Towards next generation recommender systems through generic data quality

Wissam Al Jurdi

UBFC
UNIVERSITÉ
BOURGOGNE FRANCHE-COMTÉ

UNIVERSITÉ DE FRANCHE-COMTÉ

THÈSE DE DOCTORAT

DE L'ÉTABLISSEMENT UNIVERSITÉ BOURGOGNE-FRANCHE-COMTÉ

PRÉPARÉE À L'UNIVERSITÉ DE FRANCHE-COMTÉ

École doctorale n°37

Sciences Pour l'Ingénieur et Microtechniques

Doctorat d'Informatique

par

WISSAM AL JURDI

Towards Next Generation Recommender Systems through Generic Data Quality

Vers des Systèmes de Recommandation de Nouvelle Génération grâce à la Qualité des Données Génériques

Thèse présentée et soutenue à l'IUT Nord Franche-Comté, le 26 Mars 2024

Composition du Jury :

Mme. MOKDAD LYNDA	Professeur à l'Université de Paris-est Créteil	Rapporteure
M. KARRAY HEDI	Professeur à l'Université de Technologie de Tarbes	Rapporteur
Mme. KABACHI NADIA	Maître de conférences à l'Université Lyon 1	Examinatrice
M. BOU ABDO JACQUES	Professeur assistant à l'Université de Cincinnati	Examinateur
M. BOURGEOIS JULIEN	Professeur à l'Université de Franche-Comté	Examinateur
M. MAKHOUL ABDALLAH	Professeur à l'Université de Franche-Comté	Directeur de thèse
M. DEMERJIAN JACQUES	Professeur à l'Université Libanaise	Codirecteur de thèse

N° | X | X | X |

ACKNOWLEDGEMENTS

Completing a PhD can be daunting, and many people find the journey quite arduous. However, I was fortunate to have the support of many amazing and caring individuals who helped me navigate the challenges and stay focused on my goals. Their encouragement, guidance, and steadfast belief in me made all the difference, and I am grateful for their presence in my life.

I am thankful to the jury members, Prof. Lynda Mokdad, Prof. Hedi Karray, Prof. Jacques Bou Abdo, and Prof. Julien Bourgeois, for taking the time to read the manuscript and providing me with valuable feedback. I am also very grateful to my thesis advisors, Prof. Abdallah Makhoul and Prof. Jacques Demerjian, for providing me with the opportunity to work on this research and for their continuous support, effort, and encouragement throughout the research period.

I would like to express my sincere gratitude to Prof. Jacques Bou Abdo for his unwavering trust and support, which have been the driving force behind my pursuit of a PhD. He has been an invaluable sounding board for all the ideas presented in this thesis, and working alongside him has been an absolute pleasure. He is not only an excellent colleague but also a dear friend.

I am incredibly grateful to my parents, family members, relatives, and friends for their consistent backing. I have had the privilege of meeting many brilliant colleagues, especially Dr. Mira Bou Saleh, whose continuous encouragement kept me motivated throughout this journey.

Finally, I am extremely grateful to my wife, Nataly Dalal, for her unwavering support, especially throughout the completion of this thesis. Her presence has profoundly impacted me and kept me motivated during challenging times. She has been my voice of reason and provided valuable technical support, contributing significantly to some of the ideas and implementations in my thesis. I appreciate all the moral and intellectual support she has provided me.

Contents

I

Context and Motivation

1

INTRODUCTION

1.1/ MOTIVATION AND CONTRIBUTIONS

In today's digital age, abundant online content has made recommender systems more vital. Recommendation systems have proven to be highly efficient tools for filtering online information. They can be broadly categorized into three types: content-based, collaborative, and hybrid Roy and Dutta (2022); Ricci et al. (2010). Content-based systems recommend items based on item features and user preferences. Collaborative approaches, on the other hand, offer items based on user ratings and the ratings of other users. Hybrid systems, as the name suggests, combine content-based and collaborative approaches to overcome the limitations and challenges of each, such as scalability, cold-start, and sparsity.

In summary, recommendation systems have become an essential tool for businesses and individuals alike, as they help to streamline decision-making processes and enhance user experience. In the e-commerce field Beel and Dinesh (2017), recommenders help customers find relevant products and services, increase customer satisfaction, and boost sales and revenue Roy and Dutta (2022). According to various statistics, recommender systems account for a significant portion of e-commerce site revenues. For example, a report by Barilliance Serrano (2023) found that personalized product recommendations account for 31% of e-commerce site revenues. Another study by Salesforce Skovhøj (2022) showed that product recommendations make up for 24% of orders and 26% of revenue while accounting for only 7% of e-commerce traffic. These numbers demonstrate the power and impact of recommender systems in e-commerce.

While recommenders are widely used and have numerous advantages, they have limitations that can affect their effectiveness. These limitations include not emphasizing enhancing factors like serendipity, managing noisy datasets, and utilizing more comprehensive metrics to evaluate the systems beyond just precision.

Serendipity (i.e., chance discoveries) in recommender systems can be defined as dis-

covering relevant, novel, and unexpected items for a user. That leads to positive outcomes, such as satisfaction, learning, or creativity Kotkov et al. (2023). Serendipity is often considered a desirable property of recommender systems, as it can increase user engagement, diversity, and interest Ziarani and Ravanmehr (2021). However, it is also a complex and subjective concept that depends on the user's goals, preferences, and context Yan (2020). Therefore, measuring and optimizing uncertainty in recommender systems is challenging and requires a deeper understanding of serendipitous recommendations' item characteristics and behavioral impact.

One of the factors that may influence uncertainty in recommender systems is natural noise, one of several types of noise that affect recommenders. Natural noise refers to the random or unpredictable variations in the data of the environment that affect the recommendation process Kotkov et al. (2020). Natural noise can sometimes be seen as a source of serendipity, as it can introduce some level of diversity and unexpectedness in the recommendations, which may lead to pleasant surprises for the user Al Jurdi et al. (2018); Ziarani and Ravanmehr (2021). However, natural noise can also be detrimental to serendipity, as it can reduce the relevance and usefulness of the recommendations, which may frustrate the user or make them lose trust in the system Liu et al. (2014). Therefore, it is essential to design recommender systems that can balance the trade-off between natural noise and serendipity and adapt to the user's noise tolerance and serendipity preference.

The primary objective of this thesis is to highlight the importance of serendipity in enhancing the potential of recommenders and enabling users to escape filter bubbles, i.e., receive redundant data that eventually decreases interest and harms the system. The first proposal suggests implementing serendipity-aware techniques to effectively tackle noise in recommender systems and maintain serendipity within reasonable limits to achieve this goal. It also introduces the topic of noise in the datasets and how it's linked to uncertainty.

Inconsistent user information caused by natural noise can negatively impact the performance of recommender systems despite their advanced algorithms Amatriain et al. (2009a); Martínez et al. (2016). This can be detrimental to the quality of recommendations as it directly affects the building blocks of model training. The second objective of the thesis is to describe the structure of noise management algorithms and identify significant flaws in the field. It also emphasizes the significance of an improved assessment framework through several experiments that show how a robust system can be easily infiltrated.

A significant issue with recommender databases is the difficulty in detecting noise, as argued by Martínez et al. (2016); Luo et al. (2023). This noise is caused by natural human behavior, but current evaluation methods only focus on the best-performing recommenders. As a result, critical user-oriented factors like serendipity, diversity, and engage-

ment are often overlooked when building and evaluating these systems. In the third goal of this thesis, we propose a new approach to evaluating and assessing the performance of recommenders that is user-centric and avoids the issues of natural noise and other similar factors that are often overlooked and silently affect the performance. This goal is achieved through contemporary data clustering techniques to identify hidden issues and provide a unique evaluation approach.

Powerful recommendations consider the diverse profiles of users of the systems, eliminate the effect of noise in the datasets and ensure that users get the best out of vast data catalogs by effectively presenting them with what matters to them. This thesis's final goal is a community-based architecture proposal to construct an effective recommendation mechanism that attains better performance and user engagement scores.

1.2/ THESIS CONTEXT

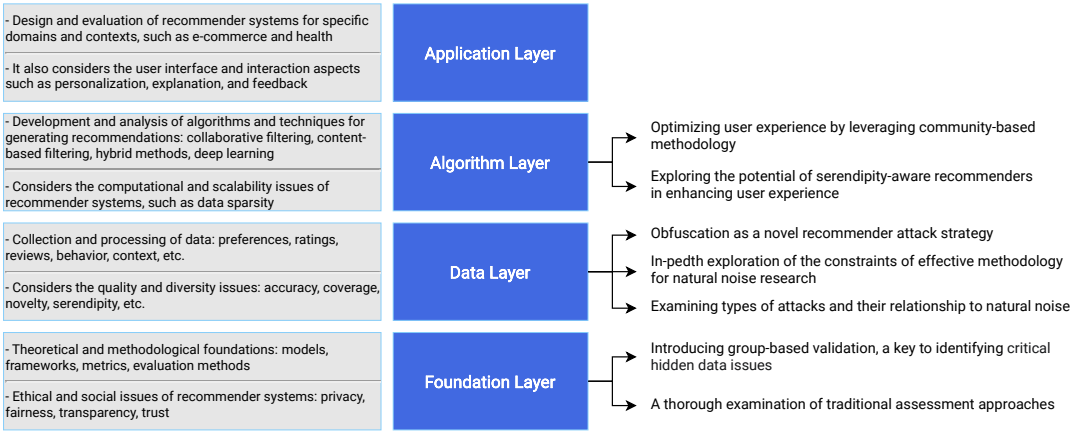

Figure 1.1: A Layered Model for Organizing Recommender Systems Research.

The study of recommender systems encompasses various areas, including information filtering, personalization, and user modeling. Computer science, social science, and domain knowledge are some contributing disciplines, as noted in Ricci et al. (2010); Aggarwal et al. (2016). To create a well-organized framework for this thesis and structure these fields, we developed a possible theoretical layered model that resembles the OSI model used in network communication, a concept touched on in Roy and Dutta (2022). Each layer in this model represents a specific level of abstraction and functionality within recommender systems. In this thesis, we address three layers (out of the four) in the proposed model illustrated in Fig. 1.1:

– In the foundation layer, we present a novel method of group-based validation, enabling us to detect and resolve crucial data problems often overlooked by conventional methods. This method is based on a comprehensive analysis of the strengths

and limitations of existing assessment approaches.

– In the data layer, we thoroughly examine the types of attacks on recommenders and highlight their relationship to natural noise. Further, we conduct an in-depth exploration of the constraints of effective methodology for natural noise research and finally propose ”obfuscation” as a novel recommender attack strategy.
– In the algorithm layer, we investigate how serendipity-aware recommenders, which can surprise and delight users with unexpected and relevant recommendations, can boost user satisfaction and engagement. We also propose a community-based methodology to optimize user experience by considering the social and contextual factors influencing user preferences and choices.

1.3/ THESIS ORGANIZATION

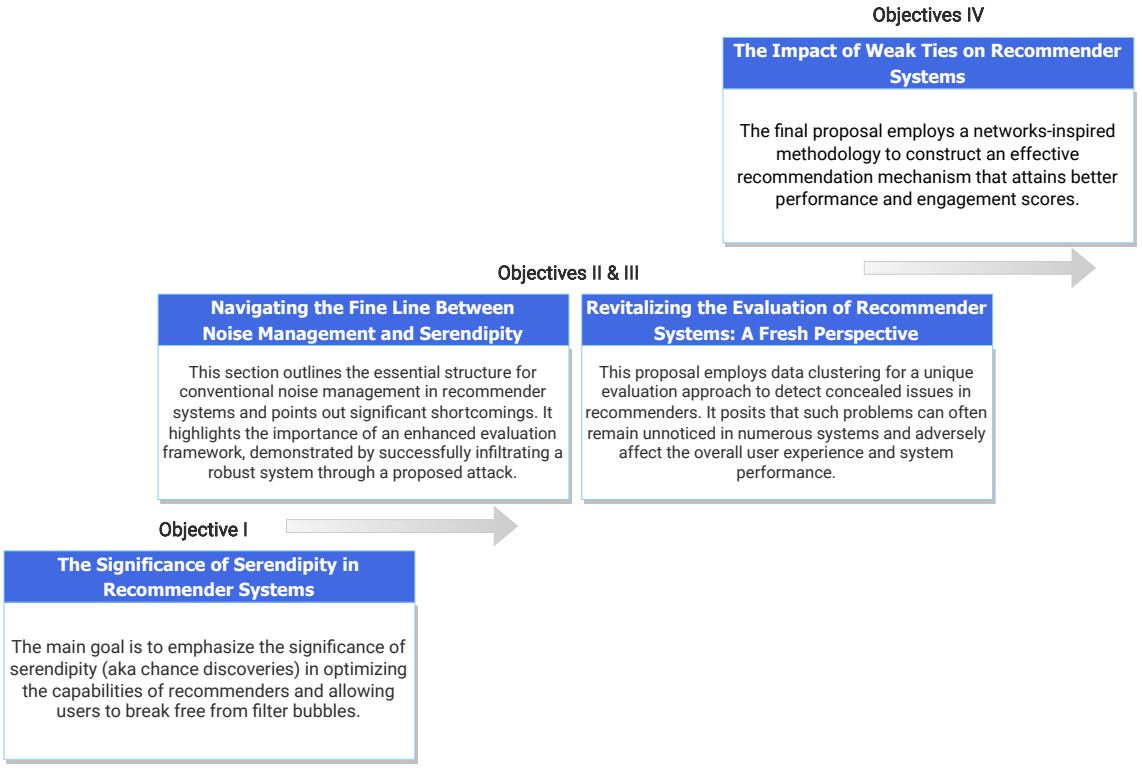

Figure 1.2: Thesis Organization.

The structure of the thesis is shown in Figure 1.2 and consists of four main objectives, which are ordered from left to right. Objectives two and three are closely interrelated and complement each other. There are six sections in total:

– Part II briefly introduces recommender systems, including their various types and applications in academic research and practical settings. Additionally, we discuss

key themes addressed in this thesis, such as serendipity, natural noise, and evaluation methods and the connections between them.

- Part III highlights the importance of serendipity, which refers to finding something valuable or exciting by accident in enhancing the performance and diversity of recommenders. By improving and adapting to serendipity, we can help users discover new and relevant items that they might not have considered otherwise and avoid the problem of filter bubbles, which limit users' exposure to diverse and challenging information.
- In Part IV, the essential components of traditional noise management methods in recommender systems are described, and significant limitations are identified. The need for a better evaluation framework is emphasized by showing how a proposed attack can breach a resilient system.
- Part V introduces a novel evaluation method based on data clustering employed to uncover hidden problems in recommenders. It is argued that these problems can frequently escape detection in many systems and negatively impact user satisfaction and system efficiency.
- In Part VI, an effective recommendation mechanism that achieves higher performance and engagement scores is constructed using a community-based methodology in the final proposal.
- Our study's results and potential implications are ultimately presented in Part VII of this dissertation, highlighting the significance of the research conducted.

II

General Overview

This section offers an overview of recommender systems and how they work using various methods. It explains the ideas addressed in this thesis: serendipity, managing natural noise, assessing the effectiveness of recommenders (including different evaluation techniques), and enhancing their performance through community-based clustering techniques. This chapter establishes the foundation for upcoming chapters and clarifies the concepts examined in subsequent research and experiments.

2

An Overview and Classification of Recommender Systems

2.1/ Introduction

As briefly introduced in the previous chapter, recommender systems are information filtering systems that predict users' preferences for a set of items. They are widely used in various domains, such as e-commerce, social media, and entertainment, to provide personalized recommendations to users. There are three main types of recommender systems: content-based CBF, collaborative CF, and hybrid Roy and Dutta (2022). Figure 2.1 shows the anatomy of different recommendation filtering techniques Isinkaye et al. (2015). It is important to note that in our work on the various topics of uncertainty, noise and evaluation, and general user data status in the system, we use different types of recommenders, and the main aim wasn't to select a "most efficient" recommender for a given task. That is because we have researched and proposed strategies to evaluate systems according to specific criteria and in a non-conventional way. Hence, the approach we apply is not standard, where we select specific models and verify their performance on a given task and across a particular set of metrics.

2.2/ Recommender Types

CBF recommender systems recommend items similar to those that a user has liked in the past. They use the attributes of the items to create a profile for the user and recommend items with similar features. These systems work well when there is sufficient data about the user's preferences and when the items have well-defined attributes. CF systems recommend items based on the likes of other users who have similar tastes to the target user. They use the ratings or reviews of other users to predict the target user's preferences. These systems work well when there is not enough data about the user's

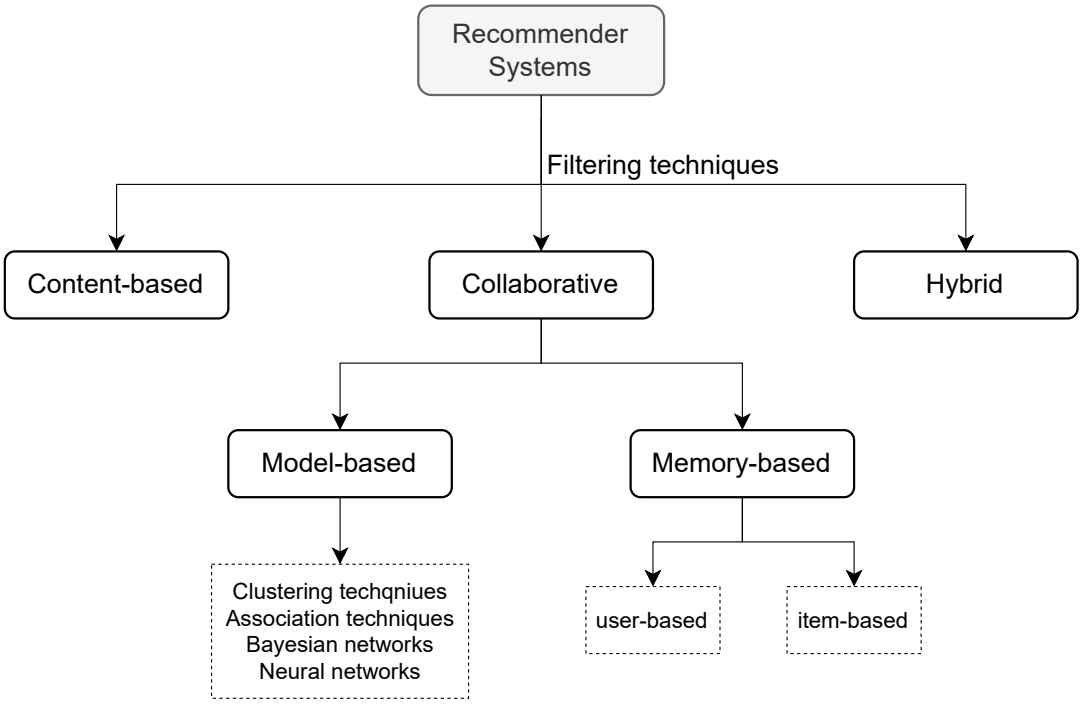

Figure 2.1: A High-level Representation of the Recommender System Techniques.

preferences and when there is a large number of users. Lastly, Hybrid recommender systems combine CBF and CF techniques to provide recommendations. They use both item attributes and user ratings to create recommendations. These systems work well when sufficient data about the users and the items exists. The different parts of this thesis focus on various types of collaborative-based models.

2.2.1/ CONTENT-BASED FILTERING

CBF is a domain-dependent algorithm that analyzes the attributes of items to generate predictions. CBF is the most successful technique when recommending web pages, publications, and news documents. In CBF, recommendations are made based on user profiles using features extracted from the content of items the user has evaluated in the past. Items mostly related to positively rated items are recommended to the user. CBF uses different models to find similarities between documents and generate meaningful recommendations. It could use a Vector Space Model such as Term Frequency Inverse Document Frequency (TF/IDF) Robertson (2004), or Probabilistic models such as Naïve Bayes Classifier Murphy et al. (2006), Decision Trees, or Neural Networks to model the relationship between different documents within a corpus. These techniques make recommendations by learning the underlying model with either statistical analysis or machine learning techniques. CBF does not need the profiles of other users since they do not influence recommendations. Also, if the user profile changes, CBF still has the potential to

adjust its recommendations within a very short period. The major disadvantage of this technique is that it requires an in-depth knowledge and description of the features of the items in the profile.

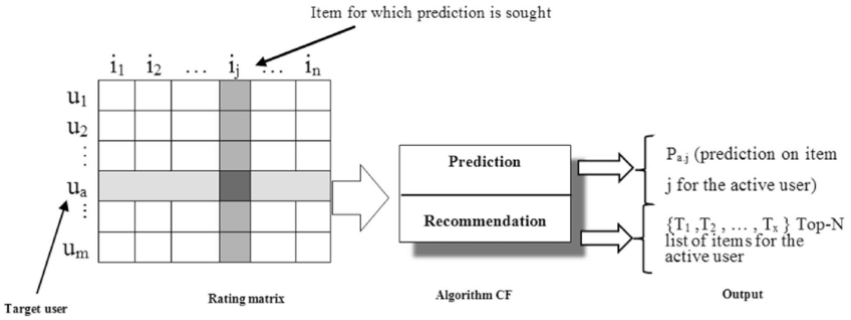

Figure 2.2: High-level Representation of the Collaborative Filtering Process in Recommenders.

2.2.2/ Collaborative Filtering

CF is a prediction technique recommending items to users based on their preferences. Figure 2.2 Atashkar and Safi-Esfahani (2020) shows a high-level representation of the general CF process. It is beneficial for content that metadata, such as movies and music, cannot easily describe. The technique builds a database of user-item preferences, represented as a matrix. The matrix then matches users with similar interests and preferences by calculating similarities between their profiles. Users with similar interests are grouped in what is called a neighborhood. Recommendations are made to users based on items they have not rated before but that were positively rated by users in their neighborhood. CF can produce either predictions or recommendations. Predictions are numerical values that express the predicted score of an item for a user, while recommendations are lists of top-N items that the user will like the most. CF can mainly be divided into two categories: memory-based and model-based.

2.2.2.1/ Memory-based Techniques

Memory-based techniques are popular CF algorithms that leverage the user-item rating matrix to make recommendations. These techniques operate on the assumption that users who have had similar preferences in the past are likely to have similar preferences in the future Roy and Dutta (2022). Memory-based techniques can be broadly classified into two categories: user-based and item-based. User-based techniques seek out similar users to the target user and suggest items that these similar users have liked in the past. On the other hand, item-based techniques identify comparable items to those the target user has previously expressed interest in and recommend these similar items.

The main elements of memory-based techniques are:

- User-item rating matrix: This matrix contains all the ratings users give to different items. It is used to compute similarities between users or items.
- Similarity measure: This measure calculates the similarity between two users or items. Various similarity measures, such as Cosine similarity, Pearson correlation, and Jaccard similarity, can be used.
- Neighborhood selection: This step involves selecting a subset of users or items most similar to the target user or item.
- Rating prediction: This step involves predicting the rating of an item for a target user based on the ratings of similar users or items.
- Recommendation generation: This step involves generating a list of recommended items for the target user based on their predicted ratings.

2.2.2.2/ Model-based Techniques

Model-based techniques are a class of CF algorithms that use statistical models to generate recommendations. These techniques are based on the assumption that an underlying model exists that explains the user-item rating matrix. Model-based techniques can be divided into matrix factorization Koren et al. (2009) and probabilistic models Ahmadli (2022). The idea behind matrix factorization is to represent the user-item rating matrix as a product of two low-rank matrices. This helps in estimating the missing values in the user-item rating matrix.

On the other hand, probabilistic model-based techniques estimate the parameters of a probabilistic model that explains the observed data. These models are used to generate recommendations based on the parameters learned from the data. One of the advantages of model-based techniques is that they can handle sparse data effectively and create recommendations based on a small number of user-item interactions.

The main elements of model-based techniques are:

- Latent factors: These are hidden variables that represent the characteristics of users and items. Model-based techniques aim to learn these latent factors from the user-item rating matrix.
- Model training: This step involves learning the model parameters using the user-item rating matrix. Various optimization algorithms can be used, such as stochastic gradient descent, alternating least squares, and Bayesian inference.
- Rating prediction: This step involves predicting the rating of an item for a target user based on the learned latent factors.
- Recommendation generation: This step involves generating a list of recommended items for the target user based on their predicted ratings.

2.3/ Data Types and Structure

For a recommender system to function accurately, it is essential to construct a well-defined user profile/model. The system requires as much information as possible from the user to provide reasonable recommendations from the beginning and avoid the cold-start problem Lam et al. (2008). Recommender systems rely on different types of input Aggarwal et al. (2016), including explicit feedback, which is the most convenient high-quality feedback and the one that is mainly used in our work in this thesis. This feedback includes precise input by users regarding their interest in an item or implicit feedback by inferring user preferences indirectly through observing user behavior. Hybrid feedback can also be obtained through both explicit and implicit feedback. In an E-learning platform, a user profile is a collection of personal information associated with a specific user. This information includes cognitive skills, intellectual abilities, learning styles, interests, preferences, and interaction with the system. The user profile is typically used to retrieve the necessary information to build up a model of the user. Thus, a user profile describes a simple user model. The success of any recommendation system depends mainly on its ability to represent the user's current interests. Accurate models are indispensable for obtaining relevant and precise recommendations from any prediction techniques. Figure 2.3 represents the critical differences between implicit and explicit feedback, the two types most commonly used in recommender applications Zhao et al. (2018). To summarize the types of feedback pertinent to the work accomplished in this thesis:

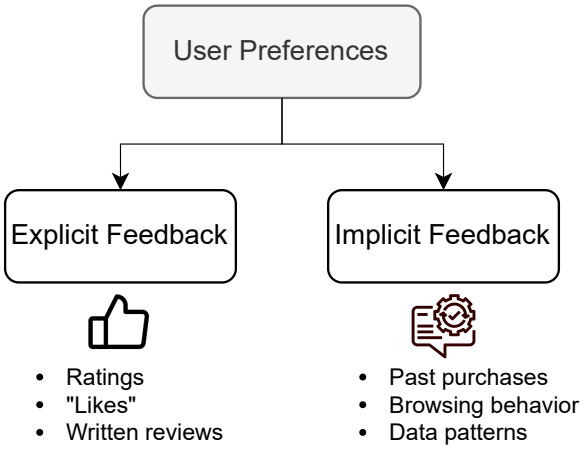

Figure 2.3: Key Differences between Implicit and Explicit Feedback in Recommender Datasets.

– Explicit feedback: The system interface usually prompts the user to provide item ratings to construct and improve the user model. The accuracy of recommendations depends on the quantity of ratings the user provides. However, this method has a shortcoming: it requires effort from users, and they are not always willing to pro-

vide enough information. Explicit feedback is more reliable than implicit feedback because it does not involve extracting preferences from actions. Although explicit feedback requires more effort from users, it provides transparency into the recommendation process, resulting in a slightly higher perceived recommendation quality and more confidence in the recommendations Buder and Schwind (2012).

– Implicit feedback: The system monitors user actions such as purchase history, navigation history, time spent on web pages, links followed by the user, the content of e-mails, and button clicks to infer the user's preferences automatically. This method of inferring choices is called implicit feedback and reduces the burden on users. Although it does not require effort from the user, it is less accurate than explicit feedback. However, some researchers argue that implicit preference data might be more objective than direct feedback because there is no bias arising from users responding in a socially desirable way, and there are no self-image issues or any need for maintaining an image for others Buder and Schwind (2012); Gadanho and Lhuillier (2007).

Table 2.1 presents highlights of the most commonly used MovieLens datasets in recommender systems, including the datasets used in most of our experimentations to achieve the goals of this thesis. Note that the sparsity of a dataset is defined as the ratio of the number of missing ratings to the total number of possible ratings. A higher sparsity indicates that the dataset has more missing values and is more challenging to work with.

The rating scale used in a dataset can impact the recommendations generated by a recommender system. A half-star rating scale provides more granularity and allows for more nuanced ratings, which can lead to better recommendations Uher (2018). On the other hand, a 1-5 rating scale is simpler and easier to understand, but it may not capture the full range of user preferences Aggarwal et al. (2016). However, the choice of rating scale depends on the specific use case and the nature of the data. In general, using a rating scale that is appropriate for the data and provides enough information to generate accurate recommendations is recommended.

Table 2.1: Information about the most commonly used MovieLens datasets for recommender systems.

Dataset Name	No. of Ratings	No. of Movies	No. of Users	Rating Scale	Sparsity	Release Year
MovieLens 100K	100,000	1,682	943	1-5 stars	93.70%	1998
MovieLens 1M	1,000,209	3,900	6,040	1-5 stars	95.53%	2000
MovieLens 10M	10,000,054	10,681	71,567	Half-star scale	98.30%	2009
MovieLens 20M	20,000,263	27,278	138,493	Half-star scale	99.46%	2016

2.4/ Evaluating Recommender Systems

As previously discussed, recommender systems in the context of this thesis are those that are mainly used to provide personalized recommendations to users based on their past behavior, preferences, and interests. Evaluating their performance is crucial as it involves the user's trust in the system. There are various incompatible assessment methods used for the evaluation of recommender systems. Still, the proper evaluation of a recommender system needs a particular objective set by the recommender system Gunawardana et al. (2012). The most commonly used evaluation metrics are accuracy, coverage, novelty, and uncertainty. Precision measures how well the system predicts users' preferences, while coverage measures how many items the system recommends. Novelty measures how diverse the recommended items are, while serendipity measures how surprising or unexpected the recommendations are Roy and Dutta (2022).

Most researchers who suggest new recommendation algorithms compare the performance of their latest algorithm to a set of existing approaches. Such evaluations are typically performed by applying some evaluation metric that ranks the candidate algorithms (usually using numeric scores). Most recommenders have been evaluated and rated based on their prediction power — their ability to accurately predict the user's choices. However, it is widely agreed that accurate predictions are crucial but insufficient to deploy a good recommendation engine Herlocker et al. (2004); Jurdi et al. (2021). In many applications, people use a recommender system for more than an exact anticipation of their tastes. Users may also be interested in discovering new items (serendipity), rapidly exploring diverse items, preserving their privacy, the fast responses of the system, and many more properties of the interaction with the recommendation engine.

Three levels of experiments are generally used to compare several recommenders: offline experiments, online experiments, and user studies Gunawardana et al. (2012). Offline experiments are conducted on historical data and compare algorithms based on their predictive accuracy. Online experiments on live systems measure an algorithm's real-time performance. User studies involve collecting user feedback about their experience with a recommender system.

2.5/ Conclusion and Areas of focus

We introduced the initial objectives of recommenders and their implementation and evaluation mechanisms. Despite their usefulness and advancement, recommender systems have limitations and challenges, such as cold start, scalability, and sparsity. Cold start refers to the problem of recommending items for new users who do not have any rating history. Scalability refers to the problem of handling large datasets with millions of users

and items. Sparsity refers to the problem of having insufficient data about some users or items. Several mathematical equations are used in recommender systems, such as cosine similarity, Pearson correlation coefficient, and Euclidean distance. Matrix factorization techniques such as singular value decomposition (SVD) and non-negative matrix factorization (NMF) are also used in CF.

In this thesis, we intend to advance the evaluation of recommender systems, specifically in the context of noise in the datasets. As introduced previously, evaluation is a vital task to measure a system's effectiveness. However, the data can sometimes mislead the results and allow a non-accurate conclusion of a good performance. In reality, the results are skewed due to the presence of such data in the systems.

Our target is to unravel the ideas of natural noise and hidden behaviors that harm recommender performance and aren't usually detected by conventional evaluation strategies. For that, there is an urgent need for a unique evaluation strategy that overcomes those data issues and allows a better assessment of the recommender's performance before being deployed and utilized.

We also intend to combine the knowledge from the evaluation, assess the main target of recommenders with the proper evaluation strategies, and propose a better recommender strategy that can maximize user engagement and recommend more engaging and suitable items for users. The aim is to align with the ultimate goal of recommenders introduced in the thesis's introduction, allowing users to explore new information and expand their knowledge and interests.

III

The Significance of Serendipity in Recommenders

This first part of the thesis highlights serendipity as a crucial factor in recommender systems and debates how there is yet to be a clear definition for it in current research fields. Experiments prove that uncertainty in the list of recommendations, alongside some relevant recommendations, improves user satisfaction. This section also addresses natural noise in recommender datasets and how noise detection algorithms attempt to free the systems from noise without considering that natural noise and serendipity overlap in their definition, disregarding the importance of serendipity. An algorithm has been developed to eliminate noise while allowing serendipitous results. The effectiveness of the algorithm's output is measured using the top-N adjusted metric.

3

Adaptive Serendipity for Recommender Systems

3.1/ Overview

Nowadays, recommender systems are widely implemented to predict the potential objects of interest for the user. With the wide world of the internet, these systems are necessary to limit the problem of information overload and make the user's internet surfing a more agreeable experience. However, a very accurate recommender system creates a situation of over-personalization where there is no place for adventure and unexpected discoveries: the user will be trapped in filter bubbles and echo rooms. Serendipity is a beneficial discovery that happens by accident. Serendipity alone can be easily confused with randomness; this takes us back to the original problem of information overload. Hypothetically, combining accurate and serendipitous recommendations will result in higher user satisfaction. In this section, we aim to prove the following concept: including some uncertainty at the cost of profile accuracy will result in higher user satisfaction and is, therefore, more favorable to implement. We will test a first-measure implementation of serendipity on an offline dataset that lacks serendipity implementation. By varying the ratio of accuracy and uncertainty in the recommendation list, we will reach the optimal number of serendipitous recommendations to be included in an accurate list.

3.2/ Introduction

Nowadays, with the internet being used worldwide and for many applications, the user is exposed to a substantial quantity of information. Consumers are suffering from what is called information overload. The need to bridge the gap between the demand and the supply becomes of urging importance. Recommender systems arise to predict what the user might need and recommend it to him, consequently narrowing his choices. Person-

alization of the internet's content or information filtering is essential in knowledge management Reviglio (2019). Personalization happens in two ways: explicitly through rating or implicitly through activity monitoring using artificial intelligence and machine learning. Personalization is somewhat dangerous, especially when done implicitly since it is imposed on the user who might not desire it. It creates filter bubbles and echo rooms. In the filter bubbles, the user continues to see and listen to what reinforces his interest and opinion.

While the echo room is a group situation where information, ideas, and beliefs are amplified like the actual echoing phenomenon if used up to a certain extent, personalization brings satisfaction to most users; however, if techniques continue to diverge towards further enhancing it, the result would be a dangerous over-personalized environment having users that are addicted to their comfort zone Reviglio (2019). Customers of e-retail businesses will view only their familiar items without being exposed to new items that they don't even know exist, even though these new items may solve problems that customers face. They aren't aware that these problems are solvable. Serendipitous items will satisfy customer's needs and increase sales. That's why "beyond-accuracy" objectives are essential in recommender systems. Kaminskas and Bridge analyze these objectives: diversity, serendipity, novelty, and coverage Kaminskas and Bridge (2016).

Serendipity is commonly described as a pleasant surprise, unintended finding, accidental discovery, or simply an "Aha!" experience Sun et al. (2013). The term was first used in 1754 by Horace in his book The Three Princes of Serendipity, whose adventure was full of unexpected happy discoveries. Simply put, serendipity is knowing what the user doesn't know they like: a challenging task. The item inside the user's mind can be divided into two categories (for simplicity): what they know and ignore. Each category can be divided into two subcategories: what they like or dislike for the known items and what they would like or dislike for the unknown. Serendipity lies in the subcategory of the items the user ignores but would like. According to the definition in Kaminskas and Bridge (2016), serendipity is discovering enjoyable or valuable things by chance.

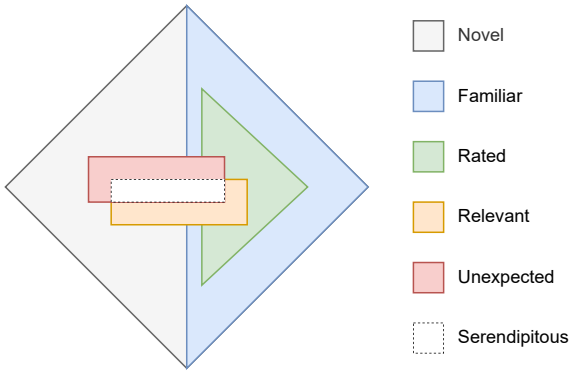

Figure 3.1: Users Perception of Recommended Items.

Serendipity is simultaneously the intersection of what is unexpected and relevant, as shown in Figure 3.1. Users enjoy what is relevant and accurate, unaware that there might be an entirely new world that they might be interested in but have never discovered. For all the previously mentioned reasons, and considering the importance of serendipity in a world so accurate that it is becoming tedious and redundant, we suggest integrating some serendipitous items in the recommendation list. First, this chapter simply aims to show that serendipity can increase user satisfaction even in offline datasets not linked to serendipity studies. The second goal is to test the optimal number of unexpectedly relevant items among others that are accurate. The following parts of the chapter are divided as follows: section 2 discusses the background and the related work. Then, we show the implementation environment, including the algorithm in its steps and the dataset. The experimental results will be presented in the last section, followed by the limitations.

3.3/ Background and Related Work

This section presents an overview of the previous studies and works on serendipity. Serendipity is a concept that is hard to define, and this complexity in the definition impacts the possibility of implementation. Ge et al. Ge et al. (2010) indicate that experimental studies of serendipity are scarce since it is not only hard to define but, in parallel, hard to measure. This difficulty in defining and measuring surprise and unexpectedness was mentioned in other surveys and studies Kaminskas and Bridge (2016). As previously mentioned, many research studies are trying to grasp the meaning of this happy surprise; they all admit that it is somewhere between the unexpectedness, the novelty, and the relevance or what is also called utility or usefulness.

Kotkov et al. Kotkov et al. (2016) in their survey list state-of-the-art recommender approaches that suggest serendipitous items. They point at the re-ranking algorithm, opposite to the accuracy-based algorithms, where obvious suggestions are given a low ranking. This algorithm can use any accuracy algorithm to provide the result, and in case we desire a serendipity-oriented modification, specific algorithms are to be used. At the same time, novelty does not rely on any standard accuracy algorithm. These algorithms can be improved by pre-filtering, modeling, and post-filtering.

Iaquinta et al. Iaquinta et al. (2008) proposed introducing serendipity in a content-based recommender system, creating consequently a hybrid recommender system that joins both the content-based algorithm and the serendipitous heuristics. According to them, the strategies to induce serendipity are as follows: implement it via ”blind luck”, i.e., randomly, via user profile in what is called the Pasteur Principle, or via poor similarity measures, or even via reasoning by analogy without any particular implementation. Therefore, some

content-based recommender systems, like Dailylearner, filter out the items that are too different and similar to the user's previously rated items.

The Pasteur Principle previously mentioned, as Pasteur himself states, "chance favors only the prepared mind", was used by Gemmis et al. in their approach De Gemmis et al. (2015). The knowledge infusion process can improve the ability of the algorithm to produce uncertainty. Their study showed a better balance between relevance and unexpectedness, which was better than other CF and CBF algorithms for recommendation. An attractive characteristic of their research was the measure of surprise done actively by analyzing the user's facial expressions. This analysis is performed using Noldus FaceReaderTM. That way, implicit feedback about the users' reactions will be gathered towards the recommendations that they are given.

In his model for news recommendations Jenders et al. (2015), Jenders suggests many ranking algorithms and models and compares them. The serendipitous ranking uses a boosting algorithm to re-rank articles. Those articles were previously ranked according to an unexpectedness model and another model based on the cosine similarity between the items and a source article. This ranking system gained the highest mean surprise ratings per participant.

In his study, Reviglio Reviglio (2019) states that uncertainty cannot be created on demand. Instead, it should be cultivated by creating opportunities for it. These opportunities would be present in a learning environment that can be physical or digital. He elaborates on his concept through social media. He affirms that by pushing the user to burst from the bubble, we give the people the power to discover, and by doing this, we create balance by providing freedom and mystery. As a continuation of what was previously said, Sun et al. Sun et al. (2013), through their observation, noted that micro-blogging communities provide a suitable context to observe the presence and effect of serendipity. Their experiment revealed a high ratio of serendipity due to retweeting. They remarked that this serendipitous diffusion of information positively affects the user's activity and engagement.

Some practitioners are trying to create systems where the design enhances serendipity. Two examples are Google's theoretical serendipity engine and eBay's test in serendipitous shopping Sun et al. (2013). Another recommender framework that tries to introduce serendipity is Auralist Zhang et al. (2012). This system attempts to balance accuracy, diversity, novelty, and serendipity in music recommendations and improve them. Observation of the systems reflects how users willingly sacrifice some accuracy to improve all the rest.

To better expect the unexpected, Adamopoulos et al. Adamopoulos and Tuzhilin (2014) proposed a method to generate surprising recommendations while maintaining accuracy. In our study, we utilized their algorithm and will explain it in the following Chapter.

3.4/ Implementation Environment

This section discusses the algorithm and dataset utilized for conducting the experiments.

3.4.1/ Strategies

To test the optimal number of serendipitous recommendations in the accurate list of recommendations, we started by choosing an algorithm for both our base and serendipity strategies. For the base strategy, we picked a non-personalized single-heuristic approach. Our base study, which is supposed to generate accurate recommendations, is based on popularity. In this strategy, the items are selected in descending order of popularity (i.e., number of ratings).

The serendipity strategy, which is personalized, considers three factors when selecting an item and adding it to the recommendation list: quality, unexpectedness, and utility. Certain restrictions and boundaries are placed on testing if the item's quality is above a specific lower limit and if it is farther enough from the user's expectations.

Six cases were subject to our testing. In each case, we varied the number of recommendations generated by each of the previously mentioned strategies. From case one, where all the items are caused by the base strategy, until the last case, where all items are serendipitous, we changed a few items following a varied approach. The procedures are summarized in table 3.1.

Table 3.1: Testing different recommendation strategies with varying numbers of generated items.

	Recommendation Strategy	
Case	**Base**	**Serendipity**
1	10	0
2	8	2
3	6	4
4	4	6
5	2	8
6	0	10

3.4.1.1/ Serendipity Algorithm

As previously mentioned, we utilized the algorithm implemented by Adamopoulos and Tuzhilin (2014). Below, we will summarize the workflow and briefly discuss the main concepts.

Step 1: Quality Calculations: First, we fix a lower limit on the quality of the recommended items. The first test compares the item's quality and the lower limit. If its quality is higher, it continues to the next step.

Step 2: Unexpectedness Calculation: The second step is to compute Eu's expected recommendations. Then, a lower limit and an upper limit are set on the distance of recommended items from expectations. This is the range of unexpectedness. Once we compute the unexpectedness of a specific item, we check if it belongs to the range. Otherwise, the item is dropped from the recommendation list.

