Eduardo Jorge da Rosa Bürgel

ACCELERATED INCREMENTAL LISTWISE LEARNING TO RANK FOR COLLABORATIVE FILTERING

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação para a obtenção do Grau de Mestre em Ciência da Computação.
Orientadora: Prof[a]. Jerusa Marchi, Dr[a].
Coorientador: Prof. Eduardo Jaques Spinosa, Dr.

Florianópolis

2017

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Bürgel, Eduardo Jorge da Rosa
Accelerated Incremental Listwise Learning to Rank for Collaborative Filtering / Eduardo Jorge da Rosa Bürgel ; orientadora, Jerusa Marchi, coorientador, Eduardo Jaques Spinosa, 2017.
75 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós Graduação em Ciência da Computação, Florianópolis, 2017.

Inclui referências.

1. Ciência da Computação. 2. Sistemas de Recomendação. 3. Filtragem Colaborativa. 4. Aprendizagem de Máquina. 5. Aprendizagem Incremental. I. Marchi, Jerusa. II. Spinosa, Eduardo Jaques. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. IV. Título.

Eduardo Jorge da Rosa Bürgel

ACCELERATED INCREMENTAL LISTWISE LEARNING TO RANK FOR COLLABORATIVE FILTERING

Esta Dissertação foi julgada aprovada para a obtenção do Título de "Mestre em Ciência da Computação", e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Florianópolis, 16 de maio 2017.

Prof. José Luís Almada Güntzel, Dr.
Coordenador

Prof. Eduardo Jaques Spinosa, Dr.
Coorientador

Banca Examinadora:

Prof[a]. Jerusa Marchi, Dr[a].
Orientadora

Prof. Eduardo Camponogara, Dr.
Universidade Federal de Santa Catarina

Prof. Jomi Fred Hürner, Dr.
Universidade Federal de Santa Catarina

Prof[a]. Carina Friedrich Dornelles, Dr[a].
Universidade Federal de Santa Catarina

I would like to dedicate this work to my lovely mother Natalina da Rosa and faithful father Erno Bürgel (in memoriam).

ACKNOWLEDGEMENTS

I would like to thanks my advisor, Jerusa Marchi, and my co-advisor, Eduardo Jaques Spinosa, for all the support and specially for the patience during the journey of this work.

Many thanks to the colleagues of IATE/UFSC for the interesting philosophical and scientific discussions.

My sincere and sweet gratitude to my girlfriend Karina Rech, for always be present and for encouraging me to be my best.

"We cannot solve our problems with the same thinking we used when we created them."

Albert Einstein

RESUMO EXPANDIDO

Introdução

Os avanços nos computadores, nas tecnologias da informação e na Internet nos levaram a produzir um enorme volume de informações. O fácil acesso a dispositivos como desktops, notebooks, tablets e smartphones é uma realidade para uma parcela significativa da população mundial. As redes sociais e uma variedade de aplicações para produzir e compartilhar conteúdo são comuns em um dia típico de pessoas que vivem atualmente. A combinação desses fatores cria o que alguns autores denominam como sobrecarga de informações, ou seja, dificuldade de tomar decisões como resultado do grande volume de informações e da capacidade limitada de processamento.

Para superar o problema de sobrecarga de informações, sistemas de suporte à decisão foram propostos. O objetivo desses sistemas é ajudar as pessoas a lidar com a enorme quantidade de informações e melhorar os resultados de suas decisões. Sistemas de recomendação são um tipo de sistema de filtragem de informação no qual o objetivo é recomendar a informação mais relevante para a necessidade específica de um usuário. Para conseguir isso, o sistema de recomendação precisa ter algum conhecimento sobre a relevância da informação para qualificá-la como uma resposta à necessidade do usuário. A função de relevância permite ao sistema de recomendação classificar a informação em relevante ou não a um determinado usuário. Um grupo de técnicas, conhecido como filtragem colaborativa, foi proposto para inferir a função de relevância baseada nas avaliações feitas pelos usuários em relação aos items.

Com base na aprendizagem de máquina descrevemos como os sistemas de recomendação são capazes de aprender a relevância de algum item para resolver a necessidade de um usuário. Naturalmente, a noção de utilidade ou preferência das coisas nos permite ranqueá-las em alguma ordem de relevância. Essa ideia orientou o desenvolvimento de uma área especializada em aprendizagem de máquina, responsável pela aprendizagem de ranqueamento de informação.

Pouca atenção foi dedicada a escalabilidade dos algoritmos na literatura de aprendizagem de ranqueamento. No entanto, essa questão tornou-se cada vez mais importante hoje em dia, especialmente devido à disponibilidade de dados em grande escala que podem ser usados para o treinamento de modelos de aprendizagem de ranqueamento.

