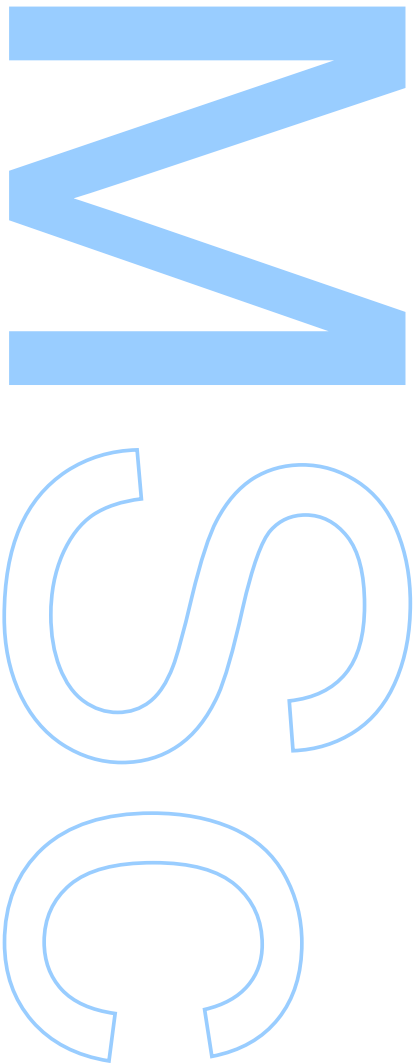
Measuring Forgetting in Stream-Based Recommender Systems

Klismam Pereira
Mestrado em Ciência de Dados
Departamento de Ciência de Computadores
2022

Orientador
Dr. João Vinagre, Faculdade de Ciências



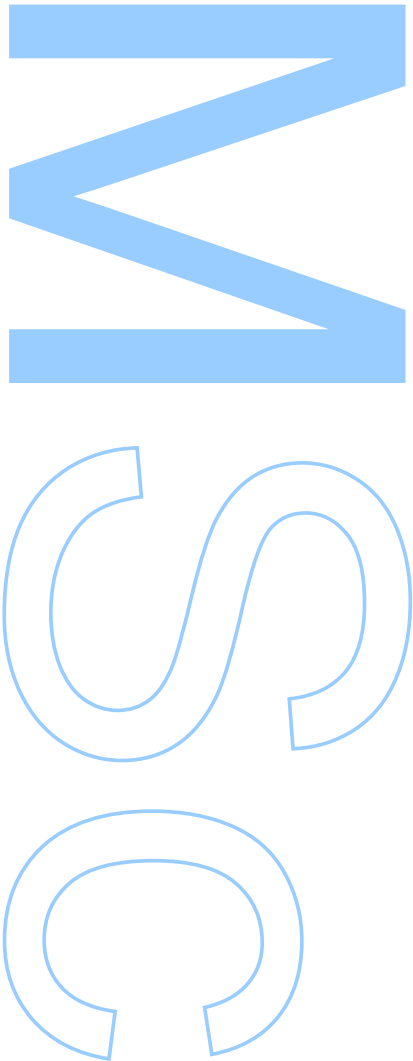
U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, _____/_____/_____



UNIVERSIDADE DO PORTO

MASTERS THESIS

Measuring Forgetting in Stream-Based Recommender Systems

Author:
Klismam PEREIRA

Supervisor:
João VINAGRE

A thesis submitted in fulfilment of the requirements
for the degree of MSc. Data Science

at the

Faculdade de Ciências da Universidade do Porto
Departamento de Ciência de Computadores

December 12, 2022

Declaração de Honra

Eu, Klismam Franciosi Pereira, inscrito no Mestrado em Ciência de Dados da Faculdade de Ciências da Universidade do Porto declaro, nos termos do disposto na alínea a) do artigo 14.º do Código Ético de Conduta Académica da U.Porto, que o conteúdo da presente dissertação reflete as perspetivas, o trabalho de investigação e as minhas interpretações no momento da sua entrega.

Ao entregar esta dissertação, declaro, ainda, que a mesma é resultado do meu próprio trabalho de investigação e contém contributos que não foram utilizados previamente noutros trabalhos apresentados a esta ou outra instituição.

Mais declaro que todas as referências a outros autores respeitam escrupulosamente as regras da atribuição, encontrando-se devidamente citadas no corpo do texto e identificadas na secção de referências bibliográficas. Não são divulgados na presente dissertação quaisquer conteúdos cuja reprodução esteja vedada por direitos de autor.

Tenho consciência de que a prática de plágio e auto-plágio constitui um ilícito académico.

Klismam Franciosi Pereira

Porto, 30 de Setembro de 2022

Acknowledgements

I am grateful to my partner, with whom I share this fleeting journey with affection, patience and happiness.

I am grateful for my supervisor and professors, whose guidance and affinity were essential for this expedition.

I am grateful for all the friends I have made along the way, the ones who are still here and all who left.

I am grateful to strangers in whom the virtues of empathy and humanity shine.

Most of all, I am grateful for every show of support and every kind word - being human before anything else is a beautiful state of being.

UNIVERSIDADE DO PORTO

Abstract

Faculdade de Ciências da Universidade do Porto
Departamento de Ciência de Computadores

MSc. Data Science

Measuring Forgetting in Stream-Based Recommender Systems

by Klismam PEREIRA

Lifelong Learning is an active and growing research topic. Nevertheless, machine learning algorithms that learn continually often suffer from forgetting learned information and may struggle to perform in previous tasks. Stream-based recommender systems are among the methods liable to suffer from forgetting. However, most works assessing information transfer are related to image classification tasks with incremental neural networks. This work presents a framework for assessing information transfer in stream-based recommendation methods. Stream-based methods with different learning schemes are assessed in real-life datasets from the e-commerce, music, and movie streaming domains. Noteworthy results indicate that the similarity-based method can better retain helpful information from past examples while showing increased forward transfer capabilities; overall, the further a model learns, the more it forgets past information; performance is often higher for examples in the near future than otherwise; finally, the information transfer metrics should be used together with recall heatmaps. Moreover, results and limitations are discussed, giving way to several possible future research topics.

UNIVERSIDADE DO PORTO

Resumo

Faculdade de Ciências da Universidade do Porto
Departamento de Ciência de Computadores

Mestrado em Ciência de Dados

Measuring Forgetting in Stream-Based Recommender Systems

por Klismam PEREIRA

Lifelong Learning é um tópico de pesquisa ativo e crescente. No entanto, algoritmos de aprendizado de máquina que aprendem continuamente comumente sofrem com o esquecimento das informações aprendidas e podem ter dificuldades para desempenhar razoavelmente em tarefas anteriores. Sistemas de recomendação baseados em fluxos de dados estão entre os métodos passíveis de sofrer com o esquecimento. No entanto, a maioria dos trabalhos que avaliam a transferência de informações está relacionada a tarefas de classificação de imagens com redes neurais incrementais. Este trabalho apresenta uma estratégia para avaliar a transferência de informações em métodos de recomendação baseados em fluxos de dados. Métodos baseados em fluxos de dados com diferentes esquemas de aprendizado são avaliados em conjuntos de dados dos domínios de *e-commerce*, música e *streaming* de filmes. Resultados indicam que o método baseado em similaridade é capaz de reter melhor as informações úteis de exemplos anteriores, ao mesmo tempo em que mostra maior capacidade de transferência de informações para o futuro; em geral, quanto mais um modelo aprende, mais ele esquece informações passadas; o desempenho costuma ser maior para exemplos em um futuro próximo do que no distante; e finalmente, as métricas de transferência de informações devem ser usadas em conjunto com mapas de calor de *recall*. Além disso, os resultados e limitações são discutidos, dando lugar a vários possíveis tópicos de pesquisa futura.

Contents

Declaração de Honra iii

Acknowledgements v

Abstract vii

Resumo ix

Contents xi

List of Figures xiii

List of Tables xv

Glossary xvii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Research Goals . . . 2
- 1.3 Contributions . . . 2
- 1.4 Dissertation Structure . . . 3

2 Background 5
- 2.1 Machine Learning . . . 5
- 2.2 Recommender Systems . . . 6
  - 2.2.1 Types of Feedback . . . 7
  - 2.2.2 Collaborative Filtering Recommender Systems . . . 8
  - 2.2.3 Content-Based Recommender Systems . . . 8
  - 2.2.4 Knowledge-Based Recommender Systems . . . 9
  - 2.2.5 Demographic Recommender Systems . . . 9
  - 2.2.6 Hybrid and Ensemble-Based Recommender Systems . . . 9
  - 2.2.7 Stream-based Recommender Systems . . . 9
  - 2.2.8 Evaluating Recommender Systems . . . 10
    - 2.2.8.1 Evaluating Stream-Based Recommender Systems . . . 12
- 2.3 Lifelong Machine Learning . . . 12
  - 2.3.1 Catastrophic Forgetting . . . 13

2.3.2 Lifelong Learning and Catastrophic Forgetting in Recommender Systems . . . 14

3 State of the Art **17**
3.1 Forgetting Assessment in Image Classification . . . 17
3.2 Forgetting Assessment in Recommendation . . . 21

4 Methodology **23**
4.1 Information Transfer Assessment Scheme . . . 23
4.1.1 Intervals and Holdouts . . . 25
4.1.2 Training and Evaluation . . . 25
4.1.3 Assessing Performance, Backward Transfer and Forward Transfer . . . 25
4.2 Datasets . . . 27
4.3 Stream-Based Recommendation Methods . . . 28

5 Results and Discussion **31**
5.1 Results per Dataset . . . 32
5.1.1 Amazon Kindle Store . . . 32
5.1.2 Amazon Digital Music . . . 33
5.1.3 Palco 2010 . . . 34
5.1.4 Movielens . . . 35
5.2 Results per Method . . . 36
5.2.1 ISGD, RAISGD and RSISGD . . . 36
5.2.2 BPRMF . . . 37
5.2.3 UKNN . . . 38
5.3 Limitations . . . 38

6 Conclusion **41**
6.1 Future Work . . . 42

A Amazon Kindle Recall@20 Heatmaps **45**

B Amazon Digital Music Recall@20 Heatmaps **49**

C Palco 2010 Recall@20 Heatmaps **53**

D Movielens Recall@20 Heatmaps **57**

Bibliography **61**

List of Figures

4.1 Methodology Scheme. The figure shows the steps taken to assess the performance of an incremental model in holdouts relative to training datasets separated by time intervals. E.g. intervals 1, 2, and 3 could represent data from January, February, and March. 1 - The dataset is separated into time-based intervals, from which interactions are held; 2 - A streaming recommender system learns from an interval, and; 3 - is assessed in all holdouts; 4 - The process repeats for all available intervals; 5 - Results are aggregated into a matrix, from which average performance, backward transfer and forward transfer can be computed. 24

4.2 Example results matrix from an experiment with $N = 3$ intervals. The left side shows intervals; the bottom shows holdouts. The matrix is populated by entries $R_{i,j}$ that represent the scores, for each holdout H_j, of model states trained on I_i and previous intervals I_k where $k < i$. 26

5.1 Results Summary. The plots show the mean diagonal (Diag), BWT, and FWT scores (in blue, orange, and green) obtained by the algorithms on each dataset. The horizontal axis represent the methods, and the vertical axis the score values. 32

A.1 ISGD recall@20 heatmap for Amazon Kindle dataset 45
A.2 RAISGD recall@20 heatmap for Amazon Kindle dataset 46
A.3 RSISGD recall@20 heatmap for Amazon Kindle dataset 46
A.4 BPRMF recall@20 heatmap for Amazon Kindle dataset 47
A.5 UKNN recall@20 heatmap for Amazon Kindle dataset 47

B.1 ISGD recall@20 heatmap for Amazon Digital Music dataset 49
B.2 RAISGD recall@20 heatmap for Amazon Digital Music dataset 50
B.3 RSISGD recall@20 heatmap for Amazon Digital Music dataset 50
B.4 BPRMF recall@20 heatmap for Amazon Digital Music dataset 51
B.5 UKNN recall@20 heatmap for Amazon Digital Music dataset 51

C.1 ISGD recall@20 heatmap for Palco 2010 dataset 53
C.2 RAISGD recall@20 heatmap for Palco 2010 dataset 54
C.3 RSISGD recall@20 heatmap for Palco 2010 dataset 54
C.4 BPRMF recall@20 heatmap for Palco 2010 dataset 55
C.5 UKNN recall@20 heatmap for Palco 2010 dataset 55

D.1 ISGD recall@20 heatmap for Movielens dataset 57
D.2 RAISGD recall@20 heatmap for Movielens dataset 58

D.3 RSISGD recall@20 heatmap for Movielens dataset 58
D.4 BPRMF recall@20 heatmap for Movielens dataset 59
D.5 UKNN recall@20 heatmap for Movielens dataset 59

List of Tables

4.1 Dataset Description 28
4.2 Datasets' Intervals and Holdouts 28

5.1 Results Summary. The table shows the mean diagonal (Diag), BWT, and FWT scores obtained by the algorithms in the four datasets. For each dataset, the symbols ↑ and ↓ represent the best and worst results for that score — the symbol ∗ represents an insufficient result for comparison, meaning the algorithm performed too poorly to allow a fair comparison. 31

Glossary

ADM	Amazon digital music
AKS	Amazon kindle store
ANN	artificial neural network
BPRMF	incremental bayesian personalized ranking matrix factorization
BWT	backward transfer
CF	collaborative filtering
DNN	deep neural network
FWT	forward transfer
GEM	gradient episodic memory
i.i.d.	independent and identically distributed
ISGD	incremental stochastic gradient descent
LML	lifelong machine learning
ML	machine learning
MLE	Movielens
MLP	multi-layer perceptron
P10	Palco 2010
RAISGD	recency adjusted incremental stochastic gradient descent
RSISGD	randomly sampled incremental stochastic gradient descent
UKNN	user k-nearest neighbors

Chapter 1

Introduction

1.1 Motivation

Catastrophic Forgetting is a phenomenon identified in the neural networks community that affects models that learn incrementally. The issue is intimately related to the stability-plasticity trade-off - model weights must be flexible enough to include new knowledge but sufficiently stable to retain learned information. In summary, information learned in the past is overridden (i.e. forgotten) when acquiring new knowledge, which can result in performance deterioration on previous tasks. It is an active and growing research topic due to the recent interest in *Lifelong Learning* (i.e. continual learning). In this paradigm, a model learns continually while retaining and accumulating knowledge, which is used to aid in learning new tasks.

While lifelong neural networks methods aim to learn and perform several tasks efficiently, most incremental recommendation methods focus on learning a single task, i.e. providing valuable recommendations based on users' past interactions. Thus, recommendation methods should model short and long-term user preferences to present meaningful suggestions. However, learning incrementally from the most recent interactions might favour dynamic user behaviour and lead to forgetting long-term preferences. Consequently, this effect can result in performance degradation and less valuable recommendations. The concern lies in comprehending the efficiency of transferring valuable past knowledge to future recommendations.

Stream-based recommender systems are designed to learn incrementally from *data streams*. Data streams are enormous amounts of potentially unbounded user feedback data that must be processed as fast as it arrives and only once. This class of methods

can successfully adapt to *concept drifts* (i.e. changes in data distribution), thus modelling users' short-term volatile preferences. On the other hand, this could lead to forgetting long-term preferences. Therefore, this work will investigate information transfer as users' behaviours are modelled through time. In this sense, information transfer encompasses the issue of retaining helpful information to model both past and future user preferences. This endeavour is critical to enabling Lifelong Learning for incremental recommender systems.

To the author's best knowledge, studies that undertake information transfer belong to the neural networks domain and focus on image classification, with fewer studies related to incremental neural networks-based recommender systems. While a lack of consensus on measuring forgetting is common to both tasks, a few studies propose evaluation benchmarks for measuring forgetting in image classification. At the same time, most methods designed to attenuate forgetting present relative performance efficiency instead of measuring it explicitly. Outside of the neural networks domain, no studies in which forgetting is assessed have been found.

