes and the value of the target attribute. This can be done by classification techniques. Since the trigger factors that will be identified from these patterns should correspond to some real-life characteristics, we assume that interpretable classification approaches will suit better. Thereby the first part of this chapter is dedicated to the analysis of basic classification approaches with the aim to choose the one interpretable, which afterwards will be used for the identification of the class-specific patterns.

After considering such classification approaches as probabilistic classification, artificial neural networks, support vector machines and rule-based classifiers, we chose the latter one as it is the most intuitively understandable. Next, we proceeded to the discussion of some topics related to rule-based classification patterns: evaluation measures, variations of the patterns (contrast patterns and treatment learning) and the way rule-based classification patterns can be mined. We formally prove that under some conditions contrast patterns are equivalent to classification

rules (that is every contrast pattern corresponds to a classification rule and vice versa).

Treatment learning, which is essentially a weighted class contrast learner, can be considered as a method of trigger factors identification. Indeed, it searches for the factors that can change the per-class distribution of the elements of the dataset, that is those factors, that can potentially stimulate the transition of the elements between classes. However, treatment learning assumes that a small number of attribute features actually impacts the value of the target attribute and searches for only small (in terms of size) treatments. But we consider that different subgroups of the dataset can have different trigger factors; such trigger factors cannot be identified by treatment learning.

In the end of the second part of this chapter, we described the Apriori algorithm whose adaptation (CAR-Apriori) we choose to use for the identification of rule-based classification patterns. We also briefly discuss the problem of rules redundancy and our view of its possible solutions.

The next chapter of this thesis is dedicated to the description of the approach that we propose to use for the trigger factors identification. This approach is essentially based on the analysis of the rule-based classification patterns and can be considered as a contrast pattern.

Chapter 5

Proposed Solution: A Technique for Automatic Identification of Trigger Factors

Contents

This chapter is dedicated to the description of a technique which we propose for automatic identification of trigger factors. As it was mentioned in the previous chapter, this technique is based on the heuristics that *changing the values of the feature attributes to those that are typical for the target class can stimulate the transition of the data elements between classes.*

5.1 Preliminaries

Let us consider the following example. Assume that authorities of a state want to increase the birth rate and thus are interested in the identification of those factors, that can encourage people of the state to bear children. It is obvious that material welfare influences the ability to have children. However, there are multiple axes of welfare: income level, availability of vehicles, type of the house etc. and each of them can have different significance levels in different subgroups (social, cultural, ethnic etc.). Assume that after the analysis of the census dataset following association rules were identified:

1. for young families

- own house → children;
- rented house → **no** children;

2. for old families
   - two cars → children;
   - no vehicles → **no** children.

Analysing these two groups of patterns we can say that for young families the presence of an own house can trigger the childbirth. However, for old families, the presence of vehicles is more important. Thereby *presence of an own house* and *presence of two cars* are trigger factors for young and old families respectively.

In the following sections of this chapter, we present the new pattern 'sets of contrasting rules', which formalises the presented above reasoning and which we propose to use for the identification of trigger factors. We analyse the proposed pattern and formally prove that it defines a contrast pattern. Through the experimental evaluation, we also show that the proposed pattern is actually capable of identifying reasonable trigger factors.

Note that this chapter presents the 'proof of the concept' rather than the complete scientific approach. That is we aim here to show that the proposed pattern can identify meaningful factors capable of stimulating the transition of data elements between the classes. However, we plan to continue this work in the future, in particular, we want to work on the specialised mining algorithm and to perform extended experimental evaluations proving in such a way not only the concept but the robustness of our approach as well.

5.2 A New Pattern 'Set of Contrasting Rules'

5.2.1 Definitions

The definition of the SCR pattern relies on the introduction of two new types of attributes: *varying* attributes and *invariant* attributes. An attribute is considered to be varying if its value can be changed externally to the system within the specified application task, and invariant otherwise. For example, when analysing census data the attribute *income_level* can be considered as varying if, for instance, the government can provide financial assistance to the citizens. At the same time, the value of the parameter *ancestry* cannot be changed, that is it belongs to the set of invariant attributes. So, we divide the set of all feature attributes (all attributes except the target attribute A^G, see Section 4.2 for corresponding notations) into two subsets: the set of varying attributes and the set of invariant attributes.

In this work we provide the definitions and proofs for the case when only two classes G_1 and G_2 are defined on the dataset. Indeed, we are interested in the identification of factors that can stimulate the transition of data elements from one class to another. It means that we can limit ourselves to the analysis of only two classes: the class to which we want to transfer the elements and the class from which the elements should be transferred.

Definition 5.1. *For a specified parameter α ($\alpha > 0.5$), a pair of rules $R1$ and $R2$ is called a pair of α-contrasting rules if:*

1. $conf(R1) \geq \alpha$ & $conf(R2) \geq \alpha$;
2. *both rules are classification rules corresponding to different classes; that is with different values of A^G in the pair (in our case G_1 and G_2);*

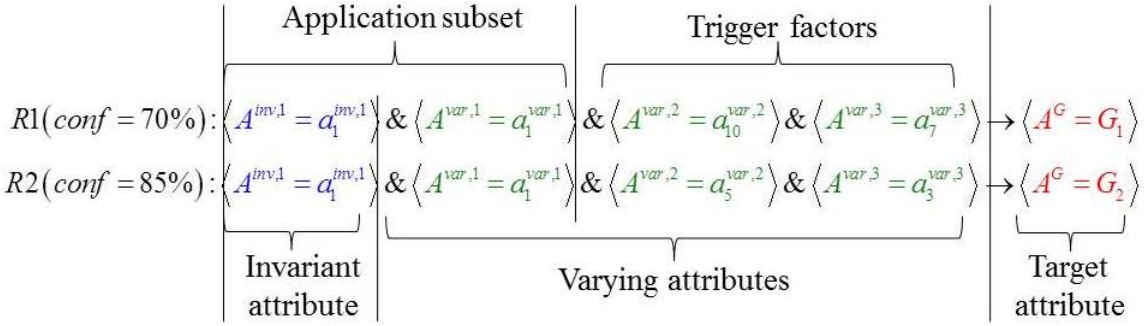

Figure 5.1: Example of a Pair of Contrasting Rules

3. *the antecedents of the rules are made up of the same attributes, within which there are at least one varying and one invariant attribute;*

4. *the values of all invariant attributes are the same for both rules;*

5. *at least one varying attribute has different values in the pair of rules.*

In this definition, we call the rule $R1$ a *contrasting pair* for the rule $R2$ and vice versa. Let us now consider the example of a rule pair $R1$ and $R2$, presented in Figure 5.1.

1. both rules are highly confident, with the minimum confidence value $minConf = 0.7$ (> 0.5);

2. both rules are classification rules with different values of A^G in the consequent;

3. the antecedents of the rules are composed of the same attributes, among which one is invariant ($A^{inv,1}$) and three are varying ($A^{var,1}$, $A^{var,2}$ and $A^{var,3}$);

4. the values of the invariant attribute $A^{inv,1}$ are the same for both rules ($A^{inv,1} = a_1^{inv,1}$), as well as the values of one of the varying attributes ($A^{var,1} = a_1^{inv,1}$ for both $R1$ and $R2$);

5. the values of the two other varying attributes are different ($a_{10}^{var,2}$ and $a_7^{var,3}$ for rule $R1$ and $a_5^{var,2}$ and $a_3^{var,3}$ for rule $R2$);

Thereby, we can conclude that the pair of rules $R1$ and $R2$ form a pattern 'pair of α-contrasting rules' with $\alpha = 0.7$.

Denote by D^{R1} the set of elements in D covered by the antecedent of the rule $R1$, then $\left|D^{R1}\right|$ is the number of these elements. By analogy, the number of elements covered by the antecedent of the rule $R2$ is $\left|D^{R2}\right|$. By $D^{R1,R2}$ we denote the set of elements of the dataset D covered by the union of the antecedents of the two contrasting rules $R1$ and $R2$. Then equation (5.1) holds.

$$\left|D^{R1,R2}\right| = \left|D^{R1}\right| + \left|D^{R2}\right| \tag{5.1}$$

Let $D_{G_1}^{R1,R2}$ be the set of elements in $D^{R1,R2}$ belonging to the class G_1, and $D_{G_2}^{R1,R2}$ - the set of elements in $D^{R1,R2}$ belonging to the class G_2. Then equation 5.2 holds.

$$\left|D^{R1,R2}\right| = \left|D_{G_1}^{R1,R2}\right| + \left|D_{G_2}^{R1,R2}\right| \tag{5.2}$$

5.2.2 Set of Contrasting Rules as a Contrast Pattern: a Proof

Now we aim to show that the pair of contrasting rules $R1$ and $R2$ defines a contrast pattern on $D^{R1,R2}$. As discussed in Section 4.3.2, the contrast patterns are defined as those having a support that changes significantly from one class to another. We will use the *GrowthRate* (equation (4.16)) to check if the proposed pattern can be considered as a contrast pattern. We seek to prove the following statement:

$R1$ has a significant *GrowthRate* from $D_{G_2}^{R1,R2}$ to $D_{G_1}^{R1,R2}$ and $R2$ has a significant *GrowthRate* from $D_{G_1}^{R1,R2}$ to $D_{G_2}^{R1,R2}$.

Let us prove the statement formulated above for the rule $R1$ (the proof for the rule $R2$ can be done in a similar way).

As the threshold of minimum confidence for both rules is set to α and as there are only two classes defined on D, the number of elements covered by the rule $R1$ in $D_{G_1}^{R1,R2}$ ($\left|D_{G_1}^{R1}\right|$), will satisfy the inequality $\left|D_{G_1}^{R1}\right| \geq \alpha \left|D^{R1}\right|$, and the number of elements covered by the rule $R2$ in $D_{G_1}^{R1,R2}$ will satisfy inequality $\left|D_{G_1}^{R2}\right| \leq (1-\alpha)\left|D^{R2}\right|$. The total number of elements in $D_{G_1}^{R1,R2}$ will be $\left|D_{G_1}^{R1,R2}\right| = \left|D_{G_1}^{R1}\right| + \left|D_{G_1}^{R2}\right|$. Then the support of the rule $R1$ on $D_{G_1}^{R1,R2}$ can be estimated as shown in the equation (5.3) with γ' standing for $\frac{|D^{R2}|}{|D^{R1}|}$.

$$\begin{aligned} supp_{D_{G_1}^{R1,R2}}(R1) = \frac{\left|D_{G_1}^{R1}\right|}{\left|D_{G_1}^{R1}\right| + \left|D_{G_1}^{R2}\right|} = \frac{1}{1 + \frac{\left|D_{G_1}^{R2}\right|}{\left|D_{G_1}^{R1}\right|}} \geq \\ \geq \frac{1}{1 + \frac{(1-\alpha)}{\alpha}\frac{|D^{R2}|}{|D^{R1}|}} = \left[\gamma' \equiv \frac{|D^{R2}|}{|D^{R1}|}\right] = \frac{1}{1 + \frac{(1-\alpha)}{\alpha}\gamma'} \end{aligned} \tag{5.3}$$

In a similar way, we can estimate the support of the rule $R1$ on $D_{G_2}^{R1,R2}$. Indeed, the number of elements covered by the rule $R1$ in $D_{G_2}^{R1,R2}$, that is $\left|D_{G_2}^{R1}\right|$, will satisfy the inequality $\left|D_{G_2}^{R1}\right| \leq (1-\alpha)\left|D^{R1}\right|$ and the number of elements covered by the rule $R2$ in $D_{G_2}^{R1,R2}$ will satisfy the inequality $\left|D_{G_2}^{R2}\right| \geq \alpha\left|D^{R2}\right|$. The total number of elements in $D_{G_2}^{R1,R2}$ will be $\left|D_{G_2}^{R1,R2}\right| = \left|D_{G_2}^{R1}\right| + \left|D_{G_2}^{R2}\right|$. The support of the rule $R1$ on $D_{G_2}^{R1,R2}$ can be estimated as shown in equation (5.4).

$$\begin{aligned} supp_{D_{G_2}^{R1,R2}}(R1) = \frac{\left|D_{G_2}^{R1}\right|}{\left|D_{G_2}^{R1}\right| + \left|D_{G_2}^{R2}\right|} = \frac{1}{1 + \frac{\left|D_{G_2}^{R2}\right|}{\left|D_{G_2}^{R1}\right|}} \leq \\ \leq \frac{1}{1 + \frac{\alpha}{(1-\alpha)}\frac{|D^{R2}|}{|D^{R1}|}} = \left[\gamma' \equiv \frac{|D^{R2}|}{|D^{R1}|}\right] = \frac{1}{1 + \frac{\alpha}{(1-\alpha)}\gamma'} \end{aligned} \tag{5.4}$$

Now using (5.3) and (5.4) we can estimate the value of the *GrowthRate* for the rule $R1$ from $D_{G_2}^{R1,R2}$ to $D_{G_1}^{R1,R2}$ with the equation (5.5), which after some mathematical transformations can be rewritten in the form (5.6).

$$GrowthRate\left(R1, D_{G_1}^{R1,R2}, D_{G_2}^{R1,R2}\right) = \frac{supp_{D_{G_1}^{R1,R2}}(R1)}{supp_{D_{G_2}^{R1,R2}}(R1)} \geq$$
$$\geq \left[\frac{1}{1+\frac{(1-\alpha)}{\alpha}\gamma'}\right] / \left[\frac{1}{1+\frac{\alpha}{(1-\alpha)}\gamma'}\right] = \quad (5.5)$$
$$= \frac{1+\frac{\alpha}{(1-\alpha)}\gamma'}{1+\frac{(1-\alpha)}{\alpha}\gamma'}$$

$$GrowthRate\left(R1, D_{G_1}^{R1,R2}, D_{G_2}^{R1,R2}\right) = 1 + \gamma' \frac{2\alpha - 1}{1+\frac{1-\alpha}{\alpha}\gamma'} \quad (5.6)$$

Let us now analyse the value of the second summand in equation (5.6). As $\gamma' \geq 0$ and $\alpha > 0.5$, $\gamma' \frac{2\alpha-1}{1+\frac{1-\alpha}{\alpha}\gamma'} \geq 0$. The second summand is equal to zero only when $\gamma' = 0$, that is $\left|D^{R2}\right| = 0$ or the number of elements covered by the antecedent of the rule $R2$ is equal to zero. However in this case the support of the rule is equal to 0, and thus this rule will not be mined as there are no instances covered by the rule.