Step 3: Utility Calculation: When the item passes the quality and unexpectedness tests, we need to estimate its utility for the user. The items with the highest utility will be recommended. Considering that the study is done offline, the users' ratings are used as a proxy for the utility of the recommendations.

3.4.1.2/ ACCURACY ALGORITHM

We used the algorithm implemented by Chaaya et al. (2017), initially suggested by Elahi et al. (2014).

R is our dataset. It is a matrix containing the items, the users, and their ratings for some of the items. The user rating is presented by r_{ui} where i is the rated item by user u.

Four main steps are used to implement the accuracy algorithm.

Step 1: Dataset Partitioning. Divide R into three datasets in a random way:

- Dataset S (System): it contains the user's ratings to the system.
- Dataset Q (Queries): it contains the ratings for items unknown by the system, but the user will simulate that.
- Dataset E (Evaluation): as its name indicates, the purpose of this dataset is evaluation through accuracy calculation.

A specific rating in the database will be present in only one of these three datasets (if the rating is not zero). In other words, there are no duplications. The not null ratings in R were divided randomly into the following percentages: around 0.5% in S, 69.5% in Q, and 30% in E. Initially, S contains very few ratings, reflecting what would happen in a real-life recommender system: the system possesses little information. This is the cold start problem faced by the recommender systems Kunaver and Požrl (2017).

Step 2: Rating Elicitation. We have the set S_u, which stands for system unknown. All the items not rated in S for every user are considered anonymous information for the system. They will be placed inside S_u. Through active learning, a certain number (L) among those items will be given to the user so they can rate the item in question. The ratings will be

retrieved from the dataset Q. Afterward, they will be transferred to S. Since there is no duplication, once those items are moved to S, they will be removed from Q. The user will rate no item twice: all L items are removed from S_u (System unknown). In the used algorithm, L is set to 10.

Step 3: Training Prediction Model. For every user in S, the prediction model is trained. The objective of training the prediction model is to predict the ratings of the unrated items. In the study Chaaya et al. (2017), the authors used a neighborhood-based technique to predict the ratings. First, the similarity between each two users is computed using Pearson correlation and summing over I_{uv}, the set of items rated by both users, u and v:

$$sim(u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r_u})(r_{vi} - \bar{r_v})}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r_u})^2 \sum_{i \in I_{uv}} (r_{vi} - \bar{r_v})^2}} \tag{3.1}$$

This value is then used to predict the ratings of the unrated items for user u, supposing that two similar users will rate the same item similarly. The predicted ratings r_{ui} are calculated using the following formula:

$$\hat{r}_{ui} = \bar{r_u} + \frac{\sum_{v \in N_i(u)} sim(u,v)(r_{vi} - \hat{r_v})}{\sum_{v \in N_i(u)} |sim(u,v)|} \tag{3.2}$$

Where u is the set of users similar to u and who rated the item i.

Step 4: Metrics Calculation. Many metrics exist to measure the success of the recommender system. Serendipity is deeply related to the user's satisfaction, which is hard to measure or define. Our experiment is done offline and is non-personalized. In other words, it does not include users. We will evaluate our technique using existing metrics. This is a common practice used when trying to assess the results, where the generated recommendations are compared with a baseline primitive recommendation system, and measurements are done through the use of saved ratings Kaminskas and Bridge (2016).

The evaluation was done using two predictive accuracy metrics: MAE and RSME. The Mean Absolute Error (MAE) computes the deviation between the actual and predicted ratings. Every prediction error is weighted in the same way.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |p_i - r_i| \tag{3.3}$$

The Root Mean Square Error is similar to MAE but emphasizes a more significant deviation.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - r_i)^2} \tag{3.4}$$

The MAE and RSME metrics are calculated on E. The algorithm then repeats the second, third, and fourth steps N times, N being the number of times every user logs in to the system. While repeating step three, the set S_u is new and should be considered.

3.4.2/ Dataset

For the data, we selected the 100K MovieLens dataset. This dataset contains 100k ratings made by 943 users on 1682 movies. A 5-point rating scale with the set 1, 2, 3, 4, 5 is considered. Every user has at least twenty ratings.

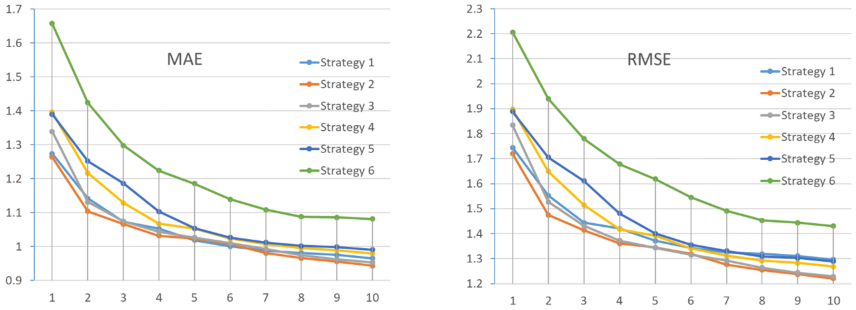

Figure 3.2: Evaluation of the Strategies with MAE and RMSE.

3.4.3/ Experimental Results

In this section, we will compare the different strategies using the selected metrics. The graphs of Figure 3.2 show the performance after every iteration from 1 to 10 for both MAE and RSME. We limited our study to 10 iterations for many reasons. First, the dataset size is small, and the strategies tend to behave similarly after a certain period. Second, users tend to rate a few items. Therefore, by limiting our iterations to ten, we are being more realistic.

The first observation is that the sixth case, where all the items are recommended serendipitously, performs the worst. This is expected and logical and was encountered by other researchers Chaaya et al. (2017). When all the items are serendipitous, the algorithm will behave identically to a random strategy, where accurate recommendations are not considered. Cases five and four have similarly bad results since the number of serendipitous recommendations is still high. However, with case three, we start seeing some better results. In the first three iterations, it still performs poorly, but after that, it starts behaving almost the same as case one, where all items are ”supposedly” accurate.

Table 3.2: Detailed values of the evaluation of the strategies using MAE.

	Strategy					
j	**1**	**2**	**3**	**4**	**5**	**6**
1	1.273	1.264	1.339	1.396	1.390	1.656
2	1.141	1.104	1.131	1.216	1.251	1.424
3	1.073	1.066	1.074	1.129	1.863	1.298
4	1.052	1.032	1.043	1.067	1.103	1.223
5	1.018	1.023	1.026	1.052	1.053	1.185
6	1.0	1.007	1.009	1.023	1.026	1.139
7	0.988	0.980	0.992	1.007	1.012	1.108
8	0.981	0.966	0.973	1.995	1	1.088
9	0.975	0.956	0.962	0.988	0.998	1.086
10	0.965	0.943	0.953	0.979	0.990	1.081

Table 3.3: Detailed values of the evaluation of the strategies using RMSE.

	Strategy					
j	**1**	**2**	**3**	**4**	**5**	**6**
1	1.744	1.720	1.183	1.897	1.889	2.206
2	1.553	1.474	1.527	1.649	1.705	1.940
3	1.443	1.414	1.430	1.514	1.611	1.780
4	1.420	1.360	1.372	1.417	1.481	1.678
5	1.371	1.344	1.343	1.391	1.400	1.618
6	1.343	1.318	1.315	1.340	1.355	1.545
7	1.325	1.276	1.292	1.312	1.330	1.491
8	1.318	1.254	1.263	1.291	1.309	1.453
9	1.311	1.237	1.243	1.282	1.303	1.444
10	1.297	1.220	1.228	1.268	1.290	1.430

The first three cases are close in performance. If we take a good look, strategy two has the best performance. A detailed table of the values resulting in each of the ten iterations for both metrics for every strategy is shown in Tables 3.2 and 3.3. Therefore, according to this study, eight accurate recommendations teamed with two serendipitous ones gave the best result in the given environment and conditions.

Despite promising results in the experiments, a few limitations are worth noting. Serendipity can be implemented using many algorithms and in different ways, and it strongly affects the user's satisfaction, which is difficult to understand or measure. An online study may be more relevant to how uncertainty affects the recommendations. Implicit feedback is required for a better assessment, like in the work of Gemmis et al. De Gemmis et al. (2015), where facial expressions were considered the key to measuring surprise. Moreover, the recommendation list size was fixed to ten, which is not always true. This goes without mentioning all the limitations that always occur in recommender systems studies

where many factors cannot be generalized, and the experiment restricts the results.

3.5/ CONCLUSION

Serendipity is an essential factor in the recommender system that is still under construction. A clear definition is yet to be unified, but we can say that it is a happy surprise. The system is asked to predict the unpredictable to expect the relevant unexpected. Many studies are interested in finding a way to measure uncertainty and, even more, to create it. In this part, we proved that serendipity in the list of recommendations, alongside some relevant recommendations, will improve user satisfaction. The future chapters will address these topics closely, including a definition of serendipity and assessment approaches when noise is present in datasets. Serendipity is a vast world worthy of discovering and a face for a recommender system that deserves investment.

4

Serendipity-Aware Noise Recommenders

4.1/ Overview

In the modern age, recommender systems are becoming a prominent and essential solution to the information overload problem. However, they are immensely susceptible to two significant types of noise: malicious noise caused by attacks and natural noise due to human error. Many detection algorithms attempt to solve the noise problems, but since natural noise and serendipity overlap in their definition, removing noise eliminates serendipity. Serendipity is the happy surprise of unexpectedly finding something relevant and is vital for the over-personalization caused by most recommender systems. This study aims to implement a serendipity-aware noise detection algorithm that will serve as a pre-processing phase to a recommender system application and protect the user's trust without leaving him in a "filter bubble". Finally, a new metric called top-N adjusted is used to measure the effectiveness of the proposed algorithm.

4.2/ Introduction

In the current era of information, the massive quantities of data render the users unable to apprehend what's presented to them. This problem was dubbed "The Information Overload Problem" Melinat et al. (2014). Many solutions try to tackle this problem depending on its type, but mainly, all solutions are based on machine learning procedures, the most famous of which are recommender systems. Like many machine learning techniques, a recommender system makes predictions based on users' historical behaviors Luo (2018). Specifically, it predicts user preference for items based on past experience. To reduce the consumer's cognitive efforts and increase the quality of their decisions (and consequently, their satisfaction), decision-support systems (DSS) are implemented

in the shape of recommendations. A recommender system aims to filter, rate, and rank the enormous chunks of information to predict the user's tastes and eventually support inefficient decision-making Aljukhadar et al. (2010). As mentioned in Part II, the two most popular approaches to building a recommender system are CBF and CF. All the various techniques adapted to enhance the systems' capabilities have one goal: to gain the user's trust; however, this trust is at risk in many cases since the datasets are highly susceptible to noise.

There are two types of noise in recommender systems datasets: shilling noise (aka profile injection attacks) Gunes et al. (2014) and natural noise. The former is usually caused deliberately by the attacker to have a certain degree of prediction advantage of some products, for example. At the same time, the latter is associated with unintentional human behavior when rating or giving reviews online, usually due to fast decisions, etc.

As a solution to this problem that generally affects most datasets in any online system, many detection algorithms were proposed with one aim in mind: to remove the noise altogether from the system. Indeed, removing the noise from a dataset can lead to better accuracy results and prediction performance of a recommender system. However, this approach eliminates the possibility of serendipitous ratings. Serendipity in recommender systems is an area lacking decent investigation, and its definition is somewhat ambiguous as there is no consensus on it Kotkov (2018). Serendipity can generally be summarized as the happy coincidence of finding a relevant item (product, article, video, etc.). This is a precisely valuable and often overlooked aspect of a recommender algorithm that serves very well in helping maintain the prime aim of a recommender system. It protects the user from an over-personalized profile that eventually causes him to lose interest in a system that predicts nothing but items that they are already familiar with. Building recommender systems is more of an art since it involves understanding users' tastes and predicting what they might like or dislike. Therefore, accuracy, noise, uncertainty, and the user's trust must all be delicately handled in the system.

Our study aims to create an algorithm that considers serendipity while dealing with noise in the dataset. It is an extension of our previous work that showed how serendipity increases the user's satisfaction in a recommender system Badran et al. (2019a). The algorithm is built with two main modules: the first detects and deals with the possible noise in a given dataset, and the second extends the code's architecture and functions as a filter that discerns between noise and uncertainty, creating the ultimate balance for better predictions. Furthermore, to measure the effectiveness of the top-N predicted ratings offline, we propose a new technique focusing on their quality and significance for each target user.

The rest of the work is organized as follows: Section two explains the state-of-the-art about noise in its two forms and serendipity. Section three describes the proposed algo-

rithm, shedding light on the constructed modules and the new metric used in our method; section four contains the experiment details and the simulation results, and section five concludes this work.

4.3/ Literature Review

4.3.1/ Noise

The natural noise in a dataset is the collection of ratings that the user does unintentionally Castro et al. (2017), so they are inaccurate and do not reflect the user's correct preferences and biases. Natural noise is related to the methods by which the system infers the user preferences or collects them. There are two types of ratings used by recommender systems depending on the type of the recommender engine: explicit ratings, where the user is asked to rate items, and implicit ratings, where the user preferences are induced from his actions on the web. Both types are prone to behavioral errors that eventually transform into noise in a dataset, causing predictions' quality to decrease gradually Amatriain et al. (2009a). However, the implicit rating mechanism is highly susceptible to natural noise Castro et al. (2017). Pham et al. Iaquinta et al. (2008) and Amatriain et al. Yamaba et al. (2013) summarize the phenomenon of natural noise in datasets as:

- Alteration in user preference over some time.
- Impact of several factors on a user like personal conditions, social media influence, emotional states, and context.

Many different types of proposed algorithms attempt to detect and remove natural noise. Amatriain et al. Amatriain et al. (2009b) propose a novel algorithm with many repetitions of user ratings, i.e., re-rating items. The algorithm's job is to compare the different ratings of the same user for the same item and, based on some possible conditions, give the output. This type requires extra input and effort from the user's end, which might be its biggest flaw. O'Mahony et al. O'Mahony et al. (2006) proposed a method based on a threshold comparison. They define the consistency of a rating as the MAE (Mean Absolute Value) between the actual and the predicted ratings. As long as it is within a predefined threshold, it is accurate and noise-free. If it exceeds this threshold, it will be considered as noise. Yera et al. Toledo et al. (2015) proposed a method that only requires the rating matrix; it does not require any additional information and solely relies on the ratings in the usual user-item matrices. Our proposed algorithm is based on this study, so the details of this technique will be provided in the next section.

4.3.2/ SERENDIPITY

Serendipity in recommender systems is an area that is still ambiguous and hard to measure. In general, it can be summarized as the happy coincidence of finding relevant items (product, article, video, etc.), and this can be an essential aspect of a recommender engine that serves very well in helping maintain the prime aim of a recommender system, which is mainly to predict the best items to a particular user Sun et al. (2013). The components of uncertainty are relevance, novelty, and unexpectedness (Fig. 4.1), and to better understand serendipity, we ought to study its components Kotkov et al. (2016).

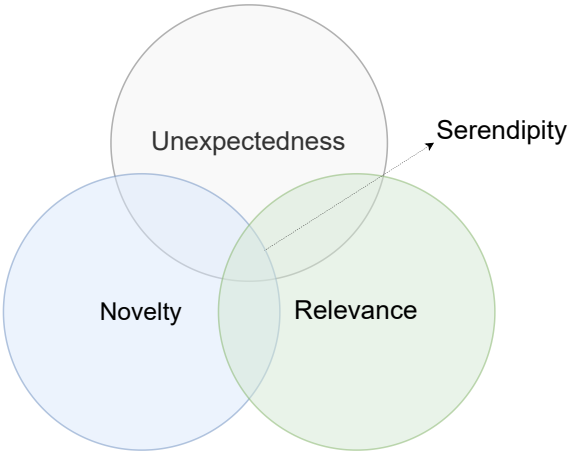

Figure 4.1: The Elements of Serendipity.

In their re-ranking serendipity-oriented algorithm, Adamopoulos et al. Adamopoulos and Tuzhilin (2014) assign overall scores to items based on unexpectedness and quality metrics. To create a recommender system that respects diversity, novelty, and uncertainty, Zhang et al. Zhang et al. (2012) propose an algorithm called Full Auralist. It comprises three algorithms that individually return a ranked list of items that will be integrated using linear combination to produce the final output. Jenders et al. Jenders et al. (2015) proposed a recommender system for newspapers based on the concept of serendipity. Moreover, Kawamae Kawamae (2010) proposed an algorithm based on the estimated search time. The work explains how the probability of an item being serendipitous varies proportionally to the difficulty of finding this item. The easier it is to find it, the less serendipitous it is. Iaquinta et al. Iaquinta et al. (2008) a strategy called anomalies and exceptions to introduce serendipity. They create a hybrid recommender system that joins both the CBF algorithm and the serendipitous heuristics, providing serendipitous recommendations alongside classical ones. The TANGENT algorithm was proposed by Onuma et al. Onuma et al. (2009) to broaden user tastes while retaining the accuracy of recommendations. TANGENT detects like-minded groups of users and suggests items to users from different groups. The algorithm proposed by De Gemmis et al. De Gemmis et al. (2015) uses an item similarity graph with an uncommon similarity measure.

4.4/ Proposed Algorithm

4.4.1/ Detection Module

To detect possible noise in a dataset, we built on top of the algorithm proposed by Toledo et al. Toledo et al. (2015). Their algorithm allows the detection of both natural noise and shilling attacks. The classification of the ratings is based on a combination of items, users, and ratings of users on items *r(u, i)*. First, the classification thresholds are set, and three categories are proposed for each separate group, users, items, and their ratings; every category has its unique classification thresholds:

- User
  - Weak-average user-dependent threshold *ku*
  - Strong-average user-dependent threshold *vu*
- Item
  - Weak-average item-dependent threshold *ki*
  - Strong-average item-dependent threshold *vi*
- Rating
  - Weak-average threshold *k*
  - Strong-average threshold *v*

To categorize the users and items in the dataset, all the ratings are compared to the threshold values and grouped based on the above divisions. The resulting groups would be as follows:

- User
  - Weak: $r(u,i) < ku$
  - Average: $ku <= r(u,i) < vu$
  - Strong: $r(u,i) >= vu$
- Item
  - Weak: $r(u,i) < ki$
  - Average: $ki <= r(u,i) < vi$
  - Strong: $r(u,i) >= vi$

Afterward, the categorized rating matrix is used to label users and items based on the number of occurrences of the user or the item in a particular category (Weak, Average, or Strong). For example, if the sum of occurrences of a user as weak, average, or strong was higher than the sum of his occurrences in different categories, then said user will be labeled as critical, average, or benevolent, respectively. Let W, A, and S be the sets that represent the total number of occurrences of a user as weak, average, or strong:

- $W > A + S$ implies a critical user
- $A > W + S$ implies an average user

- $S > W + A$ implies a benevolent user

The same logic is applied to items in the given dataset:

- $W > A + S$ implies a weakly-preferred item
- $A > W + S$ implies an averagely- preferred item
- $S > W + A$ implies a strongly preferred item

Labeling ratings as possible noise lies in combining the previous classifications of users and items with the ratings' values. The rating is characterized as possible noise if the three are not synchronized. For instance, if a benevolent user is rating a strongly preferred item, the rating should be considerably high; if this is not the case, this rating is a possible noise. Table 4.1 shows the combinations that are marked as possible noise:

Table 4.1: Possible noise based on the user/item category.

User/Item	WP	AP	SP
Critical	$r(u,i) >= k$		
Average		$r(u,i) < k$ or $r(u,i) >= v$	
Benevolent			$r(u,i) < v$

This proposal relies heavily on the values chosen for the weak-average parameters k, ku, ki and the average-strong parameters v, vu, vi. These parameters are highly domain-dependent. Therefore, having their optimal value predetermined is not straightforward. Nevertheless, a strategy can be defined to assign acceptable initial values. A global perspective approach is used for the initialization: different studies have concluded that this method for calculating the parameters is the best. Because three possible classes are considered, the thresholds should be picked to divide the ratings into approximately three cases. Code parameter choices:

- $ku = ki = k = minR + rnd(1/3 * (maxR - minR)) = 2$
- $vu = vi = v = maxR - rnd(1/3 * (maxR - minR)) = 4$
- $serendipityThreshold = 2$
- $n = 5$ (in top-n ratings for every user in the test set)

Serendipity intersects noise in the possible arrangements: In these cases (see Table 4.2), a differentiation should be made between natural noise and serendipity.

Table 4.2: The convergence of natural noise and serendipity.

User/Item	WP	AP	SP
Critical	$r(u,i) >= k$		
Average		$r(u,i) >= v$	
Benevolent			

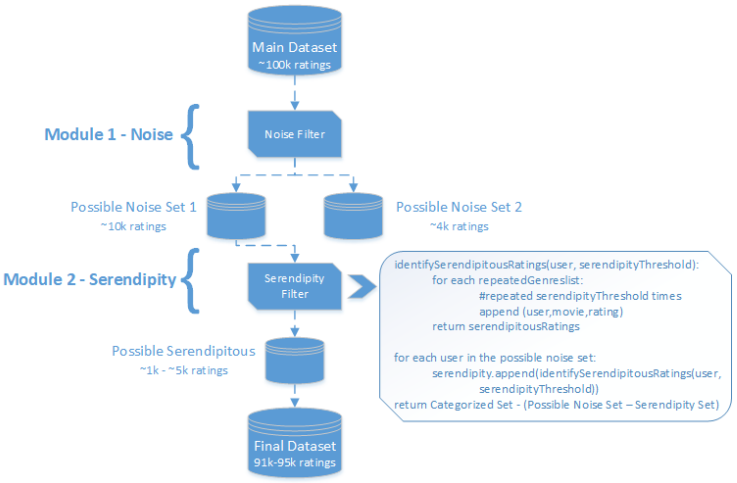

Figure 4.2: Algorithm Flow and Brief Overview of the Serendipity Module.

4.4.2/ SERENDIPITY MODULE

The serendipity module of the algorithm is responsible for filtering out the records that might be relevant to the user. To accomplish this task, we use the pipe-delimited genres field provided in the raw data of the Movielens datasets. For each user rating record in the possible noise dataset (the output of the first module of the algorithm), the unique genres list is counted to check the number of repetitions. Based on this, we can identify the user's taste in the possible noise data and filter out the records that might lead to serendipitous predictions. Figure 4.2 depicts the algorithm's two modules and briefly describes the second module. The flow of the module runs as follows:

- For every user in the noise set:
  - Retrieve the movie genre lists
  - Load the Genres into a data frame to easily calculate the count of the duplicated list of genres
  - Take only the genres that were repeated based on the serendipityThreshold variable
- The final dataset that contains all the serendipitous records from the previous step would be achieved by the following formula: *Categorized Set - (Possible Noise Set - Serendipity Set)*

The example below (see Table 4.3) shows a subset of possibly noisy ratings for a specific user. The combination of Thriller and Horror does not suit the user, while Thriller alone seems to be a good option. Therefore, the serendipity module, where $serendipityThreshold >= 2$, will keep movies 2 and 3 and remove the rest from the train set.

Table 4.3: A list of items that may contain noise for user u.

MovieId	Genre	Rating
1	thriller — horror	3
2	thriller	4.5
3	thriller	4
4	horror	2

4.5/ Experimental Results

Table 4.4: Experimental results.

Dataset	Size	MAE	RMSE	top-N	Serendipitous Items
No Noise Detection	100836	0.6732	0.8779	4.15635	-
Noise Removed - v1	90181	0.5988	0.7924	4.29859	-
Noise Removed - v2	97183	0.6606	0.8649	4.25477	-
Serendipity - v1	94479	0.6315	0.8289	4.37078	4298
Serendipity - v2	92881	0.616	0.8139	4.39314	2700
Serendipity - v3	92149	0.6165	0.8151	4.39687	1968
Serendipity - v4	91769	0.6166	0.8131	4.403010	1588
Serendipity - v5	91091	0.6102	0.8087	4.50904	910

4.5.1/ Metrics

The traditional famous accuracy metrics such as MAE and RMSE are not suitable for evaluating the output since they are accuracy-oriented. We aim to increase the serendipity of the top-N recommended lists for each user in our test set. As previously discussed, this is very hard to measure offline, albeit being quite simpler online through some forms of A/B tests.

To quantify the value of the output of our algorithm, a new metric is proposed specifically tailored to tackle the quality of the top-N recommended lists. This metric, called top-N adjusted, measures the average expected ratings for every user's top-N output and averages them out for all users. This allows us to quantify the quality and significance of the top-N ratings recommended items for a user, items that the user has not previously seen. The proposed algorithm is expected to score a high top-N average output as there is a better rating quality in the dataset after adding the possible ratings that might result in serendipitous predictions for all users in the test set and removing the ratings that are considered noise.

$$top - N\ adjusted = \frac{\sum_1^n \frac{\sum_1^N r'(u,i)}{N}}{n} \tag{4.1}$$

N is the total number of ratings in the top-N predicted recommendations for every user, n is the total number of users in the test set, and $r'(u,i)$ is the predicted rating of user u on item i.

4.5.2/ ENVIRONMENT

To show the impact of the proposed algorithm, the following scenarios were applied on the recommender base code (for more details, see Table 4.4):

- Without noise detection, passing the dataset as it is to the recommender
- With noise detection (noise removed v.1)
- Ratings that are below expectations are removed, keeping the ones that are higher than what they are supposed to be (noise removed v.2)
- With a serendipity filter, keeping items that might result in serendipitous ratings. Different thresholds are considered in this case:
  - Variation 1: $serendipityThreshold >= 2$
  - Variation 2: $serendipityThreshold >= 3$
  - Variation 3: $serendipityThreshold >= 4$
  - Variation 4: $serendipityThreshold >= 5$
  - Variation 5: $serendipityThreshold >= 9$

The proposed algorithm is applied on a Movielens dataset called ML-Latest-Small Harper and Konstan (2015). The total number of users is 609, and their ratings count is around 100k. The language used to code the algorithm is Python. The base algorithm for recommending top-N lists for every user in the test set is the SVD (Singular Value decomposition), where the split ratio of train/test is set to 0.8.

4.5.3/ RESULTS AND DISCUSSIONS

Table 4.4 summarizes the MAE, RMSE, and top-N adjusted outputs for all the scenarios of the algorithm testing. As previously argued, traditional metrics such as MAE and RMSE only account for the recommended items' accuracy without considering the recommendations' quality or relevance to the user. MAE and RMSE fail to measure the nature of a top-N recommended list, so in our study, which is mainly focused on trying to increase serendipity by treating the data in a dataset, we execute the top-N adjusted metric (see Equation 4.1). After testing the first module of the algorithm (removing noise only), we noticed that there are some variations in the top-N adjusted results, but nothing significant (note that the higher the average of top-N adjusted, the better the quality of the recommended items); however, there is significant variation in the values of MAE and RMSE, especially in (v.1) which is expected since the noise in the dataset is being com-

pletely eliminated. On the other hand, the top-N adjusted metric results start to increase after we introduce the serendipity module to reach a maximum of 4.4904 with around 9.7k noisy ratings eliminated from the set while keeping a very considerable MAE and RMSE values of 0.6102 and 0.8087 respectively. In the case of (v.1) the algorithm showed an 11% and a 9.7% decrease in MAE and RMSE, respectively, and a 5.8% increase in top-N adjusted from the raw dataset (first variation). In contrast, in the case of (v.5), the algorithm resulted in a 9.4% and a 7.9% decrease in MAE and RMSE, respectively, and an **8.5%** increase in top-N adjusted. Figure 4.3 shows the metrics plot under the tested dataset variations. These results prove that the algorithm could discern between noisy ratings and ratings that might be of tremendous importance to the user profile regarding serendipity while keeping very acceptable values of MAE and RMSE. Serendipity and accuracy are completely different in their form, where increasing the one results in the decrease of the other, and this is where our algorithm succeeds by keeping the highest values possible of both metrics, with a slight edge in top-N adjusted since we want to achieve the true aim of recommender systems.

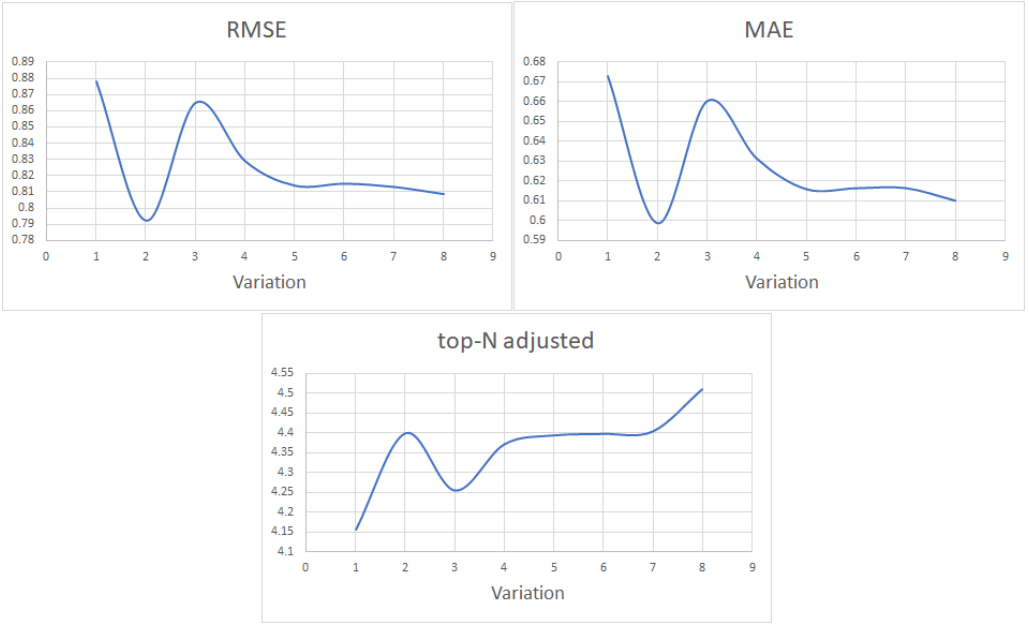

Figure 4.3: Algorithm Performance Against MAE, RMSE, and top-N adjusted. Variations 1-8 represent the scenarios in Table 4.4.

4.6/ Conclusion

The world faces the problem of information overload due to the huge amounts of data at the user's disposal. The data is often served to the user without being properly filtered. To overcome this issue, recommender systems aim to give the users what is relevant and useful to gain their trust; however, this trust is endangered by noise in most datasets. This

noise can be due to attacks or a result of natural noise that the user unintentionally makes. Noise detection algorithms free the systems from noise without considering that natural noise and serendipity overlap in their definition and, as a result, disregard the importance of serendipity. In general, recommender systems target accuracy, which naturally leads to the over-personalization of the recommended data. On the other hand, serendipity introduces the happy surprise, i.e., the unexpected relevance, and is often overlooked in recommender system implementations. Our proposed algorithm removes noise yet allows serendipity to exist in the system, and the new top-N adjusted metric measures the effectiveness of the algorithm's output. The results proved that the performance was radically enhanced. This work can be further developed by conducting online A/B tests with real subjects. Moreover, the noise in the dataset can be treated instead of being completely removed from the training set.

IV

The Fine Line Between Noise and Serendipity

In this section, we build upon previous research on managing natural noise and explore the difficulties of implementing an algorithm for natural noise management in datasets used by recommender systems. We classify all natural noise-handling algorithms and use experimental results to shed light on the reliability of the evaluation methods utilized in the suggested noise management techniques. Finally, our concluding experiment illustrates the inconsistencies of the conventional metrics employed to assess noise management techniques. In the second section of this chapter, two novel types of noise are presented, namely obfuscation and intentional user opt-out, which are difficult to detect using existing evaluation methods. The data appears to be completely normal despite these types of noise. The experiments demonstrate that the effects of this type of noise can only be observed using non-conventional evaluation techniques.

5

Critique on Natural Noise in Recommenders

5.1/ Overview

Recommender systems have been upgraded, tested, and applied in many, often incomparable ways. In attempts to diligently understand user behavior in specific environments, those systems have been frequently utilized in domains like e-commerce, e-learning, and tourism. Their increasing need and popularity have allowed the existence of numerous research paths on significant issues like data sparsity, cold start, malicious noise, and natural noise, which immensely limit their performance. Typically, the quality of the data that fuel those systems should be highly reliable. Inconsistent user information in datasets can alter the performance of recommenders, albeit running advanced personalizing algorithms. The consequences of this can be costly as such systems are employed in abundant online businesses. Successfully managing these inconsistencies results in more personalized user experiences. This article thoroughly analyzes the previous works on natural noise management in recommender datasets. We adequately explore how the proposed methods measure improved performances and touch on the different natural noise management techniques and the attributes of the solutions. Additionally, we test the evaluation methods employed to assess the approaches and discuss several vital gaps and other future improvements the field should realize. Our work considers the likelihood of a modern research branch on natural noise management and recommender assessment.

5.2/ Introduction

Over the years, recommender systems (RS) have become increasingly crucial to almost all online businesses worldwide Aggarwal et al. (2016). With various methods ranging

from prominent CF techniques to advanced latent factor models, they portray a significant role in most top-ranked commercial platforms like Amazon, Netflix, Spotify, and Last.fm Ricci et al. (2010). This emerges from the substantial problem such approaches try to tackle through highly personalized services efficiently: information overload. The underlying power of the personalized recommendations generated by various types of RSs primarily depends on the presence of generous user contributions in the forms of ratings, reviews, tags, etc. Researchers studying and enhancing RSs and their algorithms have tremendously focused on algorithmic improvements paying nominal attention to the data quality. The involvement of the human factor in the rating elicitation process is immensely prone to errors. Ratings, reviews, and other details recommender algorithms rely on holding critical information that might not always be sincere or consistent. This is recognized as noise in the datasets used by RSs to personalize information to users. If recommenders employ inaccurate data to learn user behavior, they will inevitably output inconsistent and unsatisfactory results.

There are two types of noise in RSs: malicious and natural. Simply put, noise is the rating feedback that does not reflect a user's preference or intention. This might be purposely arranged by attackers for specific reasons like biasing a recommender's output (malicious noise) Gunes et al. (2014), or it could occur naturally because of a user's inconsistent or negligent rating behavior (natural noise) Amatriain et al. (2009a,b). Malicious noise results from numerous attacks carried out on online applications that are typically powered by diverse types of RSs. This field has witnessed much attention in recent years Gunes et al. (2014).

Conversely, the natural noise domain hasn't yet received the entire focus of researchers. Natural noise occurs inherently due to user behaviors, which makes it unique. As emphasized by the very first work O'Mahony et al. (2006) and described through the publications embodying it at later stages, natural noise solely occurs due to human error that leads to data inconsistencies. It does not produce any pattern, and consequently, it's unusually complex to model. Significant improvements are required to develop a generic noise-aware recommendation algorithm capable of overcoming natural and malicious inconsistencies that might be present in the datasets of RSs.

The performance of recommenders, predominantly measured with conventional yet renowned offline tests employing accuracy metrics, such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and F1-Score, almost always records scant improvements. This poses a critical issue in the testing mechanism since evaluating RSs is inherently difficult for many reasons Herlocker et al. (2004). First, different algorithms appear to perform better or worse on varied datasets. Second, the goals for which recommenders are evaluated may differ; current works like the natural noise field primarily focus on accuracy improvements, while the proper aim of a recommender is to provide

a substantial personalized experience. Accuracy falls short of measuring the most fundamental aspect of an algorithm, which is how personalized the results are for a user Al Jurdi et al. (2018). Researchers tend to focus on amplifying the accuracy tests of the system, constantly offline, while very few target other properties that have notable effects on user personalization. Ultimately, commercial systems measure user satisfaction by the number of products purchased from recommendations and not by the score of a recommender's MAE or RMSE. Third, an authentic comparative evaluation of recommender algorithms poses a significant challenge in deciding what combination of accuracy metrics to use. The first and second reasons can be partly attributed to the fact that the quality of user interactions in the datasets used by RSs is frequently overlooked.

This work grouped natural noise studies into three main paths (Figure 5.1): the Magic Barrier path, the classical natural noise management path, and the preference-dependent natural noise management path. The term *natural noise* was introduced by O'Mahony et al. O'Mahony et al. (2006) as the inconsistencies in user data that occur without malicious intent. Subsequently, it was demonstrated by Amatriain et al. Amatriain et al. (2009a) that many users are inconsistent in the rating elicitation process. Herlocker et al. Herlocker et al. (2004) and O'Mahony et al. O'Mahony et al. (2006) are the most significant studies influencing the natural noise research topic and the three paths. A threshold termed *Magic Barrier* was speculated Herlocker et al. (2004) in which the authors argued that there seems to be a certain point where recommenders fail to get more accurate. They attributed this discovery to "inherent variability" in datasets - inconsistent user profiles. The pivotal outlook to point out, in this case, is that the authors only analyzed the evaluation methods of RSs and did not refer in any way to algorithm enhancements. This essential viewpoint was missed by the first path on natural noise that originated from Herlocker et al. (2004), where the authors debated that other types of evaluation metrics are to be engineered. They also emphasized that algorithms should be measured by how well they can communicate their reasoning to users or with how little data they can yield accurate recommendations; if this is valid, researchers require new metrics to evaluate those new algorithms. Therefore, the study in Herlocker et al. (2004) did not discuss nor prove that noise reduction induces better recommender performance. It merely proposed the concept of curating new algorithms with distinct evaluation techniques. The path that originated from Herlocker et al. (2004) in the natural noise field bore an alternative interpretation of the matter and tried to quantify the Magic Barrier limit in hopes of better accuracy results, completely missing the point of the critical study in Herlocker et al. (2004). Detached from the concept of the Magic Barrier, the second path targeted dealing with natural noise through several techniques mainly tested on CF algorithms. Some proposals typically employed classic clustering methods to identify variations in user profiles, while others resorted to more complicated fuzzy profiling techniques and matrix factorization modeling. In the third path of natural noise management, a few proposals

joined typical datasets with each other for secondary data as a natural noise management solution.

An intriguing point to note about natural noise management proposals is that throughout the three paths, the difference between identifying noise at the level of ratings and dealing with it at the level of users (noisy ratings vs. noisy users) was never technically analyzed. Toledo et al. Toledo et al. (2015) explicitly state that Li et al. Li et al. (2013) cover natural noise at the user level (identify if a user is inconsistent in his rating or not) and that it is necessary to provide a ratings-based solution. Unfortunately, no supporting evidence was provided to demonstrate how this would benefit a recommendation system regarding performance after natural noise management.

Personalizing algorithms that cater to the main aim of recommenders might be missing vital algorithmic improvements that ought to be measured by means beyond accuracy. However, those algorithms' results radically depend on the quality of the underlying dataset. Thus, accounting for natural noise in the datasets is of paramount importance and an area that requires deeper investigation. Further, suppose researchers plan to achieve improved means for measuring recommenders. In that case, they must re-evaluate the current protocols (natural noise algorithms or any other recommender approach) that previously relied on conventional evaluation metrics to judge performances and benchmark results.

The discussion on the validity of the evaluation methods used on all the natural noise approaches will be done through the functional analysis of the following two hypotheses:

- With the same recommender configuration and on various datasets, random rating removal cannot produce better performance results than a natural noise management method.
- The accuracy metric results of the above experiments always result in consistent measurements.

This article presents the following contributions to the natural noise management field:

- A detailed overview that classifies natural noise management techniques proposed since 2006 and conceptually analyzes their strengths and weaknesses.
- Analysis and critique on evaluation metrics, benchmark datasets, and recommender types used in the natural noise management proposals.
- A comparison through statistical analysis of the natural noise management mechanisms and their underlying attributes provides insight into how the natural noise path ought to sustain its development; highlights of the significant gaps in the field are also presented.
- An evaluation of the two hypotheses and a demonstration of how the uncorrelated results adversely affect the natural noise proposals path.

The remainder of the article is arranged as follows:

- Section 2. A presentation of some state-of-the-art works in the natural noise management field.
- Section 3. A discussion of all the natural noise management proposals since the field initiation in 2006. The algorithms are grouped into three primary paths.
- Section 4. Statistics and analysis of the main attributes used in the natural noise paths, like the evaluation metrics, benchmark datasets, and recommender types.
- Section 5. An investigation of the accuracy metrics used to evaluate the accuracy of CF predictions and recommendations after natural noise management. This section presents the gaps in the natural noise management paths.
- Section 6. Conclusions include areas where we feel future work is particularly warranted.

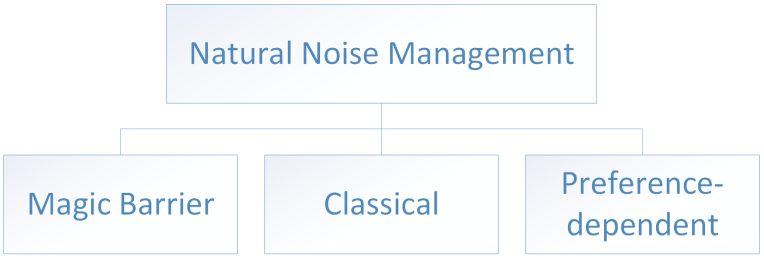

Figure 5.1: Natural Noise Management Paths.

5.3/ RELATED WORK

After the subject of malicious noise in RSs Gunes et al. (2014) has been extensively addressed, Natural Noise (NN) has lately started to spark the interest of researchers deeply. Ever since the path's introduction in O'Mahony et al. (2006), it has taken on several forms as proposals approached the problem in unique assorted ways. Till now, there has been no deep analysis of the diverse proposals on Natural Noise Management (NNM) introduced in the literature. Approaches such as those that directly deal with NNM or discuss it in terms of the concept of the Magic Barrier touched upon in Herlocker et al. (2004); moreover, there are no direct surveys on the topic. Practically all proposals in the NNM path followed the same discussion strategy throughout their idea development before introducing the approach. Summarized, they state that Amatriain et al. Amatriain et al. (2009a) deployed a re-rating method on users, and 40% of them displayed inconsistent results with their previous ratings. This confirmed that users' ratings can be irregular in that they may rate the same item differently at diverse points in time, and proved the primary idea of Hill et al. Hill et al. (1995). Their research fundamentally influenced the speculations about various evaluation techniques for a sophisticated level of personaliz-

ing algorithms in Herlocker et al. (2004) by showing users provide inconsistent ratings when asked to rate the same movie simultaneously.

Castro et al. Castro et al. (2018), and Toledo et al. Toledo et al. (2015) in the classical NNM path mentioned a few algorithms and previous work. In those connected studies that represent one type of the notable approaches to NNM, the authors categorized a few previous NNM works into two primary classes, the first that targets individual recommendations and the other that targets recommenders for groups of users Bobadilla et al. (2013). The approaches are then split into two groups: those based on crisp functions and those that introduce fuzzy profiling. These works do not mention any research from the Magic Barrier path. Subsequently, Martínez et al. Martínez et al. (2016) discussed NNM in RSs and summarized their previous approaches in Toledo et al. (2015); Castro et al. (2018). However, the study does not provide an attribute analysis of all the previous NNM approaches in the literature nor direct comparisons regarding datasets or algorithm complexities. One very recent study on NNM in RSs by Bag et al. Bag et al. (2019) introduced a sparsity-aware model by slightly amending the previous approach of Toledo et al. in Toledo et al. (2015). This study mentions research from the literature; however, it was very brief and lacked technical analysis. Most major research fields were unmentioned and dismissed from the implementations.