1.2 Research Goals

To better understand information transfer in stream-based recommender systems, this project aspires to complete the following objectives:

- For the first time, develop a framework to standardize the evaluation of information transfer in incremental recommender system methods;
- With this framework, perform a comprehensive evaluation of information transfer in state-of-the-art methods over real-life datasets;
- Given the results, identify and characterize negative and positive information transfer phenomena in recommender systems.

1.3 Contributions

The main contribution of this work is a model-agnostic framework to assess information transfer in stream-based recommender systems. The framework assesses five stream-based methods with different learning procedures in four real-life datasets from the e-commerce, music, and movie streaming domains. Some of the notable results are:

- the similarity-based method is the top performer in information transfer;
- in general, but susceptible to nuances, the further a model learns, the more it forgets from past examples;
- performance is often higher for examples that are closest in the future than otherwise;
- the devised information transfer metrics should be used together with the recall heatmaps.

The work also reviews recent research on information transfer. Finally, it provides insightful discussions on the results, which lead to constructive remarks on the framework's limitations and possible future works.

1.4 Dissertation Structure

The work is organized as follows. Chapter 2 presents a brief background on Machine Learning, Recommender Systems, and Lifelong Machine Learning. Chapter 3 overviews state-of-the-art works on information transfer assessment in image classification and recommendation. Chapter 4 describes the information transfer assessment framework (the training and evaluation schemes, metrics and formulas), datasets, and stream-based recommendation methods. Chapter 5 presents the results, discusses them per dataset and method, and touches on the limitations encountered. Chapter 6 presents the conclusion, where remarks are made over the primary outcomes, and potential future lines of research are outlined. The appendices exhibit recall heatmaps for all conducted experiments.

Chapter 2

Background

2.1 Machine Learning

Machine learning is a multidisciplinary field that aims to build computer programs that automatically increase task performance with experience - in this context, learning means the algorithm's performance improves over some determined task given experience. A learning problem is then defined given the class of tasks, the performance metric to be improved, and the source of experience. Generally, the learning problem can be described in terms of two broader categories: *supervised* and *unsupervised* learning. [1, 2]

In the supervised learning setting, there is a set of features, called *inputs* or *predictors*, used to predict real-world phenomena expressed by an outcome measure, also called *outputs* or *responses*. The goal is to find a useful approximation to the underlying function that governs the real-world phenomena. The task at hand varies with the type of outcome measure. It is usually denoted as a *regression* task for quantitative outputs, such as predicting house prices from house and neighbourhood characteristics. For categorical outputs, the task is generally seen as *classification*, for example, classifying an e-mail as spam or not given the message's content. A model is built using machine learning algorithms from a *training set* of examples. The model is then used to predict the output of new unseen examples, usually organized in a *test set*. Its performance is evaluated by comparing the predicted output against the actual response from the test set.

On the other hand, unsupervised learning deals with problems with only a set of features available but no responses to instruct the learning process. The task usually involves

discovering patterns to characterize and group data. With no output variables, performance cannot be measured as in the supervised learning setting; thus, such algorithms' evaluation often relies on heuristics and subjectivity. [2]

Over the years, several machine learning algorithms have been proposed and successfully applied to real-life problems. The same can be said of recommender systems, the focus of this work. Recommender systems, a field of study intimately related to machine learning methods and developments, are described in the next section. [1, 3, 4]

2.2 Recommender Systems

Recommender systems gained importance in the last decades as e-commerce and online content providers became widespread and universally utilized by consumers. These web applications aim to learn users' preferences to predict future items of interest, i.e. to infer users' interests. To achieve that, recommender systems leverage data from users' feedback on the items consumed. These are the two important entities in this context, and most systems assume that the history of their interactions is a good gauge to predict future preferences. These systems are based on the assumption that significant correlations exist between the activities centred on users and items. The two main paradigms in which the recommendation problem can be formulated are recommendation as *prediction* and recommendation as *ranking*. [3]

In recommendation as prediction, the objective is to predict the *rating* value users would give to items. In this framework, ratings data is structured as an m x n matrix given m users and n items. Because users generally have interacted with a limited number of items in the past, most entrances in this matrix are empty. The idea is to train a model with the observed values, i.e. past user-item interactions, and predict the missing or unobserved values. [3]

Recommendation as ranking is also known as the top-k recommendation problem. In this setting, the rating values users would give to items are unnecessary. Instead, the problem involves recommending the top-k items for a user or the top-k users for an item. While top-k lists of users and items are obtained similarly, the former is more prevalent. Even though recommendation as prediction is more general, solving the ranking problem is often more straightforward. [3]

From a broader business perspective, the target of these systems is to increase sales and consequently profit. Typically, four operational characteristics are required to accomplish this: [3]

- Relevance: the primary goal, which is to recommend items that are relevant to users;
- Novelty: recommendations are more interesting and useful if users have not seen the item before, e.g. recommend items that are not popular;
- Serendipity: differently from novelty, serendipity is related to recommending items that are truly not expected by users, even though latent interests might exist;
- Diversity: this is related to the diversity of items in the top-k list of recommendations, such that users may enjoy at least some of them.

There are several types of recommender systems that can be enhanced and altered depending on the domain of usage. The basic recommender systems methods are under the umbrella of *collaborative filtering* (CF) and *content-based* recommenders, while there are also *knowledge-based*, *context-based*, and *hybrid* recommender systems. The following sections present an overview of some of these families of recommender systems and also describe concepts related to *stream-based* systems, which are more critical in the context of this work. [3]

2.2.1 Types of Feedback

The feedback of users can be either explicit or implicit. Explicit feedback can be obtained in the form of interval scales (e.g. five-star rating systems), ordinal ratings (e.g. Hated it, Neutral, Loved it), or binary ratings (e.g. like button). On the other hand, implicit feedback is easier to obtain as it does not require a user to express her preferences. Rather, actions such as buying and browsing can be seen as positive feedback and converted to *unary* ratings (i.e. 1 for action, 0 otherwise). Nevertheless, the lack of a user-item interaction does not necessarily indicate that the user dislikes the item as she may not be aware of it. As noted previously, for m users and n items, feedback is usually structured as m x n sparse matrices. [3]

2.2.2 Collaborative Filtering Recommender Systems

These methods use user-item interaction data as explicit or implicit feedback. The idea is to leverage the interactions of several users to provide recommendations, hence the name collaborative filtering. The two main types of CF methods are: [3]

- *Memory-based methods* or *neighborhood-based collaborative filtering*: this class of methods belongs to the earliest CF methods. The idea is to predict unseen user-item interactions based on their neighbourhoods, which are defined either on the similarities between users (*user-based* CF), items (*item-based* CF), or both. They are simple to implement and generally present explainable results. That said, they underperform in the context of sparse rating matrices;

- *Model-based methods*: these methods apply predictive modelling of users' preferences through data mining and machine learning methods, such as decision trees and latent factor models (such as matrix factorization and neural networks). The parameters of models can be learned through optimization techniques, e.g. stochastic gradient descent and alternating least squares.

2.2.3 Content-Based Recommender Systems

This class of methods leverages the available information of the items a user interacts with to predict the feedback of unseen items. For example, for a system that recommends books, the information comprises books' attributes, such as genre, author, year, among others. The idea is to train a user-specific model with these attributes in order to predict her feedback (i.e. a model by *active* user). Observed items' attributes are treated as independent variables while user feedback is used as dependent variables; these constitute the training data in a supervised machine learning algorithm. If feedback is in the form of explicit ratings, the problem is seen as the prediction of rating values; a classification setting is used for implicit ratings. [3]

Content-based methods can recommend a new item with no ratings history based on the similar attributes of seen items. In contrast, CF methods require sufficient feedback data to provide sound recommendations. Nevertheless, content-based methods may present less diverse recommendations than CF, as community knowledge is not used in this context, which means recommendations are based only on what the user has seen.

They are also problematic regarding recommendations for new users, as their rating history is minimal. [3]

2.2.4 Knowledge-Based Recommender Systems

Knowledge-based and content-based methods are related, but the former includes user-defined attributes or requirements. User-defined attributes are compared against item attributes using similarity metrics based on domain knowledge to generate recommendations, while user feedback is put aside. They are helpful in situations where there are not many user-item interactions overall, such as luxury goods and real estate. As in content-based systems, knowledge-based systems can suffer from providing obvious recommendations. [3]

2.2.5 Demographic Recommender Systems

Demographic recommender systems aim to leverage user demographic profiles to train machine learning models to predict implicit or explicit feedback. They are better used as additions to hybrid or ensemble methods, as their stand-alone performance is generally not ideal. [3]

2.2.6 Hybrid and Ensemble-Based Recommender Systems

Hybrid systems intend to combine the best aspects of each type of recommender system in order to increase task performance. They are especially interesting when there are various sources of input available. The previous sections presented input sources such as user feedback, requirements, demographics, and item attributes. Ensemble models are very similar to machine learning models combined to generate more robust ensemble systems. The combined models may originate from the same or different algorithms, potentially increasing performance in both cases. [3]

2.2.7 Stream-based Recommender Systems

The vertiginous growth of online communities caused an escalation of data volume, velocity, variety, and variability. This enormous amount of data can be described as *data streams*. In this context, it is assumed that data arrive in a potentially unbounded stream or streams where each element can only be processed once; the process must be as fast as

data arrives, while arrivals happen at non-uniform rates and order. To mine helpful information from the deluge of data is complex, and search engine tools are insufficient when considering the sheer number of items. Thus, recommender systems can aid in filtering by suggesting relevant items. [5–8]

However, the recommender system methods described in the previous sections are not designed to tackle the continuous flow of information. Instead, they are adequate for batch learning, a paradigm in which a static dataset is used to obtain a model that remains unchanged until a new one is available for retraining. Models become increasingly inaccurate over time until that occurs, and retraining implies that ever-growing and potentially boundless data need to be stored and re-processed at each update, which is preposterous regarding the computational cost and scalability requirements of online systems. Not less importantly, privacy issues are also a cause for concern. [7, 9]

Stream-based recommender systems are designed to deal with data streams. User feedback is seen as a continuous data stream, while algorithms learn from it to maintain incremental models. These models are updated online as new observations are available. Ideally, these methods must: [7, 10]

- process data as fast as it is generated;
- have bounded memory requirements independent of the number of observations;
- adapt to concept drifts (e.g. changes in user preferences);
- perform a single pass over data to build the model;
- ensure the model is always available to make recommendations.

Moreover, these methods should also be able to capture users' long-term interests and model new users and items. [11]

2.2.8 Evaluating Recommender Systems

Evaluation of recommender systems is done either online or offline. The online evaluation depends on active user participation, such as using *A/B testing* to measure the *conversion rate* of users that interact with recommended items, i.e. the frequency a user chooses a recommended item. However, online evaluation is generally not viable in benchmarking and research because of the difficulty in obtaining access to user data from large-scale

systems. Furthermore, many datasets are needed to assert that a model works in several situations. Thus, offline evaluation with historical data is more commonly used in research and practice. On the other hand, offline evaluation can not assess if a system is still relevant as time passes. [3]

The evaluation paradigm varies according to the formulation of the recommendation problem. If seen as prediction, the evaluation process is similar to what is done in traditional classification and regression. For recommendation as ranking, it is analogous to evaluating retrieval effectiveness in search and information retrieval. However, the data structure used in the offline evaluation is very similar to what is done in classic machine learning in both cases, where either *hold-out* or *cross-validation* are generally used. [3]

These techniques avoid overestimating the algorithm's accuracy by splitting data into training, validation, and testing sets. Respectively, these sets are used to train models, select optimal models and hyperparameters, and assess the accuracy of the final model. In hold-out, parcels of the data are sampled into the abovementioned sets. However, the entire dataset is not used for training, and the accuracy is underestimated because there may be differences in the distribution of the held-out entries. In cross-validation the dataset is split in q equally sized sets. In q interactions, each set is used as a testing set, while the remaining are used as a training set. The accuracy is evaluated at each interaction, and its average is obtained to estimate the algorithm's actual accuracy. This method can obtain a better accuracy estimate if the number of partitions q is considerable; however, it is more computationally demanding. [3]

Accuracy metrics are more typically used for benchmarking because they are objective and easy to measure. Prediction accuracy is often obtained through *mean absolute error*, *mean squared error*, *precision*, *recall*, and similar metrics. Ranking accuracy is measured by the receiver operating characteristic curve, utility-based measures, and rank-correlation measures; the first two are used for implicit feedback. Nevertheless, evaluating recommender systems generally presents varied aspects and goals that accuracy cannot summarize by itself, even though it is usually the primary evaluation criterion. [3, 12]

Other important goals are coverage, confidence, trust, novelty, serendipity, diversity, robustness, stability, and scalability. Coverage measures how well a system covers the items' or users' space. Confidence is measured through confidence estimates, such as confidence intervals, and helps compare models. Trust measures the users' belief in the reported ratings. Novelty is related to how likely unseen items are recommended to a

user. Serendipity measures how surprising unseen successful recommendations are. Diversity measures how diverse recommendations in a list are. Robustness and stability are related to the system's resilience to attacks or changes in data distribution. Scalability is associated with the system's capacity to perform efficiently with massive data and can be measured through training time, prediction time, and memory requirements. Quantifying goals such as novelty, trust, coverage, and serendipity can be subjective and require user surveys. [3]

2.2.8.1 Evaluating Stream-Based Recommender Systems

Batch learning evaluation methods do not work with stream-based recommender systems because the algorithms must be evaluated given the data streams and stream-based model characteristics described in 2.2.7. On the other hand, *prequential evaluation* is an evaluation paradigm designed to work with stream-based algorithms. In this setting, predictions are made and evaluated for each arriving data point or bounded array/window of data points. Only then the model is optionally updated with the new data. Unlike batch learning evaluation methods, prequential evaluation allows assessing live performance over time, can be used online and offline, and does not require data pre-processing. In addition, it can include some of the goals discussed previously in 2.2.8. [12, 13]

Evaluation methods for stream-based recommender systems were proposed by researchers. Siddiqui et al. and Vinagre et al. Siddiqui et al. [13] combine hold-out and prequential evaluation in small-batches. For each arriving sequence of user-item interactions, a portion of them is used for prequential evaluation (evaluation and then learning), and the remaining is only used for evaluation. The approach by Vinagre et al. [12] operates in more straightforward steps. It receives a single user-item interaction, recommends a list of N items to the user, assesses it given the observed item, optionally updates the model with the current interaction, and continues to the next one. Additionally, Vinagre et al. [14] propose a statistical validation framework to compare pairs of stream-based recommendation algorithms.

2.3 Lifelong Machine Learning

According to Zhiyuan Chen and Bing Liu [15], the dominant paradigm in machine learning (ML) is called *isolated learning*, which involves training a model in a given dataset and

using it in a production environment without considering any form of complementary or already obtained knowledge. Unlike human learning, this paradigm cannot leverage previously acquired knowledge to learn and solve new problems, i.e. previously acquired knowledge is not retained.

Lifelong Machine Learning (LML), or *continual learning*, tries to address these issues in order to enable continuous incremental learning, in which previous knowledge can be used to learn new tasks. In a sense, the final objective of LML is to enable intelligent systems that can learn as humans do or are as close to human intelligence as possible. Its three key characteristics are: [15]

- continuous learning;
- explicit knowledge retention and accumulation;
- capability of using previously learned knowledge to help in learning new tasks.

LML is defined as a continuous learning process where a learner has executed N previous learning tasks T, each with its dataset D, and where the types and domains of said tasks can differ. Given a new task with a new dataset, this learner can use the past knowledge stored in a knowledge base to learn it. This knowledge base is updated with the knowledge obtained from learning the latest task. The performance on the new task is usually the objective to be optimized, but any previous task can be made the current objective.