So, we have shown that $GrowthRate\left(R1, D_{G_1}^{R1,R2}, D_{G_2}^{R1,R2}\right)$ is always greater than 1 if $\alpha > 0.5$. Thereby, we proved that the support of the rules forming the pair of contrasting rules changes significantly from one subset to another. Thus the proposed pattern can be considered as a contrast pattern.

Let us consider one generalizing case. Assume that the set of contrasting rules pattern is formed of 3 following rules:

1. $R1 : \langle A^{inv,1} = a_1^{inv,1} \rangle \& \langle A^{var,1} = a_2^{var,1} \rangle \rightarrow \langle A^G = G_2 \rangle$,
2. $R2 : \langle A^{inv,1} = a_1^{inv,1} \rangle \& \langle A^{var,1} = a_3^{var,1} \rangle \rightarrow \langle A^G = G_2 \rangle$,
3. $R3 : \langle A^{inv,1} = a_1^{inv,1} \rangle \& \langle A^{var,1} = a_1^{var,1} \rangle \rightarrow \langle A^G = G_1 \rangle$.

As we can see, in this case the transition of the elements from the class G_2 to the class G_1 can be triggered by changing the value of the attribute $A^{var,1}$ from $a_1^{var,1}$ to $a_2^{var,1}$ or from $a_1^{var,1}$ to $a_3^{var,1}$. These 3 rules can be rewritten in a form of 2 following disjunctive rules [Nanavati *et al.* 2001], which will also form a pair of contrasting rules:

1. $R2 : \langle A^{inv,1} = a_1^{inv,1} \rangle \& \langle A^{var,1} = \left(a_2^{var,1} \;\; OR \;\; a_3^{var,1}\right) \rangle \rightarrow \langle A^G = G_2 \rangle$,
2. $R1 : \langle A^{inv,1} = a_1^{inv,1} \rangle \& \langle A^{var,1} = a_1^{var,1} \rangle \rightarrow \langle A^G = G_1 \rangle$.

The proposed pattern and its contrast properties were presented at scientific conferences [Aleksandrova *et al.* 2016d, Aleksandrova *et al.* 2016c] and as a paper in a national journal [Aleksandrova *et al.* 2014a].

5.2.3 Quality of the Pattern 'Sets of Contrasting Rules'

Let us now check what will be the values of lift and conviction of the rules $R1$ and $R2$. According to equation (4.13), the value of $lift_D(R1)$ can be estimated as given in the equation (5.7) (recall that $\gamma \equiv \frac{|D_2|}{|D_1|}$, as defined in Section 4.3.2):

$$lift_D(R1) = \frac{conf_D(R1)}{supp_D(A^G = G_1)} \geq \frac{\alpha}{\frac{|G_1|}{|G_1|+|G_2|}} = \alpha(1+\gamma) \tag{5.7}$$

Using formula (4.14) the value of $conv(R1)$ can be estimated by the formula (5.8).

$$conv_D(R1) = \frac{1 - supp_D(A^G = G_1)}{1 - conf_D(R1)} \geq \frac{1 - \frac{|G_1|}{|G_1|+|G_2|}}{1-\alpha} = \frac{\gamma}{1+\gamma}\frac{1}{1-\alpha} \tag{5.8}$$

It can be shown that for every given value of γ, lift and conviction of the rule $R1$ will be superior to 1 if $\alpha > \frac{1}{(1+\gamma)}$. Note, that we obtained the same condition as the one given in equation (4.20). That is if the antecedents of the rules from the SCR pattern define contrast patterns not only on the subset $D^{R1,R2}$, but also on the whole dataset D, then the values of lift and convictions are guaranteed to be superior to 1, what ensures high quality of the rules and of the pattern. Moreover, for every given value of γ the minimum confidence threshold can be set in such a way, that the listed above characteristics will hold.

5.2.4 Algorithm for Mining Sets of Contrasting Rules

We now focus on the way the introduced contrast pattern can be mined. As it was discussed in Section 4.3.4 we choose to rely on CAR-Apriori algorithm [Ma 1998]. CAR-Apriori results in all possible classification rules that can be extracted from the dataset D and satisfy the user-specified minimum support and minimum confidence values (note that the value of the confidence threshold is chosen to satisfy the 1st condition of the Definition 5.1).

The pattern that we propose *'set of contrasting rules'* is made up of classification rules. However its main particularity consists in the fact that it is made up of several rules and the comparison of these rules is the main source of useful information in our case. Thereby the rules resulting after CAR-Apriori should be organised into the *'sets of contrasting rules'* following the Definition 5.1. This can be done as a post-processing step via the pairwise comparison of the rules (see Algorithm 6). In the proposed algorithm the function $isPairOfContrastingRules(R1, R2)$ is a boolean function, that returns *true* if all the conditions of Definition 5.1 are fulfilled by the pair $R1$ and $R2$, and $false$ otherwise.

5.2.5 Identification of Trigger Factors and Applications

We claimed that the proposed SCR pattern can be used to identify trigger factors, *i.e.* factors which can stimulate the transition of elements of the dataset from one class to another. Let us consider the pair of contrasting rules in Figure 5.1. Analysing these two rules, we can say that if, for the elements having $A^{inv,1} = a_1^{inv,1}$ and $A^{var,1} = a_1^{var,1}$, we force the attributes $A^{var,2}$ and $A^{var,3}$ to change their values from $a_5^{var,2}$ and $a_3^{var,3}$ to $a_{10}^{var,2}$ and $a_7^{var,3}$ respectively, then with a probability of 70%, these elements will move from the class G_2 to G_1. The move in the inverse direction will occur with a probability of 85%.

Algorithm 6 Form sets of contrasting rules

```
1: procedure formSetOfCR(L)
2:     setOfSets = {}
3:     for R ∈ L do
4:         contSet = {R}
5:         l = L − {R}
6:         for r ∈ l do
7:             if isPairOfContrastingRules(R, r) then
8:                 contrSet = contrSet ∪ {r}
9:             end if
10:        end for
11:        if size(contrSet) > 1 then
12:            setOfSets = setOfSets ∪ {contrSet}
13:        end if
14:        L = L − contrSet
15:    end for
16:    return setOfSets
17: end procedure
```

Thereby, the varying attributes with different values in the pair of contrasting rules define the trigger factors: they can stimulate the move of the elements from one class to another. The invariant attributes and those varying attributes having the same values in the pair of contrasting rules specify the application subset that is the subset of elements, that can be affected by these trigger factors. In real applications, the proposed pattern can be used to solve a wide range of tasks depending on the underlying meaning of the attributes. For example, we can make links between the proposed pattern and such research directions as *chance discovery* [Ohsawa 2006] in business and *identification of elements of habitus* [Bourdieu 1995] in sociology.

The first application task, *chance discovery*, aims to identify multiple scenarios which have an intersection point and different final states. The intersection point (which can be hidden or unobvious) is called a chance and its utility is measured as the difference of the merits of final states [Ohsawa 2006]. Depending on the application task we can consider attributes which define application subset as a starting point of possible scenarios with final states given by the target attribute. In this case, trigger factors can be viewed as chances (see Figure 5.2).

Habitus, identification of which we view as a second application task, is defined as *'a system of acquired dispositions serving as principles, which generate and organize the practices adapted for achieving certain results but do not require either conscious aiming at these results or special skills'* [Bourdieu 1995]. In other words, habitus is a *set of interconnected patterns* essential to a certain *social group* which is formed as a result of the adaptation of the group members to the living conditions and influences (or defines) the *behaviour* of the group members in different situations. The proposed pattern can be used for the identification of elements of habitus in the following way: application subset defines a *social group*; trigger factors correspond to the *set of interconnected patterns* that essentially defines habitus; finally the values of the target attribute corresponds to the *behaviour* essential for the *social group* (see Figure 5.2).

We presented and discussed these interconnections at the specialised scientific events: the workshop on chance discovery [Aleksandrova *et al.* 2016b] and a sociological conference [Chertov and Aleksandrova 2015].

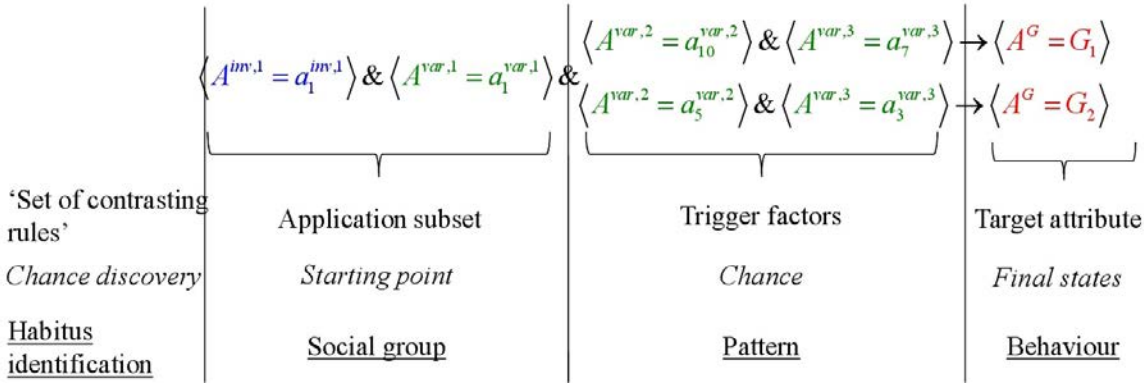

Figure 5.2: Links Between the Pattern 'Set of Contrasting Rules' and the Tasks of Chance Discovery and Identification of Elements of Habitus

5.3 Experimental Evaluation

The goal of the experiments conducted here is to show to what extent 'set of contrasting rules' patterns allow to automatically discover trigger factors.

5.3.1 Problem Formulation and Data Pre-processing

To test experimentally the validity of the proposed approach for the trigger factors identification we set the following experimental task: *identify demographic and socio-economic factors that can increase the birth-rate using demographic data.*

We choose to conduct our experiments on a publically available 5-percent sample of the California census dataset for the year 2000[12]. This dataset contains records of 610,369 family households (we choose to ignore subfamilies, as the number of households with subfamilies corresponds to only 3.6% of the initial sample).

Initially the census dataset contains many attributes (more than 100), most of which are not related to the posed problem. Indeed, it is obvious that the attribute *year_building_built*, that indicates the year when the building of the house was built, probably has no impact on the families' desire to have a child. It is true that traditional association rule mining algorithms do automatically discard statistically insignificant attributes. However, for the sake of simplicity we manually filter out the attributes.

The list of the considered attributes contains the 12 following items. Their possible values, as well as their types, are given in the Table 5.1.

- home ownership ($HouseOwn$),
- type of building ($HouseType$),
- number of vehicles available ($Vehicle$),
- husband's total income in 1999 ($HIncome$),
- spouse age (2 attributes: $HAge$ and $WAge$ for husband and wife respectively),

[12] https://www.census.gov/prod/cen2000/doc/pums.pdf

Table 5.1: Possible values of the attributes and their type; p =10,000$.

ATTRIBUTE	TYPE	DOMAIN
$HouseOwn$	varying	yes / no
$HouseType$	varying	NoStatHome / Apart / Att (attached house) / Det (detached house)
$Vehicle$	varying	0 / 1 / 2 / 3 / ≥ 4
$HIncome$	varying	$]-\infty, 0]$ / $]0, 1p]$ / $]1p, 2p]$ / $]2p, 4p]$ / $]4p, 6p]$ / $]6p, 8p]$ / $]8p, 10p]$ / $]10p, 20p]$ / $]20p, 30p]$ / $]30p, 40p]$ / $[40p, \infty[$
$HAge$	invariant	young (24-27) / middle-young (28-29) / middle (30-31) / middle-old (32-34) / old (35-38)
$WAge$	invariant	young (22-25) / middle-young (26-27) / middle (28-30) / middle-old (31-32) / old (33-37)
$(H/W)Edu$	invariant	noSchool / school / noCollege / college associate / bachelor / master / doctor
$(H/W)Anc$	invariant	WestEurope / EastEurope / Mexico / Latino / CentralAmericaIslands / NorthAfricaAndSouthAsia / otherAfrica / otherAsia / Australia / Pasific Afro-American / OtherAmerica / NonDef
$(H/W)WorkClass$	invariant	NoWork / PrivWork / GovWork / SelfEmployed
$Child$	target	YES / NO

- spouse education (2 attributes: $HEdu$ and $WEdu$ for husband and wife respectively),
- spouse ancestry (2 attributes: $HAnc$ and $WAnc$ for husband and wife respectively),
- spouse class of worker (2 attributes: $HWorkClass$ and $WWorkClass$ for husband and wife respectively).