NN in recommender systems appears to lack a well-defined track of research approaches. It is pursued from many viewpoints O'Mahony et al. (2006); Herlocker et al. (2004); Kluver et al. (2012) as seen in every publication/article in the divergent paths. NNM needs a well-defined course for addressing inconsistencies in any recommender dataset. To efficiently overcome the issue of NN in RSs, researchers must develop an algorithmic program that works on any data, is ergonomic, has a reasonable execution time, and significantly impacts key personalization metrics (Beyond accuracy metrics Al Jurdi et al. (2018)). In this work, we extensively cover all the NNM approaches, the techniques used in developing the algorithms, and the statistical analysis of the attributes deployed with them. We categorize all studies based on the paths they took, the complexity of the algorithms, and the dependence on supplemental data that might be unavailable in regular datasets. We start from the point where NN was first introduced in O'Mahony et al. (2006) and targeted in the path that was influenced by Herlocker et al. (2004). Furthermore, we provide strategic directions about the NN field and discuss numerous gaps and essential critical points, considering the notion of uncertainty in recommenders.

5.4/ Approaches to Natural Noise Management

There were several attempts to change the research evaluation process of RSs from offline accuracy-focused studies to online user-personalized tests Herlocker et al. (2004).

Nevertheless, up until now, the evaluation retains the initial accuracy-based solutions. This is the case in all the NNM paths discussed in this section. This is likely attributed to the Netflix prize competition that evaluated the performance of the winning algorithm based on better accuracy results (RMSE) [1]. Netflix did not use the winner algorithm since they realized that better RMSE does not strictly mean superior personalized recommendations [2]. Netflix's reason for disregarding the winning algorithm was precisely what was discussed by Herlocker et al. Herlocker et al. (2004). The authors introduced the concept of the Magic Barrier limit and speculated how the evaluation process should be re-visited from a peculiar angle. One prominent feature they stressed is discarding accuracy metrics and focusing on engineering new metrics for user-oriented approaches, such as how well an RS communicates its reasoning to users or with how little data it can yield accurate recommendations.

The first study on NN emerged side by side with Herlocker et al. (2004) in 2016 by O'Mahony et al. O'Mahony et al. (2006), and since then has attracted the attention of researchers, gaining much popularity very recently. It was backed up by a study conducted in 1995 Hill et al. (1995) and took on several definitions, such as the noise that results from user preference change over time or the inherent rating inconsistencies of users. However, the studies from Herlocker et al. (2004) and O'Mahony et al. (2006) were inconsistent and held various approaches to the problem. The first path from Herlocker et al. (2004) does not address NN in RSs datasets. Simply put, it proposes an attempt to merely assess the quality of RSs based on the concept of inherent noise and variations in user rating over time. There were no solutions to how improvements to RSs beyond this calculated limit (Magic Barrier) can be achieved, but only how to calculate the limit based on accuracy standards such as RMSE. The following sections will introduce the NNM research categorized into three significant paths based on several criteria, especially how the researchers approached the problem.

The first path that emerged from the Magic Barrier study in Herlocker et al. (2004) mainly focused on calculating the Magic Barrier of RSs. This wasn't an approach to deal with NN but an attempt to open up a way for further improving RSs from an accuracy outlook. The second path dealt directly with NN, and most papers throughout the path proposed methods to either eliminate noise or correct it. This path was named the classical NNM, as most algorithms were based on discrete formulas and worked on datasets containing users and their ratings only. The last approach was termed the preference-dependent path. With more sophisticated algorithms, it mainly focuses on information that further extends the essential data in the most widely used RSs datasets, such as reviews, director information (in the case of movie datasets), etc. All the published studies in this first and second paths are laid out in timelines with summaries in tables 5.1 and 5.4, respectively.

[1] https://www.thrillist.com/entertainment/nation/the-netflix-prize, accessed: 12/01/2019
[2] https://www.wired.com/2012/04/netflix-prize-costs, accessed: 12/01/2019

5.4.1/ The Magic Barrier - Logic vs. Accuracy

The study of natural noise all started with the term Magic Barrier, which was speculated in Herlocker et al. (2004) and presented a significant challenge in deciding what combination of measures to use to evaluate recommenders in general. All enhancements and tuning on the algorithms that constitute RSs appeared to produce similar output qualities regarding the MAE accuracy metric – "many researchers find their newest algorithms yield an MAE of 0.73 (on a five-point rating scale) on movie rating datasets. Though the new algorithms often appear to do better than the older algorithms they are compared to, we find that when each algorithm is tuned to its optimum, they all produce similar measures of quality". Hence, the expression Magic Barrier was introduced as the point where natural variability may prevent us from getting any more accurate.

From then forward, the NN research has taken several paths, the first being a series of publications by the same authors where they tried to measure and quantify the Magic Barrier of Herlocker et al. (2004). Their path that originated from Said et al. (2012b) appears to have taken its approach under the NNM route and defined the Magic Barrier from their perspective and terms. It exhibited little correlation and few comparisons with other techniques that explicitly dealt with NNM in the datasets of recommenders. On top of that, it can be observed that the researchers tried tackling the Magic Barrier of Herlocker et al. (2004) by defining it as the point at which the performance and accuracy of a recommender algorithm cannot be further enhanced due to inherent noise in the data. Every improvement in accuracy (exclusively measured by MAE or RMSE) might denote an over-fitting and not a more competent performance. In addition, they quantified this definition by the notion that a mathematical characterization of the Magic Barrier that was speculated in Herlocker et al. (2004) is missing. They presented this characterization of the Magic Barrier based on RMSE. They claimed it allows us to assess a recommender's authentic performance and compute a precise room for improvement. The research path that branched from this study is explained and analyzed below and will be referred to as the *Accuracy Barrier* while the concept that Herlocker et al. (2004) introduced will be referred to as the *logic barrier* to separate the two and avoid confusion for future research on the topic as they are different at their core.

Before presenting the Accuracy Barrier path, it is substantial to note that the authors explicitly state that this approach represents a mere attempt to estimate the logic barrier. It is impossible to directly determine the logic barrier because it involves an optimal rating function, which is usually unavailable Bellogín et al. (2014); Said et al. (2012a).

5.4.1.1/ MAJOR PATH ON THE SUBJECT

In their early model Said et al. (2012b), the authors experimented with a user study scenario to assess and quantify the Accuracy Barrier of a recommendation system. The experiment included designing an online form to gather users' opinions on items they previously rated using the MoviePilot recommender and dataset [3]. The difference between the opinions and ratings was defined as the Accuracy Barrier of the dataset powered by the RMSE metric. The ultimate assumption was that the Accuracy Barrier of RSs can be better assessed by noise estimation. They presented a preliminary model for the Accuracy Barrier and the level of accuracy a recommender system can achieve without over-fitting to the noise in the data. The authors assumed the existence of additional transactions for $r_{u,i}$ given at different points in time and called them $o_{u,i}$ (opinion of user u on item i); recall that $r_{u,i}$ is the rating of user u on an item i. After that, the error between those ratings was defined as $\epsilon_{u,i} = o_{u,i} - r_{u,i}$ and the first attempt towards the Accuracy Barrier estimation was proposed in equation 5.1.

$$E(f * |R) = \sqrt{\frac{1}{|R|} \sum_{(u,i)\epsilon R} (o_{u,i} - r_{u,i})^2} \tag{5.1}$$

where $f*$ is an unknown rating function that knows the true opinions $o_{u,i}$ of each user u about any item i. Equation 5.1 refers to the estimated RMSE of the function $f*$. The authors continue to stress the idea that there might be a rating function f that results in a lower RMSE on R; however, those tend to over-fit the given rating set R and are likely to degrade the recommendation performance, and that is why equation 5.1 defines their Accuracy Barrier point.

Their idea was further developed and backed up in another work Said et al. (2012a). In it, they expanded the analysis and the case study with a commercial movie recommender and investigated the inconsistencies of the user ratings. In addition, they provided an estimate of the Accuracy Barrier to attain their goal of assessing the genuine quality of a recommender. The exact mathematical characterization of the Accuracy Barrier was further developed and expanded yet still based solely on RMSE as in equation 5.1; according to the authors, that allows the assessment of the authentic performance of a recommender as well as the amount of room for improvement. They reveal how the Accuracy Barrier represents the standard deviation of inherent rating inconsistencies in user ratings and present a noise model before deriving it. After estimating the Accuracy Barrier for MoviePilot, the authors concluded that said estimate helps assess the quality of a recommendation method and reveals room for improvement. Recommenders with a prediction accuracy close to the estimated Accuracy Barrier can be regarded as *optimal*. They con-

[3] https://www.moviepilot.de/

tinue to state that further improvements on such recommenders are meaningless. The mathematical representation of their estimate of the Accuracy Barrier was further developed in a procedure Said et al. (2012a) and took the following final form based on the average: $B_x = \sqrt{\frac{1}{|X|}\sum_{(u,i)\epsilon X}\epsilon^2_{u,i}}$; where X is a randomly generated subset of user-item pairs and $\epsilon^2_{u,i}$ is the variance of the ratings. With a similar logic as before, the authors added there might be a rating function $f\epsilon F$ that results in better RMSE scores; however, this is considered over-fitting and meaningless improvements. The results of the experiment show that the recommender system of MoviePilot can be better enhanced since the Accuracy Barrier yielded a value of 0.61 (close to the numerical step of the rating scale of MoviePilot) while the RMSE of MoviePilot's recommendation engine is about 1.8.

Subsequently, Bellogin et al. Bellogín et al. (2014) continued approaching the problem in an alternative way. They defined an experimental method to calculate the coherence of users in a dataset and revealed how the results are correlated with the Accuracy Barrier of Said et al. (2012a) in RSs. They utilized an external source to achieve this goal, with which one can measure the inconsistencies in the ratings by describing them in terms of specific features like genres (the authors adopted movie datasets like MovieLens). The formulation of the coherence of a user u based on a set of item features F was formulated according to equation 5.2.

$$c(u) = -\sum_{f\epsilon F}\sigma_f(u) \tag{5.2}$$

The authors adopted the standard deviation for calculating the coherence of user profiles where $\sigma_f(u)$ is defined as:

$$\sigma_f(u) = \sqrt{\sum_{i\epsilon I(u,f)}(r(u,i) - \bar{r}_f(u))^2} \tag{5.3}$$

Where $\bar{r}_f(u)$ corresponds to the average rating within the set of items rated by user u that belong to feature f. It is evident here that $\sigma_f(u)$ is the standard deviation used by the authors to represent the variation between the user's rating and a specific feature f, and $c(u)$ measures the variance of an individual's rating relative to the feature space by which items are defined. Based on the formulation of equation 5.2, the users are clustered into two groups: easy and difficult. This will then constitute the training set of groups to train a recommender algorithm. Employing a User-based CF (UB-CF) approach with 5-fold cross-validation, the authors evaluated their method using RMSE because it is related to the concept of the Accuracy Barrier in Said et al. (2012b)Said et al. (2012a). Note that the alternative is an Item-based CF (IB-CF) approach. The results of their experiment revealed that:

- The user coherence of equation5.2 provides good predictions of the Accuracy Barrier for a recommender.
- It is possible to utilize the user coherence groups to build different training and test models to decrease the error for every user (accuracy error).

In their latest study Said and Bellogín (2018), the authors provided a more explicit representation of the Accuracy Barrier expressed in Said et al. (2012a), along with a correlation with Bellogín et al. (2014). There are no other contributions to their Accuracy Barrier approach. Still, further experiments demonstrated that being statistically coherent regarding rating deviation within an item's attribute space (genres, in this case) can convey enough information to predict the users' inconsistencies. The study also concluded how an RS could be trained differently depending on the users' inconsistencies predicated by their rating coherence Said et al. (2012a) (equation 5.2); this allowed cheaper (less computation power, time, and tuning) recommendation cycles for the easy users' group (those with high coherence). Furthermore, the experiments also revealed that the prediction performance can be improved by 10% to 40% when only training with easy users. At the same time, the group labeled as difficult will receive worse recommendations in general.

5.4.1.2/ Path Influenced by the Magic Barrier

Amatriain et al. Amatriain et al. (2009a), backed up by a small proposal Amatriain et al. (2009b) done in 2009, addressed the problem of analyzing and characterizing the noise in user ratings. They presented a user study to quantify the noise that originates from inconsistencies in those ratings. The research tried to answer the following vital queries on the subject of NN:

- Are users inconsistent when providing ratings?
- How large is the error due to such inconsistencies?
- What are the factors that have an impact on user inconsistencies?

This study performed three trials and involved 118 users who were asked to rate items from a calculated subset of the Netflix Prize dataset to analyze the user inconsistencies in items they had rated. They came up with three primary variables that produced a significant impact on the user inconsistencies:

- The rating scale. Ratings are more consistent at the ends of the scale and significantly less consistent in the middle of it.
- Item order. A rating interface that groups movies likely to receive similar ratings should help minimize user inconsistencies.
- User rating speed. This might sound counter-intuitive; however, the smaller the time interval between ratings in a row, the fewer the user inconsistencies are in a dataset.

It is unclear how the authors related their study of rating inconsistencies and user stability metrics through RMSE to the logic barrier of Herlocker et al. (2004); however, what's clear is that this study influenced the path of the Accuracy Barrier that started with Said et al. (2012b,a) especially the RMSE approach for calculating user rating inconsistencies.

The study by Yu et al. Yu et al. (2016) was moderately influenced by the Accuracy Barrier, and the authors used the same clustering method of Said et al. (2012a) in their approach to overcoming the issue of NN datasets. Unlike the previously discussed studies in the path, this research provides a broader solution to RSs. It explored directly dealing with noise in datasets, generating recommendations, and measuring performance improvements. The authors proposed a generic framework to harness different pre-processing seamlessly and recommendation approaches for ratings of unique users. The users in a dataset are classified into several groups based on the quantity and quality of their ratings by several data pre-processing strategies. After that, the authors suggest a transfer latent factor model to convey trained models between groups in the training phase.

Additionally, it was argued that recommenders who take all user information as input suffers from two significant challenges: data quantity, a computational challenge, and data quality, an NN challenge. The primary idea of the approach is to process diverse types of users when training RSs variably. This is because users possess dissimilar rating quantities, and the effect of those inherently varies with behavior. Moreover, some users maintain consistent rating behavior while others suffer from inconsistencies. The critical steps of the approach are shown in Figure 5.2 and summarized as follows:

- Classifying user groups.
  Users were split into six groups based on two primary criteria, the number of ratings a user has (quantity) and the coherence measure of a user (quality). The authors adopted the coherence approach proposed in Bellogín et al. (2014) (equation 5.2). Tables 5.2 and 5.3 show the user groups generated from this approach.
- Processing noisy ratings.
  The noise detection method was also inspired by the proposal of Bellogin et al. in Bellogín et al. (2014). The authors adopted from Bellogín et al. (2014) the idea of item features (based on genres) and implemented the following equation which calculates the rating noise degree: $RND_{(r_{ui})} = \sum I\left(\sum_{fi} \frac{|r_{ui}-\bar{r}_{uf}|}{\bar{r}_{uf}} > \vartheta\right) \Big/ \|f_i\|$, where f_i represents the item features similar to equation 5.3 from Bellogín et al. (2014)Said et al. (2012a), and I the total number of individual recommendations. RND expresses the relationship between features with a significant relative deviation (more than the threshold ϑ compared with its same item feature set) and the aggregate number of features. The processing of the noisy ratings was handled differently, considering that removing ratings from the light users' group would worsen the sparsity problem. Accordingly, the authors adopted three options. First, no noise processing

was done for medium and easy users. Second, the noise was removed for heavy users only, and third, noise correction was implemented in the light users' group. The correction method was done based on the average rating of items that had the same features according to the equation: $r'_{ui}(corrected) = \frac{\sum_{fi \epsilon F} \bar{r}_{uf_i}}{\|F_i\|}$. A sampling phase was implemented for heavy users because their ratings contain redundant and repetitive information. The authors adopted the harmonic mean of entropy, replacing entropy with variance and inverse frequency to account for items in the long tail.

- Transferring models between user groups.
  In the final step of their proposed approach, the authors observed the data quality and quantity varied sharply between the groups of users. As a result, they offered to transfer the trained item latent factor models between those groups. The results of the protocol (Figure 5.2), measured by RMSE and precision, conveyed how the recommendation performance was significantly enhanced.

To improve the recommendation output of RSs, Saia et al. Saia et al. (2016) introduced a new approach based on a previous proposal in Saia et al. (2014). They measured the similarity between two items from a user profile and discarded those that appear as highly dissimilar. The authors argue that by eliminating those incoherent items from a user profile, the metrics' (RMSE and Average Difference) accuracy improvements will be genuine and not over-fitting or useless; however, their approach requires item text descriptions in the datasets. It requires four steps: data pre-processing, semantic similarity evaluation, dynamic coherence-based modeling, and item recommendations.

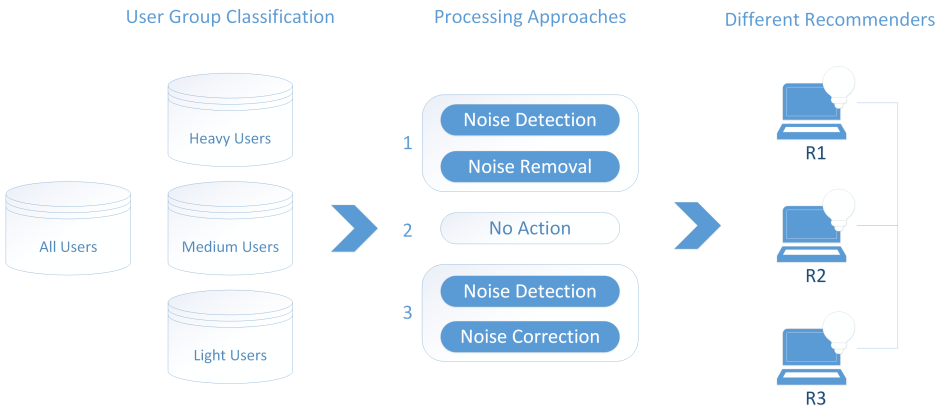

Figure 5.2: The Proposed Framework of Yu et al. (2016) Including a Natural Noise Management Mechanism.

5.4.2/ The Classical Natural Noise Path

Arguably, the most famous NN approach was proposed in 2006 by O'Mahony et al.O'Mahony et al. (2006), in parallel with the prominent Herlocker et al. (2004) that had

Table 5.1: Timeline of the Accuracy Barrier path after the study on evaluation in Herlocker et al. (2004)

2006	**Herlocker et al. (2004)**. The concept of a Magic Barrier is speculated. Ways beyond accuracy should be implemented for superior evaluation of recommender systems are discussed. The approach is referred to as the logic barrier
***2009**	**Amatriain et al. (2009b,a)**. The problem of analyzing and characterizing the noise in user feedback through ratings of movies is introduced. The noise that originates from inconsistencies in ratings is quantified
2012	**Said et al. (2012b,a)**. The authors extend the idea presented in Herlocker et al. (2004). A measure using RMSE is estimated and referred to as the Magic Barrier. This study dubs their approach as the Accuracy Barrier. It requires re-rating items by the same users
2014	**Bellogín et al. (2014)**. The authors propose a user classification approach that predicts the Accuracy Barrier of datasets in Said et al. (2012b,a). It analyzes user ratings using other factors in datasets, such as genres. The study complements the ideas of the Accuracy Barrier approach and uses RMSE as a metric for recommender evaluation
***2016**	**Yu et al. (2016); Saia et al. (2016)**. Influenced by the user classification method of Bellogín et al. (2014), a noise management algorithm that requires feature availability in datasets (such as genres) is proposed by Yu et al. (2016). The study Saia et al. (2016) introduces a coherence-based method to identify inconsistent items
2018	**Said and Bellogín (2018)**. The authors introduce an extension of works Said et al. (2012a), in terms of the Accuracy Barrier experiment, and Bellogín et al. (2014), in terms of the user coherence formulation

* These publications are not directly linked to the Magic Barrier path but are directly/indirectly influenced by it.

influenced the Accuracy Barrier path. The authors introduced the term *Natural Noise* in RSs' datasets for the first time and defined it as the noise that arises from imperfect user behavior when rating and reviewing items. Predominantly, the noise in datasets was grouped into two significant categories:

- Natural: That results from human activity errors when rating items they view/purchase.
- Malicious: This results from deliberately biasing reviews in a system to increase the recommendation frequency.

In this case, their core objective was to develop techniques to identify NN and discard it to improve the accuracy performance of a recommender. In addition, their solution also accounted for one type of malicious noise, but in this study, we are only focusing on NNM. Broadly, the authors measured the consistency of a particular rating $r_{u,v}$ as the MAE between the actual rating of a user and the predicted rating ($p_{u,i}$) of said user. The predicted rating can be identified using a particular recommendation algorithm (G) trained with a trusted user data set meticulously selected by a system administrator. This consistency was formulated as follows:

$$c(G,T)_{u,i} = \frac{|r_{u,i} - p_{u,i}|}{r_{max} - r_{min}} \tag{5.4}$$

r_{min} and r_{max} are the minimum and maximum ratings in a given rating scale, respectively. A rating was considered noise if $c(G,T)_{u,i}$ was more significant than some threshold th. The authors argued their approach allowed the possibility of analyzing the ratings of a neighborhood (k) of a particular user we wish to recommend items. The experiment results were conducted on several selected training sets and revealed how MAE improved with minor coverage development when NN was completely eliminated according to equation 5.4.

Table 5.2: Different user-item classification groups adopted in study Toledo et al. (2015) in the classical natural noise path.

Group	Description
Critical user	$\|W_u\| \geq \|A_u\| + \|S_u\|$
Average user	$\|A_u\| \geq \|W_u\| + \|S_u\|$
Benevolent user	$\|S_u\| \geq \|W_u\| + \|A_u\|$
Weakly-pref item	$\|W_i\| \geq \|A_i\| + \|S_i\|$
Averagely-pref item	$\|A_i\| \geq \|W_i\| + \|S_i\|$
Strongly-pref item	$\|S_i\| \geq \|W_i\| + \|A_i\|$

Li et al. Li et al. (2013) target NN in social RSs and refer to it as noisy but non-malicious users (NNMU). Their idea was based on the assumption that the ratings provided by the same user on closely correlated items should produce similar scores. The authors

Table 5.3: Different user-item classification groups adopted in study Yu et al. (2016) in the classical natural noise path.

			Description
Group	**Ratings**	**Consistency**	**Group Information**
HEUG	High	High	Users with high ratings and high consistency
HDUG	High	Low	Users with high ratings and low consistency
MEUG	Medium	High	Users with medium ratings and high consistency
MDUG	Medium	Low	Users with medium ratings and low consistency
LEUG	Low	High	Users with few ratings and high consistency
LDUG	Low	Low	Users with few ratings and low consistency

proposed a method for NNMU detection by capturing and accumulating individuals' self-contradictions. Formulating it as a constrained quadratic optimization problem, they defined those self-contradictions as the cases where a unique user provides very different rating scores on closely correlated items. Unlike the previous approach O'Mahony et al. (2006), this method identified noisy users. If a certain user appears to have made too many self-contradictions, the noise degree of his profile will rise, and he will automatically be classified as an NNMU. The optimization problem had the following input and output:

- Input
  - G - item-item correlation graph
  - y_L - all ratings in the test user profile
- Output
  - $\rho \epsilon [0, 1]$ - amount of noise in y_L (high $\rho \implies$ more likely to be a NNMU). Where $\rho = \frac{1}{K} \sum_{k=1}^{K} |\hat{\xi}_k|$ and $R_{min} - y_L \leq \xi \leq R_{max} - y_L$

Toledo et al. Toledo et al. (2013), with an extended publication by the same authors in Toledo et al. (2015), propose an alternative approach to deal with NN on the rating level in recommenders' datasets. The proposed framework includes two phases:

- Noise detection: Verifying if a rating is considered as noise based on a user-item profile classification scheme.
- Noise correction: Employing a classic CF method to predict a new rating to replace the noisy ratings when necessary.

Based on a group of ratings ($r_{u,i}$) classes (weak, mean, and strong) for both items and users, the authors define three particular sets for each group that constitute the preferences for each user/item: W_u, A_u and S_u for users and W_i, A_i and S_i for items. The thresholds used to group the ratings and users into the three sets are defined in equations 5.5 and 5.6, which are applied for both the item and the user sets.

$$\begin{aligned} W &= |\{r_{u,i} < k\}| \\ A &= |\{k \leq r_{u,i} < v\}| \\ S &= |\{r_{u,i} \geq v\}| \end{aligned} \tag{5.5}$$

$$\begin{aligned} k &= r_{min} + \{\tfrac{1}{3}(r_{max} - r_{min})\} \\ v &= r_{max} - \{\tfrac{1}{3}(r_{max} - r_{min})\} \end{aligned} \tag{5.6}$$

Subsequently, the classification for each group is performed based on Tables 5.2 and 5.3, and after that, the possible noisy ratings ($r_{u,i}$) are corrected ($r^*_{u,i}$) using a traditional UB-CF algorithm with PCC, $k = 60$ neighbors, and the original training set. The rating will be replaced if $|r_{u,i} - r^*_{u,i}| > \delta$. Experiments were applied to several parameter variation options: global-pv, user-based-pv, and item-based-pv. The results revealed improvement in MAE and F1, and the results were compared with O'Mahony et al. (2006) and Li et al. (2013), NNM protocols, and two other algorithms that target malicious noise.

Afterward, Castro et al. Castro et al. (2017) in a study in 2016 targeted dealing with NN under a different recommendation approach known as the group recommendation systems (GRSs). GRSs represent variations of the normal recommender strategies where individual recommendations or preferences are aggregated to form personalized recommendations for a group of users (grouping strategies) De Pessemier et al. (2014). The authors argued that GRSs employ explicit ratings and possess varying levels of information in their datasets and, therefore, are susceptible to NN that biases the recommendations. In this work, the core algorithm of Toledo et al. (2015) was used and modified to account for group preferences as part of the variations introduced for local data (the preferences belonging to the group members) and global data (the preferences belonging to all the users in the entire dataset). The results showed how NNM of the group ratings provides slight improvements to the group recommendation performance, while when applied to the entire dataset, it increases the performance of the GRS. Furthermore, the authors demonstrated how their hybrid approach aggregating a cascade of global and local approaches that manage NN (first at the global level - entire dataset - and then at the local level - group ratings) had superior performance results.

In a parallel study by Yera et al. Yera et al. (2016), the same authors argued that all the current NNM solutions cannot properly manage the inherent uncertainty and vagueness of customers' preferences. Accordingly, they proposed a novel fuzzy method to address this issue and improve, yet as well, the recommendation accuracy of recommenders. They added that the problem with previous approaches of NNM was that they solely represented and managed inherent rating uncertainties by means of crisp values, which implied an obvious lack of robustness. The authors added that the previously proposed approaches, like their own works in Castro et al. (2018), Toledo et al. (2015) etc.,

are not flexible and robust enough to deal with the uncertainty and vagueness of both the ratings and the NN. This proposal adapted the same workflow for the users in a recommender's dataset: profiling (instead, using fuzzy sets this time), noise detection, and noise correction. The steps of the approach are summarized as follows:

- Fuzzy profiling: obtain the fuzzy profiles of users, items, and ratings.
- Noise detection: apply a noise classification process on the previous profiles.
- Noise correction: noisy ratings are processed if needed.

It is comparatively explicit this approach is the exact replica of their previous technique Toledo et al. (2015) in terms of flow and logic; however, the particular difference, in this case, is that the recent method was implemented using fuzzy tools. The authors compared it to O'Mahony et al. (2006),Li et al. (2013) and Toledo et al. (2015), and in almost all cases, the MAE and F1 measures revealed better results with fuzzy profiling.

Latha et al. Latha and Nadarajan (2015) proposed an approach that assigns lesser popularity scores to users not providing good ratings for exceedingly desired items. Users with a popularity score of less than a certain threshold are identified as noisy users. The steps of the approach are summarized as follows:

- Identify popular items in a dataset using the random walk approach.
- Assign popularity scores to the users based on their ratings of popular items.
- Identify noisy users and discard them from the training set.

The popularity score of a user is calculated based on equation: $popularity\_score_u = \frac{|PI_u|}{|PI|} \times log\frac{|I_u|}{|PI_u|}$, where PI is the set of popular items, PI_u is the set of popular items rated by user u and I_u is the set of items rated by user u. The first component of the equitation considers how the user rates popular items, while the second checks whether said user also rates unpopular items. Compared with only Li et al. (2013) and O'Mahony et al. (2006), this approach showed better MAE, RMSE, and F1 metrics results.

In a more recent study in 2017 by Castro et al. Castro et al. (2018), preceded by a survey on fuzzy tools in RSs Yera and Martinez (2017), the authors combined their ideas of NNM that were presented in Castro et al. (2017) and Yera et al. (2016) and proposed an NNM for GRSs based on fuzzy tools. The approach follows the exact same approach in Yera et al. (2016) where they compared the new NNM (for GRSs) using fuzzy tools with NNM (also for GRs) using crisp values of Castro et al. (2017). They used only MAE, which ultimately resulted in improvements in most evaluation scenarios, while a few groups exhibited decay in the recommendation quality.

Choudhary et al. Choudhary et al. (2017) aimed to handle the issue of NNM in multi-criteria recommendation systems (MCRS) with nothing but the ratings of users in recommenders' datasets. An MCRS is a technique that provides recommendations by modeling a user's utility for an item as a vector of ratings along with several criteria Ricci et al.

(2021). The authors asserted how all the previous works on NNM up until this point had been done on overall ratings based on a sole criterion. The approach they followed in dealing with noise in the datasets was the exact same approach proposed in Toledo et al. (2015), and it deals with the classification of user ratings and items and the detection and correction of noisy ratings. The authors did not contribute anything to NNM; they merely used the approach in Toledo et al. (2015) and supplied the noise-free dataset to a multi-criteria recommender approach.

The study by Bag et al. Bag et al. (2019) came as an improved attempt to the series of proposals Toledo et al. (2015); Yera et al. (2016) and Castro et al. (2018). It was the first to approach sparsity Huang et al. (2004) as a major challenge in the whole NNM paths, and the authors asserted that removing NN can significantly amplify the data sparsity issue. Formerly, the sparsity issue was touched upon briefly in Yu et al. (2016); however, no systematic approaches addressed the problem when dealing with NN. They merely stated the number of noisy ratings the noise correction algorithm eliminated from the training set. The proposed method used the same approach presented in Toledo et al. (2015). They grouped ratings according to Tables 5.2 and 5.3 with slightly modifying the noise correction methodology. Rather than predicting a new rating in the correction phase, they utilized the concept of self-contradiction, which replaced the user's rating with the classified group threshold (Weak, Average, or Strong) when it was spotted to be self-contradicting. As for the sparsity issue, they integrated the Bhattacharyya similarity measure in the UB-CF approach Patra et al. (2015). The results show improved MAE and RMSE values and a better time complexity compared to Toledo et al. (2015) since they eliminated the re-prediction step to correct noisy ratings.

In their recent proposal, Yera et al. Yera et al. (2020) presented extended research on two of their previous ideas. In it, they connected the two fuzzy models for NNM previously proposed in Yera et al. (2016) (RRs for individuals) and Castro et al. (2018) (RSs for groups) that guaranteed robust modeling for the uncertainty associated with the user profiles (i.e., NN in datasets). Their experiments compared NN-Crisp Toledo et al. (2015) with NN-FT and NNMG-Crisp Castro et al. (2017) with NNMG-FT. Put differently, the authors provided a deeper study on their previously proposed classical NNM approaches for individual and group recommenders Toledo et al. (2015) and Castro et al. (2017), and their proposed fuzzy approaches Yera et al. (2016) and Castro et al. (2018).

5.4.3/ The Preference-dependent Path

All the research that depends on external data that aren't typically available in recommender systems' dataset fall under this path. Till now, there are only two interconnected proposals that directly address NNM. In research Pham and Jung (2013); Pham et al.

(2012), Pham et al. proposed a matching method between the user preferences and the dataset items to determine whether a certain item's ratings are reliable. Through two manually constructed small datasets (portions of MovieLens and Netflix joined with IMDB), the authors used item attributes to detect inconsistencies in tastes by comparing the actual preference value provided by the user with the rating predicted by a model. The inconsistencies are then corrected using preferences provided by expert users that exhibit overlapping tastes with the target user.

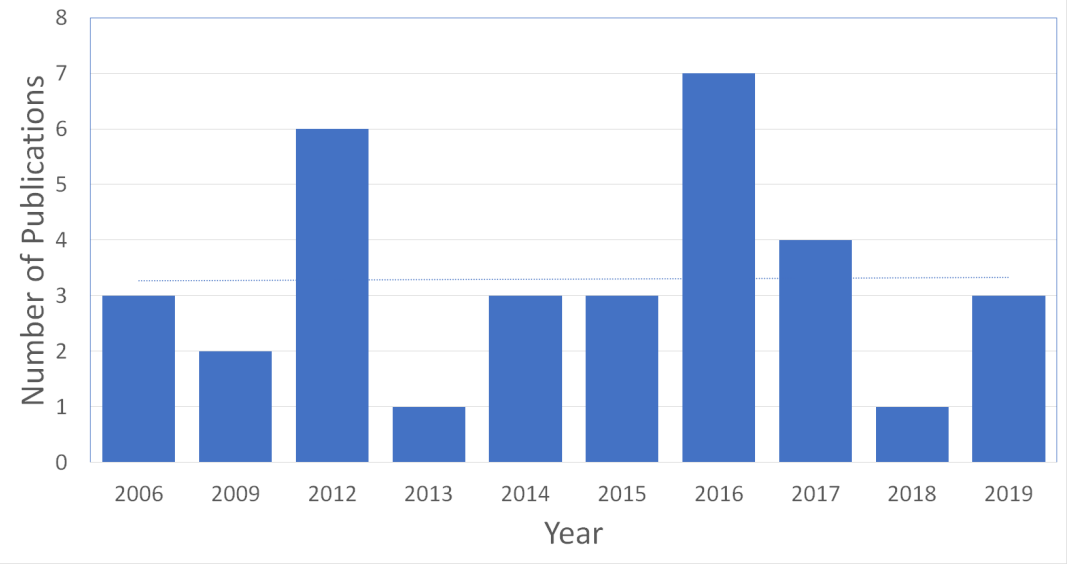

Figure 5.3: Number of Publications on Natural Noise since Inception in 2006.

5.4.4/ Natural Noise vs. Malicious Noise

The preceding works include all the proposals that targeted NNM in recommender datasets. What makes natural noise special is that it results from a user attitude and, even in exceptional cases, can go unnoticed by a recommender Zhou et al. (2015). When inconsistent and misleading behavioral patterns stack up in the dataset, they cause the recommender to learn from those anomalies rather than side-stepping them and provide recommendations that they inspire. Since it's challenging for a recommender to detect natural noise - mainly since it does not portray any suspicious or defined pattern - the results laid out to users are biased and, in many cases, highly inaccurate. This can be detrimental to systems that rely heavily on recommenders for sales, especially in commercial online stores.

Contrary to NN, malicious noise results from numerous attacks on online applications that are typically powered by diverse types of RSs, and it has witnessed much of the research attention in the past few years Castro et al. (2018). Attack patterns are usually defined, and the system can be trained to filter out the anomaly signatures. Further, malicious

Table 5.4: Timeline of the classical natural noise management path after the introduction of the concept in O'Mahony et al. (2006)

2006	The first study to address natural noise in recommender datasets has been proposed O'Mahony et al. (2006). Noise is defined as the variance between a rating and its predicted value using a recommender algorithm
2012	A comparison with O'Mahony et al. (2006) was done by Pham et al. (2012). Unlike in O'Mahony et al. (2006), the authors define the noise in terms of a user profile as a quadratic optimization problem
2014/15	In Toledo et al. (2013) and Toledo et al. (2015), the authors introduce a noise detection and correction method. It is based on user profile classification without any additional information to the dataset. Research Yera et al. (2016) introduces the same strategy as in Toledo et al. (2015) but with fuzzy profiling. Study Latha and Nadarajan (2015) introduces a noisy user detection approach by a popularity score method
2016/17	Study Toledo et al. (2015) is managed in proposal Castro et al. (2017) to implement a natural noise management (NNM) algorithm for group recommender systems (GRSs). It is improved more significantly in Castro et al. (2018) following a similar logic as Yera et al. (2016) and Castro et al. (2017) to propose an NNM based on fuzzy tools for GRSs. Research Choudhary et al. (2017) adopts the approach in Toledo et al. (2015) to employ it to a multi-criteria recommender
2019	A modified user classification and noise correction approach that takes dataset sparsity into consideration is proposed by Bag et al. (2019) based on Toledo et al. (2015). Research Yera et al. (2020) combines Yera et al. (2016) and Castro et al. (2018) in an extended study for fuzzy tools with NNM

noise is primarily the result of an external adversary aiming to carve the recommender's output rather than being generated in the dataset and by the system itself, like in the case

Table 5.5: Specification details of natural noise management approaches.

Study	Target Recommendations	NNM	Target	Category	Recommenders	Datasets	Evaluation Method
Castro et al. (2018)	Group	D & C	Ratings	Classical	UB-CB	Ml-100k, Nf-Tiny	MAE
Li et al. (2013)	Individual	D & R	Users	Classical	UB-CF	Ml-100k, BC, EM	Precision, Recall
Toledo et al. (2015)	Individual	D & C	Ratings	Classical	IB-CF, Matrix Factorization	Ml-100k, MT	MAE, F1
O'Mahony et al. (2006)	Individual	D & R	Ratings	Classical	UB-CF	Ml-100k, EM	MAE, Cov.
Yu et al. (2016)	Individual	D & C	Ratings	Pref-dependent	IB, Matrix Factorization	Ml-Latest-Full	RMSE, Precision
Yera et al. (2016)	Individual	D & C	Ratings	Classical	UB-CF, IB-CF, SlopeOne	Ml-100k, MT, Nf-Tiny	MAE, F1
Castro et al. (2017)	Group	D & C	Ratings	Classical	UB-CF, IB-CF	Ml-100k, Nf-Tiny	MAE
Yera et al. (2020)	Individual	D & C	Ratings	Classical	UB-CF, IB-CF, SlopeOne	Ml-100k, Nf-Tiny	MAE, F1
Latha and Nadarajan (2015)	Individual	D & R	Users	Classical	IB-CF	Ml-100k, Jester	MAE, RMSE, F1
Bag et al. (2019)	Individual	D & C	Ratings	Classical	UB-CF	Ml-1m	MAE, RMSE, F1, Precision, Recall
Pham and Jung (2013)	Individual	D & C	Ratings	Pref-dependent	None	ml*, nf*	RMSE

Figure 5.4: Distribution of Studies Across Datasets and Metrics.

of natural noise.

Due to this vast difference between natural noise and malicious noise, the methods Gunes et al. (2014) that had successfully worked on managing and eliminating malicious attacks (Probe, Bandwagon, Segment, Crawling, etc.) proved ineffective and inapplicable in the natural noise case Toledo et al. (2015); Bellogín et al. (2014). As demonstrated in the previous sections, the peculiar form of NN - mainly composed of inconsistencies resulting from user behavior - makes it hard to define a pattern for it instead of the specific outlined nature of malicious attacks. That's why in most cases, the studies throughout the path resorted to a classification method to categorize both users and their interactions.

5.5/ Statistical Analysis of the Paths

In this section, we curate substantial information about the previously discussed studies by providing some statistical figures. This allows a more thorough understanding of how researchers gradually approached NN. First, the number of publications over the years is presented in Figure 5.3, where it can be noticed that the pace of publications was almost steady overall between 2006 and 2009, with peaks arising in the years 2012 and 2016. The first spike mainly included proposals that attempted to reinforce the initial arguments on the concept of NN (see Tables 5.1 and 5.4), while the second included a series of research that introduced solving the NN issue with a marginally distinctive form of recommenders (GRSs), and predominantly employed fuzzy tools for NNM. This steady pace can be somewhat attributed to the intricate nature of NN compared to malicious noise

and the fact that it is more recent than malicious noise, which in turn averaged a different publication pattern over the years Gunes et al. (2014) compared to its counterpart. The study of NN was influenced by several works such as Hill et al. (1995), which conferred about rating inconsistencies in datasets, and Herlocker et al. (2004), which deeply impacted the entire logic barrier path. All researchers are still hinting that the study of NN is fairly new, and it is explicit that the field is yet open to many advances and proposals to address the missing gaps that will be discussed in the subsequent section. The NNM approaches and comprehensive specifications are summarized in Table 5.5.

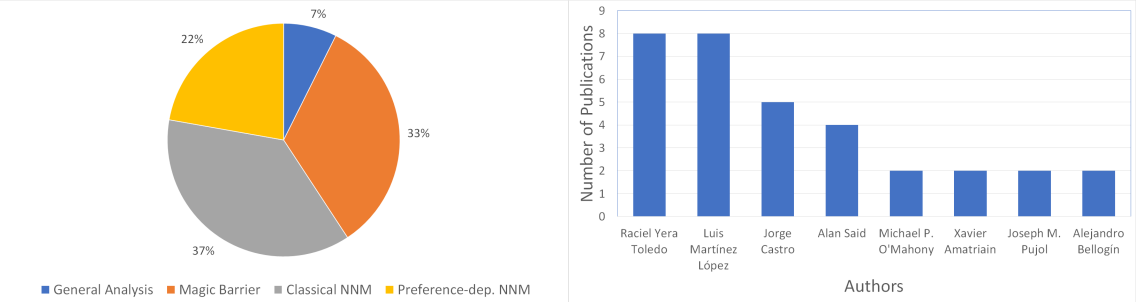

Figure 5.5: Percentages of natural noise research directions (left) and major researchers in the natural noise management field (right).

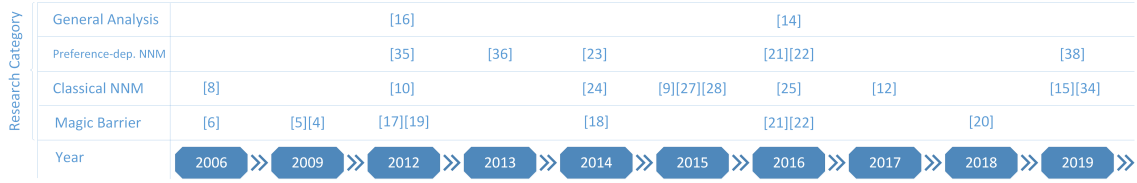

Figure 5.6: Timeline of Key Studies in Natural Noise Research.