Even though this study area has been gaining importance in the last years, several challenges remain in developing systems capable of performing lifelong learning across multiple tasks and domains. Some of the constraints are the need for a systemic approach, transferring knowledge between domains and tasks, using prior knowledge in successful methods (e.g. deep learning), accumulating large quantities of knowledge, defining what knowledge to retain, how to retain it and how to use it, among others. [15]

2.3.1 Catastrophic Forgetting

Catastrophic forgetting was first identified in the neural networks field of research. Examples of neural networks ML algorithms are multi-layer perceptrons (MLPs) and deep neural networks (DNNs), which have been successfully applied to tasks in several domains. [15]

The issue occurs when the objective is to learn continuously using these algorithms, i.e. when neural networks are used to learn a sequence of tasks. Specifically, training on new tasks causes the weights learned on previous tasks to be overridden, effectively deteriorating the performance on these tasks, which means models forget the information obtained from previous training. [15]

The objective is not to leverage old knowledge to learn new tasks (one of the LML requirements) but to learn new tasks incrementally. Nevertheless, more work is necessary to allow neural networks to be used in the context of LML. [15]

Abraham and Robins [16] discuss memory retention in biological systems and artificial neural networks (ANN) through the trade-off between stability and plasticity. They propose that a balance between stability and plasticity of learned weights is needed for memory retention. While learning new cases, weights must be flexible enough to include new information and adapt the representation of previous tasks, but not too flexible to forget. Catastrophic forgetting is a direct consequence of the stability-plasticity dilemma. [9, 15]

2.3.2 Lifelong Learning and Catastrophic Forgetting in Recommender Systems

Stream-based recommender systems described in previous sections are widely studied in recent research. They can integrate new information in real-time without retraining a model entirely, therefore being efficient in online environments. [15, 17]

That said, state-of-the-art RecSys capable of online learning do not fulfil the two remaining requisites that characterize LML - perform explicit knowledge accumulation, and use previous knowledge to help learn new and possibly different tasks. While continual learning methods are concerned with performance over several learned tasks, the focus of incremental recommender systems is generally the performance on future tasks, i.e. the efficient transference of valuable past knowledge for future recommendations. As an exception, Yuan et al. [18] made a recent contribution in which they modelled user representations over several recommendation domains. [19, 20]

Users' preferences are usually volatile, meaning that items favoured by a user at some point in time might not be the same in the future. These changes in the underlying data

distribution are known as *concept drifts*. By being continuously updated, incremental recommender systems learn new user preferences, or concepts, while retaining some information about the old concepts; thus, they can adapt to concept drifts. The study by Matuszyk et al. [17] aims to tackle the phenomena by proposing forgetting techniques to aid stream-based matrix factorization systems adapt to changes instead of relying solely on incorporating new information. They argue that some of the old feedback given by users is not representative anymore and would reduce the model's performance. [15]

Nevertheless, recommendation models should ideally capture short-term and long-term user preferences. While recent interactions contain users' dynamic preferences, previous interactions contain information on long-term preferences useful for providing meaningful recommendations. Incremental recommender models that learn only from the most recent interactions might experience performance degradation as old concepts that contain useful information are overwritten. [19, 21]

Thus, it is important to address catastrophic forgetting in recommender systems so that they can achieve lifelong learning in the future, therefore being able to learn continually and without human mediation.

Chapter 3

State of the Art

Most research in lifelong learning focuses on image classification and reinforcement learning domains. In contrast with the usual recommendation scheme, these domains present well defined tasks.

3.1 Forgetting Assessment in Image Classification

Overall, continual learning approaches that tackle catastrophic forgetting can be categorized into three families. *Replay* methods use stored samples of previous tasks or pseudo-samples from a generative model to reduce forgetting. These are either used as inputs (also called rehearsing) while learning a new task or to avoid interference with previous tasks by inhibiting loss optimization. *Regularization-based* methods use an additional regularization term in the loss function to solidify previously learned parameters. *Parameter isolation* methods fix weights for each task learned so forgetting is not possible. For each task, weights are fixed by growing new branches, using dedicated model copies, or assigning parts of a permanent architecture to learn them. [9, 22, 23]

The paper by Ashley et al. [24] reinforces that more robust evaluation methods are needed to assess catastrophic forgetting. They provide evidence that the optimization algorithm used to train a neural network can affect forgetting and compare four metrics used to quantify catastrophic forgetting, showing that they might lead to different conclusions for the same experiment. Their experiments are in the image classification and reinforcement learning domains. They conclude by suggesting that more robust experimental methods are needed to study catastrophic forgetting and propose that both *retention* and *relearning* be measured while evaluating inter-task forgetting in supervised

learning settings. They also recommend measuring *pairwise interference* to evaluate intra-task forgetting in reinforcement learning settings.

Retention metrics measure the performance variation on previous tasks after learning a new task. In the two-task setting, a model is trained on one task until it achieves proficiency, then is trained on a second task until it is proficient, and the new performance on the first task is recorded. Relearning metrics measure the time to relearn a task after learning a new task. The two-task setting is similar to retention, but with the addition of a final training session on the first task. The difference in learning time between the last and first sessions is reported. Pairwise interference measures the interference caused by learning from a sample in learning another sample, be it positive or negative. [24]

In a review of continual lifelong learning with neural networks, Parisi et al. [22] compile the significant challenges related to lifelong learning and compare methods that aim to tackle catastrophic forgetting. They affirm that comparing methods' performance, forgetting, and knowledge transfer is troublesome given the heterogeneity and limitations of evaluation frameworks. The choice of benchmark datasets and metrics is not unanimous, even though the datasets generally used to assess lifelong learning belong to the image classification task domain. They note the need for robust and flexible methods and thorough evaluation schemes to deal with complex real-life situations.

Lopez-Paz and Ranzato [25] proposed the Gradient Episodic Memory (GEM) model and developed an evaluation framework for continual learning models. The datasets belong to the image classification domain. Their framework uses either a single pass or mini-batch setting, where each example includes a task descriptor that identifies the associated task. The tasks are outlined as learning permutations of pixels, image rotations and new classes. A test set is available for each task. Model performance is measured through average accuracy and the capability of transferring knowledge across tasks. The test accuracy is evaluated on all tasks every time the model finishes learning a task, and the result is subsequently stored.

Backward transfer is the influence learning a new task has on the performance of previous tasks. It is the average difference between the accuracy on a task after learning a new task and when it was first learned. Forgetting occurs if backward transfer is negative. *Forward transfer* is the influence learning a new task has on the performance of future tasks. Díaz-Rodríguez et al. [26] modify the metrics proposed by Lopez-Paz and Ranzato [25] to consider the model's performance at every timestep, with the objective of better

characterizing the dynamic aspects of continual learning. Lovón-Melgarejo et al. [27] contribute by studying catastrophic forgetting in neural ranking models and also use a modified version of the backward transfer measure to compare different neural models.

Chaundhry et al. [28] propose an incremental learning method for classification and metrics to measure accuracy, forgetting and intransigence - the inability of an algorithm to learn new tasks. The last two are expected to be negatively correlated. Accuracy on held-out test sets is calculated for all tasks j, where $j \leq k$, after training on a task k. Forgetting is calculated through the *Forgetting Measure F*, which is the difference between a task j maximum accuracy over the course of the learning process and the accuracy after learning the current task k. They argue that using the maximum accuracy instead of the accuracy immediately after training j, such as in Lopez-Paz and Ranzato [25], allows the metric to consider the effect of positive backward transfer in the learning process. Intransigence is measured as the difference between the accuracy of a standard classification model trained on data from all tasks on held-out data from task k, and the accuracy of a incremental model trained up to task k. The authors suggest that negative values of intransigence signify positive forward transfer, while positive values imply negative forward transfer.

Hayes et al. [29] describe three evaluation paradigms and metrics to assess continual learners. Data is either unordered (i.i.d.), ordered by class, or the stream is organized by instances where classes can be revisited. The learner receives one sample at a time and can only see it once, and the model accuracy is evaluated every n samples on test data. They suggest using metrics proposed by Kemker et al. [30] and propose evaluating overall performance through the averaged accuracy on all of the test data seen at test time t weighted by the accuracy of an offline i.i.d. model on all of the training data seen at test time t. The datasets utilized belong to the image classification domain.

Kemker et al. [30] proposed benchmark experiments and metrics for comparing incremental neural network methods that mitigate catastrophic forgetting. The datasets belong to the image and audio classification domains. They use a study session (task) setting, where session batches are learned sequentially and in order. Each task has a test dataset.

The metrics reported over each experiment are Ω_{base}, Ω_{new}, and Ω_{all}. Ω_{base} is the average test accuracy on the first task after each task is learned. It evaluates how well a model keeps information learned in the first task, e.g. long-term memory. Ω_{new} is the average

test accuracy of tasks when they are learned. This metric assesses the model performance on recent tasks, e.g. if the model is still learning. Ω_{all} is the average test accuracy of all tasks learned. It is helpful to evaluate both recall and learning of previous and new information, respectively. To facilitate comparisons between datasets, both Ω_{base} and Ω_{all} are normalized by the offline MLP test accuracy of the first task.

Lomonaco and Maltoni [31] proposed continuous learning benchmark experiments and a new object recognition dataset. The task learning scenarios focus on new instances, new classes, and both. Forgetting is qualitatively assessed by comparing accuracy for the cumulative strategy - where data from the current task and all the previous are used for training - and incremental strategies - where tasks are learned sequentially.

Zhou & Cao [23] propose a continual learning method based on graph neural networks. Their experiments use datasets from the node and image classification domains. Tasks are learned sequentially and are composed of new unseen classes. Forgetting is the difference between the performance of a task after learning it and the performance after learning the following tasks.

The paper by Masarczyk et al. [32] explores forgetting by evaluating the internal representations of neural networks. The experiments use datasets from the image classification domain. Catastrophic forgetting is assessed through accuracy comparison and the index of representations similarity between tasks. Feature transferability and reconstruction loss from the first task to others are also evaluated.

Serrà et al. [33] propose a task-based hard attention method to preserve the information of previous tasks while not affecting current learning. The method is evaluated with datasets from the image classification domain. Forgetting is measured through the forgetting ratio ρ. After learning a task t, test accuracy on a previous task τ is calculated and reduced by a lower bound from a classifier randomly initialized from its classes; the result is divided by the accuracy of a high bound multitask classifier also reduced by the lower bound. The measure represents how close performance is to the lower or upper bounds. The average ρ for every task learned is reported.

De Lange et al. [9] present an extensive overview of continual learning methods that deal with catastrophic forgetting. They also propose a framework to evaluate the plasticity-stability trade-off continuously. However, they strongly relax the stream-based and continual learning settings - image classification tasks are presented sequentially but the entire batch dataset for each task is available for training for several epochs at each

step, tasks have different output layers, and they assume previous knowledge of the task of a test sample. Tasks are delimited based on the classes available for training and shifts in the images domain, and are evaluated through accuracy and forgetting. The forgetting measure of a task is the difference between the accuracy when it was first learned and after training one or more tasks. Average accuracy and average forgetting across all tasks are obtained after learning the entire sequence of tasks.

3.2 Forgetting Assessment in Recommendation

The research on continual learning recommender systems often does not measure forgetting explicitly. Forgetting evaluation is based on the performance comparison between methods. Moreover, most methods use variations of an incremental setting where user interactions are sorted by time, split into batches, and learned sequentially. [19–21, 34–38]

Mi et al. [39] propose a continual learning method for session-based recommendation (i.e. based on short-term interactions). The datasets used are DIGINETICA and YOOCHOOSE - 5 and 6 months e-commerce click-streams data. A prequential training scheme is used, where a model is evaluated on a new batch and then learns from it. Batches are composed of weekly and daily data. Accuracy is assessed through Recall@k (recall at k recommendations) and MRR@K (Mean Reciprocal Rank at k recommendations) for different baseline methods. Forgetting is not explicitly measured, being assessed by comparing the methods' accuracy. Moreover, the authors note that forgetting is not a significant issue in the less dynamic dataset YOOCHOOSE, where old items reappear frequently.

In a different approach, Yuan et al. [18] propose a neural networks-based framework to learn user representations continuously. Unlike other recommender systems, their method can learn tasks from different domains. Two datasets are used, Tencent TL (TTL) and Movielens (MLE). TTL comprises six datasets connected by users' IDs, three of which are designed for item recommendation - news and video watching interactions, clicking interactions, and thumbs-up interactions. The authors process the MLE dataset to resemble a continual learning setting, dividing observations over tasks based on the rating value - it is assumed that predicting higher-ranked items is more complicated. Tasks are batch learned sequentially, and their test accuracy is calculated after learning all tasks. In addition, a separate model is learned for every task. Top-N recommendation accuracy is measured through MRR@5. Forgetting is indirectly assessed by comparing the task accuracy of different learning methods.

Chapter 4

Methodology

This chapter describes the datasets, stream-based recommendation algorithms, and overall methodology used to assess information transfer.

4.1 Information Transfer Assessment Scheme

Inspired by the works described in Section 3, backward transfer is assessed by comparing performances of models on past interactions against models recently trained on these interactions, and forward transfer is computed by averaging the performance of models on future interactions. It is important to note that, unlike the works from the image classification domain, the examples in each interval do not represent a single concept; they contain various user and item interactions with dynamic preferences and characteristics.

The evaluation process is organized in a few steps. Figure 4.1 presents a schematic of the process, which is described below:

1. First, the dataset is processed into time-based intervals, *interval 1, 2, 3, ...*, and their respective holdouts, *holdout 1, 2, 3, ...*; Interactions are ordered by their timestamp;
2. A streaming recommendation model (*RecSys*) is trained on the first interval;
3. Its performance is assessed in the current holdout and available previous and future holdouts;
4. The model continues to learn and be assessed in this manner until there are no more intervals;
5. Results are organized in a matrix that illustrates the performances of different states of a model in all holdouts, and information transfer scores are computed.

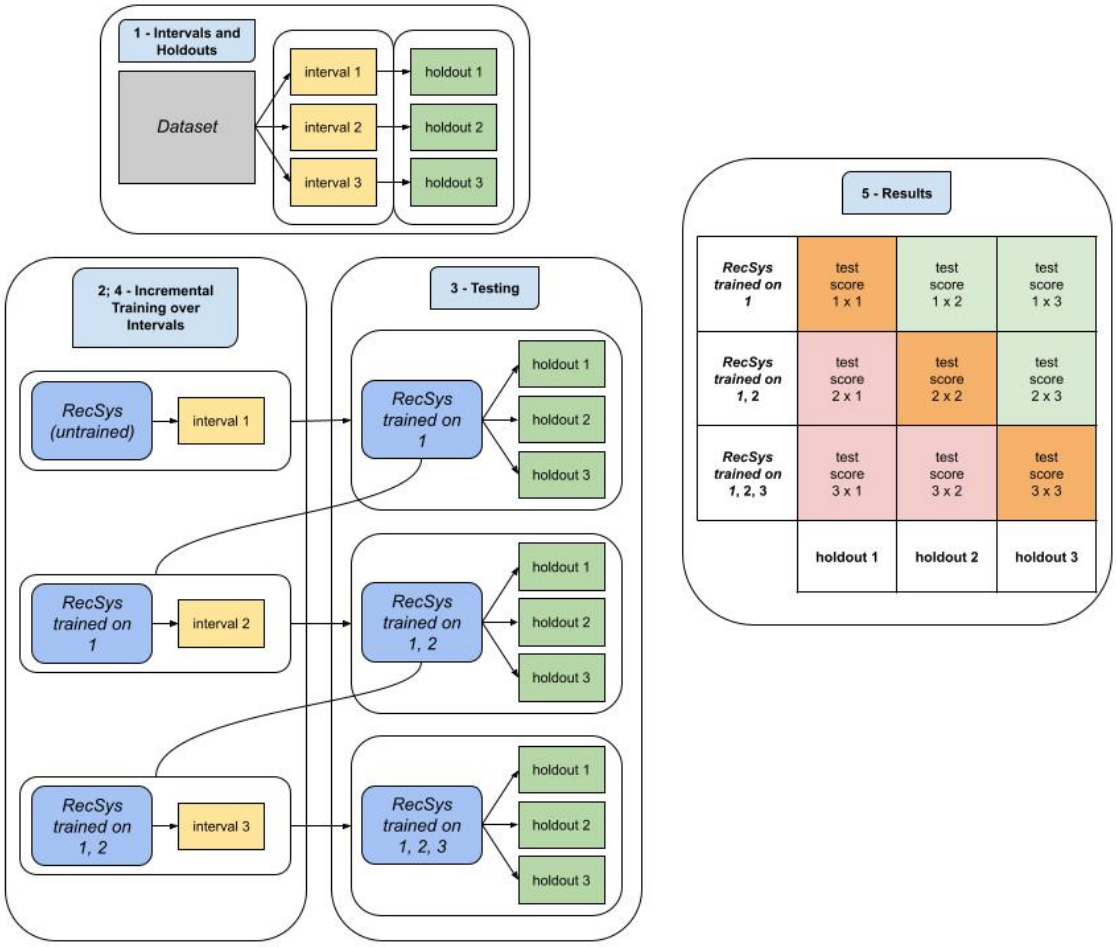


FIGURE 4.1: Methodology Scheme. The figure shows the steps taken to assess the performance of an incremental model in holdouts relative to training datasets separated by time intervals. E.g. intervals 1, 2, and 3 could represent data from January, February, and March. 1 - The dataset is separated into time-based intervals, from which interactions are held; 2 - A streaming recommender system learns from an interval, and; 3 - is assessed in all holdouts; 4 - The process repeats for all available intervals; 5 - Results are aggregated into a matrix, from which average performance, backward transfer and forward transfer can be computed.