In order to find an answer to the question that formulates our experimental goal, we form two classes G_1 and G_2 from the dataset. The first class G_1 is made up of families (elements) with one or two children aged from 0 to 2 years. The second class G_2 contains families without any children. These restrictions on the children age are imposed in order to track the change in family state from a childless family to a family with a small child (or children). We do not consider families with elder children, as it is difficult to identify which factors triggered the child appearance some years back. Thereby, we add to the dataset the target attribute $Child$ (A^G), indicating the presence or not of small children in the family, with $domain\,(Child) = \{YES, NO\}$ (see last line of the Table 5.1). The dataset is thus made up of 13 attributes.

To increase the reliability of the obtained results, we choose to impose some additional restrictions:

- all the families must be complete: the presence of both spouses is mandatory;
- both husband and wife must be without disabilities;
- spouse age must be within the most favourable period for having babies.

Table 5.2: Mining SCR pattern on different sub-datasets

subD	γ	$G_1 : minLift$, see eq. (5.7)	$G_1 : minConv$, see eq. (5.8)	*minSupp* %	# of rules		# SCR	η,%
					tot	G_1,%		
$D^{y,y}$	1.38	1.67	1.93	0.3	18,487	27	52	0.3
$D^{y,yM}$	1.78	1.94	2.13	0.6	10,064	12	24	0.2
$D^{yM,yM}$	1.86	2.0	2.17	0.8	8,041	8	23	0.3
$D^{yM,m}$	1.70	1.89	2.1	0.4	10,362	5	49	0.5
$D^{m,m}$	1.56	1.79	2.03	0.4	9,310	7	64	0.7
$D^{oM,m}$	1.38	1.67	1.93	0.4	6,435	15	67	1.0
$D^{oM,oM}$	1.33	1.63	1.9	0.5	5,650	17	74	1.3
$D^{oM,o}$	1.22	1.56	1.83	0.4	5,863	25	88	1.5
$D^{o,o}$	1.50	1.75	2.0	0.2	13,629	11	118	0.9

These conditions are quite relevant and obvious. For instance, it is clear that illness of the potential parents affects significantly their willingness and ability to have children. The bounds of the most favourable age for giving birth to babies (24 to 38 for men and 22 to 37 for women) as well as 5 age intervals for both husband and wife given in the Table 5.1 are taken from our previous works related to this dataset [Chertov and Aleksandrova 2013]. Considering all the imposed above restrictions, the size of the dataset is reduced. The number of elements with $Child = YES$ and $Child = NO$ in the resulting dataset equals to 8,299 and 12,249 elements respectively, which gives $\gamma = \frac{|G_2|}{|G_1|} = 1.48$.

If we divide the original dataset into sub-datasets according to the age of husband and wife, we get 25 sub-datasets. For example, we form the sub-dataset $D^{y,m}$ that corresponds to the families with a young husband (first position in the subscript of $D^{y,m}$) and middle-aged wives (second position in the subscript of $D^{y,m}$). In order to identify different patterns specific to a certain age, we conducted our analysis on these 25 sub-datasets separately. We present here the results for 9 of them, that contain the largest number of elements, namely for the following sub-datasets: $D^{y,y}$, $D^{y,My}$, $D^{My,My}$, $D^{My,m}$, $D^{m,m}$, $D^{Mo,m}$, $D^{Mo,Mo}$, $D^{Mo,o}$, $D^{o,o}$.

5.3.2 Mining and Analysing Sets of Contrasting Rules

We use the CAR-Apriori algorithm [Ma 1998] to mine classification rules in the sub-datasets of D, with a minimum confidence threshold equal to 0.7. As the sizes of these sub-datasets are different, we use different values of the minimum support threshold for each sub-dataset. Table 5.2 presents the general information about the chosen 9 sub-datasets.

The second column for each sub-dataset represents the ratio γ of the number of elements in the second class (number of elements, for which $Child = NO$) to the number of elements in the first class (number of elements, for which $Child = YES$). We can see that the number of elements in the second class is at least 1.2 times larger than the number of elements in the first class. However, the mean value of γ is 1.5, what support the same tendency in the whole dataset. Given the value of γ and the minimum confidence threshold $minConf = 0.7$ we can estimate the minimum values of lift and conviction for the rules describing the class G_1 using formulae (5.7) and (5.8) (see the third and the fourth columns). Note that both lift and conviction are always superior to 1, what proves the quality of the rules. We restrict ourselves to estimation the

lift and conviction only for rules describing G_1. Indeed, our goal is to transfer the data elements from G_2 to G_1. That is we are interested in trigger factors that will be identified from the rules describing the first class.

The fifth column of the table indicates the minimum support value used for each sub-dataset. We can see that the minimum support is quite low (between 0.8% and 0.2%). However, as it was mentioned in [Dong and Li 1999], contrast patterns with a large support are usually well-known, that is why it is interesting to search for contrast patterns with a small support (e.g. 5% or even 0.1%): the unexpected ones.

The sixth column presents the total number of classification rules found using the CAR-Apriori algorithm; the seventh column reveals the percentage of those classification rules that have the target attribute $Child = YES$ in the consequent. We can see that the percentage of such rules for the class G_1 is quite small. Each rule mined by the algorithm CAR-Apriori is considered to be a pattern. The eight column presents the number of SCR patterns found in every sub-dataset. The last column indicates the η coefficient, that is the ratio of the number of sets of contrasting rules to the general number of classification rules.

We can conclude that, when mining 'sets of contrasting rules' patterns, the number of patterns to analyse is dramatically reduced. For example, for the first sub-dataset, only 52 SCR patterns have to be analysed, instead of 18,487 patterns (each one represented by one association rule). Each of these 52 patterns is made up of more than 1 classification rule. Every rule directly indicates trigger factors, as well as the subgroups of elements of the dataset for which these factors can be applied. Thus, in this sub-dataset, the number of patterns to analyse corresponds to only 0.3% of the number of original patterns (classification rules).

However, the total number of obtained SCR patterns for all sub-datasets is more than 500. It can be still difficult for a human to analyse all of them. Thereby, multiple ordering strategies can be used to facilitate the analysis process, for example, analysing rules with high support values at first.

Now we proceed to show what kind of information can be obtained with the help of the proposed patterns. As an example, in Table 5.3 we present sets of contrasting rules for 5 different sub-datasets $D^{y,y}$, $D^{yM,yM}$, $D^{m,m}$, $D^{oM,oM}$ and $D^{o,o}$, that correspond to the sub-datasets where both husband and wife belong to the same age group. The antecedents of the rules in the patterns are represented in the second and third columns of the table. As stated in Definition 5.1, the antecedents of the pair of contrasting rules have a common part that specifies the subgroup of the elements. This common part is presented in the second column with invariant attributes given in bold. The antecedents have another part that differs in the values of varying attributes (this part represents the trigger factors). It is presented in the third column of the table. The fourth column indicates the value of the consequent (the attribute *Child*) of each rule in our patterns, and the fifth and the sixth columns reveal the confidence and the support values of the corresponding rules. As discussed before, the support of the rules is quite low, usually $< 2\%$. Thereby, following [Dong and Li 1999] we can expect that the proposed pattern reveals unforeseen knowledge.

Table 5.3: Some chosen 'sets of contrasting rules' patterns, p =10,000$

subD	Antecedent		Conseq.	conf	Supp,
	Subgroup	Trigger Factors	Child=		%
$D^{y,y}$	**WEdu=school** & HIncome=]1p,2p]	Vehicle=1	YES	0.71	1.43
		Vehicle=0	NO	0.70	0.33
	WEdu=noCollege & HouseOwn=no	HIncome=]2p,4p] & HouseType=Det	YES	0.73	1.38
		HIncome=]0,1p] & HouseType=Apart	NO	0.70	0.90
	HAnc=Mexico & HouseType=Det &... ...& HIncome=]4p,6p]	HouseOwn=yes	YES	0.75	0.81
		HouseOwn=no	NO	0.82	0.38
$D^{yM,yM}$	**WEdu=associate** & **WAnc=WestEurope** &... ... & **HAnc=WestEurope** & HIncome=]4p,6p]	HouseType=Det	YES	0.71	1.41
		HouseType=Apart	NO	0.92	1.41
	HAnc=otherAmerica & **WAnc=otherAmerica** &... ...& **WWorkClass=PrivWork**	HouseOwn=yes	YES	0.8	0.94
		HouseOwn=no	NO	0.88	1.76
$D^{m,m}$	**HAnc=Mexico** & **WAnc=Mexico** & ... ...& **WWorkClass=PrivWork**	HouseType=Det	YES	0.71	2.32
		HouseType=Att	NO	0.89	0.53
	HAnc=OtherAmerican	HIncome=]2p,4p]	YES	0.72	1.20
		HIncome=]1p,2p]	NO	0.86	0.80
	WAnc=Latino	HouseOwn=yes & HouseType=Det	YES	0.71	1.0
		HouseOwn=no & HouseType=Apart	NO	0.75	0.80
$D^{oM,oM}$	**HEdu=noSchool**	Vehicle=2	YES	0.70	1.48
		Vehicle=0	NO	1	0.78
	HEdu=associate & **WEdu=college**	HouseOwn=yes	YES	0.8	0.93
		HouseOwn=no	NO	0.7	0.54
$D^{o,o}$	**HEdu=associate** & **WEdu=bachelor** &... ...& **HWorkClass=PrivWork** & HouseType=Det	HouseOwn=yes	YES	0.71	1.41
		HouseOwn=no	NO	0.88	0.28
	WEdu=associate	Vehicle≥4 & HouseType=Det	YES	0.70	1.02
		Vehicle=2 & HouseType=Apart	NO	0.73	0.43
	WEdu=bachelor & **HAnc=other Asia** & ... ...& HouseOwn=yes	HIncome=]10p,20p]	YES	0.70	0.55
		HIncome=]8p,10p]	NO	0.70	3.0

When analysing the sets of contrasting rules given in Table 5.3, we can note that it can correspond to very precise recommendations for specific subgroups of elements in the dataset. For example, let us look at the first pattern obtained for the sub-dataset $D^{y,y}$. It indicates that if we provide young families ($HAge =]24, 27]$ and $WAge =]22, 25]$) in which the wife's education level is $WEdu = school$ and husband's income is in the range $]10000, 20000]$ with a vehicle, then with a high probability (70%) they will decide to have a baby. However, in another subgroup of the same sub-dataset, which is composed of families where the wife has started the college but did not finish education there ($WEdu = noCollege$) and that do not have their own house ($HouseOwn = no$), it is not the number of vehicles that can trigger a childbirth, but rather the combination of the type of the house (it should be changed from 'apartment' to 'detached house') and the increase of the income level. Also, we can see that subgroups and trigger factors (or combinations of attributes that form the trigger factors) are different for different sub-datasets. This proves the ability of the proposed pattern to extract valuable knowledge from the dataset.

5.4 Conclusion

In this chapter, we introduced a new pattern 'set of contrasting rules' which we propose to use for the automatic identification of trigger factors. This pattern relies on the introduction of the notions of invariant and varying attributes. It also has the characteristic of being made up of several rules, at the opposite of the majority state-of-the-art patterns, made up of only one itemset, or one rule. The proposed pattern allows to automatically discover not only the trigger factors but also application subsets of the original dataset for which these factors can be applicable.

We showed that the SCR pattern belongs to the framework of supervised descriptive rule induction and the antecedent of every rule of the set of contrasting rules form a contrast pattern on a sub-dataset defined by the rules forming the pattern. Thereby **it can be considered as a possible solution to the problem of the rules redundancy** because this pattern, as all other patterns of the supervised descriptive rule induction framework, aims to discover only highly informative knowledge within a specified task. **We also mathematically prove that if each rule of the pattern forms a contrast pattern on the entire dataset as well, then the values of the lift and conviction of the rules are guaranteed to be superior to 1.** This ensures high quality of the rules and reduces the influence of spurious associations.

We perform the construction of the pattern via the pairwise comparison of all discovered classification rules and filtering out those, that have no contrasting pairs. It may seem that in such a way we can loose some valuable rules. However, only one classification rule does not show which exactly feature attributes can be considered as trigger factors and to what subset of the dataset they can be applied. Thereby, as our goal is to propose an approach for the automatic identification of trigger factors, single rules are not of interest for us and thus can be filtered out.

Depending on the underlying meaning of the attributes, the proposed pattern can be used to solve a wide variety of application tasks. The possibility to use SCR pattern for *chance discovery* in business and identification of *elements of habitus* in sociology was discussed on the specialised scientific events [Aleksandrova *et al.* 2016b, Chertov and Aleksandrova 2015].

We showed on a real dataset of census data, that trigger factors can be actually identified, and that they can be easily interpreted and used to reach the desired objective. We also show how the number of patterns to analyse is reduced when the SCR pattern is used.

As it was discussed in the Section 4.4, to the best of our knowledge treatment learning is the only counterpart of our method that also allows to discover factors that can stimulate

the transition of elements from one class to another (trigger factors). However, this approach is completely based on the assumption that in the dataset there is a small number of funnel attributes that actually affect the value of the target attribute. Thus, contrary to our pattern treatment learning does not solve the posed task completely, as it is impossible to discover trigger factors specific for different subgroups of the dataset.

Part III

General Conclusions and Perspectives

Chapter 6

Conclusions and perspectives

Contents

6.1 General Motivation

One of the main features of the information society in which we live today: constantly growing amount and dimensionality of data and the importance of information that can be extracted of it. The high dimensionality of this data can reduce the quality of the data processing methods performance. This fact increases the importance of *dimensionality reduction* techniques.