Um tópico que orienta o desenvolvimento de técnicas de aprendizagem de máquina está relacionado a disponibilidade dos dados no processo de apren-

dizagem. No paradigma incremental, os dados são recebidos continuamente pelo sistema e são usados para ampliar o conhecimento do modelo existente. A principal vantagem desta abordagem está relacionada à sua baixa complexidade de memória o que possibilita escalabilidade, uma vez que apenas a informação processada precisa estar em memória - no paradigma em lote, todas as informações precisam estar em memória. A desvantagem está no comprometimento da velocidade de aprendizagem devido ao fato de possuírmos um número restrito de informações.

A restrição entre baixa complexidade de memória e velocidade de aprendizado na aprendizagem incremental para classificar os modelos aplicados no problema de filtragem colaborativa é o foco deste trabalho. Nosso objetivo é melhorar a velocidade de aprendizado do algoritmo, mantendo sua baixa complexidade de memória e obtendo um ganho na precisão do modelo de forma mais rápida.

Objetivos

Propor uma versão acelerada de um algoritmo incremental de aprendizagem de ranqueamento no contexto do problema de filtragem colaborativa.

Os objetivos específicos do trabalho são: delinear e formalizar o problema de filtragem colaborativa no contexto de aprendizagem incremental; descrever e discutir os trabalho relacionados; propor um algoritmo acelerado aderente ao problema; avaliar o algoritmo proposto em instâncias reais do problema e analisar os resultados dos experimentos.

Metodologia

A investigação deste trabalho foi orientada pelo pressuposto de que a taxa de convergência do algoritmo base poderia ser melhorada pela adoção de uma técnica de aceleração de otimização.

Especificamente, aplicamos uma técnica de aceleração ao método de otimização do algoritmo base no contexto de aprendizagem de ranqueamento incremental aplicado ao contexto de filtragem colaborativa.

Resultados

Os resultados obtidos através dos experimentos realizados confirmam estatisticamente a efetividade da técnica de aceleração aplicada ao algoritmo base.

O principal benefício obtido nesse contexto pode ser visualizado no desempenho superior do algoritmo proposto nas iterações iniciais, demonstrando uma aceleração no processo de aprendizagem.

RESUMO

O enorme volume de informação hoje em dia aumenta a complexidade e degrada a qualidade do processo de tomada de decisão. A fim de melhorar a qualidade das decisões, os sistemas de recomendação têm sido utilizados com resultados consideráveis. Nesse contexto, a filtragem colaborativa desempenha um papel ativo em superar o problema de sobrecarga de informação. Em um cenário em que novas avaliações são recebidas constantemente, um modelo estático torna-se ultrapassado rapidamente, portanto a velocidade de atualização do modelo é um fator crítico. Propomos um método de aprendizagem de ranqueamento incremental acelerado para filtragem colaborativa. Para atingir esse objetivo, aplicamos uma técnica de aceleração a uma abordagem de aprendizado incremental para filtragem colaborativa. Resultados em conjuntos de dados reais confirmam que o algoritmo proposto é mais rápido no processo de aprendizagem mantendo a precisão do modelo.

Palavras-chave: Sistemas de Recomendação. Filtragem Colaborativa. Aprendizagem de Máquina. Aprendizagem Incremental.

ABSTRACT

The enormous volume of information nowadays increases the complexity of the decision-making process and degrades the quality of decisions. In order to improve the quality of decisions, recommender systems have been applied with significant results. In this context, the collaborative filtering technique plays an active role overcoming the information overload problem. In a scenario where new ratings have been received constantly, a static model becomes outdated quickly, hence the rate of update of the model is a critical factor. We propose an accelerated incremental listwise learning to rank approach for collaborative filtering. To achieve this, we apply an acceleration technique to an incremental collaborative filtering approach. Results on real-word datasets show that our proposal accelerates the learning process and keeps the accuracy of the model.

Keywords: Recommender Systems. Collaborative Filtering. Machine Learning. Incremental Learning.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

LIST OF SYMBOLS

CONTENTS

1 INTRODUCTION

Advances in computers and information technologies and the Internet led us to produce a huge volume of information. Easy access to computers like desktops, notebooks, tablets and smartphones is a reality to a significant portion of the world population. Social networks and a variety of applications to produce and share content are common in a typical day of people living today. The combination of these factors create what some authors denominate as *information overload* (EPPLER; MENGIS, 2004). Also known as *infobesity* or *infoxication* (HIMMA, 2007), it refers to the difficulty of making decisions as a result of the limited capacity of process information.

To overcome the information overload problem some *decision support systems* have been proposed. The goal of these systems is to help people to deal with the massive amount of information and to improve the results of their decisions. This can be done in two ways: helping in the rational process of decision, when all available information are considered in a prescriptive manner and the best decision is suggested; or focusing in the descriptive aspects, trying to reveal the process of decision. *Information filtering systems*, a kind of decision support system, assist judgements recommending relevant information based on the understanding of the user's decision process (HANANI; SHAPIRA; SHOVAL, 2001).

Recommender systems are a type of information filtering system whom goal is recommend the most relevant information for a user's specific necessity. To accomplish that, the recommender system needs to have some knowledge about the information relevance in order to qualify it as an answer to the user's necessity. Seems reasonable that the system needs to learn a *relevance function*, enabling it to classify the information in some manner. The capacity of learning the relevance function was the natural evolution of recommender systems (BOBADILLA et al., 2013).