The overall performance of a model is computed by averaging the matrix's diagonal, in orange, representing a model's performance on holdout data closest to the most recently learned interval. Backward transfer is computed by comparing models' performances on previous holdouts - scores in the red lower triangle of the results matrix - to the diagonal scores directly above them. Forward transfer is computed by averaging models' performances on future holdouts, represented by the green upper triangle of the results matrix. A negative backward transfer score means forgetting occurs. A low forward transfer score might indicate that present concepts are not as helpful in modelling future preferences. Therefore, forgetting is more likely to be beneficial.

The following sections provide more details on the process.

4.1.1 Intervals and Holdouts

Intervals are obtained based on a predefined period. This work uses months to separate interactions into intervals. Other time periods might be used, but it is essential to consider the number of available interactions in each interval.

The holdouts are created from the last interaction of each user present in the interval. However, there are some conditions. If the user was not seen before and only had a single interaction in the interval, then the interaction is used for training and not for evaluation. Moreover, interactions common to both holdouts and intervals are removed from the respective holdout and kept only in the respective interval.

4.1.2 Training and Evaluation

The streaming-based recommendation algorithms are trained incrementally. Interactions in an interval are ordered by timestamp and are observed by the model one by one. After learning the last case in an interval, the model is assessed on each holdout and starts learning the next interval.

Model performance is gauged through the *Recall@20* score. For each user-item interaction in a holdout, the model provides a vector of twenty recommendations. The score is equal to one if the held-out item is among the ones recommended for the user; otherwise, it is equal to zero. Finally, the average score for a holdout is returned.

Notice that if a particular model state has not seen a user yet, no recommendations are made for her. The absence of recommendations may occur when a model state performs recommendations for future holdouts and is not seen as a score equal to zero. For example, a model trained solely on the first interval will not have seen a user with its first interaction on the second interval; thus, it cannot perform recommendations for this user, and the average score computation ignores this interaction.

The scores are organized in a matrix, as in step 5 of Figure 4.1.

4.1.3 Assessing Performance, Backward Transfer and Forward Transfer

The following results matrix represents an example with $N = 3$ intervals (4.2). The left of the matrix presents intervals I_i with subscript i=1, 2, 3; The bottom of the matrix presents holdouts H_j with subscripts j=1, 2, 3. Each entry $R_{i,j}$ represents the Recall@20 score on a holdout H_j of a model trained on I_i and previous intervals I_k where $k < i$.

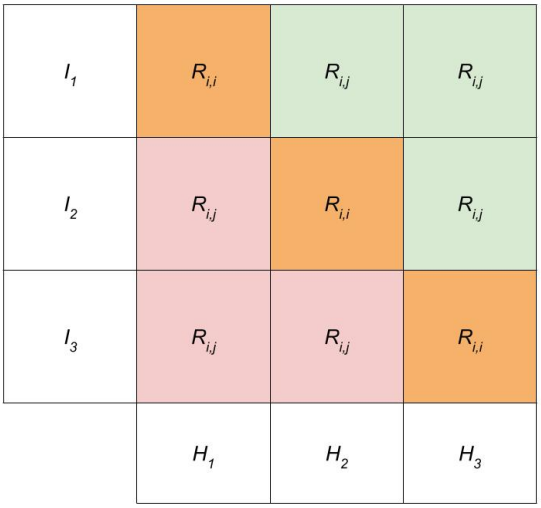


FIGURE 4.2: Example results matrix from an experiment with $N = 3$ intervals. The left side shows intervals; the bottom shows holdouts. The matrix is populated by entries $R_{i,j}$ that represent the scores, for each holdout H_j, of model states trained on I_i and previous intervals I_k where $k < i$.

Values are expected to be higher in the main diagonal (orange); it represents the scores $R_{i,i}$ of models recently trained on the interval I_i associated with the holdout H_i. The diagonal average provides a way to assess an algorithm's overall performance in a given dataset. It is calculated as:

$$DiagonalScore = \frac{1}{N}\sum_{i=1}^{N} R_{i,i} \tag{4.1}$$

Backward Transfer (BWT) is seen as the influence learning an interval has on the performance on previous intervals. BWT is computed by comparing the optimal scores $R_{i,i}$ with the scores $R_{i,j}$ after training with data from other intervals (red lower triangle of the matrix). A score in the diagonal is compared to the scores in the same column, i.e. scores from future models tested on the same holdout. In this sense, BWT is always relative to the models' performance on recent interactions $R_{i,i}$. Negative BWT means the model has lower average results than the diagonal (i.e. forgetting). Positive BWT means the model has higher average results than the diagonal (i.e. positive backward transfer). As in Días-Rodriguez et al. [26], backward transfer is computed as:

$$BWT = \frac{\sum_{i=2}^{N}\sum_{j=1}^{i-1}(R_{i,j} - R_{j,j})}{\frac{N(N-1)}{2}} \tag{4.2}$$

Forward transfer (FWT) is seen as the influence learning an interval has on the performance on future intervals. FWT is computed by averaging the scores $R_{i,j}$ on the green

upper triangle of the matrix. As in Días-Rodriguez et al. [26], forward transfer is computed as:

$$FWT = \frac{\sum_{i<j}^{N} R_{i,j}}{\frac{N(N-1)}{2}} \tag{4.3}$$

4.2 Datasets

Four datasets are used:

- Amazon Kindle Store (AKS): book reviews from the Amazon Kindle platform. Almost 5.7 million user ratings. Ratings range from 0 to 5;
- Amazon Digital Music (ADM): digital music reviews from the Amazon e-commerce platform. Around 1.6 million user ratings. Ratings range from 0 to 5;
- Palco 2010 (P10): music streaming dataset from Palco Principal. Around 584 thousand user interactions.
- Movielens (MLE): movie ratings dataset. Around 226 thousand user interactions.

The Amazon Datasets [40] contain tuples $< user, item, rating, timestamp >$. Only ratings equal to 5 are retained as the algorithms used work with positive feedback. The Palco 2010 and Movielens datasets contains tuples $< user, item, timestamp >$. Only interactions of users that have at least five interactions are kept to reduce noise from spurious users.

Each dataset is sampled to reduce experiment time. The period is chosen based on the number of interactions available. The first two months of 2014 are sampled from Amazon Kindle Store. The first three months of 2014 are sampled from Amazon Digital Music. The first four months of 2010 are sampled from Palco 2010. Months 5, 6, and 7 of 2000 are sampled from Movielens. Moreover, the Amazon Kindle Store dataset is resampled to maintain 50% of users, and the Amazon Digital Music and Palco 2010 datasets are resampled to maintain 75% of users. Users are randomly chosen with probabilities based on their frequency in the period considered. The user sampling is also done to diminish model training and testing times.

The description of the sampled datasets used in the experiments can be seen in Table 4.1. The Palco 2010 dataset presents lower proportion of items per event and users per event than the Amazon datasets; this indicates the Amazon datasets present more user and item variability, suggesting they are highly dynamic and that their events are more

complex to model. The datasets are processed into the following intervals and holdouts (i.e. train and test sets):

TABLE 4.1: Dataset Description

Dataset	Domain	Application	Events	Repeated	Users	Items
AKS	e-commerce	rating	77145	yes	13902	59197
ADM	e-commerce	rating	29781	yes	9991	21645
P10	music	streaming	435621	yes	2921	22103
MLE	movies	streaming	50742	no	1427	2492

TABLE 4.2: Datasets' Intervals and Holdouts

Dataset	Interval	Interval Size	Holdout Size
AKS	1	35763	6231
	2	27265	7886
ADM	1	9564	1397
	2	7107	2016
	3	7195	2502
P10	1	78313	699
	2	110572	768
	3	131966	874
	4	111482	947
MLE	1	20313	664
	2	12706	442
	3	16160	457

4.3 Stream-Based Recommendation Methods

Information transfer is assessed in five stream-based recommendation methods:

- Incremental User K-Nearest Neighbors (UKNN);
- Incremental Stochastic Gradient Descent (ISGD);
- Incremental Bayesian Personalized Ranking Matrix Factorization (BPRMF);
- Recency Adjusted Incremental Stochastic Gradient Descent (RAISGD);
- Randomly Sampled Incremental Stochastic Gradient Descent (RSISGD).

UKNN is an incremental user-based CF algorithm (2.2.2) for implicit binary ratings. This algorithm looks for users that are most similar to a user u in the historical database, recommending items preferred by these neighbours that were not yet seen by u. The

cosine similarity between users is computed and updated incrementally after each session instead of being computed from scratch. However, the algorithm requires caching and updating factors to compute similarities as interactions are available. The parameter k, the number of nearest neighbours, is set to ten for all experiments. [41, 42]

ISGD is an incremental matrix factorization algorithm for positive-only feedback presented by Vinagre et al. [43]. It is an incremental CF algorithm (2.2.2) that adapts stochastic gradient descent to update the user and item factor matrices one interaction at a time. The idea is to minimize the L_2-regularized squared error between the known ratings and the prediction, correcting the factor matrices in the inverse direction of the gradient of the error.

RAISGD is an adaptation of ISGD where a recency-based scheme is used to introduce negative examples artificially. The method is attractive because, with positive-only feedback, the absence of an interaction does not necessarily mean a user dislikes an item. Basically, for each user-item interaction (u, i), a set of negative feedback $\{(u, j_1), ..., (u, j_l)\}$ is introduced using the l items that were seen the farthest back in the data stream. RSISGD also introduces negative examples artificially, but these are selected randomly. [44]

BPRMF is a matrix factorization algorithm for implicit feedback that uses a different optimization criterion. This criterion is based on a Bayesian framework and is used to optimize ranking directly, in contrast to its counterparts. In this work, its incremental adaptation is used, where the algorithm iterates over user-item pairs one at a time. The implementation of BPRMF used in this work is available in the *Python* package *Flurs*. [45, 46]

The hyperparameters of Palco 2010 and Movielens experiments with ISGD and its variations were the same used by [7]. For the remaining experiments, except the ones with UKNN, the first 5% of each dataset is used to determine hyperparameters using a grid-search scheme with prequential evaluation.

Chapter 5

Results and Discussion

This section displays the results of experiments run with the datasets and algorithms described in Chapter 4. Experiments were run in a 16-core, 2400 MHz Intel Core Processor (Haswell, no TSX) machine with 65.86 GB RAM.

Table 5.1 presents the experiments' outcomes. The table shows the scores achieved for each dataset and algorithm combination, computed as described in Section 4.1.3. Moreover, the results are arranged by dataset in Figure 5.1.

TABLE 5.1: Results Summary. The table shows the mean diagonal (Diag), BWT, and FWT scores obtained by the algorithms in the four datasets. For each dataset, the symbols $\uparrow$ and $\downarrow$ represent the best and worst results for that score — the symbol $*$ represents an insufficient result for comparison, meaning the algorithm performed too poorly to allow a fair comparison.

Data	Score	ISGD	RAISGD	RSISGD	BPRMF	UKNN
AKS	Diag	1.69×10^{-2}	$\downarrow 9.84 \times 10^{-3}$	1.23×10^{-2}	$*4.48 \times 10^{-4}$	$\uparrow 3.99 \times 10^{-2}$
	BWT	$\downarrow -1.93 \times 10^{-3}$	-9.63×10^{-4}	-1.6×10^{-3}	-1.6×10^{-4}	$\uparrow 1.4 \times 10^{-2}$
	FWT	3.43×10^{-3}	$\downarrow 2.12 \times 10^{-3}$	3.27×10^{-3}	3.80×10^{-4}	$\uparrow 7.68 \times 10^{-3}$
ADM	Diag	4.17×10^{-3}	1.52×10^{-3}	1.63×10^{-3}	$\downarrow 1.24 \times 10^{-3}$	$\uparrow 9.26 \times 10^{-3}$
	BWT	$\downarrow -9.73 \times 10^{-4}$	-4.96×10^{-4}	$\downarrow -9.73 \times 10^{-4}$	-2.39×10^{-4}	$\uparrow 1.43 \times 10^{-3}$
	FWT	0	0	0	0	$\uparrow 5.26 \times 10^{-4}$
P10	Diag	2.28×10^{-1}	3.60×10^{-1}	$\uparrow 3.68 \times 10^{-1}$	$*3.58 \times 10^{-4}$	$\downarrow 1.59 \times 10^{-1}$
	BWT	-1.74×10^{-1}	$\downarrow -2.18 \times 10^{-1}$	-2.14×10^{-1}	0	$\uparrow -3.25 \times 10^{-2}$
	FWT	$\downarrow 1.26 \times 10^{-2}$	2.07×10^{-2}	2.2×10^{-2}	3.67×10^{-4}	$\uparrow 2.86 \times 10^{-2}$
MLE	Diag	2.3×10^{-2}	4.7×10^{-2}	6.33×10^{-2}	$\downarrow 1.66 \times 10^{-2}$	$\uparrow 7.2 \times 10^{-2}$
	BWT	-6.53×10^{-3}	-1.88×10^{-2}	$\downarrow -3.47 \times 10^{-2}$	3.52×10^{-3}	$\uparrow 9.03 \times 10^{-3}$
	FWT	$\downarrow 7.15 \times 10^{-3}$	2.4×10^{-2}	2.21×10^{-2}	1.63×10^{-2}	$\uparrow 5.22 \times 10^{-2}$

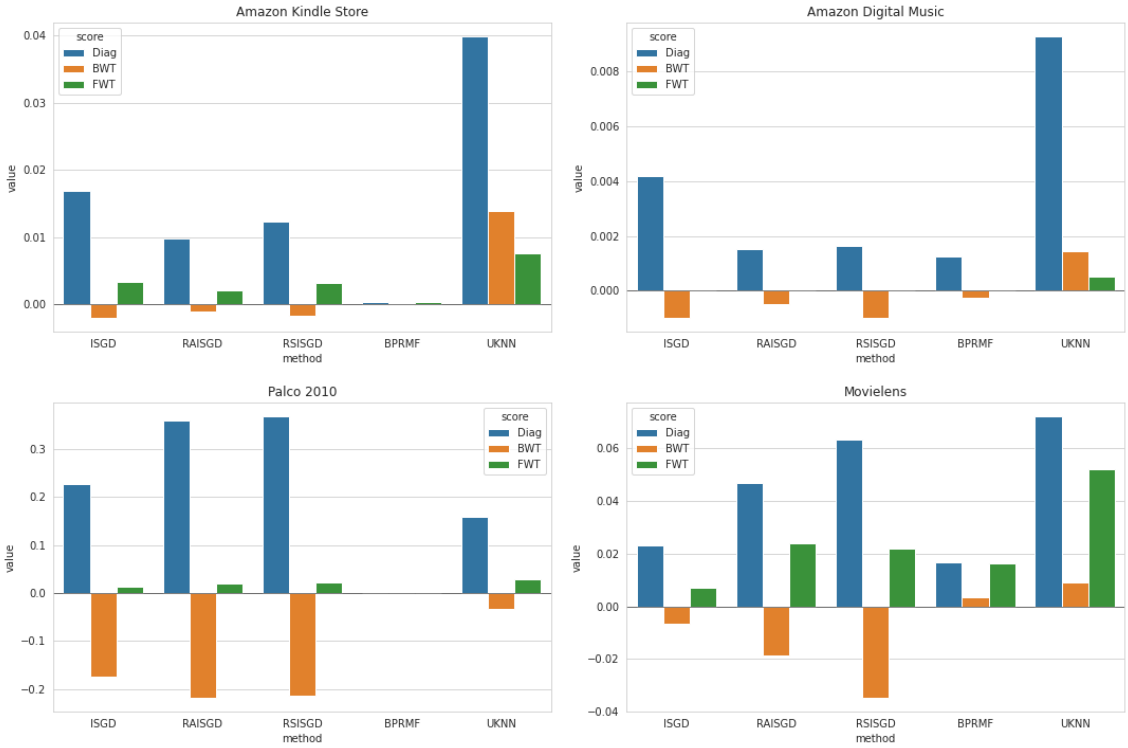


FIGURE 5.1: Results Summary. The plots show the mean diagonal (Diag), BWT, and FWT scores (in blue, orange, and green) obtained by the algorithms on each dataset. The horizontal axis represent the methods, and the vertical axis the score values.