Next, we plot the percentage of issues in each of the three categorized research tracks on NNM in Figure 5.5 (left). Classic NNM has the lead with 37% of the total publications, registering the highest publication rate, followed by the Magic Barrier with 33% of the total publications. The classic path includes more discrete formulations that define easy and effective ways to deal with natural noise in the datasets. Further, the Classic NNM and the Magic Barrier proposals chiefly depend on the exceedingly common recommender datasets (user-item matrix type) that typically do not require external features. It's reasonably expected that the preference-dependant NNM would obtain a more reduced rate as the dependency of the proposed algorithms and solutions was specific to certain features of datasets that aren't universally available nor conveniently accessible by all researchers in the field. The general analysis, amounting to 7% of the total publications, included studies that briefly discussed the problems of NN in recommenders' datasets without proposing any solution or method to deal with it. For conciseness, the most significant publications across the three tracks are laid out in the comprehensive timeline of Figure 5.6, starting with the inauguration of the concepts of NN and the Magic Barrier in

the year 2006. This timeline clearly presents the specific and most influential publications that contributed to the research peaks of 2012 and 2016 and offers a general idea of the NN track flow over the years. More specifications for the most famous NNM algorithms have been gathered and grouped together more comprehensively in Table 5.5. The data in this table indicates how every algorithm targets recommender datasets (whether it's for group or standard recommenders), the NNM method used, and whether it's built for correcting the noise after detection - Data Correction ($D\&C$) or merely removing it - Data Removal ($D\&R$), and the path class of the approach. Further, the recommender type, datasets used, and evaluation metrics implemented to measure the corresponding study have also been appended to the table. Additionally, It is apparent from Table 5.5 that the most used recommender type in the NNM paths was CF in both its forms, UB and IB, registering an appearance in almost all the proposals. Lastly, for researchers who aspire to reinforce the works on NNM, it is always beneficial to have an idea about the individuals who were contributing generously to the field up until this point; Figure 5.5 (right) plots the authors versus their respective number of publications.

On a more specific level, the datasets used in the studies across the NN track are detailed in Table 5.6. They constitute the famous open-source recommender datasets utilized in the Classic NNM and most Magic Barrier proposals. The Netflix Tiny and EachMovie datasets are currently no longer available. As previously mentioned, the data used in the Preference-dependent track is not publicly available and can be exceedingly delicate to reproduce in many cases. We also preview the datasets for each corresponding publication in Table 5.5 and present the metrics and datasets used in Figure 5.4. The Ml-100k dataset, arguably one of the most famous datasets in the overall recommender system research field Beel (2020), registered the highest usage percentage (36%), followed by Nf-Tiny (23%). Interestingly, those two datasets are considerably small (100k vs. 56k ratings) compared to their peers in Table 5.6. It's also surprising how the choice of datasets for testing NNM algorithms (through the metrics indicated in Table 5.5) in the Classic NNM and Magic Barrier was not done in a pre-calculated manner where it seemed that the authors merely chose the most famous recommender datasets to test novel approaches; this begs the question, are the post-NNM recorded accuracy improvements dataset-dependent? (More on the gaps and issues in Section 5.3)

5.6/ Analysis and Hypotheses Testing

Throughout the NNM paths that were detailed in Section 3, it has become apparent now that the common accuracy metrics, especially MAE and RMSE (Table 5.5), portrayed a significant role in deciding whether an NNM method improved a recommender's performance or not. In this section, we evaluate the two hypotheses introduced at the be-

Table 5.6: The details of the datasets used in the natural noise approaches.

Name	Category	User × Item	Range	Step	Ratings	Status
ML Latest	Movies	23k x 30k		0.5	21m	Available
Jester	Jokes	73k x 100		1	4.1m	Available
EachMovie	Movies	73k x 1.7k		1	2.8m	Retired
Book-Crossing	Books	279k x 272k		1	1.2m	Available
ML 1M	Movies	6k x 4k		1	1m	Available
MovieTweetings	Movies	21k x 12.6k		1	140k	Available
ML 100k	Movies	943 x 1.7k		1	100k	Available
Netflix Tiny	Movies	4.4k x 10k		1	56k	N/A

ginning of this work by conducting two experiments. The first introduces the concept of randomness and helps us interpret the relationship between the noise predicted by NNM measures and the metrics used to test the general performance after the noise has been managed (discarding or correcting it). The second experiment allows us to extend the first notion and checks whether the accuracy metrics used in the NNM publications provide consistent results. The recommender we employed in the two tests is a user-based CF algorithm with a cosine similarity measure, one of many measured for KNN-based algorithms Al Hassanieh et al. (2018).

5.6.1/ A Randomness-based Natural Noise Method

This experiment was uniquely designed to test the first proposed hypothesis, i.e., whether a recommender's accuracy performance can be positively affected by arbitrary rating removal from a target recommender dataset; should this claim hold, it would prove that the NNM approaches that traditionally presented accuracy improvement outcomes to show how a method is more effective than the other require radical revisions. That said, If a random-based straightforward process such as this could indeed achieve a similar performance to an NNM technique, then the evaluation approaches used in the NN field need to be adequately addressed; this does not mean that the NNM proposals are completely wrong, but it evidently would signify that the foundation that the NNM path is basing on might not be totally correct. To investigate this, our experiment ran on four varied datasets, Ml-Latest-Small, Ml-100k, Ml-1m, and Hetrec-Ml, that were selected based on the popularity in the NNM field (Table 5.5). For each simulation round, we implemented the following data removal schemes, which are chiefly based on randomness and logical intuition:

- $Random - N$: Remove random N ratings
- $Lowest - N$: Remove the lowest N ratings based on the corresponding rating scale
- $Highest - N$: Remove the highest N ratings

Table 5.7: Accuracy results for each natural noise mechanism across four different datasets

		Natural Noise Management Mechanism					
					Random		
Dataset	**N/Metric**	**Original**	**NN Filter**	**Random-N**	**Highest-N**	**Middle-N**	**Lowest-N**
Ml-Latest-Small	N	0	10655	10655	10000	10000	10000
	MAE	0.6937	0.6161	0.6880~0.7030	0.6543	0.6926	**0.5640**
	RMSE	0.9077	0.8216	0.8960~0.9214	0.8537	0.9205	**0.7077**
Ml-100k	N	0	12071	12071	12000	12000	12000
	MAE	0.7575	0.6791	0.7522~0.7744	0.7238	0.7732	**0.6285**
	RMSE	0.9607	0.8726	0.9516~0.9789	0.9180	0.9892	**0.7766**
Ml-1m	N	0	128916	128916	120000	120000	120000
	MAE	0.7473	0.6580	0.7475~0.7529	0.7057	0.7576	**0.6133**
	RMSE	0.9392	0.8401	0.9394~0.9454	0.8843	0.9647	**0.7521**
Hetrec-Ml	N	0	92634	92634	92000	92000	92000
	MAE	0.6254	0.5626	0.6230~0.6280	0.5799	0.6332	**0.4699**
	RMSE	0.8160	0.7465	0.8140~0.8202	0.7521	0.8382	**0.5858**

- $Middle - N$: Remove the ratings that are in the middle of the rating scale

The aggregate number of ratings (N) to be removed in every mechanism differs between datasets and was determined through the use of the most commonly used NNM algorithm (Toledo et al. Toledo et al. (2015)), which we employed in this experiment. In the case of the first scheme ($Random - N$), we typically had to re-train the data every time the arbitrary N ratings were removed since, unlike in the other random methods, the N ratings will vary. We need to measure the accuracy each time a different proportion from the dataset was removed to prevent biased results (in the other methods, N is typically constant, which implies that the accuracy results will be constant under the given conditions of the recommender). Accordingly, the simulation was conducted for 150 iterations in the case of $Random - N$. The final accuracy outputs of the aforementioned mechanisms were compared to the results from the NNM protocol and the original dataset without any noise management or rating removal. The results of the test are presented in Table 5.7, which details the variations of MAE and RMSE in each round for every dataset. The table also shows the value of N for every method and presents the minimum and maximum of the 150 $Random - N$ iterations. For a clearer presentation, Figure 5.7 depicts the $Random - N$ plot of the iterations where in each round, a value of $N = 10,655$ (Table 5.7) arbitrary ratings are being eliminated from the dataset. What's intriguing in the results is that the $Lowest - N$ scheme showed the best accuracy outcome across all four datasets. Surprisingly, $Random - N$ showed very acceptable scores that fluctuated between the $Original$ and $Middle - N$ schemes, sometimes achieving better output than both, such as in Ml-Latest-Small (0.6880 - 0.8960), Ml-100k (0.7522, 0.9516) and Hetrec-Ml (0.6230, 0.8140). The $Middle - N$ almost always had the worst MAE and RMSE results across all the datasets. The NNM scheme of Toledo et al. (2015) always came in the middle,

registering slightly better results than the $Highest - N$ in all the datasets.

It is evident now that utilizing the same accuracy metrics employed to evaluate the effectiveness of NNM algorithms on RSs in the previously discussed paths is controversial. Our purely random-based trial resulted in comparable significant MAE and RMSE improvements, especially with the $Lowest - N$ scheme, with relatively acceptable results for the others. This disproves the first hypothesis and clearly validates the concept that the evaluation methods for NNM that are used to assess performance and show that one is better than the other are flawed and require radical revisions. To a great extent, this also validates the fundamental opinion examined by Herlock et al. Herlocker et al. (2004), which was touched upon in the introduction of this work: algorithms should be measured in accordance with how well they can communicate their reasoning to users, or with how little data they can yield authentic recommendations, we require new metrics to evaluate those new algorithms and not merely rely on improvements that show scant enhancements in accuracy (such as MAE and RMSE) and label one better than the other.

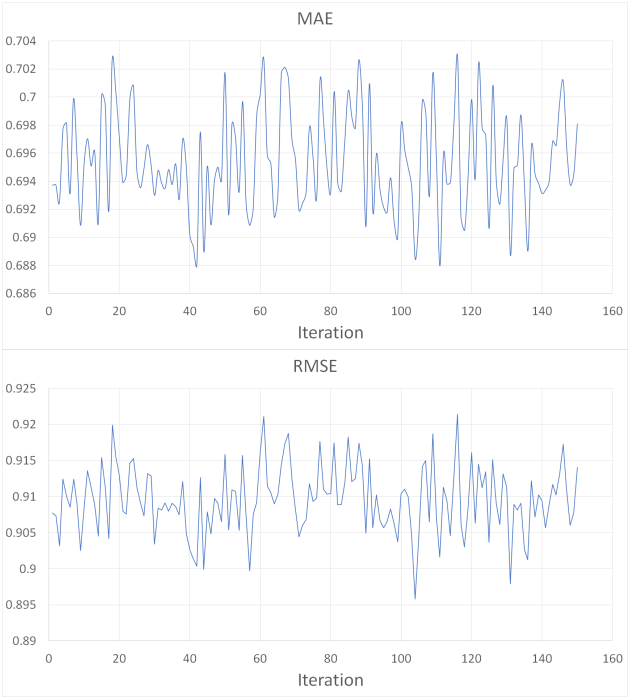

Figure 5.7: Accuracy Results of the Random-N Mechanism Applied to Ml-Latest-Small.

5.6.2/ Accuracy Consistency Test

This experiment was designed specifically to target the second proposed hypothesis, and it extends on the idea of the previous randomness notion and investigates the metrics used to test the effectiveness of the NNM methods (Table 5.5). The collective results from the earlier $Random - N$ simulation conducted are presented in Figure 5.8 for two datasets,

Ml-Latest-Small and Ml-100k. Figure 5.8 (right) depicts the values of MAE and RMSE that were previously shown in a different format in Figure 5.7. Analyzing those accuracy plots, it is apparent that there are plenty of cases throughout the 150 iterations (from the $Random - N$ scheme experiment) where the measurements portrayed conflicting results, which is the case for all four datasets used in our experiments. One notable example (marked in Figure 5.8) shows how in two consecutive runs, there was a 1.5% decrease in MAE versus a 2.5% increase in RMSE. This ultimately refutes the second hypothesis and signifies how the metrics adopted to test the success of the NNM measure are not a reliable performance measure and definitely should not be the sole test upon which the success of an NNM proposal is being evaluated.

5.6.3/ Gaps in the NNM Paths

The outcome of the two previous experiments and the analysis throughout the sections of this work have shown that the NNM field definitely lacks a well-defined, consistent approach. Further, it introduces many potential enhancement opportunities to become effective on an RS ultimately. From the randomness approach that triggered oddly comparable MAE and RMSE results to one of the best performing NNM algorithms (Section 5.1) to the inconsistency of the used evaluation methods on NNM (Section 5.2), this section summarizes all the possible gaps and weaknesses in the previously discussed publications in the NNM path. Those gaps are grouped into five primary categories.

5.6.3.1/ The Natural Noise Misconception and Inconsistency

Some proposals confuse the definition of NN in datasets and provide distinct explanations and implementation approaches, such as that of Tong et al. Tong et al. (2018). In their introduction, the authors discussed that what one user considers a mediocre rating (e.g., 1 out of 5 stars) compared to another who rates 3 out of 5 as bad is NN. In essence, this is not the appropriate definition of NN in O'Mahony et al. (2006), which was thoroughly discussed in the introduction of this study; however, it is merely an interpretation of users' unique standards and rating baselines. This effect does not occur solely across individuals but across varied cultures. Some countries, for instance, are more harsh with their ratings than others. Primitive CF methods have attempted to normalize these differences; one example would be the adjusted cosine similarity measure (ACOS) Bobadilla et al. (2013).

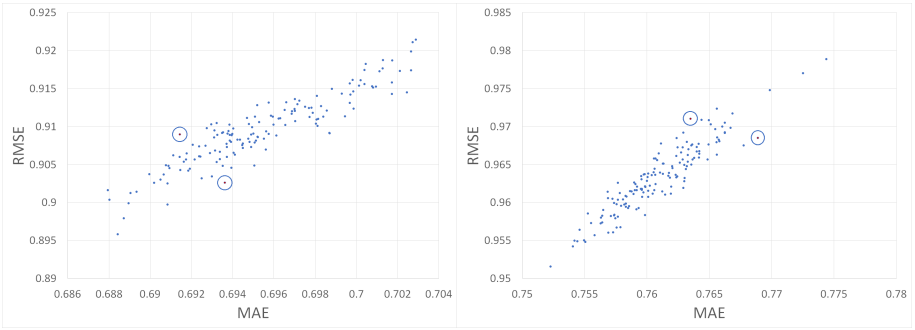

Figure 5.8: MAE vs RMSE for Ml-Latest-Small (left) and Ml-100k (right).

5.6.3.2/ The Magic Barrier - Accuracy Barrier Conflict

One of the most significant gaps in the Accuracy Barrier path is conceivably the RS evaluation methodology and the erroneous interpretation of the Magic Barrier of Herlocker et al. Herlocker et al. (2004). In Herlocker et al. (2004), the authors concluded that the accuracy metrics results of various algorithms clearly suggest that algorithm improvements in CF systems may come from divergent directions rather than just continued improvements in MAE or RMSE; it is possible that the most efficient algorithms should be measured in accordance with how well they can communicate their reasoning to users, or with how little data they can yield accurate recommendations. Lastly, the work hypothesized that modern metrics must be developed to evaluate those new algorithms. In this case, the authors are debating the effectiveness of the remarkably minute variations in MAE or RMSE (sometimes in the order of 0.01) when various RSs are evaluated against each other in terms of accuracy. Therefore, it is evident that the purpose of the revision in Herlocker et al. (2004) was never intended to introduce the Magic Barrier as the definition that was taken on by the first path (Accuracy Barrier) of the NNM approaches (the point at which the performance and accuracy of an algorithm cannot be enhanced due to noise in the data), but rather, it was abundantly evident that new metrics should be introduced to understand users in recommender system datasets better.

The Accuracy Barrier is based on the item opinions gathered from users who have already rated them at least once. Said et al. Said et al. (2012b) and Said et al. (2012a) explicitly state how to use views on previously rated items, and their initial ratings by the same users differ conceptually. However, they were essentially handled as being the same when the authors estimated the Accuracy Barrier. This proves that the Accuracy Barrier path relies on user re-ratings and an accuracy approach (mainly RMSE) to estimate the limit beyond which there can be no added improvement to the recommender's accuracy. There are two key weaknesses in this case; the first would be having to provide other opinions from users in a dataset to calculate the system's maximum performance, knowing that those opinions and their previous values differ conceptually. The second concern is the RMSE

function that was used to estimate the ultimate performance of a recommender. Is the Accuracy Barrier a proper measure of a recommender's maximum effectiveness, knowing that RMSE might not be measuring the true accuracy of a recommender (Section 5.2)?

The Accuracy Barrier track ultimately claims that a good "Magic Barrier" estimate is useful for assessing the quality of recommendations and for revealing room for improvements; however, the studies throughout the path do not propose or implement any methods that experimentally disclose those improvements nor how to improve a system after calculating the Accuracy Barrier. Markedly, Said et al. Said et al. (2012a) concluded that recommenders with prediction accuracy close to the estimated Accuracy Barrier could be regarded as "optimal systems". This conclusion also presents similar issues to what was raised in our previous point. An "optimal recommender output" represents a general term, especially when evaluating the RS's output using only MAE and RMSE metrics.

5.6.3.3/ THE ACCURACY BARRIER WEAKNESSES

As previously mentioned, calculating the Accuracy Barrier strictly depends on a primary phase, gathering users' viewpoints on previously seen and rated items. This proposal is eminent in all the works throughout the path, such as in Said et al. Said et al. (2012a), where the authors proposed that a real-world recommender system should regularly interact with users by polling opinions about items they have previously graded, allowing them the opportunity to audit their own performance and take measures to improve the recommendation engine where appropriate. This act poses a drawback in real-world applications since users must provide a second opinion on many items. Those users might have forgotten why a certain item received their dear appreciation in the first place, for instance, which would cause their second rating to become inaccurate; this begs the question, are the second user views provided in the MoviePilot experiments Said et al. (2012b,a) reliable and noise-free? Further, will those users genuinely care about enhancing the accuracy of an RS for a certain platform so much as to spend valuable time providing an accurate second opinion on products?

The authors pointed out in a later study Said and Bellogín (2018) that one major drawback of measuring and comparing the performance using only static, previously collected test data is that user behavior in the data is not always reliable. How will the second round of ratings collected from some users differ from the first? The Accuracy Barrier does not open doors for performance improvements but introduces another rating process equally (if not more) prone to NN. The rating elicitation process is intrinsically susceptible to noise due to several reasons touched upon in the introduction of this journal. In addition, subsequent studies should tackle the following missing points:

- A concrete definition of re-rating movies again after a certain period of time.

- A measure for an adequate amount of time to identify consistencies of ratings assuming this method is effective and accuracy remains our target.

5.6.3.4/ Accuracy Evaluation

As touched upon earlier in the introduction and throughout the experiment discussion, there is a certainly flawed assumption about the performance of RSs that requires further analysis of the Accuracy Barrier. All the studies Bellogín et al. (2014); Said and Bellogín (2018); Said et al. (2012a,b) define Herlocker's version of the Magic Barrier Herlocker et al. (2004) as the level of prediction accuracy that an RS can attain with the lowest possible error. Their version (Accuracy Barrier) reveals whether there is room for additional meaningful accuracy improvement or that further enhancement is typically meaningless. This approach contradicts the primary purpose of RSs, which was revealed with the introduction of the notion of the Magic Barrier in Herlocker et al. (2004). The outcomes of our experiment in Sections 5.1 and 5.2 revealed how the metrics applied to evaluate NNM algorithms of RSs result in conflicting outputs, and attempting to tackle the concept of an Accuracy Barrier for performance evaluation and improvement through the use of MAE and RMSE is counter-intuitive. Those two following points summarize the problems with accuracy and evaluation and should be considered in future proposals on the subject:

- Accuracy metrics that were employed to assess the performance of NNM methods (Table 5.5) should be re-visited to align with Herlocker et al. (2004).
- Other factors like diversity or serendipity Al Jurdi et al. (2018); Badran et al. (2019a) should be accounted for. The accuracy evaluations are still predominantly used and relied on, even in very recent proposals Li et al. (2019); Yera et al. (2020) on NNM. Serendipity and accuracy are significantly different in their nature, and increasing one leads to a decrease in the other Al Jurdi et al. (2018). Said et al. Saia et al. (2016) propose a method to calculate the similarity between items and remove the ratings of those from a user profile that are dissimilar. This can result in a sheer elimination of any form of essential serendipitous results that might occur in the recommender's output.

Throughout the fuzzy-inspired proposal that emerged in 2015 on NNM Yera et al. (2016, 2020), the authors suggested that the issue of natural noise in RSs is similar to that in fields independent from RSs such as Vaishali et al. (2015); López et al. (2013). Consequently, they were inspired to propose a user clustering mechanism based on fuzzy profiling. The accuracy problems discussed above remain in this case since the performance of NNM methods is still being assessed by the level of MAE and RMSE scores. Further to that, the fields that inspired the fuzzy NNM proposals Vaishali et al. (2015) (Noise Reduction Methods for Brain MRI Images) and López et al. (2013) (scenario of

imbalanced datasets) are intrinsically contrasting compared to RSs and their datasets. Recommendation engines' datasets are unique and, in most cases, contain explicit ratings of users. At the same time, those MRI images maintain an unrelated format and a peculiar application compared to recommendation engines.

5.6.3.5/ General Problems with the Approaches

Some general essential factors that must be taken into consideration in further proposals that might enhance the NNM paths are:

- A measure of the frequency of NN correction on datasets? Is there a certain limit beyond which applying NN becomes useless or counter-efficient?
- Almost all studies across the three paths create subsets from the datasets shown in Table 5.6. This shows that the algorithms' high-time complexities with the noise detection and correction methodologies must be properly addressed and eventually benchmarked.
- As seen at the beginning of this section, accuracy might not be the most suitable measure. Therefore, an NNM approach that is computationally demanding and produces a superior MAE or RMSE result, in the end, should be reviewed on different levels before ranking it as an optimal method for NNM compared to the others in the path.
- Most published studies in the NNM paths use CF recommender algorithms to evaluate their approaches to NN. Will a good-performing NNM method still be effective when the recommender algorithm is inevitably modified?
- There are other problems with CF approaches that researchers usually overlook. For example, Bag et al. Bag et al. (2019) argue that removing noise from the dataset amplifies their sparsity issue, which is conceptually true. They continue adding that Toledo et al. Toledo et al. (2015), who used PCC to predict ratings as a replacement for noise, have increased the issue by adopting a flawed correction measure as the environment might be sparse. PCC performs poorly in scarce environments Bag et al. (2019). PCC does indeed perform poorly in terms of accuracy for users who maintain a limited amount of ratings; however, the authors never discussed critical factors like the sum of ratings the users that were given re-ratings had, the neighborhood size, the method to decide upon a correct neighborhood size, etc. In Bag et al. (2019), the authors chose to employ another similarity measure for noise correction, ruling out PCC as an option, and still not providing any information about the critical variables of an RS, namely the neighborhood size, the number of items those noisy individuals had, their respective contribution to their neighborhood, etc.

5.7/ CONCLUSION

Implementing an effective and agile natural noise management algorithm for recommender systems' datasets is challenging due to various parameters that should be considered, especially in the evaluation process. There has been no attempt to synthesize what is traditionally known about the performance evaluation of recommender systems and natural noise management, nor to systematically recognize the implications of evaluating them for numerous tasks and diverse contexts while testing the performance of a natural noise technique. Throughout this comprehensive study, we surveyed and categorized all the natural noise-handling algorithms starting from their inauguration in 2006. In addition, we carefully introduced empirical results from two hypotheses that provided critical insight into the consistency of the evaluation methods used in the proposed noise management techniques. The first experiment illustrated how randomness could achieve comparable outcomes to one of the most conventional mechanisms, while the second proved that the metrics employed to test those techniques and rank one better than the other typically display inconsistent and unreliable results. We hope this article will naturally increase the awareness of the evaluation of recommenders, especially in the natural noise management field, and encourage the development of more standardized natural noise methods evaluated by measures beyond traditional accuracy.

As seen in the previous section, many gaps constitute the natural noise management field and undoubtedly require considerable attention in the future. The potential problems we set to address include the development of proper evaluation methods that researchers or practitioners can broadly use with any recommender technique and serve us to assess better the actual effectiveness of a recommender devoid of inconsistencies. In addition, natural noise lacks proper development regarding the type of datasets it is applied to. A practical noise management approach must be scalable and adequately work with diverse datasets irrespective of the recommendation algorithm employed. External data that administrators must retrieve from customers to implement a particular noise management approach remains an inadequate solution to the problem.

6

Strategic Attacks on Recommender Systems

6.1/ Overview

Understanding user behavior in the context of recommender systems remains challenging for researchers and practitioners. Inconsistent and misleading user information, often concealed in datasets, can inevitably shape the recommendation results in specific distorted ways despite utilizing recommender models with enhanced personalizing capabilities. Naturally, the quality of data that fuels those recommenders should be highly reliable and free of any biases that might be invisible to a model, irrespective of its type. In this article, we introduce two modern forms of noise that are intrinsically hard to detect and eliminate; one is malicious and will be termed Burst, while the other is unique in that it forms its category and will be referred to as Opt-out. Additionally, to segregate the nature of noise behind such threats, we present a distinct case study on Burst and Opt-out to illustrate how the detection of those threats can be challenging compared to that of traditional noise and with the current detection methods. Finally, we expound on the ability of such threats to bias the output of recommenders in their unique way while primarily retaining data that is not fundamentally erroneous.

6.2/ Introduction

Throughout the years, Recommender Systems (RSs) have become increasingly essential to online businesses, irrespective of their size, especially in the e-commerce field Aggarwal et al. (2016); Scholz et al. (2017). Recently, talks about methods of scaling the e-commerce sector have dominated the majority of webinars and articles, especially with the increased shift towards online-based services Meyer; Schoenauer; Columbus; Smaros and Falck (b). Comprising a varied range of implementation methods that extend

from the prominent CF techniques to advanced latent factor models, recommenders partake in most top-ranked commercial platforms like Amazon, Netflix, Spotify, Last.fm, etc. Ricci et al. (2010) and enormously contribute to their success. This originates from the substantial problem such approaches try to tackle by attempting to provide highly personalized services: information overload. As a result, it is not surprising when the demand for employing recommenders profoundly increases as the shift to online platforms registers a sharp advance.

The primary power of the personalized recommendations generated by various RSs highly depends on abundant user contributions in ratings, reviews, tags, likes, etc. Researchers studying and enhancing RSs and their algorithms have merely focused on algorithmic improvements paying nominal attention to the quality of the underlying data. The involvement of the human factor in processes such as rating elicitation renders it immensely prone to errors that might occur deliberately or naturally. Ratings, reviews, and other details recommender algorithms depend on holding critical information that might not always be genuine or reliable. This is recognized as noise in the datasets used by RSs. Naturally, if RSs train on inaccurate data to learn and predict user behavior, they will inevitably have inconsistent results.

Previous research shows two primary types of noise in RSs: malicious and natural. Briefly, the general definition of noise in datasets is the rating feedback that does not reflect a user's proper preference or intention. This anomaly might be purposely set by outsiders in the form of attacks on a system to bias the output (malicious noise) Gunes et al. (2014). It could also occur naturally due to users' inconsistent rating behavior (natural noise) Amatriain et al. (2009a,b). Malicious noise results from numerous forms of attacks carried out on online applications that are typically powered by diverse types of RSs, and it has witnessed much of the research attention in the past few years Gunes et al. (2014); conversely, the natural noise domain has not yet received a lot of focus from researchers, and lately, it has become an exciting topic in the study of anomalies in datasets Ricci et al. (2010).

In contrast to malicious noise, natural noise occurs inherently due to specific user-specific behavior, making it unique and unusually complex to model. It's completely arbitrary and user-dependent: Users can be miserable or feeling down on a particular day and rate all recommendations they encounter on their favorite platform as bad, even the genres they tend to prefer – Natural noise due to an emotional state. Significant improvements are still required to develop a generic noise-aware layer compatible with all RSs and capable of overcoming natural and malicious inconsistencies in datasets Jurdi et al. (2021). In addition, such a unified system could deal with noise irrespective of its type and independent of the deployed recommendation engine.

In this article, we introduce and discuss a new noisy user behavior that is obfuscation-

based (opting out from a system). Further, we demonstrate how this mechanism could be segmented into two main types, one of which does not belong to any of the two noise categories presented above and will maintain its class called Obfuscation, while we will refer to the noise itself as Opt-out. The other type, which we will name Burst, retains a malicious component and belongs to the adversarial noise class. This new behavioral noise is purely user-intended, a behavioral form affecting the authenticity of item feedback, and can be indirectly harmful to a recommender's output. Using the neighborhood-based assessment method Al Jurdi (2022), our study shows how the effect of Obfuscation is very detrimental to the local group of users, such as a user's neighborhood in a K-Nearest Neighbour. This implies that such noise is generally unnoticed when using conventional evaluation metrics, and only when evaluating the performance on a granular level, can we detect its actual effect.

The rest of the work is organized as follows: Section 2 discusses the obfuscation noise background, while Section 3 introduces the two newly discovered noise types in RSs. In Section 4, we present and discuss the simulation experiments and their results, and in Section 5, we conclude the work.

6.3/ Background and Related Work

In this section, we will discuss the obfuscation mechanism and how it affected recommender-related applications in the past. We will cover two significant categories of Obfuscation and touch on their relationship to RSs.

6.3.1/ Obfuscation as Twitter Phenomenon

On several occasions, such as the elections in Russia (2011) and Mexico (2012), Twitter became the court for pivotal attacks that ultimately deviated from the targeted public opinion; they came to be known as obfuscation attacks Brunton and Nissenbaum (2015). During those two incidents, people relied on Twitter to convey a specific public message and plan movements for government-targeted protests, an effective way that has become quite mainstream in many countries nowadays (Twitter revolutions Comninos (2011)). Entities wanting to oppose the information flow fell short of initiating attacks (such as Distributed Denial of Service – DDoS) on highly secured platforms such as Twitter. Instead, they resorted to a unique way of tampering with the system's algorithm. To trick the system, the attack was aimed at highly-relied-on hashtag trends and timeline recommendations and worked through injecting random and false posts under said hashtags. The plot in Figure 6.1 shows the average combined behavior from three very famous hashtags where we can observe how the number of relevant hashtag tweets, in a relatively short

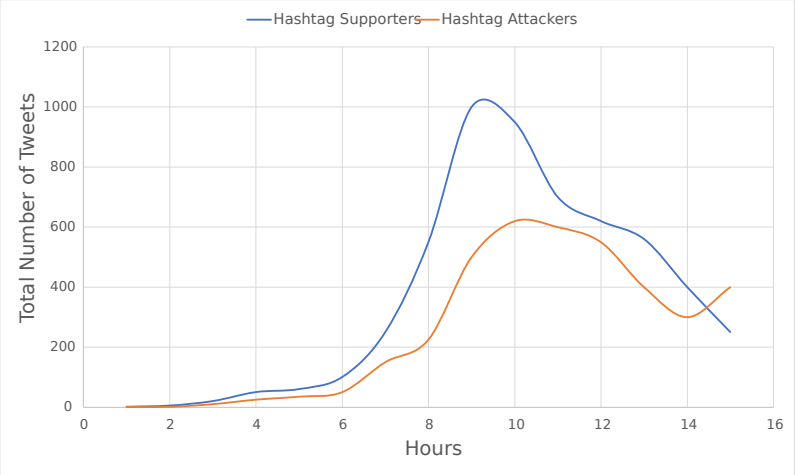

Figure 6.1: Burst attack on a Twitter hashtag.

period (an average of 9 hours), decreased quickly after the attacks flooded the timelines. Ultimately, the extraneous injected tweets dominated the stream for the target topic to an extent where those relevant to it were completely overshadowed.

6.3.2/ Obfuscation as User Weapon

Naturally, account privacy in a recommender-powered system, or any other social platform, is essential, and it's becoming much normalized as awareness about personalization protocols continues to increase Badsha et al. (2016); Cai and Zhang (2019); Zhang et al. (2019a). Those online platforms, albeit continuously reassuring about personal privacy protection, often conveyed in exaggerated ad campaigns that mask indiscriminately agreed-on privacy policies, cause users to stay connected and use them voluntarily. Deloitte's 2017 study proves it: 91% of people in the US consent to legal terms and services conditions without reading them Smaros and Falck (a). Furthermore, many of those applications, from online payment solutions to massive social networking platforms, have become integral to our routine. In this case, a different form of Obfuscation emerges Brunton and Nissenbaum (2015) as a free and elementary attack that users could leverage to opt out from those systems; it is different than the obfuscation-powered Twitter attacks in that it is primarily utilized by ordinary individuals who find themselves having shared, whether implicitly or explicitly, countless personal preferences with online systems and want to quit. That said, users who choose to do this don't just disable their accounts or refrain from logging in again. Instead, they tend to initiate a self-destructive profile behavior mechanism by introducing loads of information (in the form of ratings, likes, posts, etc. – that depends on the platform) that are not erroneous but inconsistent with their predispositions. As a result, this tricks the system and conveys false information about the content that genuinely engages this user, further amplifying a hidden form noise

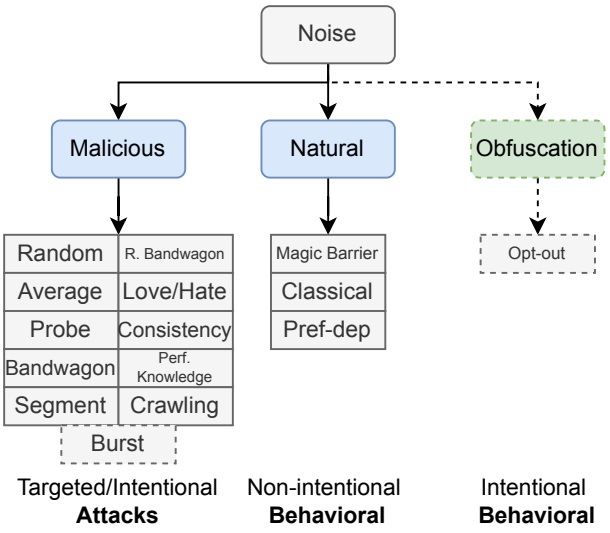

Figure 6.2: Noise Branches in Recommenders, Including New Obfuscation Forms.

in the dataset. To the system, such users maintain average profiles and merely refine or change their tastes over a certain period, but what happened was that those users subtly leveraged an opt-out obfuscation attack masking their interests and concealing their preferencesBrunton and Nissenbaum (2015).

6.4/ A NEW TYPE OF NOISE IN RECOMMENDERS

The above-introduced notions about Obfuscation allow the formation of two types of attacks on recommenders. The first is Burst, in which attackers use fake/inactive profiles to deviate from a particular opinion and tamper with the system to target a group of users specifically. The second type, Opt-out, is utilized for personal reasons should users decide to eliminate any data that might constitute their profile preferences. Those types of attacks were not mentioned in previous research about anomalies and noisy user behavior in recommenders Gunes et al. (2014); Si and Li (2020); Jurdi et al. (2021); Badran et al. (2019b). Figure 6.2 shows the noise categories with the new obfuscation types.

Burst in RSs is similar to that in the Twitter case (Section 6.3). It's even present in traditional recommender datasets extensively used in research studies. Figure 6.3 shows one example of such users. Like the Twitter bots, this user was inactive for a lengthy period before suddenly registering extensive activities and then going dormant. Typically, in an online setting, Burst can be leveraged to target a specific trending item (resembling the Twitter scenario), and the recommendation system needs to curb such behavior as it could negatively influence the general opinion. Therefore, we define the following two obfuscation-based attacks in recommenders:

- Burst attack: An RS attack strategy that targets a group of users, mainly to deviate

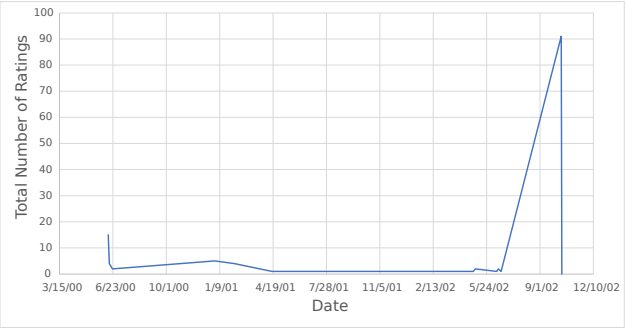

Figure 6.3: Rating Activity of a User from ML-1m.

their opinion. It utilizes fake or inactive profiles and tampers with the whole system.

- Opt-out attack: An RS attack that a single individual mainly uses to eliminate any data that might constitute his personal profile status.

6.4.1/ NOISE ALGORITHM

Natural noise in datasets can be uncovered through several approaches Jurdi et al. (2021). For that purpose, we selected the most famous natural noise management and user/item clustering methodology Toledo et al. (2015) from the natural noise path that's thoroughly discussed in Jurdi et al. (2021). After that, we combined it with the work done in Al Jurdi et al. (2018); Badran et al. (2019a) for serendipity detection and analysis and overall parameter tuning for a more optimal noise detection output. This strategy allows us to detect if a particular rating by an individual indeed deviates from his usual predilections. The principal clustering method that runs as a pre-recommender step (completely independent of the recommendation system employed) on the dataset itself classifies all users and items into distinct groups; every rating a user has will be examined against their unique overall profile if it accommodates their type then it's likely a correct rating and if doesn't, then most probably it comprises noise Toledo et al. (2015).

In addition, applying a serendipity-oriented approach, we ensured that the actual noise is not confused with uncertainty since there's a fine line between the two Al Jurdi et al. (2018); Badran et al. (2019a). In the real world, user tastes undergo alterations as they explore new items in the vast inventories. A recommender must be robust yet flexible enough to suitably handle those variations, employing acceptable ranges of churn, responsiveness, and uncertainty without overdoing it - user interests, likes, dislikes, and fashions inevitably evolve with time Aggarwal et al. (2016). Those substantial factors have been surprisingly overlooked in almost all the studies in the natural noise field where the research path became predominantly fixated on accuracy-related metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) Herlocker et al. (2004); Jurdi et al. (2021).

6.4.2/ OBFUSCATION DETECTION MECHANISM

Natural noise algorithms can help us detect noisy behaviors Jurdi et al. (2021); however, exploring the dataset for opt-out scenarios is our main aim. It might be enough to analyze the natural noise percentage in the last couple of days for the users who abandoned the system. Still, we hypothesize that it is much more accurate to examine a full retrospect of the profile of an opt-out candidate. As the last experiment will show, opt-out Obfuscation in datasets can be in different forms, and rating peaks aren't just located in the previous few days. The opt-out attack case can be equivocal and very easily overlooked by the system as it's similar to a regular activity that might even be the outcome of serendipitous discoveries (as mentioned in the introduction of Obfuscation). In an attempt to generalize an opt-out detection strategy, we propose the following equation:

$$u_{(opt-out)} = \frac{|d_{(n,u)}|}{|N_u|} > 0.5, |N_u| > 0 \tag{6.1}$$

Where $u_{(opt-out)}$ is a potential opt-out candidate, $|d_{(n,u)}|$ is the total noise on the last day of the user activities, and $|N_u|$ is the total number of noise for user u. The measure for abandoning the system can be easily achieved by ensuring that the day in $|d_{(n,u)}|$ is much older than today's date, or in case of offline datasets such as the one we are using for this example, the last day it was published online. To test the impact of opt-out malicious behaviors in the dataset and their hidden effect on the performance, we define the following characteristics of the ratings that were considered to be eliminated from the dataset for the experiments in the next section:

- A large number of ratings in a very short period (e.g., 1 or 2 days at most).
- A significant variation of taste between peak rating days and other normal days.
- A significant noise score on the peak day (Equation 6.1).

6.5/ SIMULATION AND RESULTS

In this section, we will introduce the experiment setup used to simulate the obfuscation noise in RSs and then discuss the results and the effect of such noise on the users of our system.

6.5.1/ Experimental Setup

6.5.1.1/ Datasets and Algorithms

In our experiments, the ml-latest-small dataset and the ml-1m Harper and Konstan (2015) are used to test the opt-out introduced in the previous sections. The smaller set has 610 users, 9.7k movies, and 100k ratings, while the larger has 6k users, 4k movies, and 1m ratings. The algorithm employed in the experiment is a CF recommender Aggarwal et al. (2016) that considers each user's mean ratings with the parameter k set to 40 neighbors and a Pearson Correlation similarity measure. Natural noise in datasets can be uncovered through several approaches Jurdi et al. (2021). Our experiments selected a famous natural noise management and user/item clustering methodology proposed by Toledo et al. Toledo et al. (2015). The evaluation process used for the opt-out obfuscation experiments follows the neighborhood evaluation process presented in Al Jurdi (2022) with MAE, RMSE, and the Normalized Discounted Cumulative Gain (NDCG).

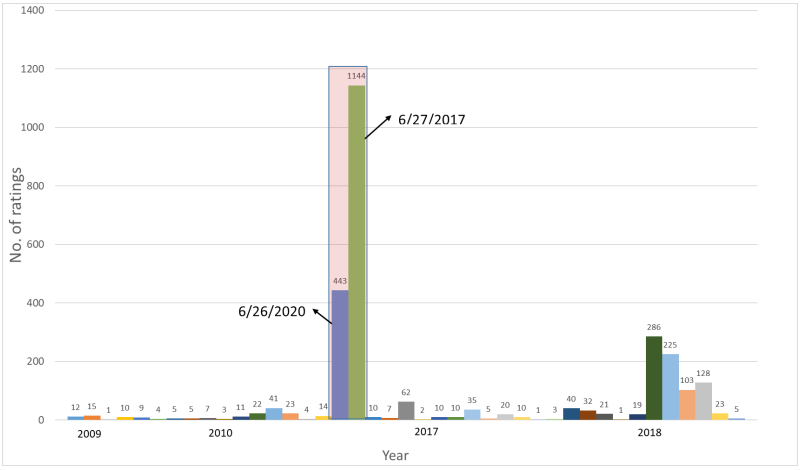

Figure 6.4: User Profile Attack Case.

6.5.1.2/ Target User Profiles

We select two profiles from each dataset that satisfy the above conditions to evaluate the performance using the neighborhood-based method. Both exhibit malicious behaviors and resemble mild obfuscation attacks where most of their ratings contain malicious or natural noise. These profiles have ratings that abruptly shift in taste and don't have a reasonable amount of ratings on certain days, which is distinctly counter-intuitive to the case of a regular user in a real-world scenario. The natural noise algorithm signaled that most of those users' ratings in their peak rating days were noise and did not fit their genuine overall profile predilection.