Dimensionality reduction techniques can be characterised in terms of two axes: according to their *aim* as dimensionality reduction for optimal data representation and dimensionality reduction for classification, and according to the adopted *strategy* as feature selection and feature extraction techniques.

Feature extraction techniques, though having usually higher descriptive and discriminative power, have one essential drawback: the lack of interpretation of the constructed feature space. This leads to first scientific problematic posed in this thesis:

SP1: how to extract interpretable latent features?

Dimensionality reduction for optimal data representation and dimensionality reduction for classification can be also viewed as non-supervised and supervised approaches. In the latter case the dimensionality reduction algorithms manage external information about the class belonging

of each datapoint, which can be used to estimate the importance of each feature, as opposite to the former case. As it was discussed in Chapter 4 we see the task of *trigger factors identification* as a direct descendant of the classification task in terms of four stages of the *class analysis* task development. As trigger factors we understand those factors that can stimulate the transition of data elements from one class to another (see Section 4.1). To the best of our knowledge, no approaches were proposed in the literature for automatic identification of trigger factors in the general case, thereby the second scientific problematic arises:

SP2: how to identify automatically factors that can cause the movement of elements from one class of the dataset to another (trigger factors)?

We choose to work on both scientific problematics in the scope of the recommender systems application domain because of two reasons. First, recommender systems aim to help users to overcome the information overload problem, which is important considering the vast amount of information available in modern society. Second, it faces both SP1 and SP2 formulated above (see Chapter 1). Indeed, a very popular in recommender systems matrix factorization technique is essentially a feature extraction method, which extracts latent features without interpretation and thus provides no means to explain the obtained recommendations. Thereby, we formulate the first application problematic as follows:

AP1: propose an automatic interpretation of latent features within the matrix factorization-based recommending models and explore if the resulting interpretation can be used to improve the recommender system performance.

After the analysis of the evolution of recommender systems we foresee the requirement for the trigger factors identification techniques in the upcoming generations of RS. We formulate the second application problematic as follows:

AP2: propose a technique that can automatically identify trigger factors and generate based on them recommendations to achieve the desired objective.

This thesis aims to solve the questions formulated in scientific and application problematics given above. This final chapter is dedicated to the analysis of the obtained results and discussion of possible future extensions of our work.

6.2 First Application Problematic: Automatic Interpretation of Matrix Factorization Recommendation Model

6.2.1 Summary of Obtained Results

The first application problematic consists in proposing an interpretation of matrix factorization-based recommender models. The analysis of the related literature (see Section 2.3.3) showed that the existing approaches either incorporate interpretation from other interpretable models, or change the structure of the basic matrix factorization model, or require human analysis. We propose to associate features of matrix factorization with real users from the system (representative users). We decide to choose those users, whose vectors in the matrix W are the closest to the canonical form. This interpretation, contrary to the state-of-the-art approaches, has the originality of being done completely automatically for the existing matrix factorization model and does not require any external information.

In order to show the validity of the proposed interpretation, we decided to test the ability of the chosen representative users to correctly represent interests of other users via the cold-start problem. Indeed, as features are considered to represent the relations between users and items, that is preferences of users on items, the feature-associated users should also be able to represent the same relation. Thereby, we proposed an approach for matrix factorization models within which the ratings of a predefined set of users on new items are used to estimate the ratings of other users on these new items.

The experimental evaluation on two real-world datasets showed that the set of chosen representative users tend to be composed of users with different behavioural patterns and their ratings can be successfully used to solve the new item cold-start problem. These statements show the validity of the proposed interpretation. Also, our approach outperforms in terms of ranking the benchmark RBMF model. It is also worth to note that the proposed cold-start problem solution does not use any content information, what makes it even more valuable in the cases when this type of information is not available.

6.2.2 Future work

Short-Term Perspectives

Following the state-of-the-art approaches, in our work we chose to rely on non-negative factorization model and use multiplicative update rules as an optimisation technique. Performing the interpretation we do not rely, however, on the non-negativity of the factor matrices. Thereby, we assume that the same interpretation can be made when other optimisation techniques are used, such as alternating least squares or stochastic gradient descend. This will allow to benefit from the proposed interpretation and the advantages of the alternative optimisation techniques. Indeed, ALS, for example, allows parallel implementation what makes it being more efficient when working with high-dimensional rating matrices. However, the validity of this interpretation, as well as the performance for the cold-start problem, have to be evaluated experimentally.

Furthermore, we suggest that the features could be associated with items (representative items) in the way it was done for user-based interpretation or with both items and users simultaneously. Although the association of features with items can be done in a straightforward way (analysing the columns of the matrix V instead of the columns of the matrix W), the second proposed association is not that evident. Indeed, in the latter case the values of both factor matrices are used to represent the association between features and representative elements (users and items), however, it is not clear how these association should be reflected. They can be represented as more complex structures than canonical vectors. One possible solution can be based on the usage of Matrix Tri-Factorization. This model presents the original rating matrix as the multiplication of three matrices, which represent the relation between users and user-related features, user-related features and item-related features, and finally item-related features and items. Thereby, analysing the first and the third matrix it is possible to define representative users and items. The second matrix, in this case, reveals the relation between these two groups or representative elements. However, to our opinion, the underlying relation between users and items do not depend on factorization model but are the underlying characteristic of every rating matrix. Thereby, such an interpretation should be possible in the case of only two factor matrices as well.

Mid-Term Perspectives

Our approach performs an association of latent features with real users from the system for the statical (in terms of time) case. However, the set of representative users may change over time. Indeed, from the mathematical point of view the original rating matrix and the one with new ratings are different, thereby the set of representative users is not guaranteed to be the same. In the future, we would like to work on the approach to identify the stable set of representative users with the course of time. We suggest that in this case the representative users should be chosen as those that can be associated with features in different time points. It can be possible to rely on the simultaneous factorization of rating matrices corresponding to different time stamps (as it is done in Collective Matrix Factorization). As compared to 'static' representative users, the 'time-stable' representative users can be used not only to predict the ratings on new items in short term, but also to follow the global change of interests of users in long term, or even as those users, who can influence the preferences of others.

6.3 Second Application Problematic: Automatic Identification of Trigger Factors

6.3.1 Summary of Obtained Results

In the Chapter 5 we presented our approach for the automatic identification of trigger factors. The proposed solution utilises classification technique, more precisely, rule-base classification.

The only known to us state-of-the-art approach for the automatic identification of trigger factors *treatment learning* is based on the assumption that a small number of attributes determines the per-class distribution of the datapoints. However, this may not always be the case. Also, we suppose that trigger factors can be different between subsets of the original dataset. Thereby, we proposed a new pattern 'set of contrasting rules'. This pattern is based on the introduction of notions of *varying* and *invariant* attributes which are used to identify trigger factors and application subsets respectively. Via experiments on a real census dataset, we showed that the identified trigger factors are reasonable and have the potential to affect the per-class distribution of the datapoints.

One of the originality of the proposed pattern is that contrary to state-of-the-art patterns it consists of several classification rules. It was shown that under some conditions the rules forming the pattern are guaranteed to be of high quality, *i.e.* lift and conviction are guaranteed to be superior to 1 (see Section 5.2.3). We also formally proved that the proposed pattern is essentially a contrast pattern and belongs to supervised descriptive rules induction paradigm (Section 5.2.2). Thereby, the SCR pattern can be considered as a partial solution to the problem of rules redundancy, as it selects only highly informative within a certain application task rules. This statement was also supported experimentally.

Depending on the meaning of the underlying attributes, the SCR pattern can be also used to solve diverse application tasks, such as chance discovery or identification of elements of habitus (see Section 5.2.5). In this case, application subsets and trigger factors correspond respectively to starting points and chances (chance discovery), and social groups and behavioural patterns (identification of elements of habitus). These interconnections of the proposed pattern with other theories we discussed on the specialised scientific events.

6.3.2 Future Work

Short-Term Perspectives

The work summarized previously is essentially a proof of concept, as we only analysed the qualities of the proposed pattern and showed experimentally that it can be used to identify the required trigger factors. However, the robustness of the proposed approach has to be further investigated. We do not consider this work as finished, thereby we aim to continue it in different directions.

In this thesis we gave the conditions under which the lift and conviction of the rules of the proposed patterns are guaranteed to be superior to 1. However, there are other quality metrics (for example, leverage and improvement [Geng and Hamilton 2006, Bhargava and Shukla 2016]). Furthermore, we provided all mathematical formulations and proofs for our pattern for the case of only two classes. Thereby, we aim to analyse mathematically the values of other evaluation metrics and extend given proofs for the case of more than two classes. This work requires additional theoretical investigations but the results that we aim to achieve will allow having the theoretical justification for more practical cases (for example, when more than 2 classes are defined on the dataset).

Also, we would like to test if it is possible to extract meaningful trigger factors from other datasets. For example, in the case of e-learning dataset it can be interesting to identify factors that can help the student to finish the started on-line course (to identify trigger factors that can transfer a student from the class of backwards students to the class of successful students). The nature of this dataset is different from the census dataset (the case considered in this thesis). Indeed, for solving the posed task it is useful to analyse the sequence of student activities (*sequential data*). Also, contrary to the census dataset, the values of some feature attributes can be missing due to the fact that the given student may not participate in all activities (*data with missing values*). Thereby, the proposed approach should be generalised for such cases. We also aim to design and perform additional experiments that will not only show the ability of the proposed pattern to identify trigger factors but will also prove the robustness of the proposed approach.

Mid-Term Perspectives

We proposed to mine the SCR pattern as a post-processing step for the CAR-Apriori algorithm. As our pattern consists of several contrasting rules, we never know if there will be a contrasting pair for a particular classification rule. Thereby, we cannot start forming SCR patterns until all classification rules are mined. However, we suppose that using depth-first search strategy we can simultaneously search for a rule and its contrasting pair. Hence, we would like to work over the specialised mining algorithm in the mid-term. In this algorithm, we want to associate each feature attribute with some numerical score. This score will represent the 'price' of changing the value of the attribute. This will allow to automatically process the invariant and varying attributes in a less rigid manner. Indeed, it can be more expensive to change the values of some varying attributes as compared to others. Hence, depending on the other feature attributes present in the rules, these attributes should be considered as invariant or varying. We would also like to adjust the proposed algorithm for the case when one target attribute can correspond to multiple classes in the original dataset. This will add additional flexibility to our approach and make it less dependent on the formalisation of the application domain. Finally, we aim to design a specialised metric for evaluation of the quality of the proposed pattern in our algorithm. This metric should measure the quality of each rule forming the pattern and the possibility of

the interclass transfer of the datapoints covered by the pattern.

6.4 Possible Contributions to Scientific Problematics and Long-Term Perspectives

Let us now have a look on the way the proposed solutions of the application problematics AP1 and AP2 can contribute to scientific problematics SP1 and SP2.

Let us start with AP1. Consider the case when the matrix to factorize represents the relation between datapoints and features. When solving the first application problem formulated in AP1 we propose to associate each latent feature with one of the real features (that will be referred to as representative feature) under the condition that the chosen representative feature approximates in the best possible way the corresponding latent feature. Thereby, we transform in such a way a feature extraction method into a feature selection method. We believe that such a transformation can be useful in other domains as well (not only in recommender systems), however, we suppose that the possibility of such an association depends on the application task. Indeed, in real life, there is usually a limited number of behavioural patterns. That is why, in our opinion, we succeeded to find a limited number of users which can be considered as representatives of the basic behavioural patterns. The other users, afterwards, can be represented as combinations of the basic behavioural patterns, that is as a combination of behavioural patterns of the chosen representative users. However, such associations may not be always possible. For example, if the matrix factorization approach is used to factorize the word-document matrix in text mining (as it is done in [Pauca *et al.* 2004]) it may be impossible to find a subset of words that can represent all other words.

The proposed approach for trigger factors identification is, in fact, a feature selection method as well. However, its aim is neither data representation nor classification but the transferring of the datapoints between classes. The latter task, from our point of view, is the direct descendant of the classification task from the perspective of four stages of the data analysis development. We also suggest that this task will be topical in the nearest future. According to the formulation of the second scientific problematic, we can say that the required technique for trigger factors identification was proposed, however, it can be further improved.

Nevertheless, we did not perform substantiation studies of the SP1 and SP2. Thereby, the possible contributions announced before should be further studies on both theoretical and experimental levels, what we see as the a long-term perspective for our work.

In general, the work presented in this thesis consists in the modelling of user preferences. More precisely, we aim to identify some basic elements of the system that are essential either for prediction of the users' behaviour (representative elements) or for stimulation of their behaviour development in the desired direction (trigger factors). However, the both proposed solutions use information about many users of the system. As it was mentioned in the first chapter of the thesis, the recommender systems of the next generation are considered to have the form of ultra-intelligent personal electronic agents. These agents will be able to take decisions in collaboration with the user and thus can accompany him for many years. Thereby, there will be a need to identify these basic elements of the system (essential for prediction and stimulation) for the case of only one user, that is for the case of ultra-personalised systems. We suggest that performing the study of our contributions to scientific problematics can help to solve this task in the frame of recommender systems.