5.1 Results per Dataset

In this section, results are analysed horizontally across models for each dataset.

5.1.1 Amazon Kindle Store

UKNN is the best performer in all scores for the AKS dataset. Its mean diagonal score of 3.99×10^{-2} is followed by ISGD's 1.69×10^{-2}. The performance of BPRMF ($*$ in the *AKS* row of Table 5.1) is two orders of magnitude lower than its counterparts; thus, it is not viable to fairly assess the method.

UKNN is the only method able to attain a positive BWT score. In contrast, ISGD presents the worst score, followed by RSISGD and RAISGD; in this case, inserting negative examples does not hurt the capability to retain past useful information even though the scores are close and doing so reduces the mean diagonal scores. UKNN also presents the highest FWT score (7.68×10^{-3}), followed by ISGD (3.43×10^{-3}). The lowest FTW score belongs to RAISGD (2.12×10^{-3}).

As with the other experiments discussed next, UKNN's caching of similarity factors is most likely what results in the high BWT and FWT scores. Moreover, the high mean diagonal performance of UKNN compared to the remaining methods is not expected, as ISGD and its variations are considered state-of-the-art.

The Recall@20 heatmaps produced for this experiment are displayed in Appendix A. The main diagonal of all graphs except UKNN (Figure A.5) presents the highest results. UKNN's Recall@20 score in the lower triangle is higher than the main diagonal, which explains its positive BWT score (Table 5.1). Moreover, its high FWT score is also elucidated when one compares its Recall@20 score in the upper triangle against the heatmaps of other methods. It can be seen by comparing the heatmaps of RAISGD and RSISGD (Figures A.2 and A.3) that, even though the latter has a lower BWT score (Table 5.1), it concurrently has a higher Recall@20 score in the lower triangle; its overall performance is superior. This outcome means scores should be used together with heatmaps to assess information transfer.

5.1.2 Amazon Digital Music

UKNN has the best results for experiments with the ADM dataset, with a mean diagonal score of 9.26×10^{-3}, followed by ISGD's score of 4.17×10^{-3}. BPRMF has the worst mean diagonal score, followed by RAISGD. In both experiments with the Amazon datasets, the ISGD algorithm has the highest mean diagonal score than its variations RAISGD and RSISGD.

A positive backward transfer score is only achieved with UKNN; ISGD and RSISGD present the worst BWT scores. Save for UKNN, models display a score of zero for FWT, meaning these could not transfer helpful information for future recommendations. Moreover, the FWT score of UKNN is one order of magnitude lower than the mean diagonal score. The performance on future holdouts is diminished because the preferences of seen users might have changed. This pattern of FWT results is observed in the experiments with the other datasets.

Overall, this dataset benefits from the memory retention promoted by UKNN to provide recommendations for previous and future holdouts. UKNN caches factors in memory to compute the similarity between users, which may be thought of as increased stability compared to the plasticity of other models. Curiously, models could not provide

better mean diagonal scores than UKNN, which suggests the sampled dataset may not be as dynamic.

The Recall@20 heatmaps produced for this experiment are displayed in Appendix B. In all graphs, the main diagonal presents the highest results, which is expected given that it represents an ideal model performance. However, RAISGD cannot perform recommendations for the first holdout independently of the model state (Figure B.2).

Even though the BWT scores of ISGD and RSISGD are the lowest for the experiment (down arrows in the ADM row of Table 5.1), their heatmaps clearly show that ISGD has higher recall values than its variations in the lower left triangle (Figures B.1, B.2, B.3). The BWT score cannot account for these nuances because it is an average of the differences between the diagonal and the lower triangle scores. According to BWT, there is no backward transfer for the first holdout in the RAISGD experiment (Figure B.2), either positive or negative. Moreover, ISGD and RSISGD have equal BWT scores, even though the former is a better performer in previous and recent holdouts (Figures B.1 and B.3). Hence, BWT must be carefully used in conjunction with a model's heatmaps and general performance scores to assess forgetting and positive backward transfer.

The heatmaps explain why BPRMF and UKNN (Figures B.4 and B.5) BWT scores are higher than their counterparts (Table 5.1). Their recall in previous holdouts is equal to or highest than the diagonal scores, even though BPRFM is a worse performer than ISGD in most cases (Figure B.1). As noted in Section 4.1.3, BWT must be interpreted relative to the methods' *ideal* performance, i.e. the recall scores in the main diagonal.

Finally, Figure B.5 exhibits non-null scores in the upper triangle, from which the mean composes the positive FWT score of UKNN shown in Table 5.1.

5.1.3 Palco 2010

The P10 dataset presents the highest scores among experiments. RSISGD presents the highest mean diagonal score and behaves very similarly to RAISGD, and UKNN displays the lowest. The performance of BPRMF ($*$ in the *P10* row of Table 5.1) is three orders of magnitude lower than its counterparts. Thus, this experiment cannot fairly assess it; this may be an issue with the algorithm implementation.

No model displays positive backward transfer; however, UKNN has a BWT score one order of magnitude lower than its counterparts, i.e. the less negative score means it

suffers less forgetting. RAISGD and RSISGD present the lowest BWT result, meaning that inserting negative examples increases forgetting in this experiment.

UKNN has the highest FWT score, but the order of magnitude is equal to the remaining models. The worst FWT score belongs to ISGD. FWT scores are one order of magnitude lower than the mean diagonal scores.

Results suggest that, for this dataset, recent interactions are better modelled through the more flexible ISGD, RAISGD, and RSISGD methods, which update model parameters through stochastic gradient descent, in contrast to UKNN, which updates the specific factors based on the active user only. UKNN has inferior performance in recent interactions but presents less forgetting while retaining more information for future recommendations.

The Recall@20 heatmaps produced for this experiment are displayed in Appendix C. The main diagonal presents the highest results in all graphs; except for BPRMF, which has reduced performance. Moreover, the recall heatmaps present an interesting and evident pattern where scores in the lower left triangle diminish from the diagonal to the bottom of the graph, column-wise and downwards. This imprint may be observed in the experiments with other datasets but not as consistently among methods and as regularly as Palco 2010. The increased number of intervals in this experiment might facilitate visualizing the trend. As expected, the pattern signifies that as each new interval is learned, forgetting diminishes the performance in previous holdouts.

The high BWT score of UKNN can be explained through its heatmap (Figure C.5), where, compared to the remaining graphs, there is a reduced difference between the recall values in the diagonal and the lower triangle. Moreover, it can be seen that while RAISGD and RSISGD models have increased performance over UKNN in the diagonal, the latter's performance is similar or superior in previous holdouts at least two intervals from the diagonal. E.g. observe the high performance of the UKNN model 3 in the holdout 1 in Figure C.5 against the same performance of RAISGD and RSISGD in Figures C.2 and C.3. The FWT scores discussed previously are also illustrated in the upper triangle of the heatmaps.

5.1.4 Movielens

UKNN presents the best results in all scores. Its mean diagonal score is equal to 7.2×10^{-2}, followed by RSISGD's score of 6.33×10^{-2}. The worst mean diagonal score belongs to

BPRMF, followed by ISGD.

UKNN presents the highest BWT score, and RSISGD displays the lowest. Moreover, UKNN and BPRMF return positive backward transfer scores; thus, learning future interactions with these methods helps to model previous ones. UKNN presents the highest FWT score, and ISGD shows the lowest. In this experiment, the FWT scores for all methods except ISGD are the same order of magnitude as the mean diagonal scores, suggesting user preferences do not change substantially in the period considered.

As with the ADM dataset, recent interactions of the MLE dataset are better modelled through UKNN, while it also presents positive backward transfer and increased forward transfer. Again, this may imply that the dataset is not as dynamic as others, for which UKNN is not the top performer.

The Recall@20 heatmaps produced for this experiment are displayed in Appendix D. The heatmap of UKNN (Figure D.5) justifies its high mean diagonal and FWT scores, as its recall values in both the main diagonal and upper triangle are higher than the remaining methods. Its high BWT score is due to the recall values in the lower triangle being higher than the the main diagonal.

Similarly, the heatmap of BPRMF (Figure D.4) explains the low mean diagonal score, as the recall values in the diagonal are lower than the remaining methods' scores. The graph also explains the positive BWT score, as the values in the lower triangle are higher than the main diagonal ones.

ISGD and its variations (Figures D.1, D.2, and D.3) present negative BWT scores as the lower triangle exhibits lower scores than the main diagonal, on average. Generally, the further away from the main diagonal, the lower the recall values, a result challenged by UKNN and BPRFM in this dataset.

5.2 Results per Method

This section discusses methods vertically (across datasets), aiming to enrich and reinforce the analysis shown in the previous section.

5.2.1 ISGD, RAISGD and RSISGD

The ISGD family of methods is not the top performer in all experiments except P10. RAISGD diagonal scores are slightly inferior to RSISGD's, and this occurs even though

RSISGD selects negative examples randomly, which is seen as a less refined technique than RAISGD - which may denounce some limitations in the sampling and evaluation strategies. As noted previously, ISGD presents higher mean diagonal scores than its variations RAISGD and RSISGD in both Amazon datasets experiments; RAISGD and RSISGD have increased performance over ISGD in the P10 and MLE experiments. Moreover, RSISGD's performance is the highest among methods in the P10 experiment, followed by RAISGD.

BWT scores for ISGD and its variations are all negative, i.e. forgetting occurs. There is no consistency in which method provides the least amount of forgetting. Overall, the results seem highly dependent on the combination of dataset and method - for the AKS and ADM datasets, ISGD's BWT score is lower than or equal to the variations' scores; and it is higher for the P10 and MLE datasets.

ISGD's FWT scores are the lowest in both P10 and MLE experiments. Except in the AKS and ADM experiments, RAISGD and RSISGD FWT scores are slightly higher than ISGD's. The methods' FWT scores for the ADM experiment are equal to zero.

5.2.2 BPRMF

BPRMF displays the lowest mean diagonal scores in all experiments. As noted before, its BWT and FWT scores cannot be fairly compared in the AKS and P10 datasets since its mean diagonal scores are orders of magnitude lower than the other methods ($*$ in Table 5.1). Even though the performance of ISGD and its variations are expected to be higher than BPRMF [7], this result may be an outcome of issues with the algorithm used and the evaluation strategy.

BPRMF presents the lowest mean diagonal results for the ADM and MLE datasets but in the same order of magnitude as other methods. Nevertheless, its BWT score is the second highest in both experiments while being positive in the MLE experiment. The method has improved performance in past holdouts by learning new interactions. The low performance of BPRMF on recent holdouts associated with the positive backward transfer may signal that it is a less plastic method than its counterparts. Finally, BPRMF displayed an FWT score of zero for the ADM dataset, similar to ISGD and its variations; the lowest score for the AKS experiment; and the second lowest for the MLE dataset. Thus, it had difficulty transferring information for future recommendations.

5.2.3 UKNN

UKNN has the highest mean diagonal scores for the AKS, ADM and MLE datasets. This result is counterintuitive given the algorithm's simplicity and previous results in [7]. Still, it also may signify that modelling the interactions of these datasets takes advantage of the memorization promoted by the method; moreover, these sampled datasets may be less dynamic than expected. In contrast, UKNN has the lowest mean diagonal score for the P10 dataset, even though it displays a score of the same magnitude as the remaining methods.

The method presents the highest BWT and FWT scores in all experiments. Its BWT scores are positive for the AKS, ADM and MLE experiments. As noted previously, while the score for the P10 experiment is negative, it is one order of magnitude lower than its counterparts, meaning it suffers less forgetting. It is the only method to achieve positive forward transfer in the ADM experiment while providing the highest FWT scores in the remaining experiments, the orders of magnitude being similar to other methods. The results suggest that the methods' characteristics aid in attenuating forgetting and improving forward transfer.

5.3 Limitations

The previous discussion provides insights into the capacity of streaming recommendation methods to retain useful information as new examples are learned. However, there are some limitations to this work that must be considered. This section aims to argue about said constraints, while it does not intend to be exhaustive.

The order of magnitude of the result scores shown in Table 5.1 hinders the discussion, given that it becomes difficult to argue that one method is more efficient than its counterparts in the performance and information transfer aspects. Moreover, the results of Vinagre [7] are favourable to the ISGD family of methods and not to UKNN. In any case, these outcomes may reflect issues with the assessment methodology, its components, the methods' efficiency, and the time limit to develop a concise body of work.

A different sampling methodology might provide disparate results. Picking other time intervals (i.e. years, months, weeks) to be included and threshold values for user interactions are complex tasks and depend on the dataset - it could result in essential changes in the datasets' characteristics and dynamics. Moreover, the datasets generally contain

an inconstant number of interactions in intervals, which could influence the information transfer assessment. The reduction of dataset size is necessary as experiments must be complete in a reasonable time, the main concern being the computational complexity of UKNN, which grows with both the number of users and items, as shown in Papagelis et al. [42]. In this sense, more efficient parallel computation techniques might be used in the future to attenuate the issue.

The assessment of streaming recommendation methods is often performed through prequential evaluation, where the model is tested in each interaction and learns afterwards. Performance assessment over holdouts with the last interaction of each user might not provide a fair and comparable appraisal. Other schemes where prequential evaluation is used might be studied.

This work simplifies the selection of hyperparameters, as it is performed using an initial parcel of the sampled datasets. The user-item dynamics ideally require that hyperparameters be adjusted over time, which is out of scope.

The results suggest that datasets are better modelled through different methods and that UKNN provides the least forgetting and increased forward transfer. Thus, it is essential to comprehend which dataset characteristics affect a method's performance over recent, previous and future interactions. Even more than that, and probably more complex, it is essential to understand what is being forgotten and remembered. The concepts modelled in incremental image classification models are well defined; in contrast, there is no such thing in the recommendation datasets. Future research can look for ways to measure dataset dynamics and determine and define concepts and their changes as streaming-based recommendation methods learn.

Chapter 6

Conclusion

This work presents a methodology for assessing information transfer in stream-based recommendation methods. The results provide insights into the information transfer behaviour of models with different learning procedures - ISGD and its variations RAISGD and RSIGD, BPRMF, and UKNN. Experiments are performed with four datasets from the e-commerce and music and movie streaming domains, fit for the recommendation task. A non-exhaustive discussion on the limitations of the project is also presented.

The results suggest that UKNN is less affected by forgetting, or negative backward transfer, than ISGD and BPRMF, at times being able to provide positive backward transfer scores, meaning learning new interactions had a positive effect on the performance of past holdouts. UKNN is also the top performer regarding forward transfer, indicating that memorization benefits future recommendations. This outcome may result from its stability compared to its counterparts, as the method caches factors used to compute the similarity between users and only alters factors relative to the active user. Moreover, it is impossible to conclude which method among ISGD and its variations suffers more forgetting, given that results seem highly dependent on the combination of dataset and method.