In addition to the two profiles selected for the tests, we run the experiments in parallel

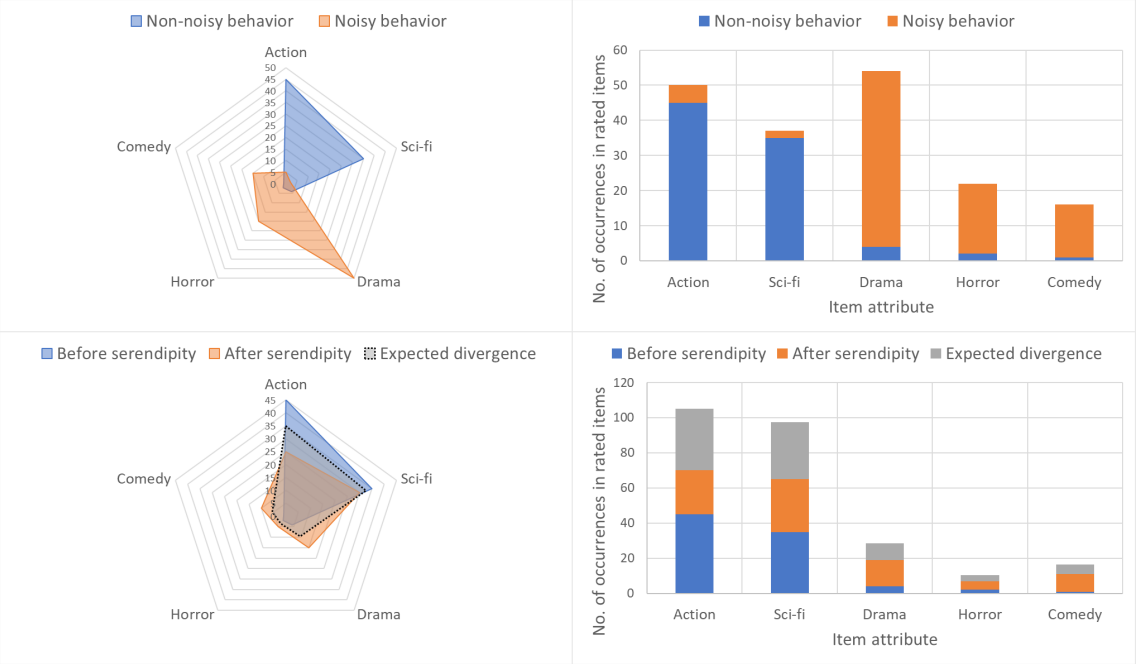

Figure 6.5: Item attribute sudden variation example (top) and a case of serendipitous discovery (bottom).

on the case of legitimate ratings to test if the neighborhood evaluation reports different results. Honest ratings are selected from other users who do not meet the above critical conditions. In contrast, the total number of ratings is selected to be equal to that of the malicious case in both datasets. In the two cases, the total number of ratings eliminated from the dataset is around 1.5k (i.e., 1% of ml-latest-small and 0.15% of ml-1m).

6.5.2/ Case Study - User Rating Noise and Sudden Taste Variation

Figure 6.4 shows the malicious ratings of a user from the ml-latest-small dataset that meets the target profile conditions presented in the previous section. There is increased activity on two specific days where the natural noise factor in them registers 82%. This goes hand-in-hand with the user taste variation on those days, as the item attributes are significantly different from the profile preferences. The abrupt change in taste is also demonstrated in the examples of Figure 6.5 (top). For brevity, only the most affected attributes are displayed. The Figure shows how items of new genres, such as drama, horror, and comedy, are given very high ratings before returning to normal. A very similar user is selected from the ml-1m dataset. Finally, we note that Badran et al. Badran et al. (2019a) and Al Jurdi et al. Al Jurdi et al. (2018) had similar observations about uncertainty and the item discoveries that users undergo in RSs. Based on their discussions on serendipity and the difference between it and noise, we can predict how a standard user profile might generally vary due to certain serendipitous discoveries. This is shown

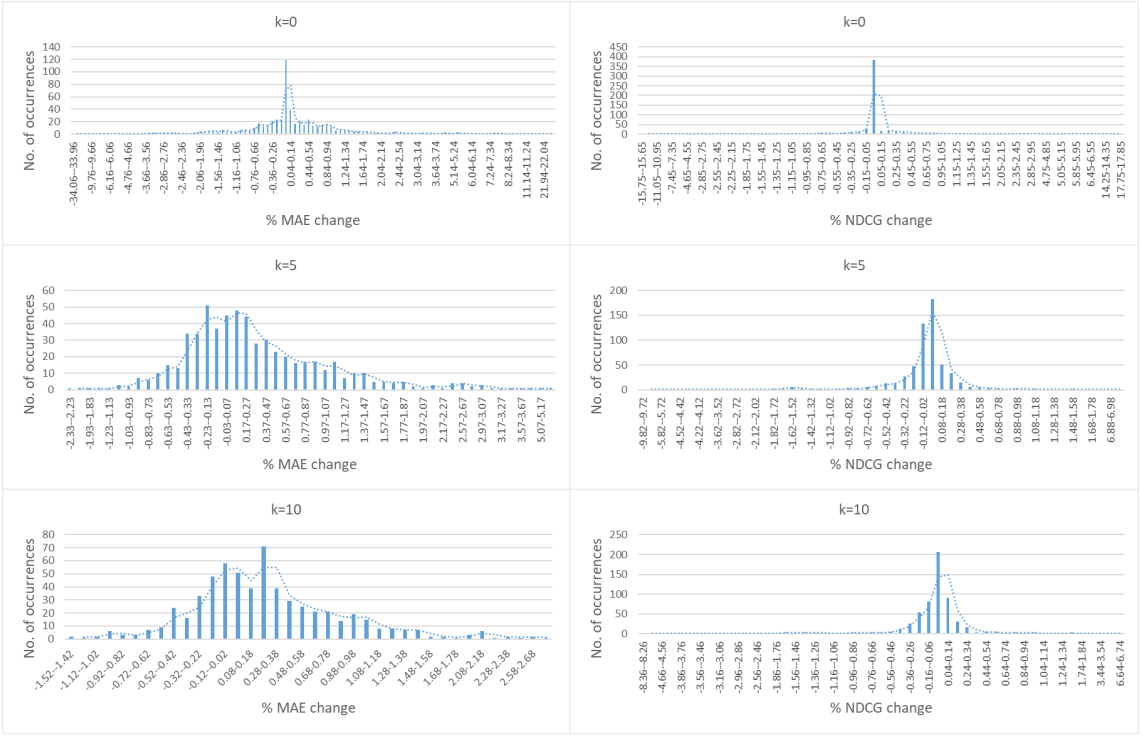

Figure 6.6: MAE (left) and NDCG (right) results on ml-latest-small using the neighborhood-based mechanism with different neighborhood sizes.

in Figure 6.5 (bottom).

6.5.3/ Effect on the System - Experimental results

After the target malicious ratings have been identified for two users in both datasets, we test the effect on the system with two methods. The first is the conventional method, which is typical offline evaluation metrics. Such approaches are used to evaluate the effectiveness of new proposals in the natural noise research Jurdi et al. (2021) and data poisoning Li et al. (2016); Fang et al. (2018). The second test is the neighborhood-based evaluation mechanism, which allows evaluation at a granular level based on neighborhood clusters.

6.5.3.1/ Impact on the System

Tables 6.1 and 6.2 summarize the results of the percentage change in the metrics before and after eliminating the identified malicious user behavior for both the conventional method and the neighborhood-based one (k in the neighborhood-based case stands for the neighborhood size). For clarity, the results for every neighborhood, Figures 6.6 (ml-latest-small) and 6.7 (ml-1m) show the percentage MAE and NDCG change for several values of k and for the case of eliminating the target malicious ratings only. First, it's clear

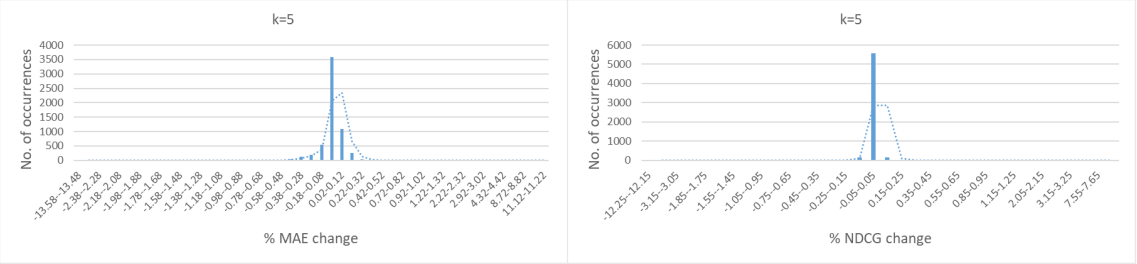

Figure 6.7: MAE (left) and NDCG (right) results on ml-1m using the neighborhood-based mechanism with $k = 5$.

Table 6.1: The impact of removing malicious ratings on the overall system metric.

		Test case - target ratings	
Dataset	**Metric**	**Legitimate ($\delta\%$)**	**Attack/Noise ($\delta\%$)**
ml-latest-small	MAE	0.01	-0.01
	RMSE	0.03	-0.01
	NDCG@10	0	0.02
ml-1m	MAE	-0.01	0.31
	RMSE	0.01	0.3
	NDCG@10	0.1	-0.04

from the data in Table 6.1 that the standard method does not report any significant impact on the system before or after removing the target malicious ratings in the two datasets. The change is nominal and registers a mere 0.01% decrease in MAE and RMSE after the attack in the case of ml-latest-small with a 0.02% increase in NDCG@10. In the case of ml-1m, the removal of the attack caused a somewhat opposite effect from the ml-latest-small dataset with around 0.3% increase in MAE and RMSE as opposed to a 0,04% decrease in NDCG@10. Finally, and as expected, the results of the malicious target ratings case are very close to that of the legitimate ratings, which also resulted in minor variations after the rating removal (Table 6.1). One cannot even know that something might be wrong with the data due to such marginal effects. The malicious case cannot be spotted; therefore, nothing appears wrong with the dataset in both cases.

Conversely, the neighborhood-based mechanism reported different results for the same cases on both datasets while it similarly registered the same findings for the legitimate target users' case. Table 6.2 shows a relatively large fluctuation in both MAE and NDCG@10 (especially for $k = 0$) in the case of malicious rating removal. With the ml-latest-small dataset, MAE scored a significant 5% increase for several neighborhoods $k = 5$ while also registering a lower 2% decrease for others. This generally means that removing the rating causes a slightly more oriented shift towards more accurate recommendations in many aspects of the dataset. On the other hand, MAE resulted in a more significant decrease in the case of ml-1m for the selected malicious ratings of the user. In this case,

Table 6.2: Examining the impact of removing negative ratings through our neighborhood-based approach with varying neighborhood sizes.

			Test case - target ratings					
			Legitimate ($\delta\%$)			**Attack/Noise ($\delta\%$)**		
Dataset	**Metric**		**0**	**5**	**10**	**0**	**5**	**10**
ml-latest-small	MAE	Avg	-0.0028	-0.0158	-0.0049	0.2104	0.302	0.2349
		Std	0.243	0.194	0.178	3.143	0.828	0.609
		Max	1.05	0	0	22.01	5.13	2.66
		Min	-1.78	-1.02	-0.8	-21.73	-2.33	-1.52
	NDCG@10	Avg	0.0038	-0.0145	-0.0042	0.0739	-0.1884	-0.1701
		Std	0.067	0.167	0.081	2.138	1.007	0.868
		Max	1.16	0.52	0.39	17.81	2.09	3.47
		Min	-0.27	-2.3	-1.27	-15.75	-9.82	-8.36
ml-1m	MAE	Avg	0.002	0.0023	0.0017	-0.0041	-0.033	-0.038
		Std	0.105	0.041	0.034	0.422	0.388	0.288
		Max	1.42	0.6	0.68	8.62	11.2	6.47
		Min	-1.17	-0.3	-0.19	-15.19	-13.58	-5.92
	NDCG@10	Avg	0.0005	-0.0002	-0.0007	0.0036	-0.0014	0.0039
		Std	0.046	0.042	0.039	0.5	0.288	0.155
		Max	1.27	0.68	0.57	12.03	9.12	5.04
		Min	-1.13	-1.38	-1.27	-17.2	-12.25	-3

removing the rating negatively affected some neighborhoods rather than being more oriented towards a positive accuracy change. We speculate that many profiles in the ml-1m exhibit natural and malicious noise, which could affect the results since they weren't eliminated. We are only evaluating the performance using the local neighborhoods of users after minimal malicious data has been eliminated. NDCG@10 registers a significant increase for $k = 0$ after the attack and a slightly smaller decrease for the others. This has to do with the nature of the ranking-based evaluation method and how Discounted Cumulative Gain (DCG) measures the relevance of ratings in the dataset. Lastly, it is worth noting that the percentage change decreases as the value of k increases and gradually diverges towards the percentage change for $k = N$ (where N is the total number of users in a dataset).

6.5.3.2/ Impact on the Neighborhood Recommendations

In the second part of the experiment, we take a closer look at the effect on the target users' neighbors. Table 6.3 shows the impact on the neighborhood of the users after the malicious ratings were eliminated. In both cases, the neighbors of the users markedly changed after the malicious rating removal. As shown in the Table, the first user (case of ml-latest-small) shows a minimal similarity (5.26%) between his two neighborhoods, while the second registers no similarity at all. We can safely say both target users had a new

Table 6.3: The impact of removing malicious noise on the community and recommendations.

Case	Correction Status	Ttop-10 Neighborhood	Recommendations (item order)
ml-latest-small	Before	{53,175,154,496,366,87,319,214,25,138}	{1,2,3,4,5,6,7,8,9,10}
	After	{2,8,11,12,26,31,35,37,44,53}	{1,2,4,11,6,9,10,12,13,14}
	Similarity Score	**5.26**	**46.7**
ml-1m	Before	{556,88,276,25,595,72,550,515,53,511}	{1,2,3,4,5,6,7,8,9,10}
	After	{2,7,10,11,12,13,14,26,29,31}	{11,6,7,12,4,5,13,14,15,16}
	Similarity Score	**0**	**25**

Table 6.4: Effect of removing malicious noise on the top-10 recommendations for the two test users' neighborhoods.

Case	Most Affected Neighbor	Malicious Ratings Status	Recommendations (item order)
ml-latest-small	276	Before	{1,2,3,4,5,6,7,8,9,10}
		After	{1,2,4,5,11,12,13,14,15,6}
		Similarity Score	**33.34**
ml-1m	154	Before	{1,2,3,4,5,6,7,8,9,10}
		After	{11,1,12,13,2,3,14,5,15,16}
		Similarity Score	**26.7**

community after the malicious ratings were eliminated. As for the recommendations for the target users before and after correction, the order and the content varied significantly, as shown in the same Table. The new neighborhood yielded four items in the top 10 of the first case and six new items in the second case. After eliminating malicious data, those new recommendations are more convenient for the user's authentic profile.

As signaled by the neighborhood results in the previous section, the primary effect of malicious ratings of the users in both datasets lies in the recommendations of the neighborhood of the target users and not just their suggestions. For that, we analyzed the recommendations of the neighborhood before and after the malicious data, and we found that they were indeed affected. Table 6.4 shows the most affected neighbor for both users, and it can be seen that the similarity between the items has considerably changed in both cases. The first was presented with five new unique items in his top 10 after the correction, while the second had six new items. The order of the items in the top-10 list has also changed drastically, proving the neighborhood evaluation results in the previous section. Digging further, we find that those recommendations differ a lot in content. Figure 6.8 displays the types of items (genres) for the most affected users before and after the correction. It is evident how the recommendations of one of the top neighbors of the user in the ml-1m case completely shifted from Drama, Romance, and Comedy to Mystery and Adventure when his profile was corrected. The most affected neighbor in the ml-latest-small case also registered a difference in the recommended content, where the new top-10 items are more oriented towards Drama, Action, and Western than Romance, Comedy, and Thriller. It's important to note that new genres popped up heavily in the recommendations after the correction was made in both cases. They are Mystery

Figure 6.8: Top-10 list of genres before and after removing the malicious rating for the most affected neighbors (276 - left and 154 - right).

and Crime in the first user's neighborhood and Western and Action in the second user's neighborhood. This shows how minor malicious variations that generally go undetected can affect the actual preferences of the communities.

6.6/ Conclusion

Implementing an effective and agile natural noise management algorithm for recommenders is challenging due to numerous parameters that should be considered, especially in the evaluation process. As demonstrated in this study, the obfuscation phenomenon created yet another challenge to the evaluation process. We have introduced two modern forms of noise that are hard to detect with current evaluation strategies and showed how the data appears to be perfectly normal. The impact was only visible when we evaluated the performance in data subgroups using the new proposed group validation process in the evaluation ecosystem of recommenders. Additionally, there has yet to be an attempt to synthesize what is known about the various noise categories in RSs, nor to systematically devise a unified protocol that would deal with noise irrespective of its type and independent of the deployed recommendation engine. Whether it's user-induced to opt out of data processing for particular security concerns or publicly injected by authorities, such as in the case of Russia, Mexico, and Lebanon, Obfuscation is a challenge that RSs should be aware of.

Opt-out attacks pave the way for multiple discussion paths that cover numerous topics; for instance, identifying a user's opt-out behavior can permit tracing back to the primary user's tastes. Additionally, data owners can develop data mining methods to discover the general trends of users opting out of the online platform.

V

Revitalizing the Evaluation of Recommender Systems

This section delves into the evaluation gaps previously discussed in the fourth chapter. We also explore the challenges of evaluating recommendation systems and testing their performance as the underlying data evolves. Our focus is on the current state of the art, which has several evaluation frameworks to standardize the assessment process. However, there is a lack of reporting on the performance of models on dataset subgroups and detecting specific kinds of potentially malicious data behaviors. In light of this, we propose a new evaluation strategy, the group validation framework, which is a valuable assessment tool for monitoring the performance of a recommender on specific critical clusters or groups of data. Our findings show that recommenders perform differently in various groups as unique data perturbations are introduced.

7

GROUP VALIDATION: MULTI-LAYER EVALUATION FRAMEWORK

7.1/ INTRODUCTION

Recommendation systems analyze information in datasets to predict what may interest users. This process utilizes available data features like search and interaction details. Although it is relatively simple to achieve with current research and suitable models or workflows, evaluating effectiveness and testing performance as machine learning models evolve with data is still a significant challenge Bellogín and Said (2021); Macdonald (2021); Ovaisi et al. (2022) that has captured considerable attention Han et al. (2021); Ferro et al. (2018); Valcarce et al. (2018). Microsoft's best practices for building recommendation systems have been helpful in this regard Argyriou et al. (2020). As a result, there have been some modern proposed approaches to tackle the evaluation challenges in recommender systems; those include the Simpson's paradox Macdonald (2021), benchmarking frameworks and toolkits Sun et al. (2020); Anelli et al. (2021); Zhao et al. (2021), dataset-oriented design Chin et al. (2022), metric selection criteria Tamm et al. (2021); Han et al. (2021), and metric adaptations (predominantly in search settings) Castells et al. (2022); Parapar and Radlinski (2021); Drosou and Pitoura (2010); Fang and Zhai (2005). Although these topics tend to be diverse, almost all revolve around one prevalent idea: the tools and techniques for assessing models still require significant improvements. In short, there have not been enough attempts to consolidate the knowledge of recommender systems nor to systematically define the implications of evaluating recommenders for different tasks and under various contexts.

The topic of evaluation challenges in recommender systems is not recent and dates back to 2004. At the time, Herlocker et al. Herlocker et al. (2004) abstracted all the factors considered in evaluating a recommender model's effectiveness and proved the presence of potential biases in most reported evaluation results. Back then, the study highlighted the pitfalls that researchers ought to avoid when attempting to implement recommendation

solutions, especially during the phase of trained models' effectiveness evaluation. Some of those issues included are undefined user goals for a particular recommender system, evaluation of models on incompatible datasets, and incorrect use of evaluation metrics that are mostly not optimized to measure the performance based on the system's prediction goals. Fast-forward to today, the same factors are still causing significant issues, as evidenced by recent comprehensive studies on the topics of noise and evaluation, such as those by Parapar et al. Parapar and Radlinski (2021), Al Jurdi et al. Jurdi et al. (2021), and Ovaisi et al. Ovaisi et al. (2022). In the research and experiments of Jurdi et al. (2021), systematic proof shows how the tools and metrics employed to test contemporary model techniques and rank them in order of performance embody inconsistent and unreliable results. Some examples of the outcome include contradicting performance metric results, a sporadic evaluation metric selection process, and unsystematic model-data combinations in experimentation.

Further, the experiments demonstrate how randomly eliminating data from a dataset (in the training and evaluation phase) could result in shockingly comparable performance results with algorithms that specialize in identifying critical noisy data that severely impact performance. Even if not directly correlated, this aspect is perfectly aligned with the recent research by Sun et al. Sun et al. (2020), which shows how a simple baseline model can outperform its superior and more complicated counterpart with the proper setup. Highly similar issues are also presented by Parapar et al. Parapar and Radlinski (2021) in their proposal of a new unified metric that combines diversity and accuracy and can be used as a replacement for the outdated and less effective traditional measures.

Building on the arguments presented in Parapar and Radlinski (2021); Jurdi et al. (2021); Ovaisi et al. (2022), and in an attempt to address the issues mentioned above debated in Jurdi et al. (2021), our work focuses on the objective of evaluating a recommender performance on sub-groups as opposed to solely relying on the traditional applications of evaluation metrics. While the current evaluation mechanisms can provide a general indicator about how the performance is expected to be in a live setting, an "averaged" evaluation, even with reasonably optimized metric results, can fail to reflect actual model behaviors on different data groups of a dataset regardless of how robust the metric is. This concept was first presented in general domain-independent (non-personalization) machine-learning studies such as the ones conducted by McMahan et al. and Chung et al. McMahan et al. (2013); Chung et al. (2018, 2019a,b) in a proposal of an automatic data slicing mechanism for evaluating subset levels of the data. The results show that it is crucial to track the performance on a more granular level to understand better the effectiveness of a particular model given a predefined set of prediction objectives. Ultimately, this aids us in better assessing the effectiveness of an algorithm on all parts of the data available for training and testing.

The concept of data slicing to uncover hidden performance issues has proven very effective Chung et al. (2019a). Such a mechanism is not yet utilized in the recommender system domain, albeit briefly touched on in the following studies Jurdi et al. (2021); Macdonald (2021); Ovaisi et al. (2022). The closest to the data slicing concept in recommenders is the research of Ovaisi et al. Ovaisi et al. (2022) since their evaluation toolkit supports sub-population evaluation (e.g., gender-based). However, it is not holistic and lacks the definition of an evaluation strategy. This sub-population evaluation scenario had been inspired by the idea of fairness in ranking evaluation discussed in the research of Singh et al. Singh and Joachims (2018). In our work, we focus on adapting a tailored and modular slicing evaluation framework, called *group validation*, to evaluate the performance of recommender models on a granular level. Our proposed mechanism will adapt the data slicing technique Chung et al. (2019a) and work through clustering datasets into groups and then identifying which of those groups a given model performs the worst using a systematic evaluation strategy. Additionally, the group validation framework allows the possibility to track a recommender's model performance across small data clusters and permits having practical applications such as:

- Enhancing automated decisions for model evolution.
- Detecting noise/fraud, such as malicious and natural noise.
- Creating a dynamic and hybrid model structure in a live environment.

The prime contribution of this chapter is implementing a modular evaluation framework for a distinct model validation process in recommender systems. Our applied method in this framework is based on data clustering[1] followed by experimental tests to identify weaknesses in model performance on certain data groups. As will be proven throughout this study, such weaknesses can usually be hidden from metric results that report overall system performances. This research extends the previous works on evaluation and noise management Jurdi et al. (2021); Macdonald (2021); Chung et al. (2019a) and adapts the slicing theory in the assessment from Chung et al. (2019a,b) to the recommender systems domain. Additionally, based on the theoretical analysis and the experiments conducted to position the framework against the current evaluation process, the following main points can be concluded:

- Clusters can be more sensitive to certain types of feedback, allowing the framework to detect adverse effects, something not possible solely with generic evaluation.
- Certain user feedback can evolve in a way that negatively affects other users. Cluster evaluation like that in the group validation framework can help localize this effect.

In the upcoming section, we present state-of-the-art works and position our work within this context, shedding light on the latest evaluation approaches in the field. Section 3

[1] The "groups" in the group validation process are mainly clusters. Both words might appear interchangeably throughout the text.

introduces our group validation framework with a comprehensive description of the proposed method, which comprises a clustering technique and a cluster assessment process. Afterward, in Section 4, we present an applied investigation of the group validation mechanism and include our experimental results and analysis. In the 5th Section, we offer some possible applications of the group validation approach and conclude the work in Section 6.

7.2/ Background and Related Work

In this Section, we cover all the studies we encountered that correlate with the proposed concept of group validation in recommender systems. The analysis is mainly grouped into three main pillars: The general slice-based evaluation in machine learning, a more related notion termed the Simpson's Paradox, and finally, a short discussion around the evaluation benchmarking in recommenders to validate whether they discuss evaluation on different parts of data or not.

7.2.1/ Slice-based Evaluation in Machine Learning

A series of studies focusing on data slicing while covering clustering methods for model evaluation was introduced by Chung et al. Chung et al. (2018, 2019b,a). This series of work is the closest to the group-based method proposed in this research as both leverage the concept of evaluating performance on smaller data pieces instead of a complete dataset evaluation. However, there are several core differences, as we will describe in what follows. The authors' final method, presented in Chung et al. (2019a) and called *SliceFinder*, is not developed in a way that would work for the recommender systems domain but rather for general classification problems. The reason for that is the use of the mechanism of features in the dataset to create subsets that make sense to the user in the end; such features are not present in recommender and personalization datasets where we usually have limited information about the users in an interactive system. SliceFinder is an interactive framework for identifying problematic slices using statistical techniques, and the slice evaluation mechanism is based on a general classification loss function.

This classification loss function returns a performance score for a set of examples by comparing h's prediction $h(x^i_F)$ with the actual label y^i. The difference between the tests conducted to identify the least performing slices in this case and the method used for determining the critical groups with our mechanism is the core loss function and the evaluation metrics employed based on the recommender optimization and goal. In the group-based validation case, the loss function is connected to the evaluation metric that will eventually be used to measure the system's performance. Moreover, we define sev-

eral loss functions based on the recommender system's optimization. As for the slicing mechanism, the authors in Chung et al. (2019b) applied several methods to implement an automated data slicing technique, such as decision trees (non-overlapping) and lattice searching (overlapping), which also generated meaningful data chunks (also referred to as literals). Our group validation framework is quite different in this regard, as the datasets for recommender systems are different and include unique features and usually much less information than the features utilized in the study of SliceFinder. We are currently using a clustering approach based on the minimal features available in the type of datasets used in this study's experiments. If we applied SliceFinder, we would end up with many data point groups that do not make sense to the user (example of an easy-to-understand slice reported by SliceFinder: $country = DE \cap gender = Male$). Such features are usually lacking when dealing with recommender system datasets. However, in a future study, we will tackle this issue by generalizing the proposed architecture to cover implicit feedback data in most e-commerce applications.

In the proposal of a comprehensive and rigorous framework for reproducible recommender evaluation by Anelli et al. Anelli et al. (2021), the authors mentioned the idea of statistical tests on data groups in the third section of the study. Throughout the work, it was emphasized that there is a need to compute fine-grained (such as per-user or per-partition) results and retain them for each recommendation model. As a result, their framework, called Elliot, was designed for multi-recommender evaluation and handling fine-grained results. Elliot brings the opportunity to compute two statistical hypothesis tests, i.e., Wilcoxon and Paired t-test, activating a flag in the configuration file. The proposal did not mention the partitioning techniques used other than the idea that the partitions might be per user. Partitioning per user is possible in our proposed method, and the groups could be generated by centering the clusters around users, similar to the k-Nearest Neighbors (KNN) Cover and Hart (1967) fashion. For instance, every user will form a distinct group. KNN is a non-parametric supervised learning method used for classification and regression where the input consists of the k closest training examples in a data set. In our proposal, the fundamental idea of the introduced framework is to report groups that might make more sense to track the recommender's performance. That way, the evaluation process allows one to check the worst-performing groups and analyze the reason behind the results.

7.2.2/ Subset Scanning

Subset Scanning is a highly efficient and accurate event and pattern detection framework that can operate on both spatial and non-spatial datasets Neill (2017); Gupta et al. (2021). The approach optimizes a score function (such as likelihood ratio statistic) over subsets of the data, making it a flexible and scalable solution that can adapt to vari-

ous real-world constraints (e.g., spatial adjacency, irregular shapes). By evaluating the score function over subsets, Subset Scanning can identify events and patterns quickly, significantly reducing the search space and making it more computationally efficient than traditional methods. Moreover, Subset Scanning builds upon spatial and space-time scan statistics, providing accurate results even when the valid spatial region of interest doesn't align perfectly with predefined search regions Neill (2012). Subset Scanning is a powerful and practical tool for researchers across various domains, offering a valuable approach for detecting events and patterns with speed and accuracy.

Subset scanning and the slice-based evaluation introduced in the previous section are two close techniques used in data analysis, each with a unique purpose and application. While subset scanning is primarily used for event detection and pattern identification, slice-based evaluation is commonly employed in machine learning for model development and validation. Subset scanning identifies subsets of data that exhibit significant deviations from the expected distribution, making it ideal for detecting clusters in epidemiology, identifying pollution hotspots in environmental monitoring, and finding influential subgroups in social network analysis. This method optimizes a score function over subsets of data and focuses on identifying regions of interest efficiently, making it scalable and adaptable to irregularly shaped clusters.

On the other hand, slice-based evaluation involves examining model behavior across specific dimensions or "slices" of the data, allowing us to assess model behavior across different groups or specific scenarios. This enhances model interpretability by revealing how it performs across various subgroups, such as understanding if a classifier works equally well for all demographic groups. Slicing data is also scalable for model evaluation, especially when assessing performance across multiple dimensions.

While both methods involve examining subsets of data, their goals and applications diverge. Subset scanning is specialized for event detection, whereas slice-based evaluation is a broader machine learning model assessment technique. Our proposal remains close in theory to subset scanning techniques and slice-based evaluation but remains specific to the recommender system features and applications context. As shown in the previous section, Group-based includes many elements from slice-based validation but can, in fact, be framed as a problem of subset scanning, especially in the following domains:

- Tackling data that exhibits significant deviations from an expected distribution
- Pattern detection in subsets

7.2.3/ The Simpson's Paradox in Recommender Systems

While not directly related to the evaluation of slices, Simpson's paradox in the offline evaluation of recommendation systems, which is covered by Jadidinejad et al. Macdon-

ald (2021), introduces a phenomenon that describes a very interesting phenomenon on granular evaluation. The research in Macdonald (2021) shows that the typical offline evaluation of recommender systems suffers from the phenomenon termed *Simpson's paradox*. Simpson's paradox is when a significant trend appears in several different subpopulations of observational data but disappears or is even reversed when these subpopulations are combined.

Although the definition of Simpson's paradox might theoretically be linked to our proposal, the authors in the study Macdonald (2021) conducted different experiments. They proposed an approach that tackles a marginally different issue while utilizing recommender systems datasets. The experiments are based on "stratified sampling" and reveal that a tiny minority of items that are frequently exposed by a deployed system (such as the system that helped generate the open source datasets, like Movielens Harper and Konstan (2015), that are used in most of the studies on recommender systems) plays a confounding factor in the offline evaluation of recommendation systems. So, the study investigates the issue of an initial recommender system, called a confounder, that influences the rating elicitation process of the users. This concept is the main difference between the study in Simpson's paradox and the group validation mechanism we are proposing, where the latter only tackles the issue with the performance of the recommender on smaller groups of users.

7.2.4/ Evaluation Benchmarking

A missing idea in the benchmarking proposals is the differentiation between different recommender algorithms and their general goal inside a particular application. In our proposal, the group-based mechanism evaluates a recommender based on the optimization of the model, which is the recommended way of approaching performance assessment Argyriou et al. (2020). As touched on in the introduction of our work, with the increase in the number of recommender algorithms proposed and different approaches to enhance them, a critical issue presents itself. A standardized approach to evaluating algorithmic enhancements as recommender proposals evolve does not exist. Some of the recent studies on evaluation focus on the benchmarking method to create a framework for the most accurate evaluation and comparison. In one of the proposals by Sun et al. Sun et al. (2020), the authors aimed to conduct rigorous (i.e., reproducible and fair) evaluation for implicit-feedback-based top-N recommendation algorithms. They reviewed several recent proposals and analyzed the different approaches for evaluating recommender systems. As expected, several inconsistencies were in what was utilized for evaluation, leading to inconsistent results when judging if a new proposal is better than its predecessor. Accordingly, the authors created benchmarks with standardized procedures and provided the performance of seven well-tuned models across six metrics on six widely used datasets.

In a similar and more recent study, Ovaisi et al. Ovaisi et al. (2022) proposed a toolkit for evaluation to assess recommender models' robustness. In this research, the authors mentioned the necessity for evaluating the model on different data slices, such as gender subgroups. In one particular example of the study, a sub-population of the test set consisted of users who were grouped by gender. The system performed much worse for females than males across all the models used in the experiment. This is proof of the importance of studying the performance of data groups versus evaluating the whole data altogether, where the negative performance tends to average out. Unfortunately, most recommenders' data lacks essential features like user information. That is why forming meaningful slices, such as those proposed in Chung et al. (2019b,a), work in theory but can be extremely difficult to apply to the standard datasets for interactive systems. Our method overcomes this for the first most common datasets in such systems: the rating-based datasets.

In general, most of the benchmarking toolkit proposals do not include studies about the quality of the data used for the recommender application nor provide detailed coverage of the concept related to analyzing the performance of smaller groups. The group validation method proposed in this study fills this gap and provides a new layer of evaluation that better tracks the performance of models across different groups.

7.3/ Group Validation in Recommender Systems

As previously indicated, data slicing for evaluation was formulated in the ML community McMahan et al. (2013) and further adapted by Chung et al. Chung et al. (2019a) to design an automated mechanism for slice identification on a general binary classification use-case. In our proposal, we re-formulate and re-structure this automatic slicing mechanism to develop an effective cluster-based evaluation process in a personalized recommender system setting. The goal is to create a tool capable of identifying performance issues on smaller clusters in the data, especially in cases where the standard evaluation methods cannot recognize such performance drops. In the following sections, we present the methodology behind our proposed framework and the adaptations we applied to the slicing evaluation study done in Chung et al. (2019a).

7.3.1/ Data Slicing and Evaluation in Machine Learning

Following the mechanism of slicing-based evaluation in Chung et al. (2019a), consider a binary classification model h, a general training dataset D with some available features F, and a value for every feature v (which could be numerical or non-numerical). A slice S is a subset of the data records in D and could be expressed in the following manner:

$\bigcap_j(F_j\ op\ v_j)$ where op could be any comparison operator and F_j a certain distinct value. One example of a random slice S could be: $(country = DE)\ \bigcap\ (gender = Female)$. The rationale behind the data slicing evaluation method in Chung et al. (2018, 2019b,a) is a trained Decision Tree (DT) coupled with a loss function that acts as the basis for which a slice's evaluation performance is measured. As the use-case implemented in the test only included a binary classification problem, the loss function used was the general logarithmic loss equation defined as:

$$\psi(h(x_F^i), y^i) = -\frac{1}{n} \sum_{(x_F^i, y^i) \in S} [y^i\ ln\, h(x_F^i) + (1 - y^i)\, ln\, (x_F^i)] \tag{7.1}$$

Where $h(x_F^i)$ is the model's prediction, y^i is the true label, and n is the total number of data points in S. The authors automated the slice identification process by setting up the DT model that utilizes a breadth-first traversal. This model travels down through the data starting at the root slice, which comprises the whole dataset and evaluates every possible easy-to-understand slice, i.e., a slice with the maximum number of feature combinations after which it could not be easily understandable Chung et al. (2019a). Ultimately, the model identifies a list of top-*k* slices where a test model performs the worst by evaluating every slice as a stand-alone dataset using Equation 7.1. A slice is considered critical and should be reported as part of the top-*k* when the following primary condition is satisfied:

$$S_{critical} : \psi(S) - \psi(S') > 0\ \ni\ C_{critical} = True \tag{7.2}$$

Where S' corresponds to the rest of the examples in the dataset D and can be calculated as $S' = D - S$. Afterward, the resulting slices of the first part of Equation 7.2 will have to satisfy a supplemental two-aspect condition, which we call in our work here $C_{critical}$, to determine if S indeed has a significantly higher loss than S'. The first condition is Welch's t-test Wikipedia contributors (2022b), which measures the existence of an effect by determining if the difference in loss is statistically significant. The second aspect is the effect size Wikipedia contributors (2022a), which complements the statistical significance and measures the magnitude of the effect. The combined usage of those two tests is adopted from the study of the effect size Sullivan and Feinn (2012). The two-aspect condition equations are summarized as follows:

$$C_{critical} = \begin{cases} \text{t-test} & H_o : \psi(S) \leqslant \psi(S') \\ & H_a : \psi(S) > \psi(S') \\ \text{effect-test} & \phi = \sqrt{2} \times \frac{\psi(S) - \psi(S')}{\sqrt{\sigma_S^2 + \sigma_{S'}^2}}\ ;\ \phi \geqslant 0.8 \end{cases} \tag{7.3}$$

Where σ_S^2 and $\sigma_{S'}^2$ are the variances of the individual example losses in S and S', respec-

tively. Equation 7.3 would result in $True$ for a given slice S only if both tests succeed, i.e., the Welch's t-test leads to the alternative hypothesis H_a passing, and the significance test equates to a large significance value on Cohen's scale Cohen (2013) - which is typically a value greater than or equal to 0.8.

7.3.2/ Group Validation in Recommenders

To adapt the concept of slicing-based evaluation to recommenders in our group validation framework, we examine several aspects of the automated data slicing mechanism. First, in the recommender domain, the datasets are of different shapes and forms compared to data from other ML problems that tend to be more general. Ordinarily, recommender datasets are naturally sparse and lack rich features found in datasets like the one used for the experiments of the data slicing mechanism of Chung et al. (2019a). Typically, a recommender system model, M, produces a list of ranked item suggestions $\vec{i} = (i_1, i_2, ..., i_n)$ selected from the whole set of items not previously seen by the user in his profile I_u. The items are usually associated with aspects $A = \{a_1, a_2, ..., a_c\}$ which could be any categorical classification of items such as genres for a particular movie in a recommender movie dataset like Movielens Harper and Konstan (2015) $(a_{action}, a_{comedy}, a_{drama}, ...)$. Second, the process of rating elicitation is highly subjective Knijnenburg et al. (2012); Jurdi et al. (2021) in the recommender domain. Typically, such feedback is translated into a 5-point Likert scale: $r_{u,i} \in \{1, 2, ..., 5\}$, where $r_{u,i}$ represents the feedback user u gives on item i. Although implicit feedback is becoming very popular lately, many systems are still utilizing the rating-based data format as evidenced by recent publications on significant issues like noise and attacks Jurdi et al. (2021), personalization enhancement Logesh et al. (2020), model evaluation and effectiveness Parapar and Radlinski (2021); Macdonald (2021), and model variations Ahuja et al. (2019). A field experiment by Zhao et al. Zhao et al. (2018) shows that blending explicit and implicit user feedback through an online learning algorithm can benefit user engagement and mitigate the increased browsing effort cost. In other words, this means that the browsing experience is generally improved, and there is much less effort on the user's side to reach the intended content.

7.3.2.1/ Data slicing

Clustering has been implemented as a baseline mechanism in the binary classification use-case experiment in the study of automated data slicing for evaluation Chung et al. (2019a). The authors argue that clustering would not be an optimal approach in the binary classification problem (which could also be a regression problem) because the clusters would be complex to interpret unless a manual investigation is conducted. Further, they add that the user has to specify the number of clusters beforehand, presenting a different

problem affecting the ability to automate the process.

However, things are different in a personalized recommender context where we will utilize a clustering method as the primary step in the group validation framework. First, if we apply the same slicing method of Chung et al. (2019a) to recommender datasets, the slices would not make sense because the features of both datasets are different, as indicated at the beginning of this section. Recommender datasets are typically sparse; other vital features like user biodata and specific extended information about items could be nonexistent. It is very challenging to effortlessly form a slice like this: $(country = DE) \bigcap (gender = Female)$ from a recommender dataset when we do not have enough data on such features. Additionally, clustering in recommender systems is very effective due to the personalized nature of the data and the way feedback is collected and used as the ground truth in training. It has been heavily leveraged to address several issues in models, such as balancing diversity, consistency, and reliability of ranked recommendations, leveling the data sparsity of user-preference matrices, and accounting for changes in user preferences over time Aggarwal et al. (2016); Beregovskaya and Koroteev (2021); Logesh et al. (2020).

As a result, in the context of group validation in recommenders, slices will be referred to as *groups* where G, the counterpart of S, now represents a subset of the recommender dataset D. The clustering technique we will use to form those dataset groups is the unsupervised learning algorithm k-means Sculley (2010); Arthur and Vassilvitskii (2006). The k-means algorithm commonly works by grouping data together in n clusters with equal variance. It does this by minimizing a parameter called the inertia or within-cluster sum-of-squares. It's essential to specify the number of clusters required for this algorithm. K-means is a popular algorithm because it performs well with many samples. However, since the group validation framework is modular, any grouping could be used in this phase, such as KNN with a Pearson Correlation Coefficient (PCC) method Mansur et al. (2017) to form groups of similar users, a simple yet effective CF approach. For the k-means method utilized in our experiments, we use the aspects A of the dataset to form a matrix with all functional aspects as features in an n-dimensional space where n is the total number of available features in a chosen dataset. Recall that A is the set of item genres used in our dataset experiments: $A = \{a_1, a_2, ..., a_c\}$.