Also, the proposed solutions perform the modelling on the basis of patterns typical for the majority of the users. Thereby, these patterns can not be used in the case of non-typical users, those users who do not match typical patterns. However, we suppose that *the proposed approach can be used for the identification of such non-typical users*. Indeed, we identify not only typical patterns but also the level of their association with users. Thereby, those users that have no considerably strong association with any of the identified typical patterns can not be described by these patterns and are, consequently, non-typical. We suggest that non-typical users can be modelled based on the following hypothesis: *non-typical users can be described by different typical patterns on different parts of the dataset*, for example, for different subset of items or for different time periods. This hypothesis raises new challenges: 1) how to identify the boundaries of those parts of the dataset on which non-typical users can be modelled using typical patterns and 2) which typical pattern or their combinations should be used on each defined part of the dataset. We suggest that the proposed in the future solutions can be inspired by recent achievements in certain scientific fields. For example, techniques from functional analysis and signal processing allow identifying the points of fundamental changes in the input information signal, which can correspond to the boundaries of those parts of the dataset where non-typical users can be modelled as typical. At the same time, the fuzzy logic techniques can be used to identify and express the 'partial matching' between non-typical users and typical patterns.

Bibliography

[Adams *et al.* 2002] Erwin Adams, B Walczak, Chris Vervaet, PG Risha, and DL Massart. Principal component analysis of dissolution data with missing elements. *International journal of pharmaceutics*, 234(1):169–178, 2002.

[Adomavicius and Tuzhilin 2005] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[Adomavicius and Tuzhilin 2011] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011.

[Adomavicius *et al.* 2011] Gediminas Adomavicius, Nikos Manouselis, and YoungOk Kwon. Multi-criteria recommender systems. In *Recommender systems handbook*, pages 769–803. Springer, 2011.

[Agarwal and Chen 2010] Deepak Agarwal and Bee-Chung Chen. flda: matrix factorization through latent dirichlet allocation. In *Proceedings of the 3d ACM International Conference on Web Search and Data Mining*, pages 91–100. ACM, 2010.

[Aggarwal 2014a] Charu C Aggarwal. Instance-based learning: A survey. In *Data Classification Algorithms and Applications*, pages 157–185. Chapman and Hall/CRC, 2014.

[Aggarwal 2014b] Charu C Aggarwal. An introduction to data classification. In *Data Classification Algorithms and Applications*, pages 1–36. Chapman and Hall/CRC, 2014.

[Aggarwal 2016a] Charu C Aggarwal. Attack-resistant recommender systems. In *Recommender Systems*, pages 385–410. Springer, 2016.

[Aggarwal 2016b] Charu C Aggarwal. Content-based recommender systems. In *Recommender Systems*, pages 139–166. Springer, 2016.

[Aggarwal 2016c] Charu C Aggarwal. Context-sensitive recommender systems. In *Recommender Systems*, pages 255–281. Springer, 2016.

[Aggarwal 2016d] Charu C Aggarwal. Ensemble-based and hybrid recommender systems. In *Recommender Systems*, pages 199–224. Springer, 2016.

[Aggarwal 2016e] Charu C Aggarwal. An introduction to recommender systems. In *Recommender Systems*, pages 1–28. Springer, 2016.

[Aggarwal 2016f] Charu C Aggarwal. Knowledge-based recommender systems. In *Recommender Systems*, pages 167–197. Springer, 2016.

[Aggarwal 2016g] Charu C Aggarwal. Model-based collaborative filtering. In *Recommender Systems*, pages 71–138. Springer, 2016.

[Aggarwal 2016h] Charu C Aggarwal. Neighborhood-based collaborative filtering. In *Recommender Systems*, pages 29–70. Springer, 2016.

[Aggarwal 2016i] Charu C Aggarwal. Social and trust-centric recommender systems. In *Recommender Systems*, pages 345–384. Springer, 2016.

[Aghaei *et al.* 2012] Sareh Aghaei, Mohammad Ali Nematbakhsh, and Hadi Khosravi Farsani. Evolution of the world wide web: From web 1.0 to web 4.0. *International Journal of Web & Semantic Technology*, 3(1):1, 2012.

[Agrawal *et al.* 1993] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, 22(2):207–216, 1993.

[Agrawal *et al.* 1994] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.

[Aguilar *et al.* 2014] Stephen Aguilar, Steven Lonn, and Stephanie D Teasley. Perceptions and use of an early warning system during a higher education transition program. In *Proceedings of the fourth international conference on learning analytics and knowledge*, pages 113–117. ACM, 2014.

[Aha *et al.* 1991] David W Aha, Dennis Kibler, and Marc K Albert. Instance-based learning algorithms. *Machine learning*, 6(1):37–66, 1991.

[Aleksandrova *et al.* 2014a] M Aleksandrova, A Brun, A Boyer, and O Chertov. Two-step recommendations: contrast analysis and matrix factorization techniques. *Mathematical machines and systems*, (1), 2014.

[Aleksandrova *et al.* 2014b] Marharyta Aleksandrova, Armelle Brun, Anne Boyer, and Oleg Chertov. Search for user-related features in matrix factorization-based recommender systems. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2014), PhD Session Proceedings*, volume 1, pages 1–10, 2014.

[Aleksandrova *et al.* 2014c] Marharyta Aleksandrova, Armelle Brun, Anne Boyer, and Oleg Chertov. What about interpreting features in matrix factorization-based recommender systems as users? In *HT (Doctoral Consortium/Late-breaking Results/Workshops)*. Citeseer, 2014.

[Aleksandrova *et al.* 2016a] Marharyta Aleksandrova, Armelle Brun, Anne Boyer, and Oleg Chertov. Identifying representative users in matrix factorization-based recommender systems: application to solving the content-less new item cold-start problem. *Journal of Intelligent Information Systems*, pages 1–33, 2016.

[Aleksandrova *et al.* 2016b] Marharyta Aleksandrova, Armelle Brun, Oleg Chertov, and Anne Boyer. Automatic identification of trigger factors: a possibility for chance discovery. In *2nd European Workshop on Chance Discovery and Data Synthesis (EWCDDS16)*, page 13, 2016.

[Aleksandrova *et al.* 2016c] Marharyta Aleksandrova, Armelle Brun, Oleg Chertov, and Anne Boyer. Sets of contrasting rules: A supervised descriptive rule induction pattern for identification of trigger factors. In *Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on*, pages 431–435. IEEE, 2016.

[Aleksandrova *et al.* 2016d] Marharyta Aleksandrova, Armelle Brun, Oleg Chertov, and Anne Boyer. Sets of contrasting rules to identify trigger factors. In *ECAI 2016: 22nd European Conference on Artificial Intelligence*. IOS Press, 2016.

[Algosaibi *et al.* 2015] Abdulelah A Algosaibi, Saleh Albahli, and Austin Melton. World wide web: A survey of its development and possible future trends. In *The 16th International Conference on Internet Computing and Big Data-ICOMP'15*, 2015.

[Alpaydin 2014] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2014.

[Amatriain *et al.* 2009] Xavier Amatriain, Neal Lathia, Josep M Pujol, Haewoon Kwak, and Nuria Oliver. The wisdom of the few: a collaborative filtering approach based on expert opinions from the web. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 532–539. ACM, 2009.

[Arisoy *et al.* 2012] Ebru Arisoy, Tara N Sainath, Brian Kingsbury, and Bhuvana Ramabhadran. Deep neural network language models. In *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, pages 20–28. Association for Computational Linguistics, 2012.

[Atzmueller *et al.* 2009] Martin Atzmueller, Florian Lemmerich, Beate Krause, and Andreas Hotho. Who are the spammers? understandable local patterns for concept description. In *Proc. 7th Conference on Computer Methods and Systems*, 2009.

[Atzmueller *et al.* 2011] Martin Atzmueller, Dominik Benz, Andreas Hotho, and Gerd Stumme. Towards mining semantic maturity in social bookmarking systems. In *Proc. Workshop Social Data on the Web, 10th Intl. Semantic Web Conference*, 2011.

[Atzmueller 2015] Martin Atzmueller. Subgroup discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(1):35–49, 2015.

[Avogadri and Valentini 2009] Roberto Avogadri and Giorgio Valentini. Fuzzy ensemble clustering based on random projections for dna microarray data analysis. *Artificial Intelligence in Medicine*, 45(2):173–183, 2009.

[Bailey *et al.* 2002] James Bailey, Thomas Manoukian, and Kotagiri Ramamohanarao. Fast algorithms for mining emerging patterns. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 39–50. Springer, 2002.

[Baker *et al.* 2001] H Kent Baker, E Theodore Veit, and Gary E Powell. Factors influencing dividend policy decisions of nasdaq firms. *Financial Review*, 36(3):19–38, 2001.

[Basu 2013] ATANU Basu. Five pillars of prescriptive analytics success. *Analytics Magazine*, pages 8–12, 2013.

[Bay and Pazzani 1999] Stephen D Bay and Michael J Pazzani. Detecting change in categorical data: Mining contrast sets. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 302–306. ACM, 1999.

[Bay and Pazzani 2001] Stephen D Bay and Michael J Pazzani. Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5(3):213–246, 2001.

[Behnke 2003] Sven Behnke. Discovering hierarchical speech features using convolutional non-negative matrix factorization. In *Proceedings of the International Joint Conference on Neural Networks*, volume 4, pages 2758–2763. IEEE, 2003.

[Benítez *et al.* 1997] José Manuel Benítez, Juan Luis Castro, and Ignacio Requena. Are artificial neural networks black boxes? *IEEE Transactions on neural networks*, 8(5):1156–1164, 1997.

[Berkovsky and Freyne 2010] Shlomo Berkovsky and Jill Freyne. Group-based recipe recommendations: analysis of data aggregation strategies. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 111–118. ACM, 2010.

[Berners-Lee 1998] Tim Berners-Lee. The world wide web: A very short personal history. https://www.w3.org/People/Berners-Lee/ShortHistory.html, 1998. Accessed: 2016-10-02.

[Bhargava and Shukla 2016] Niket Bhargava and Manoj Shukla. Survey of interestingness measures for association rules mining: Data mining, data science for business perspective. *analysis*, 6(2), 2016.

[Biem 2014] Alain Biem. Neural networks: A review. In *Data Classification Algorithms and Applications*, pages 205–243. Chapman and Hall/CRC, 2014.

[Bingham and Mannila 2001] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250. ACM, 2001.

[Bobadilla *et al.* 2013] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. Recommender systems survey. *Knowledge-Based Systems*, 46:109–132, 2013.

[Bottou 2010] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[Bourdieu 1995] Pierre Bourdieu. Structures, habitus, practices. *Modern social theory: Bourdieu, Giddens, Habermas*, pages 16–36, 1995.

[Bousquet and Bottou 2008] Olivier Bousquet and Léon Bottou. The tradeoffs of large scale learning. In *Advances in neural information processing systems*, pages 161–168, 2008.

[Breese *et al.* 1998] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[Brin *et al.* 1997] Sergey Brin, Rajeev Motwani, Jeffrey D Ullman, and Shalom Tsur. Dynamic itemset counting and implication rules for market basket data. In *ACM SIGMOD Record*, volume 26, pages 255–264. ACM, 1997.

[Brun *et al.* 2014] Armelle Brun, Marharyta Aleksandrova, and Anne Boyer. Can latent features be interpreted as users in matrix factorization-based recommender systems? In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 02*, pages 226–233. IEEE Computer Society, 2014.

[Burke *et al.* 2011] Robin Burke, Michael P O'Mahony, and Neil J Hurley. Robust collaborative recommendation. In *Recommender systems handbook*, pages 805–835. Springer, 2011.

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

[Bylesjö *et al.* 2008] Max Bylesjö, Mattias Rantalainen, Jeremy K Nicholson, Elaine Holmes, and Johan Trygg. K-opls package: Kernel-based orthogonal projections to latent structures for prediction and interpretation in feature space. *BMC bioinformatics*, 9(1):1, 2008.

[Carmagnola *et al.* 2009] Francesca Carmagnola, Fabiana Vernero, and Pierluigi Grillo. Sonars: A social networks-based algorithm for social recommender systems. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 223–234. Springer, 2009.

[Carmona *et al.* 2014] Cristóbal J Carmona, Pedro González, María José del Jesus, and Francisco Herrera. Overview on evolutionary subgroup discovery: analysis of the suitability and potential of the search performed by evolutionary algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(2):87–103, 2014.

[Castagnos 2008] Sylvain Castagnos. *Modélisation de comportements et apprentissage stochastique non supervisé de stratégies d'interactions sociales au sein de systèmes temps réel de recherche et d'accès à l'information.* PhD thesis, Université Nancy II, 2008.

[Chagoyen *et al.* 2006] Monica Chagoyen, Pedro Carmona-Saez, Hagit Shatkay, Jose M Carazo, and Alberto Pascual-Montano. Discovering semantic features in the literature: a foundation for building functional associations. *BMC Bioinformatics*, 7(1):41, 2006.

[Chan 1999] Philip K Chan. A non-invasive learning approach to building web user profiles. 1999.

[Chatterjee 2011] Deblina Chatterjee. *Eager Learning Vs Lazy Learning Methods in Classification.* PhD thesis, Jadavpur University Kolkata, 2011.

[Chen *et al.* 1989] C-C Chen, John S DaPonte, and Martin D Fox. Fractal feature analysis and classification in medical imaging. *IEEE transactions on medical imaging*, 8(2):133–142, 1989.

[Chertov and Aleksandrova 2013] Oleg Chertov and Marharyta Aleksandrova. Fuzzy clustering with prototype extraction for census data analysis. In *Soft Computing: State of the Art Theory and Novel Applications*, pages 289–313. Springer, 2013.

[Chertov and Aleksandrova 2015] Oleg Chertov and Marharyta Aleksandrova. Data mining for habitus elements identification. In *Proceedings of the International scientific conference "'State and global social changes: historical sociology of planning and resistance in modern era"', Kyiv, Ukraine*, 2015.