Overall, the recall values at previous holdouts have an inverse relation to the number of intervals a model learns since then, i.e. the further a model learns, the more it forgets from distant intervals. This relationship is more evident for ISGD and its variations and less evident for UKNN. Similarly, forward transfer scores are generally higher for closer future intervals and diminish as intervals are more distant (i.e. as user preferences change in the future).

The heatmaps provide a way to observe the performance of models in the past and future holdouts and complement the BWT and FWT scores, which are insufficient to assess information transfer as they are average scores. In addition, BWT is always relative to the *optimal* diagonal score, which can cause misinterpretations. Overall, the result heatmaps show that as each new interval is learned, forgetting diminishes the performance in previous holdouts, while UKNN is often an exception.

The limitations, discussed in more depth in Section 5.3, may be the consequence of several factors, such as the sampling and assessment methodologies, dataset characteristics, and method implementations. Even though these impediments cause concerns about the reproducibility and confidence of the results, the project provides an initial approach to analyzing information transfer in stream-based recommendation methods and valuable insights for future endeavours.

6.1 Future Work

In this section, the discussion and conclusion are reduced into bullet points that could serve as pointers for future research. More research is required to understand the phenomenon better and improve results in the context of stream-based recommendation methods. Specifically, future works could:

- Investigate different recommendation datasets from different domains and for extended periods;
- Study datasets' characteristics and dynamics from an information transfer perspective - e.g. which dataset characteristics make different models more susceptible to forgetting past information and retaining information for future recommendations?
- Deepen the assessment of sampling methods in order to retain dataset dynamics while being aware of computational time and space complexity and data availability over time;
- Evaluate different state-of-the-art stream-based recommendation methods and improve the runtime of current ones through parallelism and other available techniques;
- Consider the usage of different holdout creation schemes instead of retaining a user's last interaction in an interval;

- Consider the usage of prequential evaluation to assess information transfer instead of the holdout scheme presented in this work;
- Study techniques to select more adequate hyperparameters over time as the dataset dynamics evolve;
- Define concepts and create techniques to determine which ones are being altered, forgotten and remembered over time as stream-based recommendation methods learn.