7.3.2.2/ GROUP VALIDATION

To validate the resulting groups and determine which performs worst, we follow the general methodology of Chung et al. (2019a) regarding statistical significance assessment on the group level. The first step is to replace Equation 7.1 with a suitable evaluation metric relevant to the recommender context. Since the group-validation framework is modular,

we implement several ranking-based metrics: $nDCG$, $Precision$, $Recall$, and $\alpha\beta$-$nDCG$. The $\alpha\beta$-$nDCG$ is a unified assessment metric for measuring accuracy and diversity proposed by Parpar et al. Parapar and Radlinski (2021). It was recently adapted from an information retrieval evaluation metric called $\alpha{-}ndcg$ Clarke et al. (2008); Vargas (2014, 2015) to fit into the recommender ecosystem (just like the adaptation of our group validation proposal). The metric addresses the dilemma of having to choose what types of evaluation metrics (primitive accuracy, classical ranking Valcarce et al. (2018), diversity and novelty for highly related properties Castells et al. (2022); Chen et al. (2019); Badran et al. (2019a)) to optimize on, an issue initially presented by Herlocker et al. Herlocker et al. (2004). Adding to that, there are abundant measures to choose from and many optimization procedures for every chosen method, like simple refinements in this work to one CF approach Al Jurdi et al. (2019) (an information filtering process using techniques involving collaboration among multiple data forms). Further, this new metric satisfied all the essential axioms of modern evaluation Amigó et al. (2018), proving to be a significant upgrade from its predecessor α-$nDCG$ Vargas (2014) in the search field. Through both parameters, α and β, $\alpha\beta$-$nDCG$ is good at detecting non-optimal item order, aspect distribution and ranking, and topical redundancy accumulation. Additionally, it has been proven in the experiments conducted on the Movielens 20M dataset Harper and Konstan (2015) to behave very well in terms of discriminative power and robustness to incompleteness. A very brief description of the metric formulation is provided in the following paragraph, where the full details can be reviewed in Parapar and Radlinski (2021) or the source code Al Jurdi (2022). As the authors validated the effectiveness on the same datasets we experimented with, we kept the same tuning achieved for α and β.

As seen in the above equation, there are two new adapted parameters, α and β. The α parameter accounts for the possibility of the user being wrong in judgment. Alternatively, The β factor defines the confidence in a user's judgment value, represented by the authors as a smoothing factor accounting for user rating uncertainty. It also exhibits a secondary role that models the user's eagerness to look at items lower in the ranking (higher β implies more relevant items match the user's interests). First, the probability of an item i contributing to satisfying the user's interest in an aspect a_ϕ is defined as:

$$P(a_\phi|u,i) = \begin{cases} 0 & a_\phi \notin i \\ \alpha(u,i) & \nexists r_{u,i} \; and \; a_\phi \in i \\ \beta(u, r_{u,i}) & \exists r_{u,i} \; and \; a_\phi \in i \end{cases} \tag{7.4}$$

The authors then introduce redundancy and novelty by estimating whether or not the user is interested in position k in more items capturing a given aspect after having been shown earlier items in ranking $S = i[0, ..., k-1]$. This is defined as the gain value at position k

$G[k]$:

$$P(a_\phi|S) = P(a_\phi|u)\Pi_{i\in S}(1 - P(a_\phi|u,i)) \tag{7.5}$$

The cumulative gain computation can be done by the following equation:

$$CG[k] = \sum_{j=1}^{k} G[j] \tag{7.6}$$

Following the original $nDCG$ equation that applies a discount factor to penalize documents lower in ranking, the authors then define:

$$DCG[k] = \sum_{j=1}^{k} G[j]/log_2(1+j) \tag{7.7}$$

Returning to our adaptation and the usage of the evaluation metric $\alpha\beta$-$nDCG$, Equation 7.1 will now be adapted differently within the group validation framework and will correspond to the following rank-based metrics $nDCG$ and $\alpha\beta$-$nDCG$:

$$\psi@k = \frac{DCG[k]}{IDCG[k]} \tag{7.8}$$

After that, to identify the groups that a recommender model performs poorly on, we use the mathematical model of Equation 7.2 with some slight variations to include the new metric of Equation 7.8:

$$G_{critical} : \psi@k(G) - \psi@k(G') < 0 \ \ni \ C'_{critical} = True \tag{7.9}$$

Finally, we apply the same tests as in 7.2 to identify a critical cluster in a recommender dataset as part of the group validation process:

$$C'_{critical} = \begin{cases} \text{t-test} & H_o : \psi(G) \leqslant \psi(G') \\ & H_a : \psi(G) > \psi(G') \\ \text{effect-test} & \phi = \sqrt{2} \times \frac{\psi(G)-\psi(G')}{\sqrt{\sigma^2_G + \sigma^2_{G'}}} \ ; \ \phi \geqslant 0.8 \end{cases} \tag{7.10}$$

7.3.2.3/ GROUP WEIGHTS - A THEORETICAL MODEL

In a real-world scenario, the group validation framework procedure could also work if weights are to be assigned to groups. Specifically, certain groups in a system might be more crucial for performance monitoring due to a specific financial aspect or other significant correlation to certain outlooks. We can expand on the above system (Equations 7.9 and 7.10) and define a threshold beyond which the recommender would be performing poorly on important weight-correlated data groups. This can be represented in the

following manner:

$$\frac{\sum_{g=1}^{CG} \psi(C_g) \times w_g}{\sum_{g=1}^{CG} \psi(C_{g'}) \times w_g} \leq \lambda = 0.5 \tag{7.11}$$

In this model, C_g represents one critical group out of the total identified critical groups CG, while w_g is a special weight assigned to group g. $C_{g'}$ corresponds to the group's equivalent metric value similar to $\psi@k(G')$ in Equation 7.9. The equation's threshold λ, currently set to 0.5 as an example, can be further tweaked depending on the system's defined group weights and the strictness of the validation framework on important groups.

Table 7.1: Datasets used in the experiments of the group validation mechanism.

Dataset	Total users	Total items	Total ratings	Sparsity
ml-latest-small	610	9,742	100,836	0.983
ml-1m	6,040	3,900	1,000,209	0.957
personality	1,820	35,196	1,028,751	0.983
ml-25m	283,228	58,098	27,753,444	0.998
ml-25m*	10,000	15,316	250,000	0.995

7.4/ Group Validation Experimentation Methodology

In this section, we discuss the application methodology of the group validation framework and explain the experiment setup and procedures. We first cover the algorithms and the data used to conduct the different experiments and their primary objectives. After that, we present three types of data perturbations used to simulate potentially harmful behaviors in the dataset, and finally, we cover the mode of operation of the group validation framework in correlation with the experiments designed to showcase the validation process.

7.4.1/ Data and Algorithms

Generally, recommender systems can be categorized into two broad classes: algorithms optimized for accurate predictions and others optimized for better rankings. Since ranking is more suitable for most use-cases of deployed recommender systems compared to the less popular prediction-focused algorithms Herlocker et al. (2004), and the framework we introduced has ranking-based metrics for the group validation part (recall Equation 7.9), we mainly focus on models that are optimized for generating optimal user rankings.

The recommenders chosen for showcasing group validation are Bayesian Personalized Ranking (BPR) Rendle et al. (2012) and Singular Value Decomposition (SVD) Simon

Funk (2006). BPR uses item pairs i, j and optimizes for the correct ranking given the preference of a user u by maximizing the posterior probability. For the model's parameters, we use the generic tuning done on the same datasets here Argyriou et al. (2020) with minimal optimization and set k to 400 (dimension of the latent space), $max\text{-}iter$ to 100 (the number of iterations of the SGD procedure), $learning\text{-}rate$ to 0.01 (step size α in the gradient update rules), and $lambda\text{-}reg$ to 0.001, which controls the L2-Regularization λ in the objective function. The final objective function of the maximum posterior estimator (a probabilistic framework for solving the problem of density estimation.) is $J = \sum (u, i, j) \in D_s ln\sigma(\hat{x}_{uij}) - \lambda_\theta \|\theta\|^2$. Unlike BPR, the SVD algorithms model the user and item biases from users and items and use Stochastic Gradient Descent (SGD) as an optimization technique.

The famous SVD algorithm, as popularized by Simon Funk Simon Funk (2006) during the Netflix Prize. When baselines are not used, this is equivalent to Probabilistic Matrix Factorization Mnih and Salakhutdinov (2007); Hug (2020). All implementation details can be found in this study Hug (2020)(SVD), however, we give a quick review here for clarity:

The prediction $\hat{r}_{u,i}$ is set as:

$$\hat{r}_{u,i} = \mu + b_u + b_i + q_i^T p_u \tag{7.12}$$

If the user u is unknown, then the bias b_u and the factors p_u are assumed to be zero. The same applies to item i with b_i and q_i. The unknowns are estimated with a regularized squared error:

$$\sum_{r_{ui} \in R_{train}} (r_{ui} - \hat{r}_{ui})^2 + \lambda(b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2) \tag{7.13}$$

The minimization is then performed by an SGD:

$$\begin{aligned} b_u &\leftarrow b_u + \gamma(e_{ui} - \lambda b_u) \\ b_i &\leftarrow b_i + \gamma(e_{ui} - \lambda b_i) \\ p_u &\leftarrow p_u + \gamma(e_{ui} \cdot q_i - \lambda p_u) \\ q_i &\leftarrow q_i + \gamma(e_{ui} \cdot p_u - \lambda q_i) \end{aligned} \tag{7.14}$$

The formula $e_{ui} = r_{ui} - \hat{r}_{ui}$ is used to calculate the difference between the actual rating and the predicted rating. To train the model, the steps are performed repeatedly over all the ratings in the training set for a specified number of epochs, denoted as n_{epochs}. At the beginning of the training process, baselines are set to zero. The user and item factors are randomly initialized using a normal distribution, with the mean and standard deviation of the normal distribution being adjustable through the $init\_mean$ and $init\_std\_dev$ parameters. We can also control the learning rate γ and the regularization term λ. We used the default settings of 0.005 and 0.02, respectively.

As for the datasets used in our study, the experiments are conducted on several famous

rating-based (explicit feedback) datasets of different sizes, which are summarized in Table 7.1: MovieLens Harper and Konstan (2015) and personality Nguyen et al. (2018) where smaller counterpart versions used are marked with a (*). The data attributes used from those datasets are the user ratings for training the models and the ratings with the item attributes (such as genres) to perform the grouping.

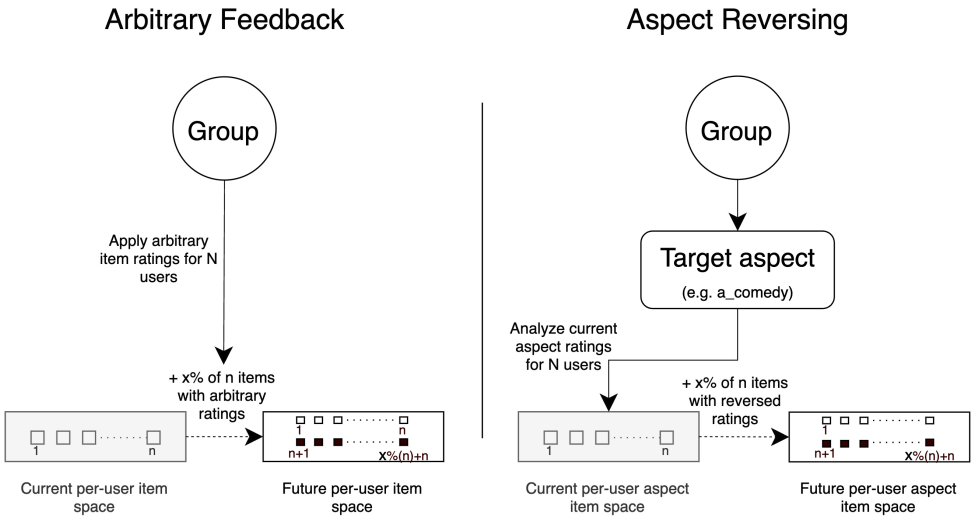

Figure 7.1: This schematic shows the two main types of synthetic data perturbation utilized in the experimentation of the group validation framework.

7.4.2/ Data Perturbations

In this section, we introduce the data modification mechanism and schemes used in the experiments of the group validation framework. First, the data alteration process is described, followed by the types of perturbations applied and how they relate to existing methods covered in the introduction of this work.

7.4.2.1/ Mechanism

Given that the introduced framework tracks the performance of smaller data groups and shows possible clusters of interest that might suffer performance degradation, we try to simulate a case of old and future dataset versions and apply both the normal and the group validation methods to both versions in an attempt to compare and showcase both performances side by side. We establish the initial dataset version as the ”old state” and a future version with newer data introduced as a ”future state” where a recommender model will be refreshed. This is a customary process in live settings where recommender systems must be periodically re-trained to refresh the model weights with new user observations. There are even new proposed efficient ways to optimize this process regard-

ing memory and speed and use advanced ways to transfer knowledge from previously trained-on data. This aids in using much bigger dataset sizes and helps avoid shrinking older data in future train runs to increase performance, as this would cause overfitting and historical forgetting issues Zhang et al. (2020).

Additional data perturbations applied on the future dataset version are introduced on targeted dataset groups, for example, on the 20^{th} out of the 50 available groups (after the clustering process has been applied). There is no particular reasoning behind a target group selection in the scope of this study; however, it is crucial to note that the general idea of the results was consistent with different chosen groups across several runs Al Jurdi (2022). This point is further affirmed in the other experiment parts in the following sections as we vary the target groups for the synthetic data introduction and lower potential biases by having a small 2% standard data progression, i.e., the data added to a system as users interact with it over time.

7.4.2.2/ Types and Parameters

Data perturbations represent the feedback variations that will be applied to selected user profiles to showcase the possibilities of the group validation framework and position it as an additional layer beside the general evaluation metrics. The two main types of perturbations utilized are arbitrary feedback and aspect reversing, followed by one baseline type, which we refer to as standard feedback. Figure 7.1 represents the methods of applying arbitrary feedback and aspect-reversing. The general idea behind every type is summarized in the following points:

- Standard: A simple process of holding out a few portions of user data in the initial dataset version and re-applying them as a form of ”future” feedback.
- Arbitrary: On the first side, Schematic 7.1 depicts how arbitrary feedback corresponds to the mechanism of *randomly/arbitrarily* applying new future ratings for N users, i.e., not following a certain profile's historical patterns and rating inclinations, which could be framed as an arbitrary form of feedback.
- Aspect reversing: As opposed to the arbitrary scheme, aspect reversing (right side in the diagram), a target aspect is initially selected (such as a_{comedy} in the MovieLens dataset cases), and the new future ratings for N users will be introduced in a fashion where their least favorite and most favorite item aspects will be reversed. This would be possible after analyzing the overall per-user item aspect space.

As reviewed in the introduction and the state-of-the-art sections, such feedback patterns like the arbitrary and aspect reversing types exist in the real world and can maintain different forms. For example, they could appear as natural noise, a process described in one of the significant studies of natural noise ratings Toledo et al. (2015). This type inspired

the arbitrary feedback perturbations and is mainly due to errors in the rating elicitation that create erratic user review data not aligned with their general feedback profile, even when serendipity is accounted for with relaxed parameters Al Jurdi et al. (2018). This type of behavior is usually challenging to identify and manage since it does not have a specific pattern and consequently cannot be modeled Toledo et al. (2015); Jurdi et al. (2021). Another real-world form that inspired the aspect reversing is opt-out, sometimes referred to as obfuscation Brunton and Nissenbaum (2015); Al Jurdi et al. (2022). Users leverage this mechanism when an opt-out option is not provided in a system, meaning they are not provided with an option to remove their data from the system's databases and revert the consent for processing it in machine learning applications. This type is usually prevalent in masking profiles when users want to hide their identity, which could be revealed from registered interactions. Similar data perturbations, in the form of aspect reversing, were used for testing the effectiveness of the $\alpha\beta$-$nDCG$ evaluation metric in Parapar and Radlinski (2021).

Table 7.2: Group validation framework's mode of operation: data perturbation methods and percentages of the magnitudes applied to the datasets. The 2% of standard data progression is implemented on all users, while the arbitrary and aspect reversing methods are only applied on a portion of the specific target group G.

Feedback Method	Min Change	Max Change	Affected Users	Notes
standard*	2%	2%	-	Normal Progression
arbitrary	2%	25%	10/G	Arbitrary Progression
aspect reversing	2%	25%	10/G	Reversed Aspects

For the arbitrary ratings and the aspect reversing data schemes, we have to set a reasonable limit in the experiments that show how group validation works on evaluating smaller data clusters and how it compares with current evaluation methods. Adding significant percentages can cause unwanted biases that might overshadow the point of showing the mode of operation of validating groups. Generally, any data modification like the ones introduced above should yield worse recommender performances. As long as the percentage of the affected data is high enough, sensitive ranking metrics like the $\alpha\beta$-$nDCG$ (or even $nDCG$ and $prediction$ in some cases Parapar and Radlinski (2021)) should be able to spot a degradation in performance. With group validation, the framework studies smaller data clusters, so any minor and potentially malicious change could negatively affect the group level (recall Section 7.3). Since group validation is intended to localize the effect of such small-scale variations, especially their potential malicious effect on different user groups, we select the behavioral synthetic data portions to be smaller in scale than those used to test new metrics like the $\alpha\beta$-$nDCG$ in Parapar and Radlinski (2021). The magnitudes applied are summarized in Table 7.2. The first four magnitudes range between 2% and 15% of added feedback of the total ratings in a target cluster. This is

quite similar to what was done in Parapar and Radlinski (2021) for the metric tests and also in Jurdi et al. (2021), where a similar range was used to introduce data anomalies that exhibit the same characteristics as the arbitrary rating scheme of our experiment. Therefore, we try to utilize a similar structure. In our process, the percentage change is less since the percentage of added feedback is based on the group size (affecting around 10 users per group - 10/G), a subset of the dataset.

Algorithm 1: Group Validation Algorithm Overview

Data: recommender model M, original dataset D, a future version of the dataset D', user feedback R, item aspects A, weight thresh. λ

Result: critical groups $G_{critical}$, performance per group P_G, system performance results P_D

The below process is repeated twice, before and after the change in data (D and D'), as described in Section 7.4

initialization
$G_{critical}$, $G_{non-critical}$ = [ ], [ ]
λ = 0.5

clusters = *apply_clustering*(D, R, A) `// all groups and their respective users`
train_data, test_data = test_split(D)
predictions = fit_score(M, train_data)

for G in *clusters* **do**
  $G' = D - G$ `// get the equivalent of G`
  $condition\_1 = metric(G) - metric(G')$ `// metric is` $ndcg@k$ `or` $\alpha\beta$-$ndcg$
  while condition_1 **do**
    group_status = *aspects_test*(g, g') `// apply the tests of Equation 7.10`
    if group_status is True **then**
      $G_{critical}$.append(g) `// The identified critical groups. The algorithm is performing poorly here`
    else
      $G_{non-critical}$.append(g)
    end
  end
end
P_D = *evaluate*(predictions, test_data)
P_G = *group_evaluate*$(G_{critical}, G_{non-critical}, \lambda)$

7.4.3/ GROUP VALIDATION EXPERIMENT PROCESS

In this section, we cover the mode of operation of the group validation process and go over the experiments presented in the following section. Algorithm 1 Al Jurdi (2022) represents the high-level code structure of the proposal's procedure. First, the primary process of the algorithm repeats twice as we target one typical real-world scenario where the data is

refreshed with a newer version for offline re-training and batch inferencing (recall Section 7.4.2.1). Therefore, in this case, the dataset's future state change would be the addition of the data perturbation schemes alongside the small percentage of standard held-out data. The small held-out data corresponds to the standard rating method introduced in the previous section. It aids in establishing a baseline and better representing a real-world scenario where newer data feedback usually occurs more visibly across a more significant portion of the dataset. Following this method, we can measure the effect different data variations (Section 7.4.2) might have on the system using standard evaluation mechanisms while on the clusters using the group validation process.

The other parts of the pseudocode in Algorithm 1 mainly follow the theoretical model of Section 7.4. The initial step is establishing the dataset groups using the k-means method and performing a typical train-test split on a target dataset. The number of clusters for each dataset has been determined in a standard way using the elbow method, resulting in around 50 clusters for smaller datasets and 100 for the bigger ones. General evaluation metrics are obtained using typical evaluation with predicted outcomes and held-out test sets with the evaluation metrics Precision, Recall, nDCG, etc. At the same time, group validation commences with studying each group and applying the procedures explained in Section 7.3.2.2.

Recall that the primary aim of the group validation method is to identify which data clusters would be negatively affected by potentially harmful behavioral patterns (in the form of data perturbations) that are typically undetectable by standard evaluation methods. We conducted three experiments to position group validation in the evaluation procedures of recommenders. Initially, we would like to establish whether the data perturbations (representing the previously discussed feedback scenarios) affect only some data groups but not the whole system. This can help us validate the first point introduced in the introduction:

1. **(1):** Clusters can be more sensitive to certain types of feedback, allowing the framework to detect adverse effects, something not possible solely with generic evaluation.

The group validation process should be able to identify which groups are negatively impacted by changes in the data, rendering the negative effect more localized and the clusters sensitive to small, potentially malicious changes. Standard metric evaluation on the whole test set might be unaffected by those schemes. To verify this, we conduct two additional experiments to generalize the results further and place both the standard validation methods alongside the group validation process as we test the different data perturbation schemes. Potentially, this can help us affirm the following second main motivation of the work:

1. **(2):** Certain user feedback can evolve in a way that negatively affects other users. Cluster evaluation like that in the group validation framework can help localize this effect.

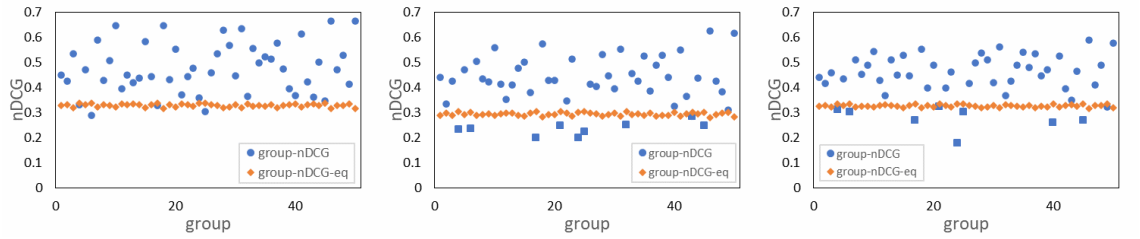

Figure 7.2: Group validation results using $nDCG$ on ml-latest-small with no data perturbations - just standard data (left), arbitrary data perturbations (center), and aspect reversing (right).

7.5/ Results and Analysis

In this section, we present a walk-through and further analysis of the experiments and findings of the group-based validation framework following the theoretical model and framework guidelines introduced in Section 7.3 and the experiment methodology presented in Section 7.4.

7.5.1/ Critical Groups due to Data Perturbations and the Simpson's Paradox

The first experiment is split into three parts, and for brevity, we present the results on one of the test datasets. In the first trial, we check the regular evaluation and group validation framework on standard data review future values (devoid of any perturbations) to establish a baseline result and verify whether or not we will encounter critical groups. This permits us to show the mode of operation of group validation against other typical evaluation methods when users provide standard data feedback consistent with their profile predilections without any potentially harmful data between the two dataset versions. The second and third tests will have the data perturbations in two forms: arbitrary feedback and profile swapping. As explained in the methodology, we have the standard data scheme of 2% regular data feedback in the three tests as a form of natural data progression between the current and future dataset versions. For the arbitrary and aspect reversing, in the last two tests of this experiment, we use a 10% rating feedback (of a target cluster's total item feedback) as a data perturbation magnitude on a random target cluster.

Figure 7.2 shows the group validation framework's outcome and mode of operation. The first test results are displayed in the left plot. The almost straight line represents the values of the core metric of the group validation of all the equivalent examples of every group G' and is denoted by $group\text{-}nDCG\text{-}eq$. In contrast, the scattered points in blue represent $n\text{-}DCG$ values of the groups of interest G and are labeled as $group\text{-}nDCG$. The plot shows a generic distribution of the groups in an acceptable zone of ranked item performance. This means that not even the first condition of the group validation model in Equation 7.9 was met. Only two groups slightly appear below their G' counterpart value, thus validating the first condition of Equation 7.9, but the second condition hasn't been met, and therefore they are still considered in the acceptable zone with a non-critical-status. The overall system metric registered an $nDCG$ value of approximately 0.338, coinciding with the $group\text{-}nDCG\text{-}eq$ values of each respective G'. We can imply that this establishes a stable state of the dataset with all groups having acceptable performance results in a case where none of the two primary perturbation schemes are applied.

In this experiment's second and third tests, displayed in the middle and the right plots of Figure 7.2, we introduce arbitrary rating and aspect-swapping data perturbations, respectively. As described, this is done on a target cluster to create a new future dataset version where the model will be trained and re-evaluated. Similar percentages are utilized: 2% standard progression for all clusters to simulate the normal data evolution at two points in time and a 10% data perturbation on another cluster number (which was group number 20 this time). It is clear from the plot how group validation identifies critical groups (presented as squares) following the theoretical model in Equations 7.9 and 7.10. Those groups are, therefore, negatively affected by the small-magnitude perturbations introduced in only one of the clusters. The results of the third test are not very different; the system metric maintained almost the same value, 0.336 versus 0.333, while several critical groups were reported. Combining the results in both cases of the groups results in the overall $nDCG$ system value. It will diminish the negative effect spotted on the groups shown in the plots - which the group validation framework translated into critical groups. This renders a Simpson's paradox scenario in effect (recall Section 7.2.3). For instance, a drop of 0.9% in $nDCG$ in a re-train/re-evaluation scenario such as this one evaluation will not be alarming if a model is regularly re-trained for offline batch-inferencing.

Recall that this experiment aims to prove that there will be negatively affected users in a dataset whenever there is undetectable and malicious behavioral data. The above tests show how group validation with $nDCG$ as the core evaluation metric of Equation 7.9 was able to identify critical groups with a 10% data perturbation magnitude. This validates the first goal (Goal 1 in Section 7.4.3) outlined in the introduction of this work. Some groups are sensitive to specific changes in the dataset, and the group evaluation process detected the effect.

Additionally, reiterating the connection with the concept of Simpson's paradox, we proved how the paradox explained the results obtained in our experiments where the group validation mechanism was able to spot and report smaller versions of the data (in the form of clusters/groups) where the model's performance is negatively affected. This negative performance will not be noticeable when we evaluate the system with legacy evaluation methods.

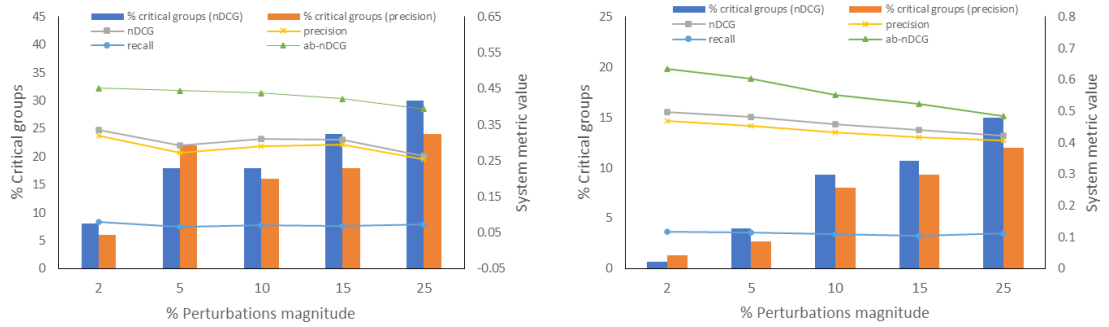

Figure 7.3: The percentage of critical groups alongside normal metric scores as data perturbation percentage (aspect reversing scheme) increases on ml-latest-small (left) and ml-1m (right). With increased perturbation magnitudes, modest metric effects are observed while critical groups increase, as spotted by the group validation process.

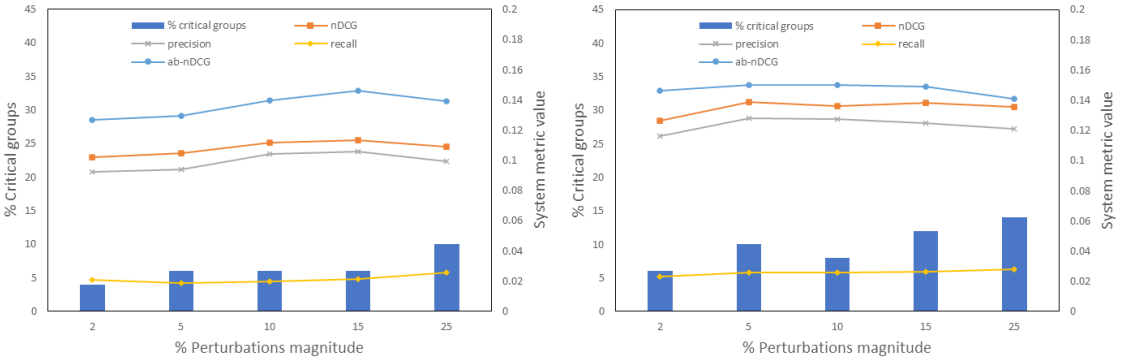

Figure 7.4: The percentage of critical groups alongside normal metric scores as aspect reversing intensity increases on ml-latest-small (left) and ml-1m (right) using the SVD algorithm.

7.5.2/ Critical Groups versus Normal Evaluation

In our second experiment, we study the effect of the chosen data perturbations on the group validation method alongside the normal model evaluation using the evaluation metrics $nDCG$, $precision$, $recall$, and $\alpha\beta$-$nDCG$. This second test expands on the previous results and tests the difference between measuring the system's performance and the group's performance as the data evolves. For this experiment, we select two datasets, the ml-latest-small, and the ml-1m, with the two algorithms, BPR and SVD. Data per-

turbation aspect reversing is used, and the method follows the same strategy as in the first experiment for the 2% normal data progression between the two dataset versions on which the tests will be applied. However, in this case, we vary the magnitude intensity and define five levels ranging from 2% to 25% of the group's total ratings. The target group for the data perturbation has also been randomly selected: 25th (out of the entire 50 groups) for the ml-latest-small and 70th (out of 150) for the ml-1m. This test aims to validate that the malicious data perturbations with various percentages will not significantly affect metric results contrary to the group validation method. Similarly, the group validation result is used for the evaluation results with $nDCG$ and $precision$ as the core evaluation metrics for the group validation process of Equation 7.9 in the first test, and $nDCG$ for the second and upcoming ones.

Figures 7.3 and 7.4 display the outcome of the metrics on the 5-magnitude data perturbations for the BPR and the SVD algorithms, respectively. We first notice that the $\alpha\beta$-$nDCG$ metric scored slightly higher in both tests than its counterpart $nDCG$. This indicates a somewhat better metric value on the rankings achieved by the BPR model with the likelihood and aspect weights (recall Section 7.3.2.2). As the intensity of perturbations increases from 2% to 25%, we notice a modest decrease in the ranking metrics, slightly sharper for the $\alpha\beta$-$nDCG$, especially in the ml-1m case. For ml-latest-small, until the 15% data perturbation magnitude, the decrease in $\alpha\beta$-$nDCG$ registered a mere 5%, while for nDCG and precision, the score is a 3% decrease only. On the other hand, the recall metric values maintained a consistent outcome of around 0.065 for ml-latest-small and 0.12 for ml-1m. Analyzing the group validation outcome, we notice that critical groups significantly increase in both cases with increased perturbation magnitudes. This indicates that even though our system metrics still show acceptable results, a significant number of affected users in the dataset are overshadowed by the effect reversal once the results are combined to generate the system metric. Even with the 15% data perturbations, we can see that ml-latest-small still registered around 18%-24% ($precision$ and $nDCG$ respectively) from the 50 total groups. The same appears for ml-1m but with a slightly lower intensity where 10% of the dataset groups are now in a critical state and will potentially experience degraded recommender ranking performance.

Figure 7.4 shows the same results but uses the SVD model as a recommender. The results appear to be not very different from the BPR model except for generally lower system metric values. The $\alpha\beta$-$nDCG$ still scores slightly higher for the same reason, while metrics exhibit a modest decrease in the ml-1m case. It is marginally different in the ml-latest-small, where we can see an increase in metrics performance as the perturbation magnitudes increased before a decrease on the last more-intensified level of 25%. Critical groups expanded from around 5% of the total groups to 10% for the highest perturbation magnitude applied. The group validation, however, still reported critical groups that increased with the increase of the perturbation magnitude, with results very close to

those achieved with the BPR model on the ml-1m dataset. The same conclusions can be drawn for this test: even though our system metrics still show acceptable results, a significant number of affected users in the dataset are overshadowed by the reversal of the effect once the results are combined to generate the system metric.

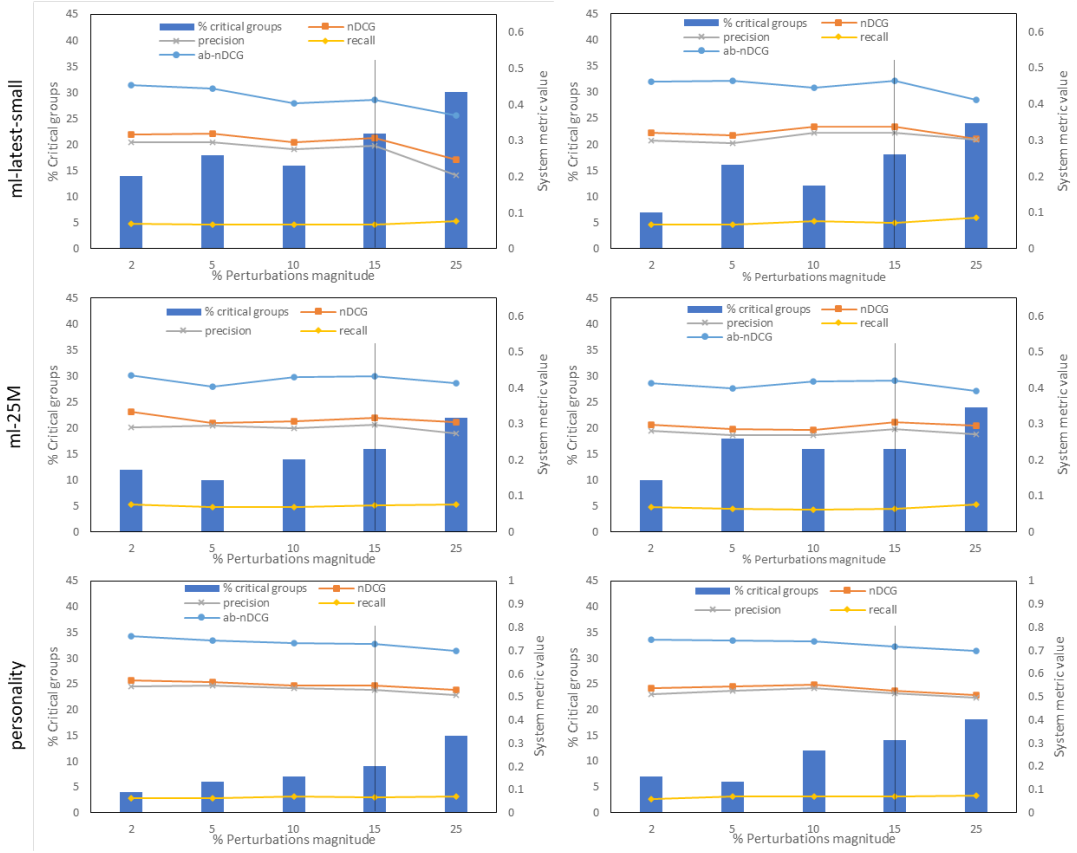

Figure 7.5: The effect of different data perturbation magnitudes on the percentage of critical group formation and metric scores. The schemes used are random (left column) and aspect reversing (right column) on the three datasets: ml-latest-small, ml-25M, and personality.

7.5.2.1/ Extended Analysis - Possible Group Validation Break-point

We extend the experiments and test the group validation method alongside the same metrics on different datasets to further validate the previous test's results and better represent the group validation versus the typical evaluation methods. Figure 7.5 depicts the outcome of this experiment on ml-latest-small, ml-25M, and personality (top to bottom). The outcome of the first two datasets is consistent with the previous results and theory. The critical groups increase as the magnitude of data perturbations increases from 2% to 25%. The random perturbations result in slightly lower percentages of critical groups, most likely due to the random variability of the feedback where some random items might be relevant to a user's profile. The aspect reversing would still be a more targeted data

change compared to the random feedback process and is guaranteed to result in a high level of ”non-relevance” for the targeted user Parapar and Radlinski (2021). Conversely, metric results showed humble decreases until the 15% magnitude of data perturbations, where the metrics start to indicate a slightly higher decline in performance. At this point, critical groups registered the highest number for all three datasets and reached around 30% for the first two. Personality is a richer dataset with many more attributes and reliable feedback. We can notice from the Figure that the performance is significantly higher for the metrics compared to the other datasets with the same BPR model and parameter tuning used in the second experiment. The critical group percentage is slightly lower, which is equally consistent with the metric results; however, group validation can still identify affected clusters. Analyzing the overall graph patterns, defining a breaking point at the 4^{th} level representing a 15% change (marked with a vertical line in the Figure's plots) of data perturbations would seem possible. At this stage, potentially malicious change of up to 20% of the target cluster's total feedback would not be reflected in the system's normal evaluation metrics (recall the data operations defined in Table 7.2).

Given the above-obtained results, we can see how the group validation tests helped us localize the adverse effects of potentially harmful behavioral feedback in the form of critical groups, thus affirming the motivation set in the second goal (Goal 2 in Section 7.4.3).

7.5.2.2/ General Conclusions - Group Validation Versus Normal Evaluation

The second section of the experiments better presented the mode of operation of the group validation process side by side with normal evaluation metrics in a scenario where data evolves from one state to the other in several datasets. The evolution is obtained with a slight uniform data progression combined with the aspect reversing data perturbation with an increased magnitude from 2% of rating data of a target cluster to 25% of its feedback data. The results show how particular user feedback can evolve in a way that negatively impacts users of specific clusters more than others, affirming the second motivation behind this proposal. Cluster evaluation like that in the group validation framework helps localize this effect and reports where the performance specifically degrades. In contrast, the normal system evaluation generally exhibits Simpson's paradox effect. Negative results are balanced out by other higher performance results that might be due to biases resulting from the newly added information. Additionally, the small perturbations that affect one group could degrade the performance of the target group and other groups in the dataset. This further affirms the second point of the motivation behind this method, where the feedback can evolve in a way that negatively impacts other users.

Applying the method's validation procedures to monitor different groups' performance

evolution of a dataset is not a replacement for the general evaluation procedure. Returning to the introduction of this work, we aim to have this mechanism run in parallel with the typical evaluation techniques as it can apply evaluation procedures on recommenders as the data evolves from one to the other from a different vantage point. It complements the general evaluation results and helps create a more robust filtering process. We further review this method's possible applications and extensions in Section 7.6.

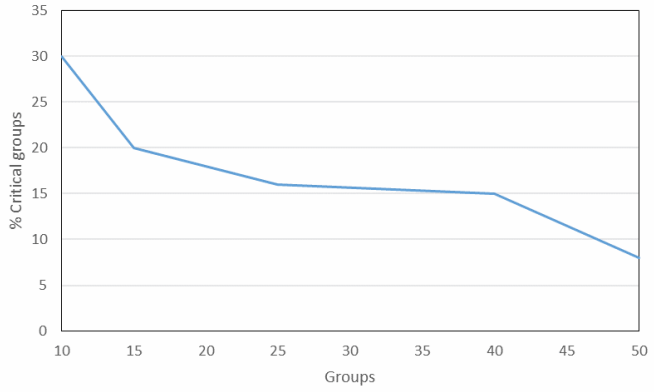

Figure 7.6: Effect of different group sizes on the percentage of the critical groups.

7.5.3/ Varying Group Sizes

In this final test, we analyze the dataset's effect on different group options. As mentioned earlier, the grouping method we apply is an example procedure that could be adapted to a different mechanism in a distinct setting depending on the features available, the metadata type, and the recommender's primary goal. However, we can still vary the number of groups with the k-means method we selected and study how different cluster values can affect the critical groups' outcome of the group validation method. The results are summarized in Figure 7.6, which shows the percentage of critical groups as we vary the total number of clusters on the ml-latest-small dataset in 9 unique runs. For this test, We utilized a constant amount of 15% random data perturbations in every run to maintain a coherent scenario across the runs. With a small number of groups (such as 10), we can notice a higher percentage of critical groups. This phenomenon implies that group validation can identify a negative effect in one of the dataset groups; however, since the group number might be small, the negative effect is not very well localized and points to a group containing a relatively significant number of users. When the number of groups increases, we can notice that the critical clusters start to decrease. We conclude from this that we can spot the negative impact of the data perturbations in a more localized cluster in the dataset. As the number of clusters decreases, we are approaching the case of the normal system metrics, i.e., evaluating the effect on the whole dataset. The correct amount of groups is subjective and should be solely based on the dataset attributes

and the recommender's goal. We provided a small theoretical adaptation of the group validation method if weights are to be assigned to groups in Section 7.3.2.3.

7.6/ Group Validation Limitations and Possible Applications

In this section, we highlight some of the limitations of our work and the possible applications enabled by the new group validation framework in the recommender system ecosystem.

7.6.1/ Limitations

The data clustering in our experiments was done based on the available item attributes of the datasets at hand. It would be interesting to investigate how the group-based evaluation approach would behave when the clustering is conducted based on a variation of different features, such as user behavior in the system. It is crucial to assess the behaviors using different cluster forms to generalize the approach to various scenarios where recommender systems are used. As different data types and features lead to the usage of different techniques in recommender systems Aggarwal et al. (2016); Roy and Dutta (2022), this can be extended to form different strategies of group formations based on rating types (e.g. implicit or explicit) and features.

7.6.2/ Applications

The different parts of this framework, such as the core metric used to identify the critical groups and the means of generating clusters, are interchangeable. This renders the foundation a little more flexible and open to further experimentation. Some of the potential applications that could be implemented on top of the group validation framework are listed below:

- Model Evolution and Fairness. With the recent vital importance of fair recommender systems Singh and Joachims (2018), it is crucial to report and analyze the performance of a specific group of users. Group validation spots a localized form of negative effects that could result from potentially harmful behavioral data. This can be an initial step in forming a metric better optimized to increase fairness in a system and monitor it across different model generations.
- Noise/Fraud Detection. The detection of fraud (which sometimes can be referred to as noise in a simpler form) involves identifying malicious behaviors that form spe-

cial patterns. Popular methods for their detection involve Graph-based anomaly detection (GBAD) Pourhabibi et al. (2020), primarily used to analyze connectivity patterns in communication networks and identify suspicious behaviors. The proposed group validation framework can be further tweaked to provide a new layer for anomaly detection in recommender systems. The granular evaluation outlook could be sophisticated enough to detect weaknesses in performance and spot malicious behavior that forms a unique connection between different groups. A similar concept has also been debated by Al Jurdi et al. Jurdi et al. (2021) in the study of natural noise, where it was shown that certain malicious patterns could affect parts of the system.

- Hybrid Model Decision System. Building on the evaluation method introduced in this work, the architecture could also be extended to include multiple models deployed in a production environment where each would be tuned for different groups based on the results from the group validation framework. Figure 7.7 shows a flow diagram of this hypothetical application. The critical clusters could be transferred to a grouping mechanism that arranges them based on the metrics portfolio similar to that employed in the experiments of Section 7.5. This potentially useful application aids in limiting degrading performance results in small groups.