[Chertov *et al.* 2015] Oleg Chertov, A BOYER, M ALEKSANDROVA, et al. Comparative analysis of neighborhood-based approache and matrix factorization in recommender systems. *Eastern-European Journal of Eenterprise Technologies*, 3(4 (75)), 2015.

[Choongo *et al.* 2016] Progress Choongo, Elco Van Burg, Leo J Paas, and Enno Masurel. Factors influencing the identification of sustainable opportunities by smes: Empirical evidence from zambia. *Sustainability*, 8(1):81, 2016.

[Chung *et al.* 2016] Ki-Sook Chung, Thai Quang Tung, and Chang-sup Keum. A design of smart docent service using hybrid recommenders. In *Ubiquitous and Future Networks (ICUFN), 2016 Eighth International Conference on*, pages 386–389. IEEE, 2016.

[Claypool *et al.* 2001] Mark Claypool, Phong Le, Makoto Wased, and David Brown. Implicit interest indicators. In *Proceedings of the 6th international conference on Intelligent user interfaces*, pages 33–40. ACM, 2001.

[Cremonesi *et al.* 2011] Paolo Cremonesi, Antonio Tripodi, and Roberto Turrin. Cross-domain recommender systems. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 496–503. Ieee, 2011.

[Cruz-Roa *et al.* 2012] Angel Cruz-Roa, Fabio González, Joseph Galaro, Alexander R Judkins, David Ellison, Jennifer Baccon, Anant Madabhushi, and Eduardo Romero. A visual latent semantic approach for automatic analysis and interpretation of anaplastic medulloblastoma virtual slides. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 157–164. Springer, 2012.

[Cybenko 1989] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[Das *et al.* 2007] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th International Conference on World Wide Web*, pages 271–280. ACM, 2007.

[Daugherty *et al.* 2014] Paul Daugherty, Prith Banerjee, Walid Negm, and Allan E Alter. Driving unconventional growth through the industrial internet of things. *New York: Accenture*, 2014.

[Davis and Herschel 2016] Melissa Davis and Gareth Herschel. Understand the spectrum of analytics capabilities. https://www.gartner.com/webinar/3237918?srcId=1-3931087981, 2016. Accessed: 2016-07-22.

[Deitel *et al.* 2002] Harvey M Deitel, Paul J Deitel, and Tem R Nieto. *Internet & world wide web*. Prentice Hall, 2002.

[Deng *et al.* 2014] Hongbo Deng, Yizhou Sun, Yi Chang, and Jiawei Han. Probabilistic models for classification. In *Data Classification Algorithms and Applications*, pages 65–86. Chapman and Hall/CRC, 2014.

[Desrosiers and Karypis 2011a] Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender Systems Handbook*, pages 107–144. Springer, 2011.

[Desrosiers and Karypis 2011b] Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook*, pages 107–144. Springer, 2011.

[Devarajan 2008] Karthik Devarajan. Nonnegative matrix factorization: an analytical and interpretive tool in computational biology. *PLoS Computational Biology*, 4(7):12, 2008.

[Dong and Bailey 2012] Guozhu Dong and James Bailey. *Contrast Data Mining: Concepts, Algorithms, and Applications*. CRC Press, 2012.

[Dong and Li 1999] Guozhu Dong and Jinyan Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 43–52. ACM, 1999.

[Dong *et al.* 1999] Guozhu Dong, Xiuzhen Zhang, Limsoon Wong, and Jinyan Li. Caep: Classification by aggregating emerging patterns. In *International Conference on Discovery Science*, pages 30–42. Springer, 1999.

[Donkers *et al.* 2015a] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. Merging latent factors and tags to increase interactive control of recommendations. In *RecSys Posters*, 2015.

[Donkers *et al.* 2015b] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. Towards understanding latent factors and user profiles by enhancing matrix factorization with tags. *i-com*, 14(1):5–17, 2015.

[Dreiseitl and Ohno-Machado 2002] Stephan Dreiseitl and Lucila Ohno-Machado. Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, 35(5):352–359, 2002.

[Duan 2014] Lei Duan. Understanding the differences among users: Opportunities and challenges for personalized recommendation. 2014.

[Duda *et al.* 2012] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[Eckart and Young 1936] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[Enrich *et al.* 2013] Manuel Enrich, Matthias Braunhofer, and Francesco Ricci. Cold-start management with cross-domain collaborative filtering and tags. In *E-Commerce and Web Technologies*, volume 152, pages 101–112. Springer, 2013.

[Erkin *et al.* 2013] Zekeriya Erkin, Thijs Veugen, and Reginald L Lagendijk. Privacy-preserving recommender systems in dynamic environments. In *2013 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 61–66. IEEE, 2013.

[Esslimani *et al.* 2013] I. Esslimani, A. Brun, and A. Boyer. Towards leader based recommendations. In *The Influence of Technology on Social Network Analysis and Mining*, pages 455–470. Springer, 2013.

[Evans and Lindner 2012] James R Evans and Carl H Lindner. Business analytics: the next frontier for decision sciences. *Decision Line*, 43(2):4–6, 2012.

[Facebook and CtrlShift 2016] Facebook and CtrlShift. The data driven economy: Toward sustainable growth. https://www.dropbox.com/s/2gu2dlv6khgahrp/Report2016. Accessed: 2016-06-24.

[Fayyad *et al.* 1996] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34, 1996.

[Fernández-Tobías *et al.* 2012] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. Cross-domain recommender systems: A survey of the state of the art. In *Spanish Conference on Information Retrieval*, 2012.

[Fischoff *et al.* 1998] Stuart Fischoff, Joe Antonio, and Diane Lewis. Favorite films and film genres as a function of race, age, and gender. *Journal of Media Psychology*, 3(1):1–9, 1998.

[Gamberger *et al.* 2003] Dragan Gamberger, Nada Lavrač, and Goran Krstačić. Active subgroup mining: a case study in coronary heart disease risk group detection. *Artificial Intelligence in Medicine*, 28(1):27–57, 2003.

[Gandomi and Haider 2015] Amir Gandomi and Murtaza Haider. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2):137–144, 2015.

[Gantner *et al.* 2010] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *Proceedings of 2010 IEEE 10th International Conference on Data Mining (ICDM)*, pages 176–185. IEEE, 2010.

[Gemulla *et al.* 2011] Rainer Gemulla, Erik Nijkamp, Peter J Haas, and Yannis Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 69–77. ACM, 2011.

[Geng and Hamilton 2006] Liqiang Geng and Howard J Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys (CSUR)*, 38(3):9, 2006.

[Godoy and Corbellini 2016] Daniela Godoy and Alejandro Corbellini. Folksonomy-based recommender systems: A state-of-the-art review. *International Journal of Intelligent Systems*, 31(4):314–346, 2016.

[Golbeck *et al.* 2006] Jennifer Golbeck, James Hendler, et al. Filmtrust: Movie recommendations using trust in web-based social networks. In *Proceedings of the IEEE Consumer communications and networking conference*, volume 96, pages 282–286. Citeseer, 2006.

[Goreinov *et al.* 2010] SA Goreinov, IV Oseledets, DV Savostyanov, EE Tyrtyshnikov, and NL Zamarashkin. How to find a good submatrix. *Matrix Methods: Theory, Algorithms, Applications, V. Olshevsky and E. Tyrtyshnikov, eds., World Scientific, Hackensack, NY*, pages 247–256, 2010.

[Gröger *et al.* 2014] Christoph Gröger, Holger Schwarz, and Bernhard Mitschang. Prescriptive analytics for recommendation-based business process optimization. In *International Conference on Business Information Systems*, pages 25–37. Springer, 2014.

[Groh *et al.* 2012] Georg Groh, Stefan Birnkammerer, and Valeria Köllhofer. Social recommender systems. In *Recommender Systems for the Social Web*, pages 3–42. Springer, 2012.

[Guermeur *et al.* 2004] Y. Guermeur, A. Lifchitz, and R. Vert. A kernel for protein secondary structure prediction. In *Kernel Methods in Computational Biology*, pages 193–206. MIT Press, 2004.

[Gunnalan *et al.* 2003] Rajesh Gunnalan, Tim Menzies, Kalaivani Appukutty, Amarnath Srinivasan, and Ying Hu. Feature subset selection with tar2less. http://www.academia.edu/2699626/Feature_Subset_Selection_with_TAR2less, 2003. Accessed: 2016-10-01.

[Guo *et al.* 2014] Zhenhai Guo, Dezhong Chi, Jie Wu, and Wenyu Zhang. A new wind speed forecasting strategy based on the chaotic time series modelling technique and the apriori algorithm. *Energy Conversion and Management*, 84:140–151, 2014.

[Guy 2015] Ido Guy. Social recommender systems. In *Recommender Systems Handbook*, pages 511–543. Springer, 2015.

[Guyon and Elisseeff 2003] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[Hagan *et al.* 1996] Martin T Hagan, Howard B Demuth, Mark H Beale, and Orlando De Jesús. *Neural network design*, volume 20. PWS publishing company Boston, 1996.

[Han *et al.* 2011] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques*. Elsevier, 2011.

[Hawkins 2004] Douglas M Hawkins. The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12, 2004.

[Hazel 2015] Davis Hazel. Smaller firms think big data. *Raconteur. The data economy*, (0324):6–7, 2015.

[Hendrickx and Van Den Bosch 2005] Iris Hendrickx and Antal Van Den Bosch. Hybrid algorithms with instance-based classification. In *European Conference on Machine Learning*, pages 158–169. Springer, 2005.

[Herlocker *et al.* 2000] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, pages 241–250. ACM, 2000.

[Hernando *et al.* 2016] Antonio Hernando, Jesús Bobadilla, and Fernando Ortega. A non negative matrix factorization for collaborative filtering recommender systems based on a bayesian probabilistic model. *Knowledge-Based Systems*, 97:188–202, 2016.

[Hillier and Rooksby 2005] Jean Hillier and Emma Rooksby. *Habitus: A sense of place*. Ashgate Aldershot, 2005.

[Holte 1993] Robert C Holte. Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11(1):63–90, 1993.

[Hornik 1991] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[Hougaard *et al.* 2013] Anders Hougaard, Faisal Amin, Anne Werner Hauge, Messoud Ashina, and Jes Olesen. Provocation of migraine with aura using natural trigger factors. *Neurology*, 80(5):428–431, 2013.

[Houlsby *et al.* 2014] Neil Houlsby, Jose M Hernandez-lobato, and Zoubin Ghahramani. Cold-start active learning with robust ordinal matrix factorization. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 766–774, 2014.

[Hu *et al.* 2008] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of 8th IEEE International Conference on Data Mining*, pages 263–272. IEEE, 2008.

[Hu *et al.* 2015] Guang-Neng Hu, Xin-Yu Dai, Yunya Song, Shu-Jian Huang, and Jia-Jun Chen. A synthetic approach for recommendation: combining ratings, social relations, and reviews. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 1756–1762. AAAI Press, 2015.

[Hu 2003] Ying Hu. *Treatment learning: Implementation and application.* PhD thesis, University of British Columbia, 2003.

[Huang *et al.* 2012] Yu-Jia Huang, Evan Wei Xiang, and Rong Pan. Constrained collective matrix factorization. In *Proceedings of the 6th ACM Conference on Recommender Systems*, pages 237–240. ACM, 2012.

[Huang 2011] S. Huang. Designing utility-based recommender systems for e-commerce: evaluation of preference-elicitation methods. *Electronic Commerce Research and Applications*, 10(4):398—-407, 2011.

[IDC 2014] IDC. Discover the digital universe of opportunities: rich data and the increasing value of the internet of things. http://www.emc.com/leadership/digital-universe/index.htm, 2014. Accessed: 2016-03-07.

[Impedovo *et al.* 1991] S Impedovo, L Ottaviano, and S Occhinegro. Optical character recognition—a survey. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(01n02):1–24, 1991.

[Jamali and Ester 2011] Mohsen Jamali and Martin Ester. A transitivity aware matrix factorization model for recommendation in social networks. In *Proceedings of IJCAI, the 22nd International Joint Conference on Artificial Intelligence*, volume 11, pages 2644–2649. Citeseer, 2011.

[John Lu 2010] ZQ John Lu. The elements of statistical learning: data mining, inference, and prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 173(3):693–694, 2010.

[Judd *et al.* 2011] Charles M Judd, Gary H McClelland, and Carey S Ryan. *Data analysis: A model comparison approach.* Routledge, 2011.

[Karatzoglou *et al.* 2010] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the 4th ACM Conference on Recommender Systems*, pages 79–86. ACM, 2010.

[Kelle and Bird 1995] Udo Kelle and Katherine Bird. *Computer-aided qualitative data analysis: Theory, methods and practice.* Sage, 1995.

[Kim and Kim 2003] Choonho Kim and Juntae Kim. A recommendation algorithm using multi-level association rules. In *Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on*, pages 524–527. IEEE, 2003.

[Kim and Yum 2005] Dohyun Kim and Bong-Jin Yum. Collaborative filtering based on iterative principal component analysis. *Expert Systems with Applications*, 28(4):823–830, 2005.

[Klašnja-Milićević *et al.* 2015] Aleksandra Klašnja-Milićević, Mirjana Ivanović, and Alexandros Nanopoulos. Recommender systems in e-learning environments: a survey of the state-of-the-art and possible extensions. *Artificial Intelligence Review*, 44(4):571–604, 2015.

[Koenigstein *et al.* 2011] Noam Koenigstein, Gideon Dror, and Yehuda Koren. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 165–172. ACM, 2011.

[Kohavi and Quinlan 2002] Ronny Kohavi and J Ross Quinlan. Data mining tasks and methods: Classification: decision-tree discovery. In *Handbook of data mining and knowledge discovery*, pages 267–276. Oxford University Press, Inc., 2002.