Appendix A

Amazon Kindle Recall@20 Heatmaps

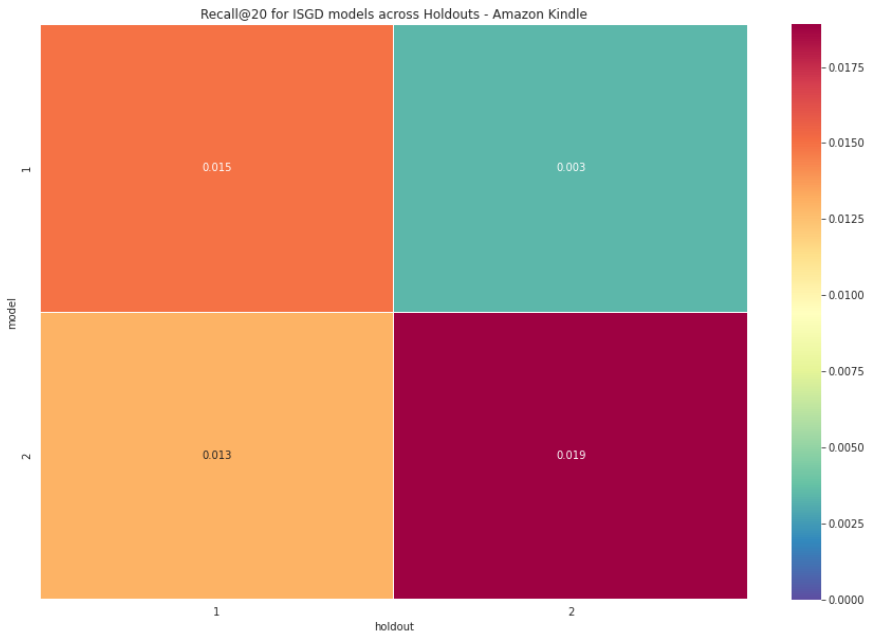


FIGURE A.1: ISGD recall@20 heatmap for Amazon Kindle dataset

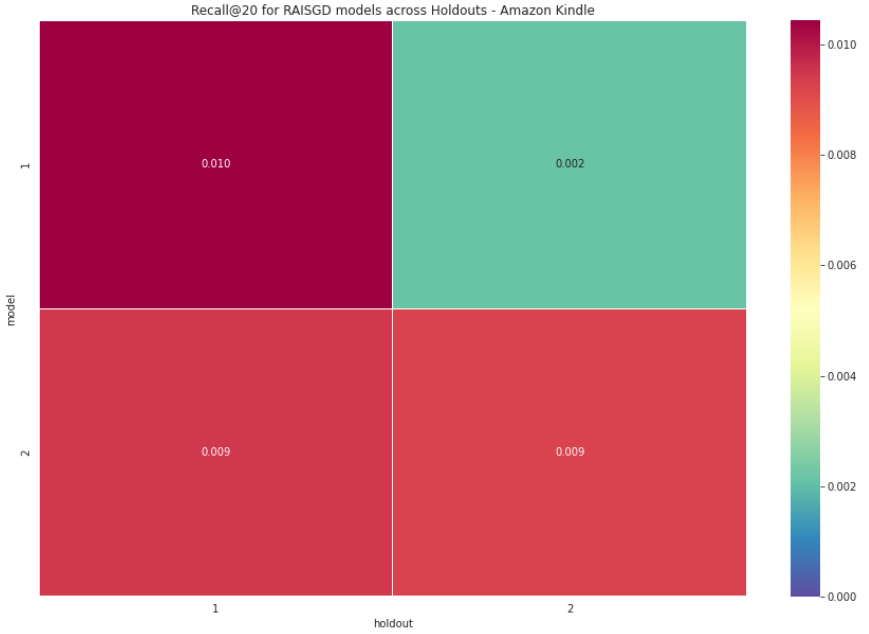


FIGURE A.2: RAISGD recall@20 heatmap for Amazon Kindle dataset

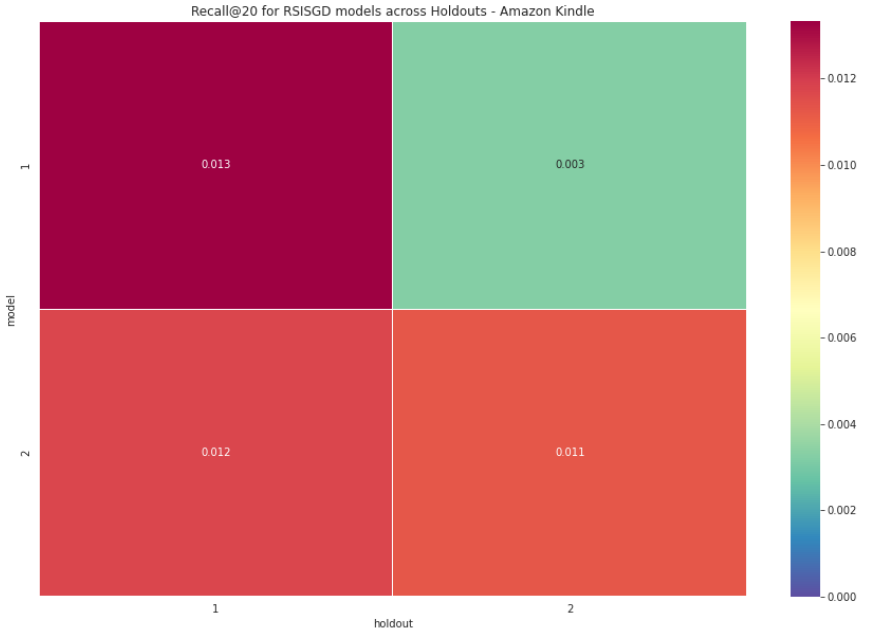


FIGURE A.3: RSISGD recall@20 heatmap for Amazon Kindle dataset

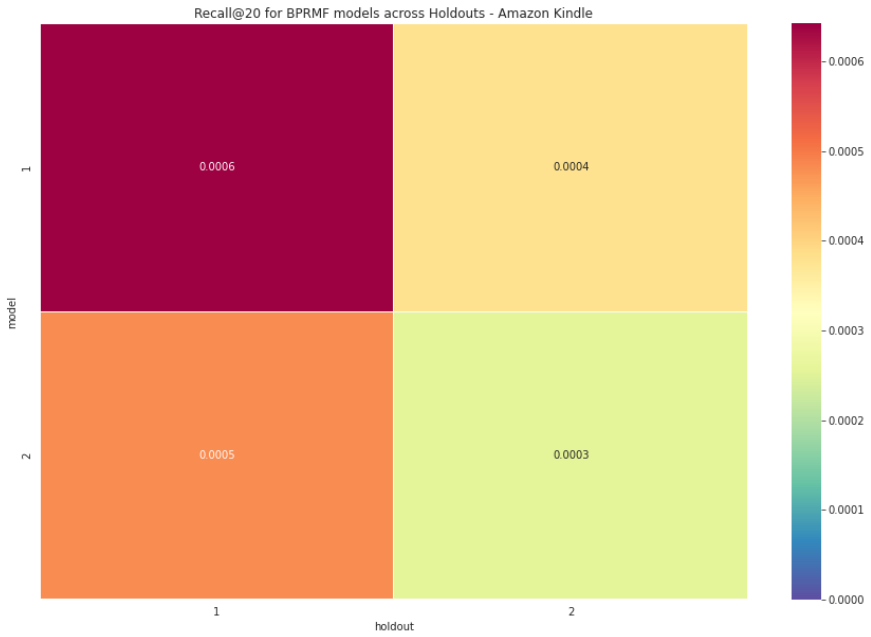


FIGURE A.4: BPRMF recall@20 heatmap for Amazon Kindle dataset

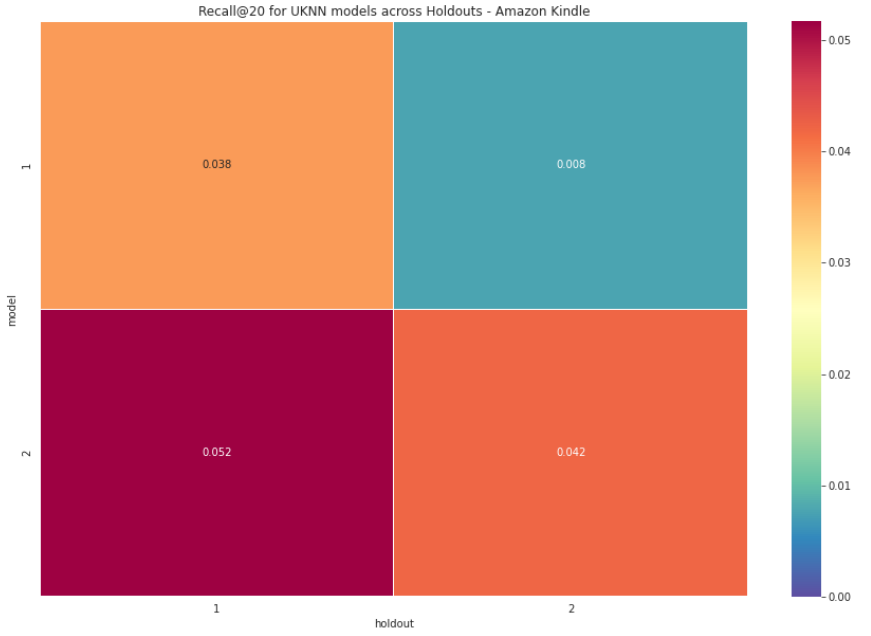


FIGURE A.5: UKNN recall@20 heatmap for Amazon Kindle dataset

Appendix B

Amazon Digital Music Recall@20 Heatmaps

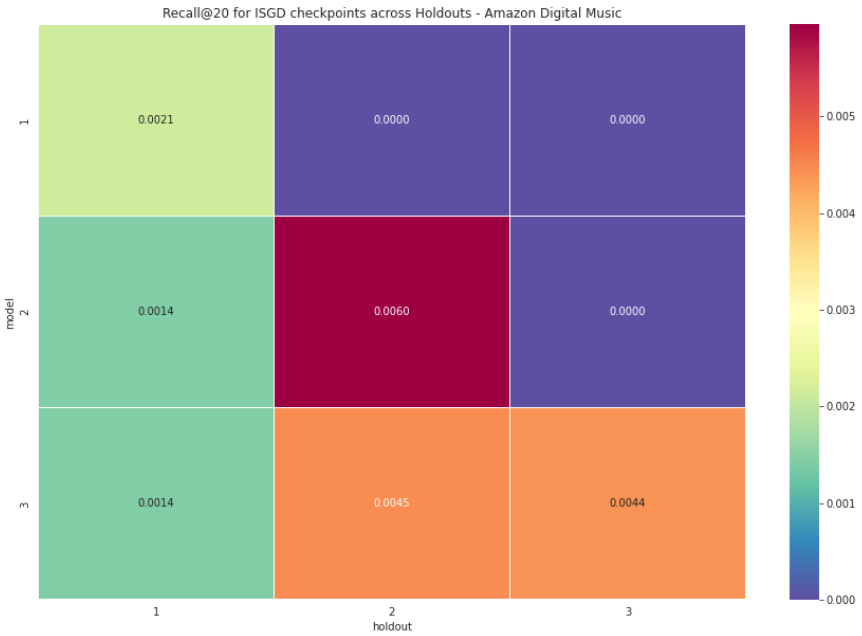


FIGURE B.1: ISGD recall@20 heatmap for Amazon Digital Music dataset

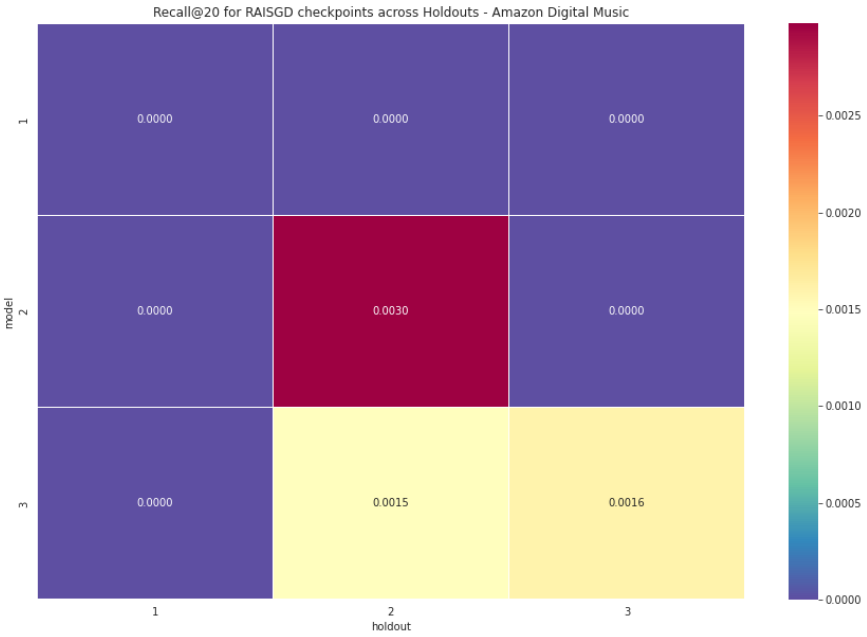


FIGURE B.2: RAISGD recall@20 heatmap for Amazon Digital Music dataset

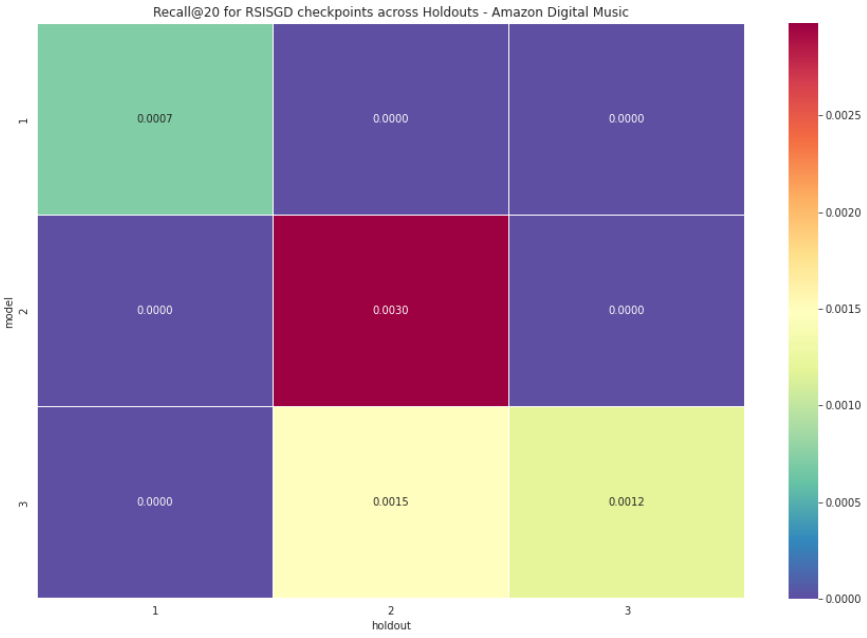


FIGURE B.3: RSISGD recall@20 heatmap for Amazon Digital Music dataset

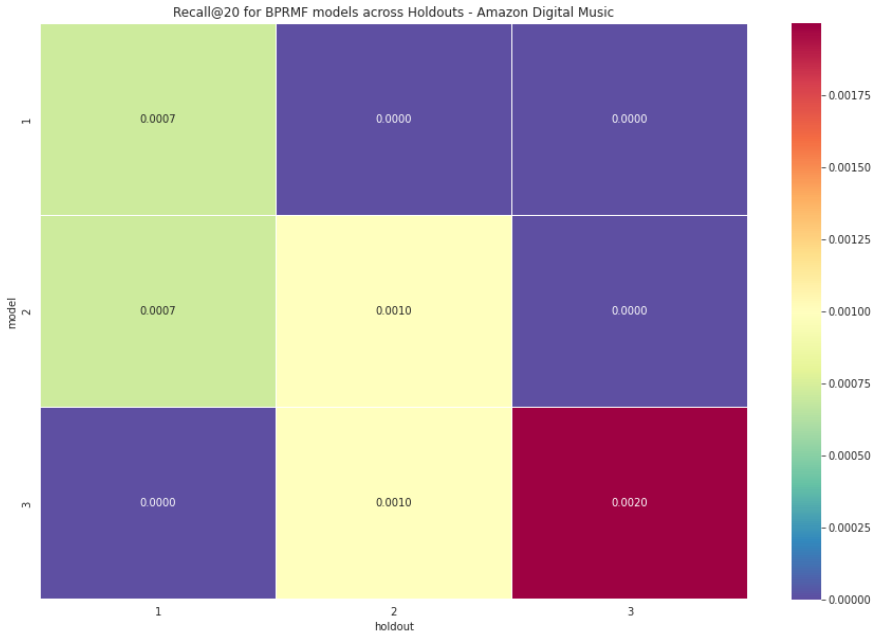


FIGURE B.4: BPRMF recall@20 heatmap for Amazon Digital Music dataset

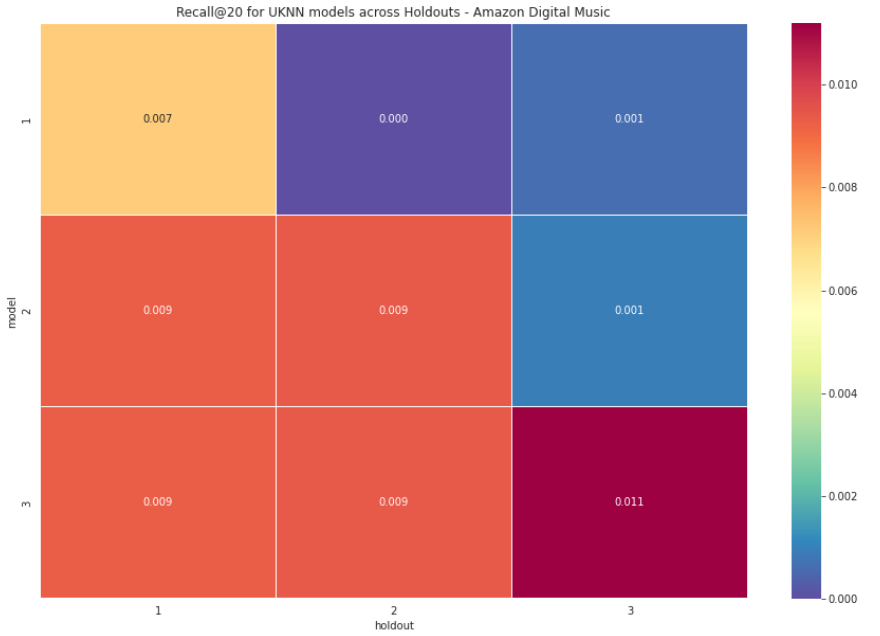


FIGURE B.5: UKNN recall@20 heatmap for Amazon Digital Music dataset

Appendix C

Palco 2010 Recall@20 Heatmaps

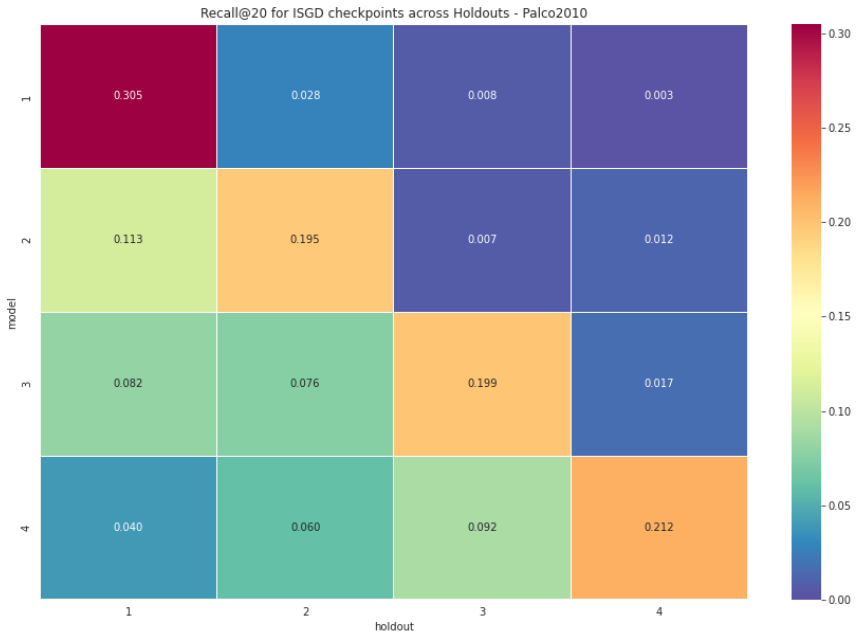


FIGURE C.1: ISGD recall@20 heatmap for Palco 2010 dataset

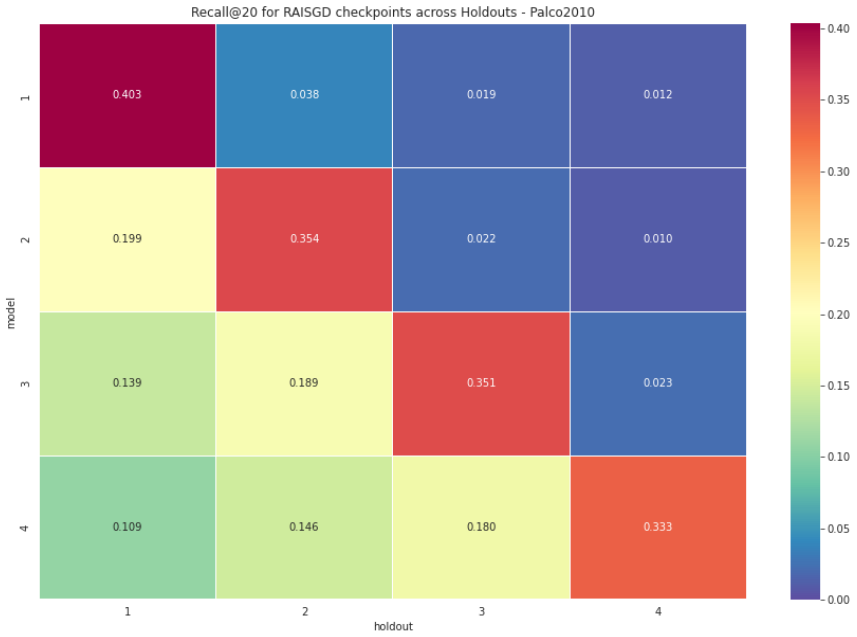


FIGURE C.2: RAISGD recall@20 heatmap for Palco 2010 dataset

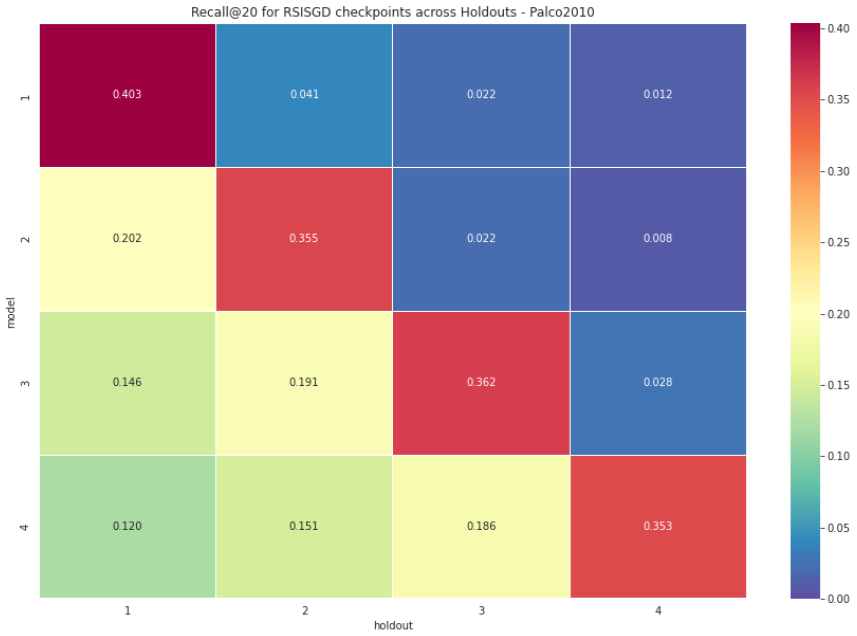


FIGURE C.3: RSISGD recall@20 heatmap for Palco 2010 dataset

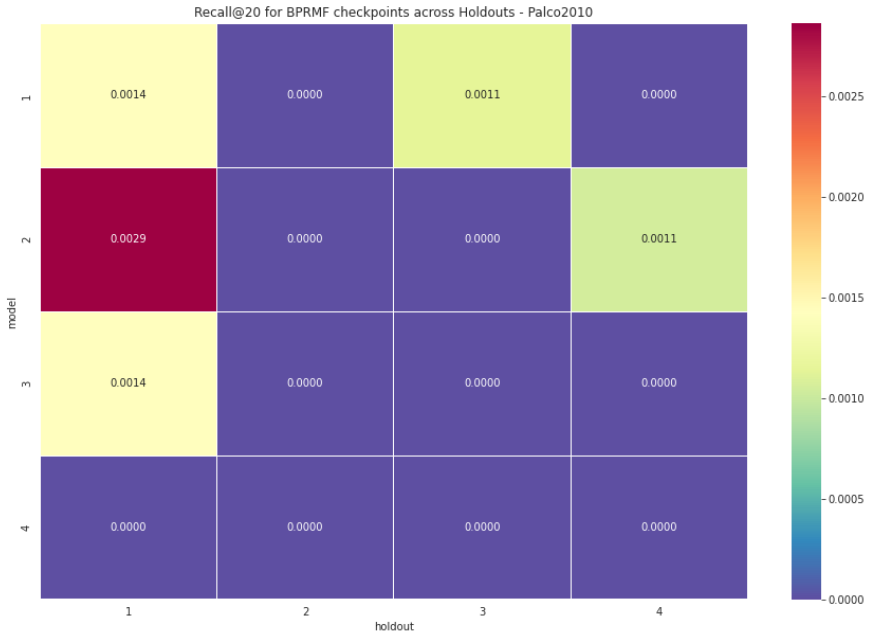


FIGURE C.4: BPRMF recall@20 heatmap for Palco 2010 dataset

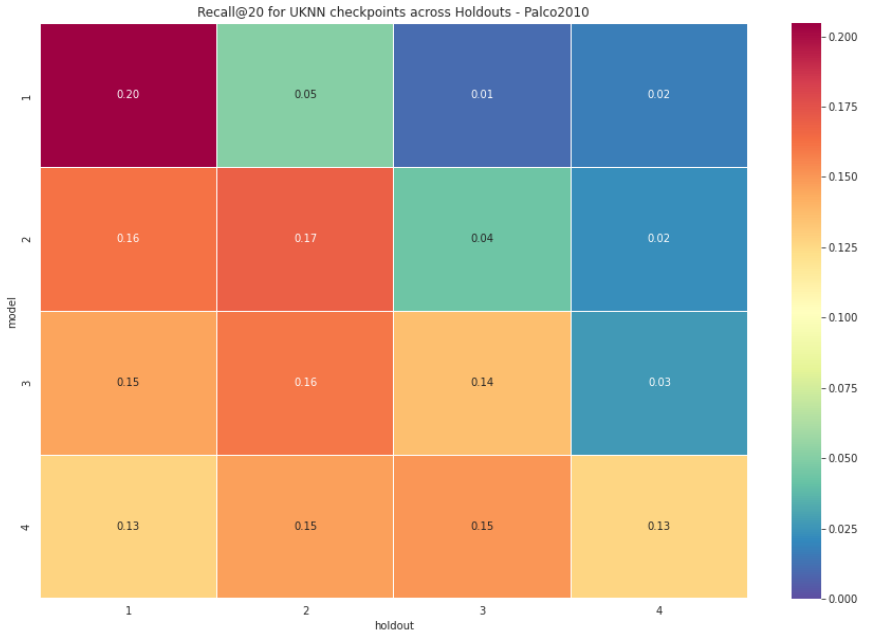


FIGURE C.5: UKNN recall@20 heatmap for Palco 2010 dataset

Appendix D

Movielens Recall@20 Heatmaps

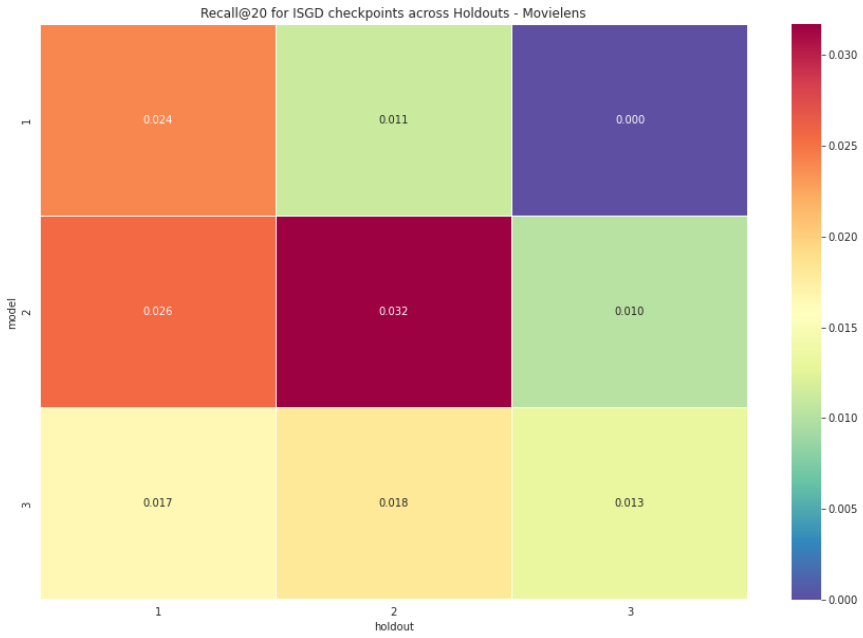


FIGURE D.1: ISGD recall@20 heatmap for Movielens dataset

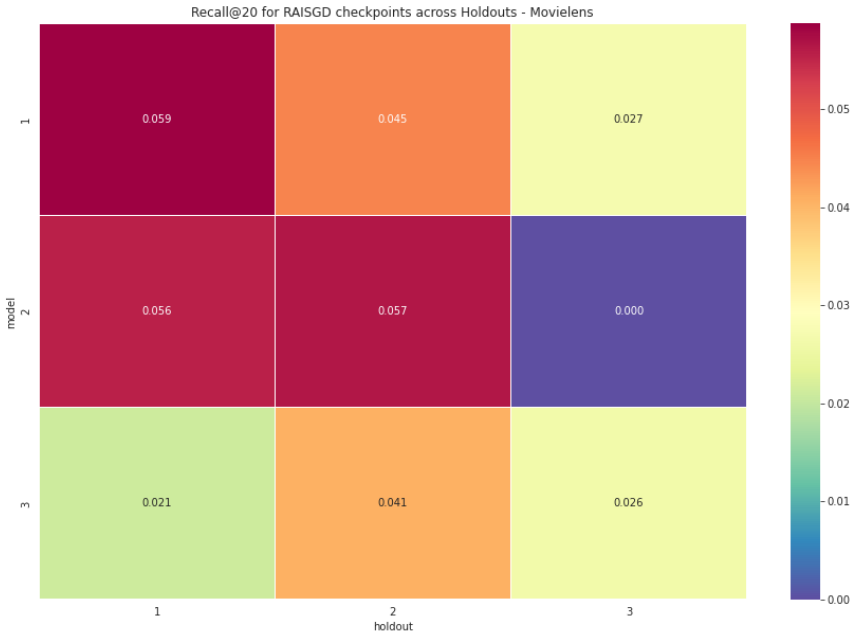


FIGURE D.2: RAISGD recall@20 heatmap for Movielens dataset

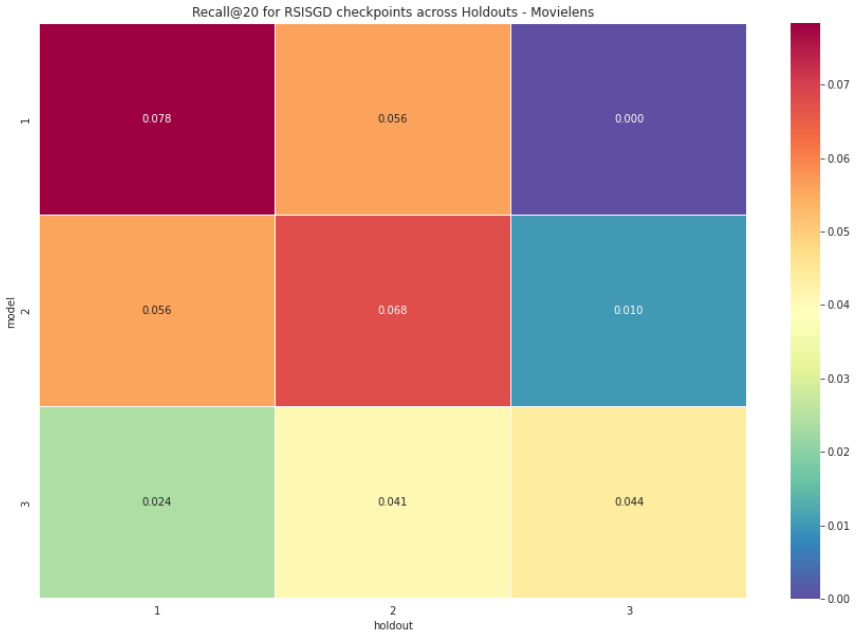


FIGURE D.3: RSISGD recall@20 heatmap for Movielens dataset

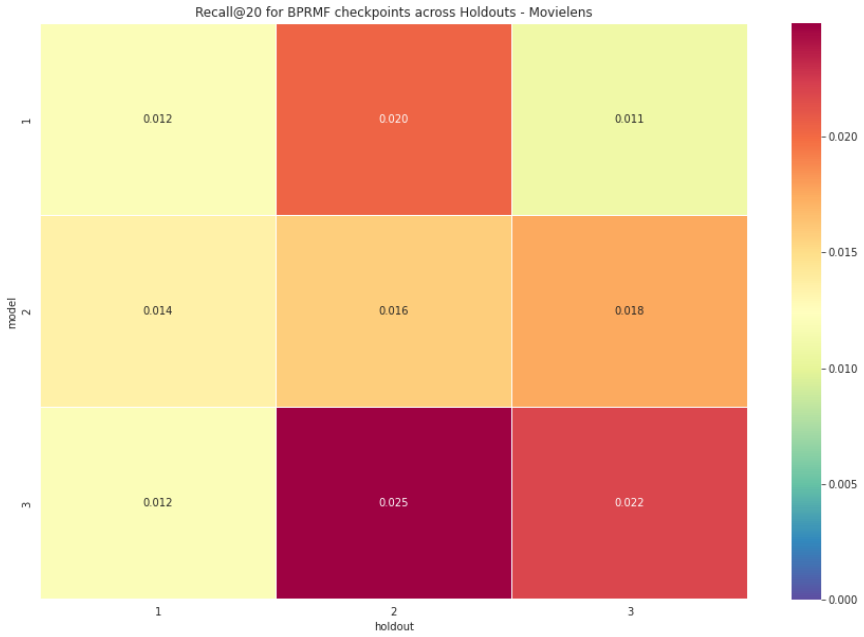


FIGURE D.4: BPRMF recall@20 heatmap for Movielens dataset

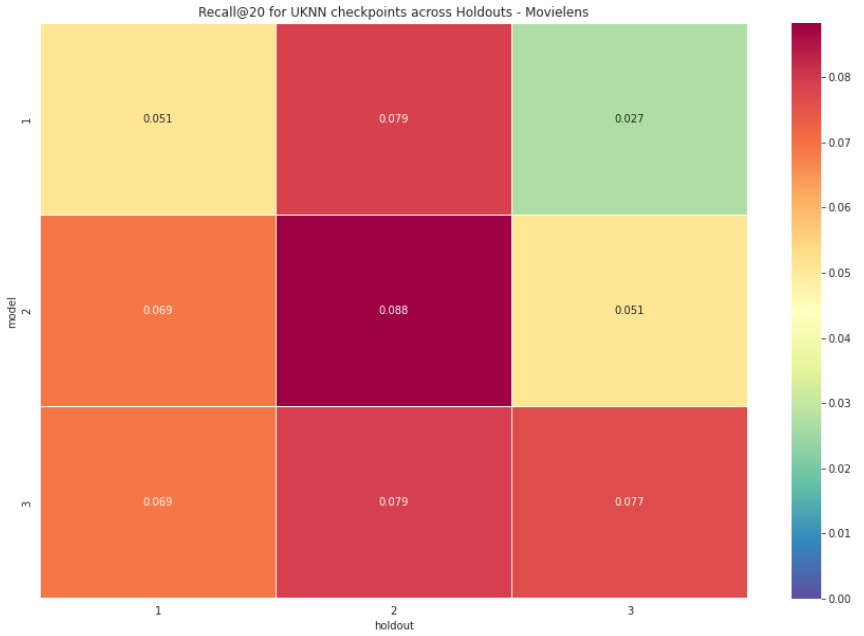


FIGURE D.5: UKNN recall@20 heatmap for Movielens dataset

Bibliography

[1] T. Mitchell, *Machine Learning*, ser. McGraw-Hill International Editions. McGraw-Hill, 1997. [Online]. Available: https://books.google.pt/books?id=EoYBngEACAAJ [Cited on pages 5 and 6.]

[2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*, ser. Springer Series in Statistics. Springer New York, 2009. [Online]. Available: https://books.google.pt/books?id=tVIjmNS3Ob8C [Cited on pages 5 and 6.]

[3] C. Aggarwal, *Recommender Systems: The Textbook*. Springer International Publishing, 2016. [Online]. Available: https://books.google.pt/books?id=GKjWCwAAQBAJ [Cited on pages 6, 7, 8, 9, 11, and 12.]

[4] F. Ricci, L. Rokach, and B. Shapira, *Introduction to Recommender Systems Handbook*. Boston, MA: Springer US, 2011, pp. 1–35. [Online]. Available: https://doi.org/10.1007/978-0-387-85820-3_1 [Cited on page 6.]

[5] J. Vinagre, A. M. Jorge, and J. Gama, "An overview on the exploitation of time in collaborative filtering," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, 2015. [Cited on page 10.]

[6] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, pp. 76–80, 2003.

[7] J. Vinagre, *Scalable adaptive collaborative filtering. Ph.D. thesis*. Universidade do Porto (Portugal), 2016. [Cited on pages 10, 29, 37, and 38.]

[8] A. Rajaraman and J. D. Ullman, *Recommendation Systems*. Cambridge University Press, 2011, p. 277–309. [Cited on page 10.]

[9] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021. [Cited on pages 10, 14, 17, and 20.]

[10] P. M. Domingos and G. Hulten, "Catching up with the data: Research issues in mining data streams," in *DMKD*, 2001. [Cited on page 10.]

[11] W. Wang, H. Yin, Z. Huang, Q. Wang, X. Du, and Q. V. H. Nguyen, "Streaming ranking based recommender systems," *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018. [Cited on page 10.]

[12] J. Vinagre, A. M. Jorge, and J. Gama, "Evaluation of recommender systems in streaming environments," *ArXiv*, vol. abs/1504.08175, 2015. [Cited on pages 11 and 12.]

[13] Z. F. Siddiqui, E. Tiakas, P. Symeonidis, M. Spiliopoulou, and Y. Manolopoulos, "xstreams: Recommending items to users with time-evolving preferences," in *WIMS '14*, 2014. [Cited on page 12.]

[14] J. Vinagre, A. M. Jorge, C. Rocha, and J. Gama, "Statistically robust evaluation of stream-based recommender systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, pp. 2971–2982, 2021. [Cited on page 12.]

[15] Z. Chen and B. Liu, "Lifelong machine learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 12, no. 3, pp. 1–207, 2018. [Cited on pages 12, 13, 14, and 15.]

[16] W. C. Abraham and A. V. Robins, "Memory retention – the synaptic stability versus plasticity dilemma," *Trends in Neurosciences*, vol. 28, pp. 73–78, 2005. [Cited on page 14.]

[17] P. Matuszyk, J. Vinagre, M. Spiliopoulou, A. M. Jorge, and J. Gama, "Forgetting techniques for stream-based matrix factorization in recommender systems," *Knowledge and Information Systems*, vol. 55, no. 2, pp. 275–304, 2018. [Cited on pages 14 and 15.]

[18] F. Yuan, G. Zhang, A. Karatzoglou, J. Jose, B. Kong, and Y. Li, "One person, one model, one world: Learning continual user representation without forgetting," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 696–705. [Cited on pages 14 and 21.]