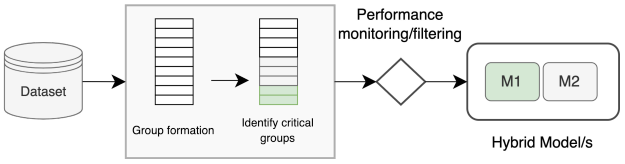

Figure 7.7: This schematic shows a hypothetical hybrid model setup of recommender systems deployed in the real world that leverages the group validation framework.

7.7/ CONCLUSION

Evaluating the effectiveness of recommendation systems and testing their performance as the underlying data evolves with time remains a remarkably tough challenge yet to be tackled. The current evaluation state of the art has reported some improvement in the benchmarking field, where researchers created several evaluation frameworks to standardize the assessment process. However, the recommender ecosystem has not covered the concept of reporting the performance of models on dataset subgroups and the potential of detecting specific kinds of potentially malicious data behaviors. This chapter described a new evaluation strategy for recommenders: *group validation framework* in recommenders. This method can be employed as an assessment tool to track the performance of a recommender on certain important clusters/groups of data. The results showed how recommenders performed differently in various groups as unique data

perturbations were introduced. The most fundamental aspect is that we could spot negatively affected sections of the dataset due to added synthetic data that was not visible with normal evaluation techniques. Group validation helped localize the errors in the data and provided a means to identify the effect of behavioral data changes in the system.

In the future, this proposed framework can be extended to cover other scenarios where dataset grouping could be effectively applied to several recommender dataset types (such as implicit feedback) and embed different recommender goals. Further, a data-oriented approach can be introduced to have a more intelligent grouping mechanism. It can automatically treat potential performance degradation in specific clusters and measure the effect on serendipity as a factor of such changes.

VI

The Impact of Weak Ties on Recommender Systems

In the final section of our thesis, we emphasize the significance of chance discoveries in recommender systems. We also incorporate the new evaluation knowledge from the previous part. To enhance the recommender performance, we have devised a technique using social network theory that leverages weak links. Our method focuses on creating communities strategically instead of randomly introducing data. Our research demonstrates that recommending items through weak-linked communities among different users promotes user engagement and surprise, as measured by a serendipity metric. We also show how the clustering and grouping process can be further improved by incorporating enhanced data that reinforces social and rating-based behaviors within communities. Moreover, we demonstrate how the measure of uncertainty is affected by the items recommended from the long tail.

8

The Power of Weak Ties on Serendipity in Recommenders

8.1/ Overview

With our increasingly refined online browsing habits, the demand for high-grade recommendation systems has never been greater. Improvements constantly target general performance, evaluation, security, and explainability, but optimizing for serendipitous experiences is imperative since a serendipity-optimized recommender helps users discover unforeseen relevant content. Given that serendipity is a form of genuine unexpected experiences and recommenders are facilitators of user experiences, we aim at leveraging weak ties to explore their impact on serendipity. Weak links refer to social connections between individuals or groups that are not closely related or connected but can still provide valuable information and opportunities. On the other hand, the underlying social structure of recommender datasets can be misleading, rendering traditional network-based approaches ineffective. For that, we developed a network-inspired clustering mechanism to overcome this obstacle. This method elevates the system's performance by optimizing models for unexpected content. By leveraging group weak ties, we aim to provide a novel perspective on the subject and suggest avenues for future research. Our study can also have practical implications for designing online platforms that enhance user experience by promoting unexpected discoveries.

8.2/ Introduction

Recommendation systems utilize complex algorithms to analyze large datasets and generate personalized user recommendations. They have proven to be highly effective in various industries, including e-commerce and media Jannach and Jugovac (2019), and have been shown to improve user engagement and satisfaction significantly Zhang et al.

(2019b). Recommenders are typically designed utilizing data features such as users' search and interaction details. Creating an effective recommendation system that aligns with the primary goal of recommenders remains a subjective and very challenging Zhou et al. (2010) task despite the abundance of research, convenient models, and diversified workflows, such as the recent release of Microsoft's best practices for building recommendation systems Argyriou et al. (2020). The main goal of recommender systems is to provide users with information tailored to their interests Ricci et al. (2021); Nabizadeh et al. (2015). Despite the significant agreement on this general definition, research pathways frequently prioritize other aspects that sometimes even impede the attainment of this goal, as the study by Herlocker et al. highlights regarding the accuracy improvement branches that stemmed from a concept in evaluation termed the ”magic barrier”, i.e., the point beyond which recommenders fail to become more accurate Herlocker et al. (2004); McNee et al. (2006).

Nonetheless, there has been recent research that focuses on further refining recommender systems by addressing crucial issues in general performance Ma et al. (2023); Yannam et al. (2023), noise and evaluation Jurdi et al. (2021); Al Jurdi et al. (2022); Jurdi et al. (2022), security Pramod (2023), and explainability Chatti et al. (2023). Despite progress, there is still a notable disparity: designing methods or frameworks that enhance user engagement and knowledge through chance discoveries and exploration of the unknown.

The concept of ”strength of weak ties” is an influential social science theory that emphasizes the role of weak associations, such as acquaintances, in spreading information and creating opportunities through social networks. Weak ties are more likely to provide novel information than strong ties, such as close friends, who tend to share similar perspectives and resources Granovetter (1973). As we have highlighted, recommender algorithms aim to suggest information likely to interest a user. Therefore, incorporating weak ties into recommender systems can enhance their effectiveness by presenting users with unforeseen recommendations they may not have otherwise discovered, as one study by Duricic et al. recently hinted at Duricic et al. (2019). This performance is not to be confused with general performance in terms of precision or accuracy; it is a little more complex to measure and set up in this case and requires methods beyond the conventional ones Herlocker et al. (2004); Jurdi et al. (2021, 2022). Surely, weak ties can also help overcome the cold-start problem, where new users have insufficient data to generate personalized recommendations. However, weak links also pose privacy challenges, as users may not want to reveal their preferences or behavior to distant or unknown connections Ramakrishnan et al. (2001).

Our approach uses social network theory to measure the impact of weak connections between user groups on serendipity, making it a vital part of the recommender ecosystem.

Narrowing the focus only on chance discoveries allows us to advance recommender enhancement for their primary objective. There are two types of recommender data sources: social and rating Shokeen and Rana (2020). Social datasets have data about user relationships or interactions, such as friendships, likes, etc. Rating datasets have data about user ratings for items or services, such as stars, preferences, etc. In our study, we target the rating-based datasets and recommenders. This research also builds on previous validation and serendipity approaches Jurdi et al. (2021, 2022); Al Jurdi et al. (2018). It adapts a new optimization framework to train recommenders oriented towards chance discoveries for users.

The following section explores the latest research while placing our work in this context. Section 8.4 introduces our unique approach involving community-based data processing and cluster assessment. We present the experimental results and analysis in Section 8.5 and conclude with Section 8.6.

8.3/ Background and Related Work

This section provides an overview of the research on serendipity in recommender systems. Currently, there is no established method to specifically improve the chances of discovering new and unexpected recommendations and to increase user involvement in recommenders. This is especially true when it comes to using weak connections between clusters.

8.3.1/ Serendipity in Recommenders

Some recent recommender system proposals aim to improve serendipity. For example, Kotkov et al. Kotkov et al. (2023) proposed a new definition of uncertainty in recommenders that considers items that are surprising, valuable, and explainable, arguing that the common understanding and original meaning of serendipity is conceptually broader, requiring serendipitous encounters to be neither novel nor unexpected. Others have proposed a multi-view graph contrastive learning framework that can enhance cross-domain sequential recommendation by exploiting serendipitous connections between different domains Wang et al. (2021).

The study by Ziarani et al. Ziarani and Ravanmehr (2021) is crucial and reviews the overall serendipity-oriented approaches in recommender systems. The authors emphasize the significance of serendipity in generating attractive and practical recommendations in recommender systems. This reinforces our introduced concept regarding the direction and primary objective of recommenders in this work's introduction. The approaches covered in the study generally discuss serendipity enhancements by introducing randomness into

the recommendation process. This can lead to discovering new and exciting items that the user may not have otherwise. In addition, serendipity can be enhanced by incorporating diversity into the recommendation process, which can help reduce over-specialization and make recommendations more interesting and engaging. The study concludes that while there is no agreement on the definition of serendipity, most studies find serendipitous recommendations valuable and unexpected.

In a study about surprise in recommenders by Eugene Yan Yan (2020), the importance of a serendipity metric in recommenders is discussed. The author argues that while accuracy is an essential metric for recommendation systems, it is not the only metric that matters. Recommender systems that solely focus on accuracy can lead to information over-specialization, making recommendations boring and predictable. The author suggests incorporating serendipity as a criterion for making appealing and valuable recommendations to address this issue. Serendipity is a criterion for making unexpected and relevant recommendations to the user's interests Ziarani and Ravanmehr (2021). The usefulness of serendipitous recommendations is the main superiority of this criterion over novelty and diversity. The article highlights that serendipity can be measured using various metrics such as surprise, unexpectedness, and relevance Yan (2020). The article further explains that serendipity-oriented recommender systems have been the focus of many studies in recent years. The author conducted a systematic literature review of previous studies on serendipity-oriented recommender systems. The review focused on the contextual convergence of serendipity definitions, datasets, serendipitous recommendation methods, and their evaluation techniques Ziarani and Ravanmehr (2021). The review results indicate that the quality and quantity of articles in the serendipity-oriented recommender systems are progressing. In conclusion, incorporating serendipity as a criterion for making recommendations can help make them more appealing and valuable. It can also help address issues related to information over-specialization and make recommendations more diverse.

One of the studies by Bhandari et al. Bhandari et al. (2013) proposes a method for recommending serendipitous apps using graph-based techniques. The approach can recommend apps even if users do not specify their preferences and can discover highly diverse apps. The authors also introduce randomness into the recommendation process to increase the likelihood of finding new and exciting items that the user may not have discovered otherwise. Therefore, similar to the studies covered in Ziarani and Ravanmehr (2021), this unique process of app recommendations also uses the same method of randomness.

8.3.2/ Recommendations and social network connections

Another proposal by M. Jenders et al. Jenders et al. (2015) introduces a CBF recommendation technique focusing on the serendipity of news recommendations. Serendipitous recommendations have the characteristic of being unexpected yet fortunate and interesting to the user and thus might yield higher user satisfaction. The authors explore the concept of serendipity in the area of news articles and propose a general framework that incorporates the benefits of serendipity and similarity-based recommendation techniques. In addition, they carried out an evaluation against other baseline recommendation models in a user study.

Based on the studies mentioned above, it is clear that enhancing serendipity is a crucial step in improving recommender systems. However, there is currently no established framework for achieving this goal besides incorporating randomization into the system.

8.4/ Community-based Mechanism

Multiple steps are involved in utilizing recommender data to establish weak-connection-based recommendations. We can ideally set two kinds of connections: social-based links Shokeen and Rana (2020) or non-social links inferred from user behavior. In our experimentation, we introduce the latter and develop an approach that could be expanded further if recommender datasets were enriched with more information, particularly those with social and rating-based components. The diagram shown in Figure 8.1 depicts the metadata level, where we aim to enhance the recommendations by processing data differently through the community-based mechanism. To achieve this, we use an approach inspired by networks theory involving grouping users and utilizing weak links between them and the communities (or groups) they belong to. We use techniques like Gower Gower (1971) to form initial user clusters and then create principal collections to establish higher-level communities. Theoretically, this should help us optimize the recommendations to provide more relevant and unexpected suggestions. Next, we refine the training process for the recommender system; this includes modifying the cluster and principal group formation parameters and generating various versions of the potential ”weak links” between groups, as depicted in Figure 8.1. The aim is to avoid prejudice or overfitting towards a particular set of communities and links.

8.4.1/ Serendipity-based Evaluation

While accuracy is vital for recommendation systems, it’s not the only metric that matters. Incorporating serendipity, defined as making unexpected and relevant recommendations,

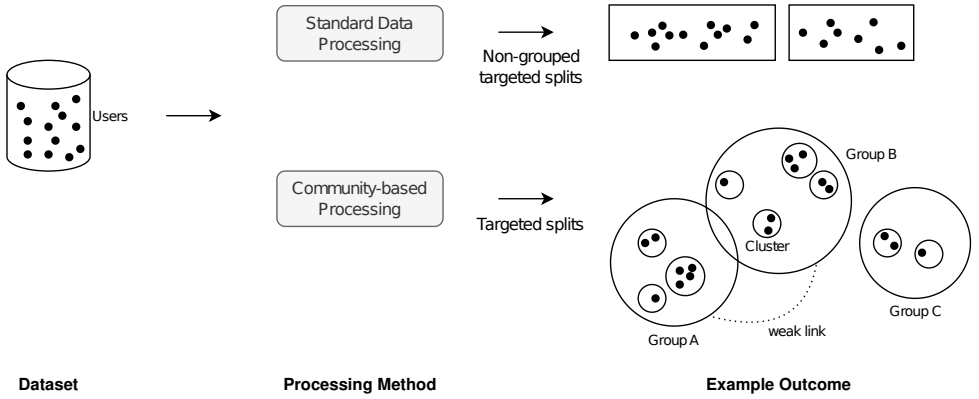

Figure 8.1: This schematic illustrates a high-level difference between normal data processing and group-based processing.

can make recommendations more appealing and valuable. Serendipity can be measured using metrics like surprise, unexpectedness, and relevance. Previous studies on serendipity-oriented recommender systems show that incorporating serendipity can help make recommendations more diverse and address issues related to information over-specialization.

Following the study by Eugene Yan Yan (2020), serendipity can be measured using the following formula:

$$serendipity(i) = unexpectedness(i) \times relevance(i) \tag{8.1}$$

Where $relevance(i) = 1$ if i is interacted upon and 0 otherwise. Alternatively, we use one of several approaches to measure the unexpectedness of recommendations Yan (2020); Panagiotis (2011). This approach considers some distance metric (e.g., cosine similarity). We compute the cosine similarity between a user's recommended items (I) and historical interactions (H). Lower cosine similarity indicates higher unexpectedness:

$$unexpectedness(I, H) = \frac{1}{I} \sum_{i \in I} \sum_{h \in H} cos(i, h) \tag{8.2}$$

The overall serendipity can be achieved by averaging all users (U) and all recommended items (I):

$$serendipity(i) = \frac{1}{count(U)} \sum_{u \in U} \sum_{i \in I} \frac{serendipity(i)}{count(I)} \tag{8.3}$$

The following section explains how we form user clusters and groups. As we measure serendipity on the group level, we use a recently proposed group-based validation technique Jurdi et al. (2022) to track performance on smaller data portions, which helps avoid

averaging results that may mask essential effects. Therefore, changes in serendipity are measured by changes in its level within groups (e.g., group A in Fig. 8.1) rather than the overall user serendipity of equation 8.3.

8.4.2/ User Clusters and Groups

In this section, we cover the process of forming clusters and higher-level groups after discussing the method of evaluating groups of users in the previous section. Two levels are involved in this process - clustering users together at the first level and forming larger groups that can connect and include user groups with weak and strong links at the second level. We experiment with various versions of weak links between user clusters, as there can be multiple variations of higher-level cluster groups. This demonstrates the method's adaptability to accommodate different datasets.

To create the first level of user clusters for datasets like ML-100k, which often have both categorical and non-categorical data, we employ the Gower distance method Gower (1971) to produce a distance matrix. This approach calculates the distance between two entities based on their mixed categorical and numerical attribute values. We then use hierarchical clustering to refine the grouping further. For some given features $x_i = x_{i1}, ..., x_{ip}$ in a dataset, the Gower similarity matrix can be defined as:

$$S_{Gower}(x_i, x_j) = \frac{\sum_{k=1}^{p} s_{ijk}\delta_{ijk}}{\sum_{k=1}^{p} \delta_{ijk}} \tag{8.4}$$

For each feature $k = 1, ..., p$ a score s_{ijk} is calculated. A quantity δ_{ijk} is also calculated with a binary possible value depending on whether the input variables x_i and x_j can be compared. $S_{Gower}(x_i, x_j)$ is a similarity score, so the final result is converted through the following equation to achieve a distance metric: $d_{Gower} = \sqrt{1 - S_{Gower}}$. For numerical variables, the score can be calculated as a simple L1 distance between the two values normalized by the range of the feature R_k:

$$s_{ijk} = 1 - \frac{|x_{ik} - x_{jk}|}{R_k} \tag{8.5}$$

For categorical variables, the score will be 1 if the categories are the same and 0 if they are not:

$$S_{ijk} = 1 x_{ik} = x_{jk} \tag{8.6}$$

Several linkage methods exist to compute distance $d(s, t)$ between two clusters s and t using the distance matrix achieved with Equation 8.4. We utilize the general-purpose

clustering algorithm proposed by Müllner Müllner (2011). The algorithm begins with a forest of clusters that have yet to be used in the hierarchy being formed. When two clusters s and t from this forest are combined into a single cluster u, s and t are removed from the forest, and u is added to the forest. When only one cluster remains in the forest, the algorithm stops, and this cluster becomes the root.

In the following section, we present experimental results for the abovementioned method. The experiment has three main goals:

- Investigating the impact of recommending items via weak-linked groups.
- Determining whether optimizations in one group can impact others.
- Showcasing the effect of utilizing weaker connections alongside group linkage tuning and whether more favorable outcomes can be achieved.

8.5/ Results and Discussions

In this section, we discuss the results of experiments on two open-source datasets, namely the ML-100k Harper and Konstan (2015) and the Epinions Richardson et al. (2003). Our work doesn't focus on a specific recommender algorithm but on the experimentation process. We use LightGCN Wang et al. (2019) as an example recommender. LightGCN is a simplified version of Neural Graph Collaborative Filtering (NGCF) that incorporates GCNs and is relatively new. We have created multiple versions of the code and experiment scenarios, all available in the source Jurdi and Abdo (2023).

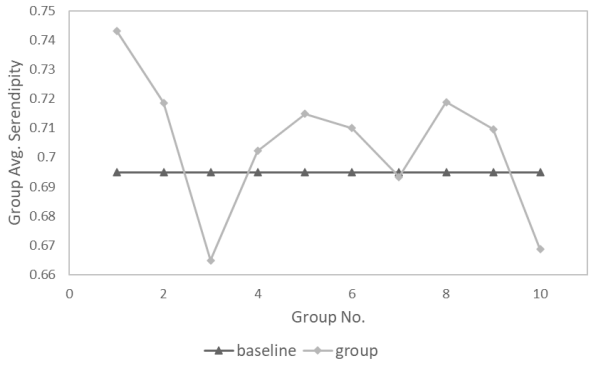

Figure 8.2: A comparison between the average serendipity value of a group and the baseline value achieved during regular data processing.

Our initial goal is to measure the impact on group serendipity. We plan to achieve this by selecting a formed group and tuning the recommender through training to allow favored recommendations from weakly-linked communities. The second objective is to determine whether this approach affects only the target group or any other group in the dataset that shares familiar users.

Figure 8.2 displays the average group serendipity obtained from one of our experiments on ML-100k. These groups were created through the process explained in Section 8.4. We observed a notable increase in the serendipity factor in multiple groups compared to the baseline process, ranging from 5% to 18%. This baseline process involved standard data processing and training using the same recommender parameters and tuning. However, only two out of the ten groups showed a decrease in the metric result.

Table 8.1: The evaluation metric values for baseline and community-based processes.

Metric	**Baseline**	**Group Average**	$\delta\%$
Precision	0.2897	0.2288	-21.04
Map	0.1424	0.0993	-30.26
NDCG	0.3716	0.2874	-22.65
Recall	0.2440	0.1866	-23.50
Coverage	0.3610	0.1741	-51.76

After analyzing the offline metric results of the system, it is evident that all of them experienced a decrease compared to the baseline run. This decrease can be attributed to the increase in serendipity, which leads to a corresponding decline in precision and recall. The results can be viewed in Table 8.1. However, we must remember that offline evaluation is not enough to determine actual relevance. Through online experimentation, we can accurately gauge the model's effectiveness Yan (2020). One of the interesting findings is a decrease in coverage. As explained in Section 8.3, increasing coverage (or introducing more randomness) in the dataset typically leads to an increase in serendipity during offline evaluation, which is a limitation of using this measure offline instead of in an A/B test. However, we took steps to minimize this effect by ensuring that our final recommendations were unbiased and that we did not filter out items from the long tail. Online tests can improve the validation of the serendipity metric. Several small tests have shown that users converge more with recommenders with lower accuracy and precision metrics Ziarani and Ravanmehr (2021); Yan (2020). Therefore, our results are in line with this trend.

Subsequently, an exploration is conducted to determine the potential impact of optimizing surprise in one group on the other by utilizing the approach above. For clarity, we have included the cluster-level outcomes (refer to Figure 8.1).

In Figure 8.3, we show the effect of the same serendipity metric but on the cluster level of the ML-100k dataset. The figure shows three cases when our approach is optimized to increase serendipity in one group while measuring the effect on the others. It can be noticed here how, with no unique tuning, the result can be better (first figure), almost the same with minor exceptions (middle figure), or worse (last figure). As mentioned in Section 8.4, forming user clusters and groups is sensitive, with multiple possibilities for weak links.

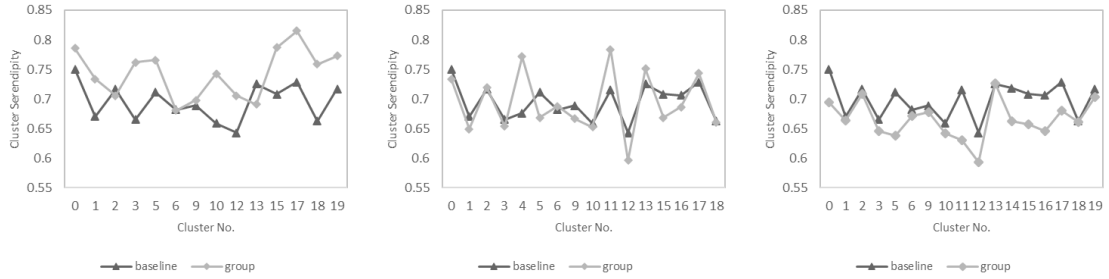

Figure 8.3: Three scenarios that compare cluster serendipity to the baseline.

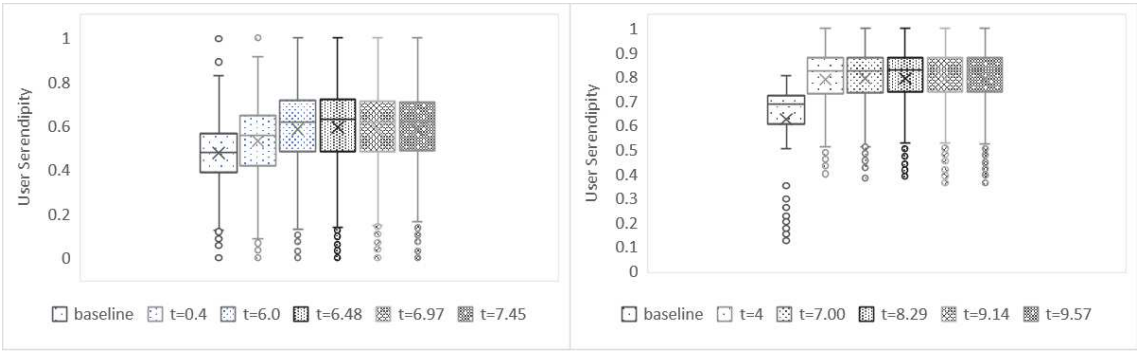

Figure 8.4: The distribution of user serendipity metric values as group formations slightly vary.

In Section 8.4.2, we discussed the hierarchical clustering method. This method can simplify the dendrogram and assign data points to individual clusters. The assigned clusters are determined by a distance threshold, denoted as t. A smaller threshold will allow even the closest data points to form a cluster, while a more significant threshold can result in too many clusters and few communities. By varying the value of t, we can produce different group representations that could affect the outcome of the metrics obtained in the initial stage of the experiments. To address this, we conduct multiple iterations that result in diverse weak-linked groups. Subsequently, we implement recommendations and re-evaluate the serendipity metric to determine any impact on the results.

The boxplot in Figure 8.4 displays the results. In the ml-100k scenario (on the left), we can observe an overall rise in user serendipity after the initial three iterations. This suggests an enhanced surprise element for most groups, as previously demonstrated in an experiment. We attain the highest value at approximately $t = 6.48$, corresponding to the optimal distance between the groups formed. This distance has a positive impact on serendipitous recommendations for almost all groups. We achieved the same outcome for the Epinions dataset, although the parameter scale and the optimal distance between groups differed slightly. The best results were obtained with values between $t = 7$ and $t = 8.5$, and we found that further adjustments did not significantly improve results for most of the clusters. Using varied weak connections between groups can improve outcomes.

Testing multiple scenarios helps find the best distance for each optimization run.

8.6/ CONCLUSION

This study emphasizes the significance of prioritizing chance discoveries for users in recommender systems. We developed a method based on social network theory, which utilizes weak links to enhance recommender performance. As non-social links inferred from user behavior have not previously been used, we created a process for it in this work. This involves strategically forming communities rather than introducing random data. Our experiment yielded a positive result in enhancing the level of unexpectedness and surprise for users within the system. We demonstrated that recommending items through weak-linked communities among different users favors surprise and user engagement, as measured by a serendipity metric. This was achieved without any intentional randomness introduced into the data.

The clustering and grouping process can be improved by tuning with enhanced data for recommender systems, specifically data that reinforces social and rating-based behaviors within communities. Alternatively, the measure of serendipity is impacted by the items recommended from the long tail. While we avoided bias in suggesting long-tail items, conducting A/B tests to confirm convergence and not rely solely on offline tests would be beneficial. Finally, future research could explore social clustering to validate whether different effects can be achieved on the serendipity of the model.

VII

CONCLUSION

9

General Conclusion

9.1/ Conclusion

The growing amount of online data has made it difficult for users to find relevant information. Recommender systems help by filtering this information and presenting users with content most likely to interest them. With the rise of generative AI applications, the amount of data available is increasing at an unprecedented rate, making recommender systems more critical than ever. These systems rely on advanced algorithms that analyze user data to identify patterns and make personalized recommendations. Using these systems saves users time and avoids the frustration of sifting through irrelevant information.

The thesis covers various important aspects such as serendipity's significance in recommender systems, natural noise in datasets, evaluation of the systems, and methods to improve and optimize performance while considering essential angles only.

Serendipity is vital in a recommender system and continues to be developed. Although a precise technical definition has not been established, it can be described as a pleasant surprise of an unknown encounter. In the first part of the thesis, we have demonstrated that including serendipitous recommendations in the list of relevant recommendations can enhance user satisfaction. Alternatively, recommender data is often used in model training without filtering. This allows forms of noise to dictate the recommendations users receive, and one very famous form is natural noise, which is the topic of focus of the second part of this thesis. This noise can be due to attacks or a result of natural noise that the user unintentionally makes. Noise detection algorithms free the systems from noise without considering that natural noise and serendipity overlap in their definition and, as a result, disregard the importance of uncertainty. In general, recommender systems target accuracy, which naturally leads to the over-personalization of the recommended data. Alternatively, serendipity is often neglected in recommender implementation. In this part, we proposed an optimized algorithm that can handle noise yet allows serendipity to exist in the system, and the new top-N adjusted metric measures the effectiveness of the

algorithm's output. The results proved that the performance was radically enhanced.

In the second part of the thesis, we focus on the issue of noise in recommenders and the problems present in their evaluation. Recommenders are complex systems, and we have found several related issues. In this section, we discuss these issues and propose optimized implementation of evaluation methods to improve their performance. Additionally, we introduce newer forms of noise, such as opt-out obfuscation, that researchers and practitioners should be aware of.

Implementing an effective and agile natural noise management algorithm for recommender systems' datasets is challenging due to various parameters that should be considered, especially in the evaluation process. There has been no attempt to synthesize what is traditionally known about the performance evaluation of recommender systems and natural noise management, nor to systematically recognize the implications of evaluating them for numerous tasks and diverse contexts while testing the performance of a natural noise technique. Throughout a comprehensive study in the fourth part of this thesis, we analyzed and categorized all the natural noise algorithms. In addition, we introduced empirical results from two hypotheses that provided critical insight into the consistency of the evaluation methods used in the proposed noise management techniques. The experiments were very promising, and one illustrated how randomness could achieve comparable outcomes to one of the most conventional mechanisms. At the same time, the second proved that the metrics employed to test those techniques and rank one better than the other typically display inconsistent and unreliable results.

Implementing an effective and agile natural noise management algorithm for recommenders is challenging due to numerous parameters that should be considered, especially in the evaluation process. As demonstrated in the second chapter of the fourth part, the obfuscation phenomenon created an additional challenge to the evaluation process. We have introduced two modern forms of noise that are hard to detect with current evaluation strategies and showed how the data appears to be perfectly normal. The impact was only visible when we evaluated the performance in data subgroups using the new proposed group validation process in the evaluation ecosystem of recommenders. Additionally, there has yet to be an attempt to synthesize what is known about the various noise categories in RSs, nor to systematically devise a unified protocol that would deal with noise irrespective of its type and independent of the deployed recommendation engine. Whether it's user-induced to opt out of data processing for particular security concerns or publicly injected by authorities, such as in the case of Russia, Mexico, and Lebanon, Obfuscation is a challenge that RSs should be aware of.

In the fifth part of this thesis, the main focus is on extending the previous work discussed in part four. The proposed approach is aimed at evaluating and assessing the performance of recommender systems in a holistic manner. Evaluating the effectiveness of

recommenders and testing their performance as the underlying data evolves with time remains a remarkably tough challenge yet to be tackled. While some improvement in the benchmarking field has been reported, the recommender ecosystem has not covered the concept of reporting the performance of models on dataset subgroups and the potential of detecting specific kinds of potentially malicious data behaviors. To address this gap, this part introduced a new evaluation strategy for recommenders called group validation framework in recommenders. This method can be employed as an assessment tool to track the performance of a recommender on certain important clusters/groups of data. The results proved that recommenders performed differently in various groups as unique data perturbations were introduced. The most critical aspect of this approach is that it could spot negatively affected sections of the dataset due to added synthetic data that was not visible with standard evaluation techniques. Group validation helps localize the errors in the data and provides a means to identify the effect of behavioral data changes in the system.

The final part of the thesis combines the ideas of the previous three sections and proposes a new strategy to obtain high-level recommendations using a community-based approach. It emphasizes the significance of prioritizing uncertainty for users in recommender systems. We developed a method based on social network theory - the power of weak ties - which utilizes weak links between individuals to enhance their experience in a system. As non-social links inferred from user behavior have not previously been used, we designed an initial process for it. It involves strategically forming communities rather than introducing random data. Our experiment yielded very positive results in enhancing the level of surprise for users within the system. We demonstrated that recommending items through weak-linked communities among different users favors user engagement. This was achieved without any intentional randomness introduced into the data.

9.2/ PERPECTIVES

The experiments, results, and knowledge acquired throughout this research work open the door to many short-, medium-, and long-term perspectives in various domains. The findings of this thesis could be implemented in many areas, including e-commerce, social networking, news recommendation systems, and healthcare systems, among others. The proposed algorithm and evaluation methods could be applied to improve the accuracy and serendipity of recommendations, reduce noise, and enhance user satisfaction in these domains. Additionally, the study's insights could lead to further research on natural (even non-natural) noise management algorithms and their performance evaluation, ultimately leading to future robust recommender systems.

In the domain of serendipity and user engagement, the clustering and grouping process

can be improved by tuning various models with enhanced data, specifically data that reinforces social and rating-based behaviors within communities. Alternatively, the measure of serendipity is impacted by the items recommended from the long tail, the items that are least exposed in a certain retailer, for example. While we avoided bias in suggesting long-tail items in the experiments of this proposal, conducting A/B tests to confirm convergence and not rely solely on offline tests would be very beneficial to proposing the correct approach in model selection, data clustering, and community formation, and of course, performance evaluation based on the formed groups. Finally, future research could explore social clustering to validate whether different effects can be achieved on the serendipity of the model.

The natural noise management part added important discussions and results to the evaluation and employment of recommender systems. The potential problems that might be addressed in future works extending those points include the development of unified evaluation methods that can broadly be used with any recommendation technique (and in various domains) and serve us to assess better the actual effectiveness of a recommender devoid of inconsistencies. In addition, natural noise lacks proper development regarding the type of datasets it is applied to. A practical noise management approach must be scalable and adequately work with diverse datasets irrespective of the recommendation algorithm employed. External data that administrators must retrieve from customers to implement a particular noise management approach remains an inadequate solution to the problem.

On the other hand, opt-out attacks pave the way for multiple discussion paths that cover numerous topics; for instance, identifying a user's opt-out behavior can permit tracing back to the primary user's tastes. Additionally, data owners can develop data mining methods to discover the general trends of users opting out of the online platform.

The group-based evaluation framework that was proposed in the fifth part of the thesis paves the way for a unified evaluation process but can be further improved to cover other scenarios where dataset grouping could be effectively applied to several recommender dataset types (such as implicit feedback) and embed different recommender goals. Further, a data-oriented approach can be introduced to have a more intelligent grouping mechanism. It can automatically treat potential performance degradation in specific clusters and measure the effect on serendipity as a factor of such changes.

BIBLIOGRAPHY

[Adamopoulos and Tuzhilin 2014] ADAMOPOULOS, Panagiotis ; TUZHILIN, Alexander: **"On unexpectedness in recommender systems: Or how to better expect the unexpected"**. In *ACM Transactions on Intelligent Systems and Technology (TIST)* 5 (2014), number 4, pages 1–32

[Aggarwal et al. 2016] AGGARWAL, Charu C. ; OTHERS: *Recommender systems*. Volume 1. Springer, 2016

[Ahmadli 2022] AHMADLI, Aydin: **"Probabilistic Models for Recommender Systems"**. (2022)

[Ahuja et al. 2019] AHUJA, Rishabh ; SOLANKI, Arun ; NAYYAR, Anand: **"Movie recommender system using K-Means clustering and K-Nearest Neighbor"**. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* IEEE (event), 2019, pages 263–268

[Al Hassanieh et al. 2018] AL HASSANIEH, Lamis ; ABOU JAOUDEH, Chadi ; ABDO, Jacques B. ; DEMERJIAN, Jacques: **"Similarity measures for collaborative filtering recommender systems"**. In *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)* IEEE (event), 2018, pages 1–5

[Al Jurdi 2022] AL JURDI, Wissam: *Group Validation in Recommender Systems: Framework for Multi-layer Performance Evaluation*. 5 2022. – URL https://github.com/wissamjur/group-validation

[Al Jurdi et al. 2022] AL JURDI, Wissam ; ABDO, Jacques B. ; DEMERJIAN, Jacques ; MAKHOUL, Abdallah: **"Strategic Attacks on Recommender Systems: An Obfuscation Scenario"**. In *2022 IEEE/ACS 19th International Conference on Computer Systems and Applications (AICCSA)* IEEE (event), 2022, pages 1–8

[Al Jurdi et al. 2018] AL JURDI, Wissam ; BADRAN, Miriam El K. ; ABOU JAOUDE, Chady ; ABDO, Jacques B. ; DEMERJIAN, Jacques ; MAKHOUL, Abdallah: **"Serendipity-aware noise detection system for recommender systems"**. In *Proceedings of the Information and Knowledge Engineering* (2018)

[Al Jurdi et al. 2019] AL JURDI, Wissam ; JAOUDE, Chady A. ; BADRAN, Miriam El K. ; BOU ABDO, Jacques ; DEMERJIAN, Jacques ; MAKHOUL, Abdallah: **"SCCF Parame-**

ter and Similarity Measure Optimization and Evaluation". In *International Conference on Knowledge Science, Engineering and Management* Springer (event), 2019, pages 118–127

[Aljukhadar et al. 2010] ALJUKHADAR, Muhammad ; SENECAL, Sylvain ; DAOUST, Charles-Etienne: **"Information overload and usage of recommendations"**. In *Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain*, 2010, pages 26–33

[Amatriain et al. 2009a] AMATRIAIN, Xavier ; PUJOL, Josep M. ; OLIVER, Nuria: **"I like it... i like it not: Evaluating user ratings noise in recommender systems"**. In *International Conference on User Modeling, Adaptation, and Personalization* Springer (event), 2009, pages 247–258

[Amatriain et al. 2009b] AMATRIAIN, Xavier ; PUJOL, Josep M. ; TINTAREV, Nava ; OLIVER, Nuria: **"Rate it again: increasing recommendation accuracy by user re-rating"**. In *Proceedings of the third ACM conference on Recommender systems*, 2009, pages 173–180

[Amigó et al. 2018] AMIGÓ, Enrique ; SPINA, Damiano ; ALBORNOZ, Jorge Carrillo-de: **"An axiomatic analysis of diversity evaluation metrics: Introducing the rank-biased utility metric"**. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pages 625–634

[Anelli et al. 2021] ANELLI, Vito W. ; BELLOGÍN, Alejandro ; FERRARA, Antonio ; MALITESTA, Daniele ; MERRA, Felice A. ; POMO, Claudio ; DONINI, Francesco M. ; DI NOIA, Tommaso: **"Elliot: a comprehensive and rigorous framework for reproducible recommender systems evaluation"**. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pages 2405–2414

[Argyriou et al. 2020] ARGYRIOU, Andreas ; GONZÁLEZ-FIERRO, Miguel ; ZHANG, Le: **"Microsoft Recommenders: Best Practices for Production-Ready Recommendation Systems"**. In *Companion Proceedings of the Web Conference 2020*, 2020, pages 50–51

[Arthur and Vassilvitskii 2006] ARTHUR, David ; VASSILVITSKII, Sergei: **"k-means++: The advantages of careful seeding"**. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2006, pages 1027—-1035

[Atashkar and Safi-Esfahani 2020] ATASHKAR, Mahdieh ; SAFI-ESFAHANI, Faramarz: **"Item-Based Recommender Systems Applying Social-Economic Indicators"**. In *SN Computer Science* 1 (2020), number 2, pages 113

[Badran et al. 2019a] BADRAN, Miriam El K. ; BOU ABDO, Jacques ; AL JURDI, Wissam ; DEMERJIAN, Jacques: **"Adaptive Serendipity for Recommender Systems: Let It Find You."**. In *ICAART (2)*, 2019, pages 739–745

[Badran et al. 2019b] BADRAN, Miriam El K. ; JURDI, Wissam ; ABDO, Jacques B.: **"Survey on shilling attacks and their detection algorithms in recommender systems"**. In *Proceedings of the International Conference on Security and Management (SAM)* The Steering Committee of The World Congress in Computer Science, Computer ... (event), 2019, pages 141–146

[Badsha et al. 2016] BADSHA, Shahriar ; YI, Xun ; KHALIL, Ibrahim: **"A practical privacy-preserving recommender system"**. In *Data Science and Engineering* 1 (2016), pages 161–177

[Bag et al. 2019] BAG, Sujoy ; KUMAR, Susanta ; AWASTHI, Anjali ; TIWARI, Manoj K.: **"A noise correction-based approach to support a recommender system in a highly sparse rating environment"**. In *Decision Support Systems* 118 (2019), pages 46–57

[Beel 2020] BEEL, Joeran: **"Recommender-Systems. com: A Central Platform for the Recommender-System Community"**. In *Proceedings of the 14th ACM Conference on Recommender Systems*, 2020, pages 600–603

[Beel and Dinesh 2017] BEEL, Joeran ; DINESH, Siddarth: **"Real-World Recommender Systems for Academia: The Pain and Gain in Building, Operating, and Researching them."**. In *BIR@ ECIR*, 2017, pages 6–17

[Bellogín and Said 2021] BELLOGÍN, Alejandro ; SAID, Alan: **"Improving accountability in recommender systems research through reproducibility"**. In *User Modeling and User-Adapted Interaction* (2021), pages 1–37

[Bellogín et al. 2014] BELLOGÍN, Alejandro ; SAID, Alan ; VRIES, Arjen P. de: **"The magic barrier of recommender systems–no magic, just ratings"**. In *User Modeling, Adaptation, and Personalization: 22nd International Conference, UMAP 2014, Aalborg, Denmark, July 7-11, 2014. Proceedings 22* Springer (event), 2014, pages 25–36

[Beregovskaya and Koroteev 2021] BEREGOVSKAYA, Irina ; KOROTEEV, Mikhail: **"Review of Clustering-Based Recommender Systems"**. In *arXiv preprint arXiv:2109.12839* (2021)

[Bhandari et al. 2013] BHANDARI, Upasna ; SUGIYAMA, Kazunari ; DATTA, Anindya ; JINDAL, Rajni: **"Serendipitous recommendation for mobile apps using item-item similarity graph"**. In *Information Retrieval Technology: 9th Asia Information Retrieval Societies Conference, AIRS 2013, Singapore, December 9-11, 2013. Proceedings 9* Springer (event), 2013, pages 440–451

[Bobadilla et al. 2013] BOBADILLA, Jesús ; ORTEGA, Fernando ; HERNANDO, Antonio ; GUTIÉRREZ, Abraham: **"Recommender systems survey"**. In *Knowledge-based systems* 46 (2013), pages 109–132

[Brunton and Nissenbaum 2015] BRUNTON, Finn ; NISSENBAUM, Helen: *Obfuscation: A user's guide for privacy and protest*. Mit Press, 2015

[Buder and Schwind 2012] BUDER, Jürgen ; SCHWIND, Christina: **"Learning with personalized recommender systems: A psychological view"**. In *Computers in Human Behavior* 28 (2012), number 1, pages 207–216

[Cai and Zhang 2019] CAI, Hongyun ; ZHANG, Fuzhi: **"Detecting shilling attacks in recommender systems based on analysis of user rating behavior"**. In *Knowledge-Based Systems* 177 (2019), pages 22–43

[Castells et al. 2022] CASTELLS, Pablo ; HURLEY, Neil ; VARGAS, Saul: **"Novelty and diversity in recommender systems"**. In *Recommender systems handbook*. Springer, 2022, pages 603–646

[Castro et al. 2017] CASTRO, Jorge ; YERA, Raciel ; MARTÍNEZ, Luis: **"An empirical study of natural noise management in group recommendation systems"**. In *Decision Support Systems* 94 (2017), pages 1–11

[Castro et al. 2018] CASTRO, Jorge ; YERA, Raciel ; MARTINEZ, Luis: **"A fuzzy approach for natural noise management in group recommender systems"**. In *Expert Systems with Applications* 94 (2018), pages 237–249

[Chaaya et al. 2017] CHAAYA, Georges ; MÉTAIS, Elisabeth ; ABDO, Jacques B. ; CHIKY, Raja ; DEMERJIAN, Jacques ; BARBAR, Kablan: **"Evaluating non-personalized single-heuristic active learning strategies for collaborative filtering recommender systems"**. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)* IEEE (event), 2017, pages 593–600