[Koren *et al.* 2009] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[Koren 2008] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 426–434. ACM, 2008.

[Kotsiantis and Kanellopoulos 2006] Sotiris Kotsiantis and Dimitris Kanellopoulos. Association rules mining: A recent overview. *GESTS International Transactions on Computer Science and Engineering*, 32(1):71–82, 2006.

[Krumeich *et al.* 2015] Julian Krumeich, Dirk Werth, and Peter Loos. Prescriptive control of business processes. *Business & Information Systems Engineering*, pages 1–20, 2015.

[Kvalheim and Karstang 1989] Olav M Kvalheim and Terje V Karstang. Interpretation of latent-variable regression models. *Chemometrics and intelligent laboratory systems*, 7(1):39–51, 1989.

[Lam *et al.* 2008] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*, pages 208–211. ACM, 2008.

[Lavrač *et al.* 2004] Nada Lavrač, Branko Kavšek, Peter Flach, and Ljupčo Todorovski. Subgroup discovery with cn2-sd. *The Journal of Machine Learning Research*, 5:153–188, 2004.

[Lee and Cho 2015] HJ Mikyoung Lee and Minhee Cho. On a hadoop-based analytics service system. *International Journal Advance Soft Computing Applications*, 7(1):1–8, 2015.

[Lee and Seung 1999] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[Lee and Seung 2001] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Proceedings of Advances in Neural Information Processing Systems*, pages 556–562, 2001.

[Lee *et al.* 2008] Tong Queue Lee, Young Park, and Yong-Tae Park. A time-based approach to effective recommender systems using implicit feedback. *Expert systems with applications*, 34(4):3055–3062, 2008.

[Lee *et al.* 2014] Victor E. Lee, Lin Liu, and Ruoming Jin. Decision trees: Theory and algorithms. In *Data Classification Algorithms and Applications*, pages 87–120. Chapman and Hall/CRC, 2014.

[Lenca *et al.* 2008] Philippe Lenca, Patrick Meyer, Benoit Vaillant, and Stéphane Lallich. On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid. *European journal of operational research*, 184(2):610–626, 2008.

[Letham *et al.* 2012] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, and David Madigan. Building interpretable classifiers with rules using bayesian analysis. *Department of Statistics Technical Report tr609, University of Washington*, 2012.

[Li and Li 2012] Lei Li and Tao Li. Meet: a generalized framework for reciprocal recommender systems. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 35–44. ACM, 2012.

[Li and Liu 2014] Xiao-Li Li and Bing Liu. Rule-based classification. In *Data Classification Algorithms and Applications*, pages 121–156. Chapman and Hall/CRC, 2014.

[Liu and Dong 2009] Qingbao Liu and Guozhu Dong. A contrast pattern based clustering quality index for categorical data. In *2009 Ninth IEEE International Conference on Data Mining*, pages 860–865. IEEE, 2009.

[Liu and Motoda 1998] Huan Liu and Hiroshi Motoda. *Feature extraction, construction and selection: A data mining perspective*. Springer Science & Business Media, 1998.

[Liu *et al.* 2011] Nathan N Liu, Xiangrui Meng, Chao Liu, and Qiang Yang. Wisdom of the better few: cold start recommendation via representative based rating elicitation. In *Proceedings of the 5th ACM Conference on Recommender Systems*, pages 37–44. ACM, 2011.

[Loekito and Bailey 2006] Elsa Loekito and James Bailey. Fast mining of high dimensional expressive contrast patterns using zero-suppressed binary decision diagrams. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 307–316. ACM, 2006.

[Lops *et al.* 2011] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.

[Lufthansa 2016] Lufthansa. Lufthansa focuses on big data and analytics technology. https://www.lufthansagroup.com/en/press/news-releases/singleview/archive/2016/march/10/article/3960.html, 2016. Accessed: 2016-07-17.

[Ma *et al.* 2011] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, pages 287–296. ACM, 2011.

[Ma 1998] Bing Liu Wynne Hsu Yiming Ma. Integrating classification and association rule mining. In *Proceedings of the fourth international conference on knowledge discovery and data mining*, 1998.

[Manning and Schütze 1999] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*, volume 999. MIT Press, 1999.

[Mao and Saul 2004] Yun Mao and Lawrence K Saul. Modeling distances in large-scale networks by matrix factorization. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 278–287. ACM, 2004.

[Masthoff 2011] Judith Masthoff. Group recommender systems: Combining individual models. In *Recommender systems handbook*, pages 677–702. Springer, 2011.

[McAuley and Leskovec 2013] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 165–172. ACM, 2013.

[McCallum *et al.* 1998] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

[Melville *et al.* 2002] Prem Melville, Raymod J. Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 187–192. American Association for Artificial Intelligence, 2002.

[Menzies *et al.* 2003] Tim Menzies, Eliza Chiang, Martin Feather, Ying Hu, and James D Kiper. Condensing uncertainty via incremental treatment learning. In *Software Engineering with Computational Intelligence*, pages 319–361. Springer, 2003.

[Miller *et al.* 2015] William L Miller, Ryan S Baker, Matthew J Labrum, Karen Petsche, Yu-Han Liu, and Angela Z Wagner. Automated detection of proactive remediation by teachers in reasoning mind classrooms. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 290–294. ACM, 2015.

[Mnih and Salakhutdinov 2007] Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *Proceedings of Advances in Neural Information Processing Systems, 2007*, pages 1257–1264, 2007.

[Momma and Bennett 2006] Michinari Momma and Kristin P Bennett. Constructing orthogonal latent features for arbitrary loss. In *Feature Extraction*, pages 551–583. Springer, 2006.

[Moore and Zuev 2005] Andrew W Moore and Denis Zuev. Internet traffic classification using bayesian analysis techniques. In *ACM SIGMETRICS Performance Evaluation Review*, volume 33, pages 50–60. ACM, 2005.

[Murphy 2006] Kevin P Murphy. Naive bayes classifiers. *University of British Columbia*, 2006.

[Nanavati *et al.* 2001] Amit A Nanavati, Krishna P Chitrapura, Sachindra Joshi, and Raghu Krishnapuram. Mining generalised disjunctive association rules. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 482–489. ACM, 2001.

[Novak *et al.* 2009a] Petra Kralj Novak, Nada Lavrač, Dragan Gamberger, and Antonija Krstačić. Csm-sd: Methodology for contrast set mining through subgroup discovery. *Journal of Biomedical Informatics*, 42(1):113–122, 2009.

[Novak *et al.* 2009b] Petra Kralj Novak, Nada Lavrač, and Geoffrey I Webb. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *The Journal of Machine Learning Research*, 10:377–403, 2009.

[Novak 2009] Petra Kralj Novak. *SUPERVISED DESCRIPTIVE RULE INDUCTION*. PhD thesis, Jozef Stefan International Postgraduate School, 2009.

[Oard *et al.* 1998] Douglas W Oard, Jinmook Kim, et al. Implicit feedback for recommender systems. In *Proceedings of the AAAI workshop on recommender systems*, pages 81–83, 1998.

[O'Donovan and Smyth 2005] John O'Donovan and Barry Smyth. Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces*, pages 167–174. ACM, 2005.

[Ohsawa 2006] Yukio Ohsawa. Chance discovery: The current states of art. In *Chance discoveries in real world decision making*, pages 3–20. Springer, 2006.

[Oracle and FSN 2012] Oracle and FSN. Mastering big data: Cfo strategies to transform insight into opportunit. http://www.oracle.com/us/solutions/ent-performance-bi/business-intelligence/mastering-big-data-cfo-strategies-1853061.pdf, 2012. Accessed: 2016-06-24.

[Ordonez *et al.* 2001] Carlos Ordonez, Edward Omiecinski, Levien De Braal, Cesar A Santana, Norberto Ezquerra, Jose A Taboada, David Cooke, Elizabeth Krawczynska, and Eenest V Garcia. Mining constrained association rules to predict heart disease. In *icdm*, page 433. IEEE, 2001.

[Ordonez *et al.* 2006] Carlos Ordonez, Norberto Ezquerra, and Cesar A Santana. Constraining and summarizing association rules in medical data. *Knowledge and Information Systems*, 9(3):1–2, 2006.

[Ortega *et al.* 2014] Fernando Ortega, Jesús Bobadilla, Antonio Hernando, and Fernando Rodríguez. Using hierarchical graph maps to explain collaborative filtering recommendations. *International Journal of Intelligent Systems*, 29(5):462–477, 2014.

[Panigrahi *et al.* 2016] Sasmita Panigrahi, Rakesh Ku Lenka, and Ananya Stitipragyan. A hybrid distributed collaborative filtering recommender engine using apache spark. *Procedia Computer Science*, 83:1000–1006, 2016.

[Park and Chu 2009] Seung-Taek Park and Wei Chu. Pairwise preference regression for cold-start recommendation. In *Proceedings of the ACM Conference on Recommender Systems (RecSys)*, pages 21–28. ACM, 2009.

[Park and Tuzhilin 2008] Yoon-Joo Park and Alexander Tuzhilin. The long tail of recommender systems and how to leverage it. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 11–18. ACM, 2008.

[Park *et al.* 2012] Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, and Jae Kyeong Kim. A literature review and classification of recommender systems research. *Expert Systems with Applications*, 39(11):10059–10072, 2012.

[Patel 2013] Karan Patel. Incremental journey for world wide web: introduced with web 1.0 to recent web 5.0–a survey paper. *International Journal*, 3(10), 2013.

[Pauca *et al.* 2004] V Paul Pauca, Farial Shahnaz, Michael W Berry, and Robert J Plemmons. Text mining using non-negative matrix factorizations. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, pages 452–456. SIAM, 2004.

[Pessiot *et al.* 2006] Jean-François Pessiot, Tuong-Vinh Truong, Nicolas Usunier, Massih-Reza Amini, and Patrick Gallinari. Factorisation en matrices non-négatives pour le filtrage collaboratif. In *Proceedings of 3rd Conference en Recherche d'Information et Applications*, pages 315–326, 2006.

[Petra 2009] Kralj Novak Petra. *Supervised Descriptive Rule Induction*. PhD thesis, Jozef Stefan International Postgraduate School, 2009.

[Pizzato *et al.* 2010] Luiz Pizzato, Tomek Rej, Thomas Chung, Irena Koprinska, and Judy Kay. Recon: a reciprocal recommender for online dating. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 207–214. ACM, 2010.

[Pudil and Novovičová 1998] Pavel Pudil and Jana Novovičová. Novel methods for feature subset selection with respect to problem knowledge. In *Feature Extraction, Construction and Selection*, pages 101–116. Springer, 1998.

[Quinlan 2014] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[Ramamohanarao *et al.* 2005] Kotagiri Ramamohanarao, James Bailey, and Hongjian Fan. Efficient mining of contrast patterns and their applications to classification. In *2005 3rd International Conference on Intelligent Sensing and Information Processing*, pages 39–47. IEEE, 2005.

[Rao 2008] K Nageswara Rao. Application domain and functional classification of recommender systems–a survey. *DESIDOC Journal of Library & Information Technology*, 28(3):17, 2008.

[Rashid *et al.* 2002] Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K Lam, Sean M McNee, Joseph A Konstan, and John Riedl. Getting to know you: learning new user preferences in recommender systems. In *Proceedings of the 7th International Conference on Intelligent User Interfaces*, pages 127–134. ACM, 2002.

[Rashid 2007] Al Mamunur Rashid. *Mining influence in recommender systems*. PhD thesis, University of Minnesota, 2007.

[Ricci *et al.* 2011] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Introduction to recommender systems handbook*. Springer, 2011.

[Rich 1979] Elaine Rich. User modeling via stereotypes. *Cognitive science*, 3(4):329–354, 1979.

[Rich 1989] Elaine Rich. Stereotypes and user modeling. In *User models in dialog systems*, pages 35–51. Springer, 1989.

[Rish 2001] Irina Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM New York, 2001.

[Rossetti 2014] Marco Rossetti. *Advancing recommender systems from the algorithm, interface and methodological perspective*. PhD thesis, Università Degli Studi di Milano - BICOCCA, 2014.

[Roweis and Saul 2000] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[Salakhutdinov and Mnih 2008] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning*, pages 880–887, 2008.

[Salakhutdinov *et al.* 2007] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798. ACM, 2007.

[Sarwar *et al.* 2000a] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Application of dimensionality reduction in recommender system-a case study. Technical report, DTIC Document, 2000.

[Sarwar *et al.* 2000b] B.M. Sarwar, G. Karypis, J.A. Konstan, and Riedl J.T. Application of dimensionality reduction in recommender system, a case study. In *Proceedings of ACM WebKDD 2000 Web Mining for E-Commerce Workshop*, page 12, 2000.

[Sarwar *et al.* 2002] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Fifth International Conference on Computer and Information Science*, pages 27–28. Citeseer, 2002.

[Saveski 2013] M. Saveski. Cold start recommendations: a non-negative matrix factorization approach. Master's thesis, Universidad Politecnica de Cataluna, 2013.

[Schafer *et al.* 2007] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The Adaptive Web*, pages 291–324. Springer, 2007.

[Schmidhuber 2015] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[Schutt and O'Neil 2013] Rachel Schutt and Cathy O'Neil. *Doing data science: Straight talk from the frontline.* " O'Reilly Media, Inc.", 2013.

[Segal and Kephart 2000] Richard B Segal and Jeffrey O Kephart. Incremental learning in swiftfile. In *ICML*, pages 863–870, 2000.