[19] D. Peng, S. J. Pan, J. Zhang, and A. Zeng, "Learning an adaptive meta model-generator for incrementally updating recommender systems," in *Fifteenth ACM Conference on Recommender Systems*, 2021, pp. 411–421. [Cited on pages 14, 15, and 21.]

[20] Y. Zhang, F. Feng, C. Wang, X. He, M. Wang, Y. Li, and Y. Zhang, "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, pp. 1479–1488. [Cited on page 14.]

[21] Y. Xu, Y. Zhang, W. Guo, H. Guo, R. Tang, and M. Coates, "Graphsail: Graph structure aware incremental learning for recommender systems," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2861–2868. [Cited on pages 15 and 21.]

[22] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019. [Cited on pages 17 and 18.]

[23] F. Zhou and C. Cao, "Overcoming catastrophic forgetting in graph neural networks with experience replay," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4714–4722. [Cited on pages 17 and 20.]

[24] D. R. Ashley, S. Ghiassian, and R. S. Sutton, "Does the adam optimizer exacerbate catastrophic forgetting?" *arXiv preprint arXiv:2102.07686*, 2021. [Cited on pages 17 and 18.]

[25] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, vol. 30, 2017. [Cited on pages 18 and 19.]

[26] N. Díaz-Rodríguez, V. Lomonaco, D. Filliat, and D. Maltoni, "Don't forget, there is more than forgetting: new metrics for continual learning," *arXiv preprint arXiv:1810.13166*, 2018. [Cited on pages 18, 26, and 27.]

[27] J. Lovón-Melgarejo, L. Soulier, K. Pinel-Sauvagnat, and L. Tamine, "Studying catastrophic forgetting in neural ranking models," in *European Conference on Information Retrieval*. Springer, 2021, pp. 375–390. [Cited on page 19.]

[28] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 532–547. [Cited on page 19.]

[29] T. L. Hayes, R. Kemker, N. D. Cahill, and C. Kanan, "New metrics and experimental paradigms for continual learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2031–2034. [Cited on page 19.]

[30] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan, "Measuring catastrophic forgetting in neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018. [Cited on page 19.]

[31] V. Lomonaco and D. Maltoni, "Core50: a new dataset and benchmark for continuous object recognition," in *Conference on Robot Learning*. PMLR, 2017, pp. 17–26. [Cited on page 20.]

[32] W. Masarczyk, K. Deja, and T. Trzcinski, "On robustness of generative representations against catastrophic forgetting," in *International Conference on Neural Information Processing*. Springer, 2021, pp. 325–333. [Cited on page 20.]

[33] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4548–4557. [Cited on page 20.]

[34] K. Ahrabian, Y. Xu, Y. Zhang, J. Wu, Y. Wang, and M. Coates, "Structure aware experience replay for incremental learning in graph-based recommender systems," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 2832–2836. [Cited on page 21.]

[35] K. Ren, J. Qin, Y. Fang, W. Zhang, L. Zheng, W. Bian, G. Zhou, J. Xu, Y. Yu, X. Zhu *et al.*, "Lifelong sequential modeling with personalized memorization for user response prediction," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 565–574.

[36] Q. Pi, G. Zhou, Y. Zhang, Z. Wang, L. Ren, Y. Fan, X. Zhu, and K. Gai, "Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2685–2692.

[37] Y. Wang, Y. Zhang, and M. Coates, "Graph structure aware contrastive knowledge distillation for incremental learning in recommender systems," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 3518–3522.

[38] Y. Ouyang, J. Shi, H. Wei, and H. Gao, "Incremental learning for personalized recommender systems," *arXiv preprint arXiv:2108.13299*, 2021. [Cited on page 21.]

[39] F. Mi, X. Lin, and B. Faltings, "Ader: Adaptively distilled exemplar replay towards continual learning for session-based recommendation," in *Fourteenth ACM Conference on Recommender Systems*, 2020, pp. 408–413. [Cited on page 21.]

[40] J. Ni, J. Li, and J. McAuley, "Justifying recommendations using distantly-labeled reviews and fine-grained aspects," in *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 2019, pp. 188–197. [Cited on page 27.]

[41] C. Miranda and A. M. Jorge, "Incremental collaborative filtering for binary ratings," in *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, vol. 1. IEEE, 2008, pp. 389–392. [Cited on page 29.]

[42] M. Papagelis, I. Rousidis, D. Plexousakis, and E. Theoharopoulos, "Incremental collaborative filtering for highly-scalable recommendation algorithms," in *International symposium on methodologies for intelligent systems*. Springer, 2005, pp. 553–561. [Cited on pages 29 and 39.]

[43] J. Vinagre, A. M. Jorge, and J. Gama, "Fast incremental matrix factorization for recommendation with positive-only feedback," in *International conference on user modeling, adaptation, and personalization*. Springer, 2014, pp. 459–470. [Cited on page 29.]

[44] ——, "Collaborative filtering with recency-based negative feedback," in *Proceedings of the 30th annual ACM symposium on applied computing*, 2015, pp. 963–965. [Cited on page 29.]

[45] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," *ArXiv*, vol. abs/1205.2618, 2009. [Cited on page 29.]

[46] T. Kitazawa, "Flurs: Streaming recommendation in python," https://flurs.readthedocs.io/en/latest/index.html, August 2022. [Cited on page 29.]