[Chatti et al. 2023] CHATTI, Mohamed A. ; GUESMI, Mouadh ; MUSLIM, Arham: **"Visualization for Recommendation Explainability: A Survey and New Perspectives"**. In *arXiv preprint arXiv:2305.11755* (2023)

[Chen et al. 2019] CHEN, Li ; YANG, Yonghua ; WANG, Ningxia ; YANG, Keping ; YUAN, Quan: **"How serendipity improves user satisfaction with recommendations? a large-scale user evaluation"**. In *The world wide web conference*, 2019, pages 240–250

[Chin et al. 2022] CHIN, Jin Y. ; CHEN, Yile ; CONG, Gao: **"The Datasets Dilemma: How Much Do We Really Know About Recommendation Datasets?"**. In *Proceed-*

ings of the Fifteenth ACM International Conference on Web Search and Data Mining, 2022, pages 141–149

[Choudhary et al. 2017] CHOUDHARY, Priyankar ; KANT, Vibhor ; DWIVEDI, Pragya: **"Handling natural noise in multi criteria recommender system utilizing effective similarity measure and particle swarm optimization"**. In *Procedia computer science* 115 (2017), pages 853–862

[Chung et al. 2019a] CHUNG, Yeounoh ; KRASKA, Tim ; POLYZOTIS, Neoklis ; TAE, Ki H. ; WHANG, Steven E.: **"Automated data slicing for model validation: A big data-AI integration approach"**. In *IEEE Transactions on Knowledge and Data Engineering* 32 (2019), number 12, pages 2284–2296

[Chung et al. 2019b] CHUNG, Yeounoh ; KRASKA, Tim ; POLYZOTIS, Neoklis ; TAE, Ki H. ; WHANG, Steven E.: **"Slice finder: Automated data slicing for model validation"**. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)* IEEE (event), 2019, pages 1550–1553

[Chung et al. 2018] CHUNG, Yeounoh ; KRASKA, Tim ; WHANG, Steven E. ; POLYZOTIS, Neoklis: **"Slice finder: Automated data slicing for model interpretability"**. In *SysML Conference*, 2018, pages 1550–1553

[Clarke et al. 2008] CLARKE, Charles L. ; KOLLA, Maheedhar ; CORMACK, Gordon V. ; VECHTOMOVA, Olga ; ASHKAN, Azin ; BÜTTCHER, Stefan ; MACKINNON, Ian: **"Novelty and diversity in information retrieval evaluation"**. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008, pages 659–666

[Cohen 2013] COHEN, Jacob: *Statistical power analysis for the behavioral sciences*. Routledge, 2013

[Columbus] COLUMBUS, L.: *How COVID-19 Is Transforming E-Commerce*. – URL https://www.forbes.com/sites/louiscolumbus/2020/04/28/how-covid-19-is-transforming-e-commerce/

[Comninos 2011] COMNINOS, Alex: **"E-revolutions and cyber crackdowns: User-generated content and social networking in protests in MENA and beyond"**. In *Global Information Society* (2011), pages 31–37

[Cover and Hart 1967] COVER, Thomas ; HART, Peter: **"Nearest neighbor pattern classification"**. In *IEEE transactions on information theory* 13 (1967), number 1, pages 21–27

[De Gemmis et al. 2015] DE GEMMIS, Marco ; LOPS, Pasquale ; SEMERARO, Giovanni ; MUSTO, Cataldo: **"An investigation on the serendipity problem in recommender**

systems". In *Information Processing & Management* 51 (2015), number 5, pages 695–717

[De Pessemier et al. 2014] DE PESSEMIER, Toon ; DOOMS, Simon ; MARTENS, Luc: **"Comparison of group recommendation algorithms"**. In *Multimedia tools and applications* 72 (2014), pages 2497–2541

[Drosou and Pitoura 2010] DROSOU, Marina ; PITOURA, Evaggelia: **"Search result diversification"**. In *ACM SIGMOD Record* 39 (2010), number 1, pages 41–47

[Duricic et al. 2019] DURICIC, Tomislav ; LACIC, Emanuel ; KOWALD, Dominik ; LEX, Elisabeth: **"Exploiting weak ties in trust-based recommender systems using regular equivalence"**. In *arXiv preprint arXiv:1907.11620* (2019)

[Elahi et al. 2014] ELAHI, Mehdi ; RICCI, Francesco ; RUBENS, Neil: **"Active learning strategies for rating elicitation in collaborative filtering: a system-wide perspective"**. In *ACM Transactions on Intelligent Systems and Technology (TIST)* 5 (2014), number 1, pages 1–33

[Fang and Zhai 2005] FANG, Hui ; ZHAI, ChengXiang: **"An exploration of axiomatic approaches to information retrieval"**. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005, pages 480–487

[Fang et al. 2018] FANG, Minghong ; YANG, Guolei ; GONG, Neil Z. ; LIU, Jia: **"Poisoning attacks to graph-based recommender systems"**. In *Proceedings of the 34th annual computer security applications conference*, 2018, pages 381–392

[Ferro et al. 2018] FERRO, Nicola ; FUHR, Norbert ; GREFENSTETTE, Gregory ; KONSTAN, Joseph A. ; CASTELLS, Pablo ; DALY, Elizabeth M. ; DECLERCK, Thierry ; EKSTRAND, Michael D. ; GEYER, Werner ; GONZALO, Julio ; OTHERS: **"The Dagstuhl perspectives workshop on performance modeling and prediction"**. In *ACM SIGIR Forum* Volume 52 ACM New York, NY, USA (event), 2018, pages 91–101

[Gadanho and Lhuillier 2007] GADANHO, Sandra C. ; LHUILLIER, Nicolas: **"Addressing uncertainty in implicit preferences"**. In *Proceedings of the 2007 ACM conference on Recommender systems*, 2007, pages 97–104

[Ge et al. 2010] GE, Mouzhi ; DELGADO-BATTENFELD, Carla ; JANNACH, Dietmar: **"Beyond accuracy: evaluating recommender systems by coverage and serendipity"**. In *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pages 257–260

[Gower 1971] GOWER, John C.: **"A general coefficient of similarity and some of its properties"**. In *Biometrics* (1971), pages 857–871

[Granovetter 1973] GRANOVETTER, Mark S.: **"The strength of weak ties"**. In *American journal of sociology* 78 (1973), number 6, pages 1360–1380

[Gunawardana et al. 2012] GUNAWARDANA, Asela ; SHANI, Guy ; YOGEV, Sivan: **"Evaluating recommender systems"**. In *Recommender systems handbook*. Springer, 2012, pages 547–601

[Gunes et al. 2014] GUNES, Ihsan ; KALELI, Cihan ; BILGE, Alper ; POLAT, Huseyin: **"Shilling attacks against recommender systems: a comprehensive survey"**. In *Artificial Intelligence Review* 42 (2014), pages 767–799

[Gupta et al. 2021] GUPTA, Roopam ; OTHERS: **"An efficient feature subset selection approach for machine learning"**. In *Multimedia tools and applications* 80 (2021), number 8, pages 12737–12830

[Han et al. 2021] HAN, Di ; HUANG, Yifan ; JING, Xiaotian ; LIU, Junmin: **"AND: Effective Coupling of Accuracy, Novelty and Diversity in the Recommender System"**. In *2021 17th International Conference on Mobility, Sensing and Networking (MSN)* IEEE (event), 2021, pages 772–777

[Harper and Konstan 2015] HARPER, F M. ; KONSTAN, Joseph A.: **"The movielens datasets: History and context"**. In *Acm transactions on interactive intelligent systems (tiis)* 5 (2015), number 4, pages 1–19

[Herlocker et al. 2004] HERLOCKER, Jonathan L. ; KONSTAN, Joseph A. ; TERVEEN, Loren G. ; RIEDL, John T.: **"Evaluating collaborative filtering recommender systems"**. In *ACM Transactions on Information Systems (TOIS)* 22 (2004), number 1, pages 5–53

[Hill et al. 1995] HILL, Will ; STEAD, Larry ; ROSENSTEIN, Mark ; FURNAS, George: **"Recommending and evaluating choices in a virtual community of use"**. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1995, pages 194–201

[Huang et al. 2004] HUANG, Zan ; CHEN, Hsinchun ; ZENG, Daniel: **"Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering"**. In *ACM Transactions on Information Systems (TOIS)* 22 (2004), number 1, pages 116–142

[Hug 2020] HUG, Nicolas: **"Surprise: A Python library for recommender systems"**. In *Journal of Open Source Software* 5 (2020), number 52, pages 2174. – URL https://doi.org/10.21105/joss.02174. DOI: 10.21105/joss.02174

[Iaquinta et al. 2008] IAQUINTA, Leo ; DE GEMMIS, Marco ; LOPS, Pasquale ; SEMERARO, Giovanni ; FILANNINO, Michele ; MOLINO, Piero: **"Introducing serendipity in a**

content-based recommender system". In *2008 eighth international conference on hybrid intelligent systems* IEEE (event), 2008, pages 168–173

[Isinkaye et al. 2015] ISINKAYE, Folasade O. ; FOLAJIMI, Yetunde O. ; OJOKOH, Bolande A.: **"Recommendation systems: Principles, methods and evaluation"**. In *Egyptian informatics journal* 16 (2015), number 3, pages 261–273

[Jannach and Jugovac 2019] JANNACH, Dietmar ; JUGOVAC, Michael: **"Measuring the business value of recommender systems"**. In *ACM Transactions on Management Information Systems (TMIS)* 10 (2019), number 4, pages 1–23

[Jenders et al. 2015] JENDERS, Maximilian ; LINDHAUER, T ; KASNECI, Gjergji ; KRESTEL, Ralf ; NAUMANN, Felix: **"A serendipity model for news recommendation"**. In *KI 2015: Advances in Artificial Intelligence: 38th Annual German Conference on AI, Dresden, Germany, September 21-25, 2015, Proceedings 38* Springer (event), 2015, pages 111–123

[Jurdi and Abdo 2023] JURDI, Wissam A. ; ABDO, Jacques B.: *Github Repository: Optimizing Recommendations: A Contemporary Networks-inspired Approach*. June 2023. – URL https://github.com/wissamjur/serendipity

[Jurdi et al. 2021] JURDI, Wissam A. ; ABDO, Jacques B. ; DEMERJIAN, Jacques ; MAKHOUL, Abdallah: **"Critique on natural noise in recommender systems"**. In *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15 (2021), number 5, pages 1–30

[Jurdi et al. 2022] JURDI, Wissam A. ; ABDO, Jacques B. ; DEMERJIAN, Jacques ; MAKHOUL, Abdallah: **"Group Validation in Recommender Systems: Framework for Multi-layer Performance Evaluation"**. In *arXiv preprint arXiv:2207.09320* (2022)

[Kaminskas and Bridge 2016] KAMINSKAS, Marius ; BRIDGE, Derek: **"Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems"**. In *ACM Transactions on Interactive Intelligent Systems (TiiS)* 7 (2016), number 1, pages 1–42

[Kawamae 2010] KAWAMAE, Noriaki: **"Serendipitous recommendations via innovators"**. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 2010, pages 218–225

[Kluver et al. 2012] KLUVER, Daniel ; NGUYEN, Tien T. ; EKSTRAND, Michael ; SEN, Shilad ; RIEDL, John: **"How many bits per rating?"**. In *Proceedings of the sixth ACM conference on Recommender systems*, 2012, pages 99–106

[Knijnenburg et al. 2012] KNIJNENBURG, Bart P. ; WILLEMSEN, Martijn C. ; GANTNER, Zeno ; SONCU, Hakan ; NEWELL, Chris: **"Explaining the user experience of recommender systems"**. In *User modeling and user-adapted interaction* 22 (2012), number 4, pages 441–504

[Koren et al. 2009] KOREN, Yehuda ; BELL, Robert ; VOLINSKY, Chris: **"Matrix factorization techniques for recommender systems"**. In *Computer* 42 (2009), number 8, pages 30–37

[Kotkov 2018] KOTKOV, Denis: **"Serendipity in recommender systems"**. In *Jyväskylä studies in computing* (2018), number 281

[Kotkov et al. 2023] KOTKOV, Denis ; MEDLAR, Alan ; GLOWACKA, Dorota: **"Rethinking Serendipity in Recommender Systems"**. In *Proceedings of the 2023 Conference on Human Information Interaction and Retrieval*, 2023, pages 383–387

[Kotkov et al. 2020] KOTKOV, Denis ; VEIJALAINEN, Jari ; WANG, Shuaiqiang: **"How does serendipity affect diversity in recommender systems? A serendipity-oriented greedy algorithm"**. In *Computing* 102 (2020), pages 393–411

[Kotkov et al. 2016] KOTKOV, Denis ; WANG, Shuaiqiang ; VEIJALAINEN, Jari: **"A survey of serendipity in recommender systems"**. In *Knowledge-Based Systems* 111 (2016), pages 180–192

[Kunaver and Požrl 2017] KUNAVER, Matevž ; POŽRL, Tomaž: **"Diversity in recommender systems–A survey"**. In *Knowledge-based systems* 123 (2017), pages 154–162

[Lam et al. 2008] LAM, Xuan N. ; VU, Thuc ; LE, Trong D. ; DUONG, Anh D.: **"Addressing cold-start problem in recommendation systems"**. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, 2008, pages 208–211

[Latha and Nadarajan 2015] LATHA, R ; NADARAJAN, R: **"Ranking based approach for noise handling in recommender systems"**. In *Multimedia Communications, Services and Security: 8th International Conference, MCSS 2015, Kraków, Poland, November 24, 2015. Proceedings 8* Springer (event), 2015, pages 46–58

[Li et al. 2013] LI, Bin ; CHEN, Ling ; ZHU, Xingquan ; ZHANG, Chengqi: **"Noisy but non-malicious user detection in social recommender systems"**. In *World Wide Web* 16 (2013), pages 677–699

[Li et al. 2016] LI, Bo ; WANG, Yining ; SINGH, Aarti ; VOROBEYCHIK, Yevgeniy: **"Data poisoning attacks on factorization-based collaborative filtering"**. In *Advances in neural information processing systems* 29 (2016)

[Li et al. 2019] LI, Dongsheng ; CHEN, Chao ; GONG, Zhilin ; LU, Tun ; CHU, Stephen M. ; GU, Ning: **"Collaborative filtering with noisy ratings"**. In *Proceedings of the 2019 SIAM International Conference on Data Mining* SIAM (event), 2019, pages 747–755

[Liu et al. 2014] LIU, Haishan ; GOYAL, Anuj ; WALKER, Trevor ; BHASIN, Anmol: **"Improving the discriminative power of inferred content information using segmented virtual profile"**. In *Proceedings of the 8th ACM Conference on Recommender systems*, 2014, pages 97–104

[Logesh et al. 2020] LOGESH, R ; SUBRAMANIYASWAMY, V ; MALATHI, D ; SIVARAMAKRISHNAN, N ; VIJAYAKUMAR, Varadarajan: **"Enhancing recommendation stability of collaborative filtering recommender system through bio-inspired clustering ensemble method"**. In *Neural Computing and Applications* 32 (2020), number 7, pages 2141–2164

[López et al. 2013] LÓPEZ, Victoria ; FERNÁNDEZ, Alberto ; GARCÍA, Salvador ; PALADE, Vasile ; HERRERA, Francisco: **"An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics"**. In *Information sciences* 250 (2013), pages 113–141

[Luo et al. 2023] LUO, Chenhong ; WANG, Yong ; LI, Bo ; LIU, Hanyang ; WANG, Pengyu ; ZHANG, Leo Y.: **"An Efficient Approach to Manage Natural Noises in Recommender Systems"**. In *Algorithms* 16 (2023), number 5, pages 228

[Luo 2018] LUO, S: *Introduction to recommender system: Approaches of collaborative filtering: Nearest neighborhood and matrix factorization*. 2018

[Ma et al. 2023] MA, Tie-min ; WANG, Xue ; ZHOU, Fu-cai ; WANG, Shuang: **"Research on diversity and accuracy of the recommendation system based on multi-objective optimization"**. In *Neural Computing and Applications* 35 (2023), number 7, pages 5155–5163

[Macdonald 2021] MACDONALD, Craig: **"The Simpson's Paradox in the offline evaluation of recommendation systems"**. In *ACM Transactions on Information Systems* (2021)

[Mansur et al. 2017] MANSUR, Farhin ; PATEL, Vibha ; PATEL, Mihir: **"A review on recommender systems"**. In *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)* IEEE (event), 2017, pages 1–6

[Martínez et al. 2016] MARTÍNEZ, Luis ; CASTRO, Jorge ; YERA, Raciel: **"Managing natural noise in recommender systems"**. In *Theory and Practice of Natural Computing: 5th International Conference, TPNC 2016, Sendai, Japan, December 12-13, 2016, Proceedings 5* Springer (event), 2016, pages 3–17

[McMahan et al. 2013] MCMAHAN, H B. ; HOLT, Gary ; SCULLEY, David ; YOUNG, Michael ; EBNER, Dietmar ; GRADY, Julian ; NIE, Lan ; PHILLIPS, Todd ; DAVYDOV, Eugene ; GOLOVIN, Daniel ; OTHERS: **"Ad click prediction: a view from the trenches"**. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pages 1222–1230

[McNee et al. 2006] MCNEE, Sean M. ; RIEDL, John ; KONSTAN, Joseph A.: **"Being accurate is not enough: how accuracy metrics have hurt recommender systems"**. In *CHI'06 extended abstracts on Human factors in computing systems*, 2006, pages 1097–1101

[Melinat et al. 2014] MELINAT, Peter ; KREUZKAM, Tolja ; STAMER, Dirk: **"Information overload: a systematic literature review"**. In *Perspectives in Business Informatics Research: 13th International Conference, BIR 2014, Lund, Sweden, September 22-24, 2014. Proceedings 13* Springer (event), 2014, pages 72–86

[Meyer] MEYER, S.: *Understanding the COVID-19 Effect on Online Shopping Behavior*. – URL https://www.bigcommerce.com/blog/covid-19-ecommerce/

[Mnih and Salakhutdinov 2007] MNIH, Andriy ; SALAKHUTDINOV, Russ R.: **"Probabilistic matrix factorization"**. In *Advances in neural information processing systems* 20 (2007)

[Müllner 2011] MÜLLNER, Daniel: **"Modern hierarchical, agglomerative clustering algorithms"**. In *arXiv preprint arXiv:1109.2378* (2011)

[Murphy et al. 2006] MURPHY, Kevin P. ; OTHERS: **"Naive bayes classifiers"**. In *University of British Columbia* 18 (2006), number 60, pages 1–8

[Nabizadeh et al. 2015] NABIZADEH, AmirHossein ; JORGE, Alípio ; LEAL, José P.: **"Long term goal oriented recommender systems"**. In *11th International Conference on Web Information Systems and Technologies* (2015)

[Neill 2012] NEILL, Daniel B.: **"Fast subset scan for spatial pattern detection"**. In *Journal of the Royal Statistical Society Series B: Statistical Methodology* 74 (2012), number 2, pages 337–360

[Neill 2017] NEILL, Daniel B.: *Subset Scanning for Event and Pattern Detection.* 2017

[Nguyen et al. 2018] NGUYEN, Tien T. ; MAXWELL HARPER, F ; TERVEEN, Loren ; KONSTAN, Joseph A.: **"User personality and user satisfaction with recommender systems"**. In *Information Systems Frontiers* 20 (2018), number 6, pages 1173–1189

[O'Mahony et al. 2006] O'MAHONY, Michael P. ; HURLEY, Neil J. ; SILVESTRE, Guénolé CM: **"Detecting noise in recommender system databases"**. In *Proceedings of the 11th international conference on Intelligent user interfaces*, 2006, pages 109–115

[Onuma et al. 2009] ONUMA, Kensuke ; TONG, Hanghang ; FALOUTSOS, Christos: **"Tangent: a novel,'surprise me', recommendation algorithm"**. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pages 657–666

[Ovaisi et al. 2022] OVAISI, Zohreh ; HEINECKE, Shelby ; LI, Jia ; ZHANG, Yongfeng ; ZHELEVA, Elena ; XIONG, Caiming: **"RGRecSys: A Toolkit for Robustness Evaluation of Recommender Systems"**. In *arXiv preprint arXiv:2201.04399* (2022)

[Panagiotis 2011] PANAGIOTIS, Adamopoulos: **"On unexpectedness in recommender systems: Or how to expect the unexpected"**. In *DiveRS 2011: Proceedings of the Workshop on Novelty and Diversity in Recommender Systems (DiveRS 2011), at the 5th ACM International Conference on Recommender Systems (RecSys 2011)*, 2011, pages 11–18

[Parapar and Radlinski 2021] PARAPAR, Javier ; RADLINSKI, Filip: **"Towards Unified Metrics for Accuracy and Diversity for Recommender Systems"**. In *Fifteenth ACM Conference on Recommender Systems*, 2021, pages 75–84

[Patra et al. 2015] PATRA, Bidyut K. ; LAUNONEN, Raimo ; OLLIKAINEN, Ville ; NANDI, Sukumar: **"A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data"**. In *Knowledge-Based Systems* 82 (2015), pages 163–177

[Pham and Jung 2013] PHAM, Hau X. ; JUNG, Jason J.: **"Preference-based user rating correction process for interactive recommendation systems"**. In *Multimedia tools and applications* 65 (2013), pages 119–132

[Pham et al. 2012] PHAM, Xuan H. ; JUNG, Jason J. ; NGUYEN, Ngoc-Thanh: **"Integrating multiple experts for correction process in interactive recommendation systems"**. In *Computational Collective Intelligence. Technologies and Applications: 4th International Conference, ICCCI 2012, Ho Chi Minh City, Vietnam, November 28-30, 2012, Proceedings, Part I 4* Springer (event), 2012, pages 31–40

[Pourhabibi et al. 2020] POURHABIBI, Tahereh ; ONG, Kok-Leong ; KAM, Booi H. ; BOO, Yee L.: **"Fraud detection: A systematic literature review of graph-based anomaly detection approaches"**. In *Decision Support Systems* 133 (2020), pages 113303. – URL https://www.sciencedirect.com/science/article/pii/S0167923620300580. – ISSN 0167-9236. DOI: https://doi.org/10.1016/j.dss.2020.113303

[Pramod 2023] PRAMOD, Dhanya: **"Privacy-preserving techniques in recommender systems: state-of-the-art review and future research agenda"**. In *Data Technologies and Applications* 57 (2023), number 1, pages 32–55

[Ramakrishnan et al. 2001] RAMAKRISHNAN, Naren ; KELLER, Benjamin J. ; MIRZA, Batul J. ; GRAMA, Ananth Y. ; KARYPIS, George: **"When being weak is brave: Privacy in recommender systems"**. In *arXiv preprint cs/0105028* (2001)

[Rendle et al. 2012] RENDLE, Steffen ; FREUDENTHALER, Christoph ; GANTNER, Zeno ; SCHMIDT-THIEME, Lars: **"BPR: Bayesian personalized ranking from implicit feedback"**. In *arXiv preprint arXiv:1205.2618* (2012)

[Reviglio 2019] REVIGLIO, Urbano: **"Serendipity as an emerging design principle of the infosphere: challenges and opportunities"**. In *Ethics and Information Technology* 21 (2019), number 2, pages 151–166

[Ricci et al. 2010] RICCI, Francesco ; ROKACH, Lior ; SHAPIRA, Bracha: . Volume 1-35. pages 1–35. In *Recommender Systems Handbook* Volume 1-35, 10 2010, DOI: 10.1007/978-0-387-85820-3_1

[Ricci et al. 2021] RICCI, Francesco ; ROKACH, Lior ; SHAPIRA, Bracha: **"Recommender systems: Techniques, applications, and challenges"**. In *Recommender Systems Handbook* (2021), pages 1–35

[Richardson et al. 2003] RICHARDSON, Matthew ; AGRAWAL, Rakesh ; DOMINGOS, Pedro: **"Trust management for the semantic web"**. In *International semantic Web conference* Springer (event), 2003, pages 351–368

[Robertson 2004] ROBERTSON, Stephen: **"Understanding inverse document frequency: on theoretical arguments for IDF"**. In *Journal of documentation* 60 (2004), number 5, pages 503–520

[Roy and Dutta 2022] ROY, Deepjyoti ; DUTTA, Mala: **"A systematic review and research perspective on recommender systems"**. In *Journal of Big Data* 9 (2022), number 1, pages 59

[Saia et al. 2014] SAIA, Roberto ; BORATTO, Ludovico ; CARTA, Salvatore: **"Semantic coherence-based user profile modeling in the recommender systems context"**. In *International Conference on Knowledge Discovery and Information Retrieval* Volume 2 SciTePress (event), 2014, pages 154–161

[Saia et al. 2016] SAIA, Roberto ; BORATTO, Ludovico ; CARTA, Salvatore: **"A semantic approach to remove incoherent items from a user profile and improve the accuracy of a recommender system"**. In *Journal of Intelligent Information Systems* 47 (2016), pages 111–134

[Said and Bellogín 2018] SAID, Alan ; BELLOGÍN, Alejandro: **"Coherence and inconsistencies in rating behavior: estimating the magic barrier of recommender systems"**. In *User Modeling and User-Adapted Interaction* 28 (2018), number 2, pages 97–125

[Said et al. 2012a] SAID, Alan ; JAIN, Brijnesh J. ; NARR, Sascha ; PLUMBAUM, Till: **"Users and noise: The magic barrier of recommender systems"**. In *User Modeling, Adaptation, and Personalization: 20th International Conference, UMAP 2012, Montreal, Canada, July 16-20, 2012. Proceedings 20* Springer (event), 2012, pages 237–248

[Said et al. 2012b] SAID, Alan ; JAIN, Brijnesh J. ; NARR, Sascha ; PLUMBAUM, Till ; ALBAYRAK, Sahin ; SCHEEL, Christian: **"Estimating the magic barrier of recommender systems: a user study"**. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 2012, pages 1061–1062

[Schoenauer] SCHOENAUER, C.: *Buying behavior after COVID-19: e-commerce boom will remain*. – URL https://www.the-future-of-commerce.com

[Scholz et al. 2017] SCHOLZ, Michael ; DORNER, Verena ; SCHRYEN, Guido ; BENLIAN, Alexander: **"A configuration-based recommender system for supporting e-commerce decisions"**. In *European Journal of Operational Research* 259 (2017), number 1, pages 205–215

[Sculley 2010] SCULLEY, David: **"Web-scale k-means clustering"**. In *Proceedings of the 19th international conference on World wide web*, 2010, pages 1177–1178

[Serrano 2023] SERRANO, Stephan: **"Personalized Product Recommendations Tactics for Profits"**. Brilliance, 2023. – Research Report. – URL https://www.barilliance.com/personalized-product-recommendations-stats/. .

[Shokeen and Rana 2020] SHOKEEN, Jyoti ; RANA, Chhavi: **"Social recommender systems: techniques, domains, metrics, datasets and future scope"**. In *Journal of Intelligent Information Systems* 54 (2020), number 3, pages 633–667

[Si and Li 2020] SI, Mingdan ; LI, Qingshan: **"Shilling attacks against collaborative recommender systems: a review"**. In *Artificial Intelligence Review* 53 (2020), pages 291–319

[Simon Funk 2006] SIMON FUNK: *Netflix Update: Try This at Home*. 2006. – URL https://sifter.org/~simon/journal/20061211.html. – [Online; accessed 16-May-2022]

[Singh and Joachims 2018] SINGH, Ashudeep ; JOACHIMS, Thorsten: **"Fairness of exposure in rankings"**. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pages 2219–2228

[Skovhøj 2022] SKOVHØJ, Fiona Z.: **"The Power of Product Recommendations: 30 Must-Know Statistics"**. Salesforce, 2022. – Research Report. – URL https://www.clerk.io/blog/product-recommendations-statistics-2022. .

[Smaros and Falck a] SMAROS, J. ; FALCK, M.: *Global Mobile Consumer Survey, US Edition*. – URL https://www2.deloitte.com/tr/en/pages/technology-media-and-telecommunications/articles/global-mobile-consumer-survey-us-edition.html

[Smaros and Falck b] SMAROS, J. ; FALCK, M.: *The New Normal in Ecommerce after COVID-19*. – URL https://www.relexsolutions.com/resources/the-new-normal-in-ecommerce-after-covid-19/

[Sullivan and Feinn 2012] SULLIVAN, Gail M. ; FEINN, Richard: **"Using effect size—or why the P value is not enough"**. In *Journal of graduate medical education* 4 (2012), number 3, pages 279–282

[Sun et al. 2013] SUN, Tao ; ZHANG, Ming ; MEI, Qiaozhu: **"Unexpected relevance: An empirical study of serendipity in retweets"**. In *Proceedings of the International AAAI Conference on Web and Social Media* Volume 7, 2013, pages 592–601

[Sun et al. 2020] SUN, Zhu ; YU, Di ; FANG, Hui ; YANG, Jie ; QU, Xinghua ; ZHANG, Jie ; GENG, Cong: **"Are we evaluating rigorously? benchmarking recommendation for reproducible evaluation and fair comparison"**. In *Fourteenth ACM conference on recommender systems*, 2020, pages 23–32

[Tamm et al. 2021] TAMM, Yan-Martin ; DAMDINOV, Rinchin ; VASILEV, Alexey: **"Quality Metrics in Recommender Systems: Do We Calculate Metrics Consistently?"**. In *Fifteenth ACM Conference on Recommender Systems*, 2021, pages 708–713

[Toledo et al. 2013] TOLEDO, Raciel Y. ; LÓPEZ, Luis M. ; MOTA, Yailé C.: **"Managing natural noise in collaborative recommender systems"**. In *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)* IEEE (event), 2013, pages 872–877

[Toledo et al. 2015] TOLEDO, Raciel Y. ; MOTA, Yailé C. ; MARTÍNEZ, Luis: **"Correcting noisy ratings in collaborative recommender systems"**. In *Knowledge-Based Systems* 76 (2015), pages 96–108

[Tong et al. 2018] TONG, Chao ; LIAN, Yu ; NIU, Jianwei ; LONG, Xiang: **"A novel rating prediction method based on user relationship and natural noise"**. In *Multimedia Tools and Applications* 77 (2018), pages 4171–4186

[Uher 2018] UHER, Jana: **"Quantitative data from rating scales: An epistemological and methodological enquiry"**. In *Frontiers in psychology* 9 (2018), pages 2599

[Vaishali et al. 2015] VAISHALI, S ; RAO, K K. ; RAO, GV S.: **"A review on noise reduction methods for brain MRI images"**. In *2015 International Conference on Signal Processing and Communication Engineering Systems* IEEE (event), 2015, pages 363–365

[Valcarce et al. 2018] VALCARCE, Daniel ; BELLOGÍN, Alejandro ; PARAPAR, Javier ; CASTELLS, Pablo: **"On the robustness and discriminative power of information retrieval metrics for top-N recommendation"**. In *Proceedings of the 12th ACM conference on recommender systems*, 2018, pages 260–268

[Vargas 2014] VARGAS, Saúl: **"Novelty and diversity enhancement and evaluation in recommender systems and information retrieval"**. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 2014, pages 1281–1281

[Vargas 2015] VARGAS, Saúl: *Novelty and diversity evaluation and enhancement in recommender systems*, PhD thesis, Universidad Autónoma de Madrid, Spain, PhD Thesis, 2015

[Wang et al. 2019] WANG, Xiang ; HE, Xiangnan ; WANG, Meng ; FENG, Fuli ; CHUA, Tat-Seng: **"Neural graph collaborative filtering"**. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 2019, pages 165–174

[Wang et al. 2021] WANG, Yingheng ; MIN, Yaosen ; CHEN, Xin ; WU, Ji: **"Multi-view graph contrastive representation learning for drug-drug interaction prediction"**. In *Proceedings of the Web Conference 2021*, 2021, pages 2921–2933

[Wikipedia contributors 2022a] WIKIPEDIA CONTRIBUTORS: *Effect size — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Effect_size&oldid=1102056368. 2022. – [Online; accessed 14-August-2022]

[Wikipedia contributors 2022b] WIKIPEDIA CONTRIBUTORS: *Welch's t-test — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Welch%27s_t-test&oldid=1091289937. 2022. – [Online; accessed 14-August-2022]

[Yamaba et al. 2013] YAMABA, Hisaaki ; TANOUE, Michihito ; TAKATSUKA, Kayoko ; OKAZAKI, Naonobu ; TOMITA, Shigeyuki: **"On a serendipity-oriented recommender system based on folksonomy and its evaluation"**. In *Procedia Computer Science* 22 (2013), pages 276–284

[Yan 2020] YAN, Eugene: **"Serendipity: Accuracy's unpopular best friend in Recommender Systems"**. In *Towards Data Science, April* (2020)

[Yannam et al. 2023] YANNAM, V R. ; KUMAR, Jitendra ; BABU, Korra S. ; PATRA, Bidyut K.: **"Enhancing the accuracy of group recommendation using slope one"**. In *The Journal of Supercomputing* 79 (2023), number 1, pages 499–540

[Yera et al. 2016] YERA, Raciel ; CASTRO, Jorge ; MARTÍNEZ, Luis: **"A fuzzy model for managing natural noise in recommender systems"**. In *Applied Soft Computing* 40 (2016), pages 187–198

[Yera et al. 2020] YERA, Raciel ; CASTRO, Jorge ; MARTINEZ, Luis: **"Natural noise management in recommender systems using fuzzy tools"**. In *Computational Intelligence for Semantic Knowledge Management: New Perspectives for Designing and Organizing Information Systems* (2020), pages 1–24

[Yera and Martinez 2017] YERA, Raciel ; MARTINEZ, Luis: **"Fuzzy tools in recommender systems: A survey"**. In *International Journal of Computational Intelligence Systems* 10 (2017), number 1, pages 776

[Yu et al. 2016] YU, Penghua ; LIN, Lanfen ; YAO, Yuangang: **"A novel framework to process the quantity and quality of user behavior data in recommender systems"**. In *Web-Age Information Management: 17th International Conference, WAIM 2016, Nanchang, China, June 3-5, 2016, Proceedings, Part I 17* Springer (event), 2016, pages 231–243

[Zhang et al. 2019a] ZHANG, Feng ; LEE, Victor E. ; JIN, Ruoming ; GARG, Saurabh ; CHOO, Kim-Kwang R. ; MAASBERG, Michele ; DONG, Lijun ; CHENG, Chi: **"Privacy-aware smart city: A case study in collaborative filtering recommender systems"**. In *Journal of Parallel and Distributed Computing* 127 (2019), pages 145–159

[Zhang et al. 2019b] ZHANG, Shuai ; YAO, Lina ; SUN, Aixin ; TAY, Yi: **"Deep learning based recommender system: A survey and new perspectives"**. In *ACM computing surveys (CSUR)* 52 (2019), number 1, pages 1–38

[Zhang et al. 2020] ZHANG, Yang ; FENG, Fuli ; WANG, Chenxu ; HE, Xiangnan ; WANG, Meng ; LI, Yan ; ZHANG, Yongdong: **"How to retrain recommender system? A sequential meta-learning method"**. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pages 1479–1488

[Zhang et al. 2012] ZHANG, Yuan C. ; SEAGHDHA, Diarmuid O. ; QUERCIA, Daniele ; JAMBOR, Tamas: **"Auralist: introducing serendipity into music recommendation"**. In *Proceedings of the fifth ACM international conference on Web search and data mining*, 2012, pages 13–22

[Zhao et al. 2018] ZHAO, Qian ; HARPER, F M. ; ADOMAVICIUS, Gediminas ; KONSTAN, Joseph A.: **"Explicit or implicit feedback? Engagement or satisfaction? A field experiment on machine-learning-based recommender systems"**. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, 2018, pages 1331–1340

[Zhao et al. 2021] ZHAO, Wayne X. ; MU, Shanlei ; HOU, Yupeng ; LIN, Zihan ; CHEN, Yushuo ; PAN, Xingyu ; LI, Kaiyuan ; LU, Yujie ; WANG, Hui ; TIAN, Changxin ; OTHERS: **"Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms"**. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pages 4653–4664

[Zhou et al. 2010] ZHOU, Tao ; KUSCSIK, Zoltán ; LIU, Jian-Guo ; MEDO, Matúš ; WAKELING, Joseph R. ; ZHANG, Yi-Cheng: **"Solving the apparent diversity-accuracy dilemma of recommender systems"**. In *Proceedings of the National Academy of Sciences* 107 (2010), number 10, pages 4511–4515

[Zhou et al. 2015] ZHOU, Wei ; WEN, Junhao ; KOH, Yun S. ; XIONG, Qingyu ; GAO, Min ; DOBBIE, Gillian ; ALAM, Shafiq: **"Shilling attacks detection in recommender systems based on target item analysis"**. In *PloS one* 10 (2015), number 7, pages e0130968

[Ziarani and Ravanmehr 2021] ZIARANI, Reza J. ; RAVANMEHR, Reza: **"Serendipity in recommender systems: a systematic literature review"**. In *Journal of Computer Science and Technology* 36 (2021), pages 375–396

List of Figures

List of Tables

VIII

APPENDIX

A

List of Contributions

A.1/ Jounrals

[1] Wissam Al Jurdi, Jacques Bou Abdo, Jacques Demerjian, and Abdallah Makhoul. "Group Validation in Recommender Systems: Framework for Multi-layer Performance Evaluation", doi/10.1145/3640820, ACM Transactions on Recommender Systems, TORS 2024.

[2] Wissam Al Jurdi, Jacques Bou Abdo, Jacques Demerjian, and Abdallah Makhoul "Critique on natural noise in recommender systems." ACM Transactions on Knowledge Discovery from Data, TKDD 15, no. 5 (2021): 1-30.

A.2/ Conferences

[1] Wissam Al Jurdi, Jacques Bou Abdo, Jacques Demerjian, and Abdallah Makhoul. "Optimizing Recommendations: A Contemporary Networks-inspired Approach." – Accepted, International Conference on Complex Networks and Their Applications, CNA 2023.

[2] Wissam Al Jurdi, Jacques Bou Abdo, Jacques Demerjian, and Abdallah Makhoul "Strategic Attacks on Recommender Systems: An Obfuscation Scenario." IEEE/ACS 19th International Conference on Computer Systems and Applications, AICCSA pp. 1-8. IEEE, 2022.

[3] Wissam Al Jurdi, Chady Abou Jaoude, Miriam El Khoury Badran, Jacques Bou Abdo, Jacques Demerjian, and Abdallah Makhoul "SCCF Parameter and Similarity Measure Optimization and Evaluation." Knowledge Science, Engineering and Management: 12th International Conference, KSEM 2019.

[4] Badran, Miriam El Khoury, Jacques Bou Abdo, Wissam Al Jurdi, and Jacques Demerjian "Adaptive Serendipity for Recommender Systems: Let It Find You." ICAART

(2), pp. 739-745. 2019.

[5] Wissam Al Jurdi, Miriam El Khoury Badran, Chady Abou Jaoude, Jacques Bou Abdo, Jacques Demerjian, and Abdallah Makhoul ”Serendipity-aware noise detection system for recommender systems.” Proceedings of the Information and Knowledge Engineering, IKE 2019.

Document generated with LaTeX and:
the LaTeX style for PhD Thesis created by S. Galland — https://github.com/gallandarakhneorg/tex-templates
the tex-upmethodology package suite — http://www.arakhne.org/tex-upmethodology/

école doctorale **sciences pour l'ingénieur et microtechniques**

Title: Towards Next Generation Recommender Systems through Generic Data Quality

Keywords: recommender systems, serendipity, dataset noise, attacks, privacy, evaluation, model validation, data clustering

Abstract:

Recommender systems are essential for filtering online information and delivering personalized content, reducing users' effort to find relevant information. They can be content-based, collaborative, or hybrid, each with a unique recommendation approach. These systems are crucial in various fields, including e-commerce, where they help customers find pertinent products, enhancing user experience and increasing sales. A significant aspect of these systems is the concept of unexpectedness, which involves discovering new and surprising items. While improving user engagement and experience, this feature is complex and subjective, requiring a deep understanding of serendipitous recommendations for its measurement and optimization. Natural noise, an unpredictable data variation, can influence serendipity in recommender systems. It can introduce diversity and unexpectedness in recommendations, leading to pleasant surprises. However, it can also reduce recommendation relevance, causing user frustration. Therefore, designing systems that balance natural noise and serendipity is crucial. Inconsistent user information due to natural noise can negatively impact recommender systems, leading to lower-quality recommendations. Current evaluation methods often overlook critical user-oriented factors, challenging noise detection. To provide powerful recommendations, it's important to consider diverse user profiles, eliminate noise in datasets, and effectively present users with relevant content from vast data catalogs. This thesis emphasizes the role of serendipity in enhancing recommender systems and preventing filter bubbles. It proposes serendipity-aware techniques to manage noise, identifies algorithm flaws, suggests a user-centric evaluation method, and proposes a community-based architecture for improved performance. It highlights the need for a system that balances serendipity and considers natural noise and other performance factors. The objectives, experiments, and tests aim to refine recommender systems and offer a versatile assessment approach.

Titre : Vers des Systèmes de Recommandation de Nouvelle Génération grâce à la Qualité des Données Génériques

Mots-clés : systèmes de recommandation, sérendipité, bruit des ensembles de données, attaques, confidentialité, évaluation, validation de modèles, regroupement de données

Résumé :

Les systèmes de recommandation sont essentiels pour filtrer les informations en ligne et fournir un contenu personnalisé, réduisant ainsi l'effort nécessaire pour trouver des informations pertinentes. Ils jouent un rôle crucial dans divers domaines, dont le commerce électronique, en aidant les clients à trouver des produits pertinents, améliorant l'expérience utilisateur et augmentant les ventes. Un aspect significatif de ces systèmes est le concept d'inattendu, qui implique la découverte d'éléments nouveaux et surprenants. Cependant, il est complexe et subjectif, nécessitant une compréhension approfondie des recommandations fortuites pour sa mesure et son optimisation. Le bruit naturel, une variation imprévisible des données, peut influencer la sérendipité dans les systèmes de recommandation. Il peut introduire de la diversité et de l'inattendu dans les recommandations, conduisant à des surprises agréables. Cependant, il peut également réduire la pertinence de la recommandation. Par conséquent, il est crucial de concevoir des systèmes qui équilibrent le bruit naturel et la sérendipité. Cette thèse souligne le rôle de la sérendipité dans l'amélioration des systèmes de recommandation et la prévention des bulles de filtre. Elle propose des techniques conscientes de la sérendipité pour gérer le bruit, identifie les défauts de l'algorithme, suggère une méthode d'évaluation centrée sur l'utilisateur, et propose une architecture basée sur la communauté pour une performance améliorée.

Université Bourgogne-Franche-Comté
32, avenue de l'Observatoire
25000 Besançon, France