[Seminario and Wilson 2014] Carlos E Seminario and David C Wilson. Assessing impacts of a power user attack on a matrix factorization collaborative recommender system. In *Proceedings of The 27th International Florida Artificial Intelligence Research Society Conference*, pages 81–86, 2014.

[Shani and Gunawardana 2011] G. Shani and A. Gunawardana. Evaluating recommendation systems. In *Recommender Systems Handbook*, pages 257–297. Springer, 2011.

[Sharma *et al.* 2016] Kshitij Sharma, Hamed S Alavi, Patrick Jermann, and Pierre Dillenbourg. A gaze-based learning analytics model: in-video visual feedback to improve learner's attention in moocs. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 417–421. ACM, 2016.

[Shibata *et al.* 2009] Ai Shibata, Koichiro Oka, Yoshio Nakamura, and Isao Muraoka. Prevalence and demographic correlates of meeting the physical activity recommendation among japanese adults. *Journal of physical activity & health*, 6(1):24, 2009.

[Shmilovici 2005] Armin Shmilovici. Support vector machines. In *Data Mining and Knowledge Discovery Handbook*, pages 257–276. Springer, 2005.

[Simon and Zha 2000] Horst D Simon and Hongyuan Zha. Low-rank matrix approximation using the lanczos bidiagonalization process with applications. *SIAM Journal on Scientific Computing*, 21(6):2257–2274, 2000.

[Singh and Gordon 2008] Ajit P Singh and Geoffrey J Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 650–658. ACM, 2008.

[Singh 2013] Arvind Singh. Is big data the new black gold? https://www.wired.com/insights/2013/02/is-big-data-the-new-black-gold/, 2013. Accessed: 2016-06-24.

[Sinha and Swearingen 2002] Rashmi Sinha and Kirsten Swearingen. The role of transparency in recommender systems. In *Proceedings of CHI'02 Extended Abstracts on Human Factors in Computing Systems*, pages 830–831. ACM, 2002.

[Snyman 2005] Jan Snyman. *Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms*, volume 97. Springer Science & Business Media, 2005.

[Song *et al.* 2014] Sa-kwang Song, Do-Heon Jeong, Jinhyung Kim, Myunggwon Hwang, Jangwon Gim, and Hanming Jung. Research advising system based on prescriptive analytics. In *Future Information Technology*, pages 569–574. Springer, 2014.

[Spivack 2006] Nova Spivack. Web 3.0: The third generation web is coming. http://lifeboat.com/ex/web.3.0, 2006. Accessed: 2016-10-04.

[Srikant *et al.* 1997] Ramakrishnan Srikant, Quoc Vu, and Rakesh Agrawal. Mining association rules with item constraints. In *KDD*, volume 97, page 67, 1997.

[Sun *et al.* 2013] Mingxuan Sun, Fuxin Li, Joonseok Lee, Ke Zhou, Guy Lebanon, and Hongyuan Zha. Learning multiple-question decision trees for cold-start recommendation. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, pages 445–454. ACM, 2013.

[Sundgot-Borgen 1994] Jorunn Sundgot-Borgen. Risk and trigger factors for the development of eating disorders in female elite athletes. *Medicine & Science in Sports & Exercise*, 26(4):414–419, 1994.

[Takacs *et al.* 2009] Gabor Takacs, Istvan Pilaszy, Bottyan Nemeth, and Domonkos Tikk. Scalable collaborative filtering approaches for large recommender systems. *Journal of Machine Learning Research*, 10:623–656, 2009.

[Tan *et al.* 2004] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right objective measure for association analysis. *Information Systems*, 29(4):293–313, 2004.

[Techapichetvanich and Datta 2004] Kesaraporn Techapichetvanich and Amitava Datta. Visual mining of market basket association rules. In *Computational Science and Its Applications–ICCSA 2004*, pages 479–488. Springer, 2004.

[Tew *et al.* 2014] C Tew, C Giraud-Carrier, K Tanner, and S Burton. Behavior-based clustering and analysis of interestingness measures for association rule mining. *Data Mining and Knowledge Discovery*, 28(4):1004–1045, 2014.

[Tinghuai *et al.* 2015] MA Tinghuai, ZHOU Jinjuan, TANG Meili, TIAN Yuan, AL-DHELAAN Abdullah, AL-RODHAAN Mznah, and LEE Sungyoung. Social network and tag sources

based augmenting collaborative recommender system. *IEICE transactions on Information and Systems*, 98(4):902–910, 2015.

[Tintarev and Masthoff 2011] Nava Tintarev and Judith Masthoff. Designing and evaluating explanations for recommender systems. In *Recommender Systems Handbook*, pages 479–510. Springer, 2011.

[Tintarev and Masthoff 2012] Nava Tintarev and Judith Masthoff. Evaluating the effectiveness of explanations for recommender systems. *User Modeling and User-Adapted Interaction*, 22(4-5):399–439, 2012.

[Trewin 2000] Shari Trewin. Knowledge-based recommender systems. *Encyclopedia of library and information science*, 69(Supplement 32):180, 2000.

[Ubaldi 2013] Barbara Ubaldi. Open government data: Towards empirical analysis of open government data initiatives. *OECD Working Papers on Public Governance*, (22):0_1, 2013.

[U.S.CensusBureau 2000] U.S.CensusBureau. Public use microdata sample. 2000 census of population and housing. https://www.census.gov/prod/cen2000/doc/pums.pdf, 2000. Accessed: 2016-05-20.

[Verbert *et al.* 2012] K. Verbert, N. Manouselis, X. Ochoa, and M. Wolpers. Context-aware recommender systems for learning: a survey and future challenges. *IEEE Transactions on Learning Technologies*, 5(4):318–335, 2012.

[Wang and Lin 2014] Po-Wei Wang and Chih-Jen Lin. Support vector machines. In *Data Classification Algorithms and Applications*, pages 187–204. Chapman and Hall/CRC, 2014.

[Wang *et al.* 2007] Qiong Wang, George M Garrity, James M Tiedje, and James R Cole. Naive bayesian classifier for rapid assignment of rrna sequences into the new bacterial taxonomy. *Applied and environmental microbiology*, 73(16):5261–5267, 2007.

[Wang *et al.* 2015] Miao Wang, Li Chen, Yanjun Huang, Lei Zhang, Zihao Zhang, Jie Ding, and Huiliang Shang. The application characteristics of traditional chinese medical science treatment on vertigo based on data mining apriori algorithm. *International Journal of Wireless and Mobile Computing*, 9(4):349–354, 2015.

[Webb and Zhang 2005] Geoffrey I Webb and Songmao Zhang. K-optimal rule discovery. *Data Mining and Knowledge Discovery*, 10(1):39–79, 2005.

[Webb *et al.* 2003] Geoffrey I Webb, Shane Butler, and Douglas Newlands. On detecting differences between groups. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 256–265. ACM, 2003.

[Weber *et al.* 2014] Jens Weber, Min-Hee Cho, Mikyoung Lee, Sa-Kwang Song, Michaela Geierhos, and Hanmin Jung. System thinking: Crafting scenarios for prescriptive analytics. In *IPaMin@ KONVENS*, 2014.

[Webster 2014] Frank Webster. *Theories of the information society.* Routledge, 2014.

[Wrobel 1997] Stefan Wrobel. An algorithm for multi-relational discovery of subgroups. In *Principles of Data Mining and Knowledge Discovery*, pages 78–87. Springer, 1997.

[Xia *et al.* 2006] Zhonghang Xia, Yulin Dong, and Guangming Xing. Support vector machines for collaborative filtering. In *Proceedings of the 44th annual Southeast regional conference*, pages 169–174. ACM, 2006.

[Xia *et al.* 2012] Feng Xia, Laurence T Yang, Lizhe Wang, and Alexey Vinel. Internet of things. *International Journal of Communication Systems*, 25(9):1101, 2012.

[Xie *et al.* 2013] Y. Xie, Z. Chen, K. Zhang, C. Jin, Y. Cheng, A. Agrawal, and A. Choudhary. Elver: recommending facebook pages in cold start situation without content features. In *Proceedings of 2013 IEEE International Conference on Big Data*, pages 475–479, 2013.

[Xin *et al.* 2015] Xin Xin, Chin-Yew Lin, Xiao-Chi Wei, and He-Yan Huang. When factorization meets heterogeneous latent topics: an interpretable cross-site recommendation framework. *Journal of Computer Science and Technology*, 30(4):917–932, 2015.

[Yao and Hamilton 2006] Hong Yao and Howard J Hamilton. Mining itemset utilities from transaction databases. *Data & Knowledge Engineering*, 59(3):603–626, 2006.

[Yoo and Choi 2009] J. Yoo and S. Choi. Weighted nonnegative matrix co-tri-factorization for collaborative prediction. *Advances in Machine Learning*, 5828:396–411, 2009.

[Yu *et al.* 2014] Hsiang-Fu Yu, Cho-Jui Hsieh, Si Si, and Inderjit S Dhillon. Parallel matrix factorization for recommender systems. *Knowledge and Information Systems*, 41(3):793–819, 2014.

[Zaki *et al.* 1997] Mohammed Javeed Zaki, Srinivasan Parthasarathy, Mitsunori Ogihara, Wei Li, et al. New algorithms for fast discovery of association rules. In *KDD*, volume 97, pages 283–286, 1997.

[Zaki 2000] Mohammed J Zaki. Generating non-redundant association rules. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 34–43. ACM, 2000.

[Zanker *et al.* 2008] M. Zanker, M. Fuchs, W. Höpken, M. Tuta, and N. Muller. Evaluating recommender systems in tourism a case study from austria. In *Information and Communication Technologies in Tourism 2008*, pages 24–34. Springer, 2008.

[Zhan *et al.* 2010] Justin Zhan, Chia-Lung Hsieh, I-Cheng Wang, Tsan-Sheng Hsu, Churn-Jung Liau, and Da-Wei Wang. Privacy-preserving collaborative recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(4):472–476, 2010.

[Zhang *et al.* 2006] Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *Proceedings of the 6th SIAM Conference on Data Mining*, volume 6, pages 548–552, 2006.

[Zhang *et al.* 2010] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52, 2010.

[Zhang *et al.* 2014] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 83–92. ACM, 2014.

[Zhang 2004] Harry Zhang. The optimality of naive bayes. *AA*, 1(2):3, 2004.

[Zhao *et al.* 2015] Dan Zhao, Junyi Wang, Andi Gao, and Pengfei Yue. Learning to recommend with hidden factor models and social trust ensemble. In *Proceedings of International Conference on Computer Science and Intelligent Communication*, pages 87–91, 2015.

[Zhou *et al.* 2008] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. In *Algorithmic Aspects in Information and Management*, pages 337–348. Springer, 2008.

[Zhou *et al.* 2011] Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. Functional matrix factorizations for cold-start recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 315–324. ACM, 2011.

[Zhuang *et al.* 2013] Yong Zhuang, Wei-Sheng Chin, Yu-Chin Juan, and Chih-Jen Lin. A fast parallel sgd for matrix factorization in shared memory systems. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 249–256. ACM, 2013.

Résumé

Dans de nombreux domaines, les données peuvent être de grande dimension. Ça pose le problème de la réduction de dimension. Les techniques de réduction de dimension peuvent être classées en fonction de leur but : techniques pour la *représentation optimale* et techniques pour la *classification*, ainsi qu'en fonction de leur stratégie : la *sélection* et *l'extraction* des caractéristiques. L'ensemble des caractéristiques résultant des méthodes d'extraction est non interprétable. Ainsi, la première problématique scientifique de la thèse est **comment extraire des caractéristiques latentes interprétables**? La *réduction de dimension pour la classification* vise à améliorer la puissance de classification du sous-ensemble sélectionné. Nous voyons le développement de la tâche de classification comme la *tâche d'identification des facteurs déclencheurs*, c'est-à-dire des facteurs qui peuvent influencer le transfert d'éléments de données d'une classe à l'autre. La deuxième problématique scientifique de cette thèse est **comment identifier automatiquement ces facteurs déclencheurs**? Nous visons à résoudre les deux problématiques scientifiques dans le domaine d'application des systèmes de recommandation. Nous proposons d'interpréter les caractéristiques latentes de systèmes de recommandation basés sur la factorisation de matrices comme des utilisateurs réels. Nous concevons un algorithme d'identification automatique des facteurs déclencheurs basé sur les concepts d'analyse par contraste. Au travers d'expérimentations, nous montrons que les motifs définis peuvent être considérés comme des facteurs déclencheurs.

Mots-clés: fouille de données, factorisation de matrices, systeme de recommandation.

Abstract

In many application areas, data elements can be high-dimensional. This raises the problem of dimensionality reduction. The dimensionality reduction techniques can be classified based on their aim: dimensionality reduction for optimal data representation and dimensionality reduction for classification, as well as based on the adopted strategy: feature selection and feature extraction. The set of features resulting from feature extraction methods is usually uninterpretable. Thereby, the first scientific problematic of the thesis is how to extract interpretable latent features? The dimensionality reduction for classification aims to enhance the classification power of the selected subset of features. We see the development of the task of classification as the task of trigger factors identification that is identification of those factors that can influence the transfer of data elements from one class to another. The second scientific problematic of this thesis is how to automatically identify these trigger factors? We aim at solving both scientific problematics within the recommender systems application domain. We propose to interpret latent features for the matrix factorization-based recommender systems as real users. We design an algorithm for automatic identification of trigger factors based on the concepts of contrast analysis. Through experimental results, we show that the defined patterns indeed can be considered as trigger factors.

Keywords: data mining, matrix factorization, recommender systems.