A group of techniques, known as *collaborative filtering*, have been proposed to infer the relevance function based only on rating data (evaluations given by users to items). The underlying assumption is that if a person has the same opinion as other person about an item, this person is more likely to have the other's person opinion on a different item than that of a randomly chosen person (RICCI et al., 2011).

Learning something can be understood as the acquisition, modification or reinforcement of knowledge about some subject. With some other processes as attention, memory, judgment, reasoning, planning, problem solving and decision making, it constitutes cognition - all the mind process studied in cognitive science area and related to knowledge found in humans, ani-

mals and machines (MILLER, 2003). Based on artificial intelligence, or more specifically through machine learning, we build the bridge that explains how recommender systems are able to learn the relevance of some item to solve a user's necessity.

Naturally, the notion of utility or preference of things allow us to arrange them in some order of relevance. This idea guided the development of a specialized area in machine learning, responsible for *learning to rank* information. Initially developed in the field of information retrieval, learning to rank techniques started to be used in recommender system to improve recommendations (LIU, 2009).

In the literature of learning to rank, researchers have paid a lot of attention to the design of model, but somehow overlooked the scalability of algorithms. This however, has become a more and more important issue nowadays, especially due to the availability of large-scale data that can be used to train the learning to rank models (LIU, 2009).

A topic that guides the development of machine learning techniques is when data becomes available to the learning process. In the *incremental learning* paradigm, data is continuously received by the system and is used to extend the existing model's knowledge (FIAT, 1998). The main advantage of this approach is related to its low memory complexity which provides scalability, since that only the information been processed needs to be in memory - in the batch paradigm, all the information needs to be in memory. The drawback is that with less available information, the speed of learning is compromised which impacts on the accuracy of the model.

The trade-off between low memory complexity and speed of learning in incremental learning to rank models applied in collaborative filtering problem is the focus of this thesis. Our goal is to improve the speed of learning of the algorithm, keeping its low memory complexity and impacting in the accuracy of the model.

1.1 GENERAL OBJECTIVE

Propose an accelerated version of an incremental learning to rank algorithm in the context of the collaborative filtering problem.

1.2 SPECIFIC OBJECTIVES

Achieving the general objective of this thesis encompasses the following specific objectives:

- Delineate and formalize the collaborative filtering problem in the incremental learning context;
- Describe and discuss previous work related to the problem;
- Propose an accelerated algorithm adherent to the problem;
- Evaluate the proposed algorithm in real-world instances of the problem;
- Analyze the evaluation results and take conclusions.

1.3 SCOPE

The scope of this thesis is in the collaborative filtering techniques, in the area of recommender systems. The same warning is directed to the listwise learning to rank approach, in the machine learning area. Lastly, the scope is delimited by the incremental learning paradigm, based on online optimization techniques.

1.4 THESIS OVERVIEW

This thesis is organized in the follow structure. In Chapter 1, an introduction of the theme is presented, with general and specific objectives, and the scope of the thesis. In Chapter 2, the background knowledge, problem definition, and previous work are presented. In Chapter 3, the proposal is presented. In Chapter 4, we describe the design of the experiment and discuss the results of the experiments. Finally, in Chapter 5, we conclude and present the future work.

2 THEORETICAL FOUNDATION

Our objective at this chapter is to provide a concise and relevant background knowledge necessary to understand the scope of this thesis, as presented at Section 1.3, provide a formal definition of the incremental collaborative filtering problem and discuss the previous work in the area. In the last section, we justify the general objective of this thesis, as described at Section 1.1.

2.1 BACKGROUND

2.1.1 Recommender System

Recommender systems are software tools responsible for suggest an item of interest to a user. The items recommended can be of many different types, like movie, music, book, product or even a new relationship in the context of a social network. Considering this, recommender systems are defined as a type of information filtering system, removing redundant and unwanted information from the information stream managing the information overload problem (RICCI et al., 2011). To accomplish this objective, a *relevance function* that describes the relevance of an item to a user is inferred and used to select the most relevant items among a huge set of it (RICCI et al., 2011).

Researching in recommender systems field is a interdisciplinary effort including disciplines like cognitive science, artificial intelligence, machine learning, data mining, mathematics, and statistics. Particular techniques are used in the design of a recommender system and are grouped according to the data that is used in the recommendation process. Recommender systems can be classified depending the type of information used to infer the model as: content-based, collaborative filtering, context-aware, knowledge-based and hybrid (JANNACH et al., 2010).

The main concept in the content-based approach is the user's profile. It is built with content and metadata information of the items that user has liked or interacted in the past. This profile indicates prefered items of user and is used to recommend new items that are in accordance with it. In the recommendation process, the search for new items to recommend are made looking for items with an elevate degree of similarity with the user's profile (JANNACH et al., 2010).

The collaborative filtering approach uses only data from ratings of

users about items. This ratings can be in the form of numerical, binary (like or dislike) or unary values (an interaction with an item). There is two main approaches used in collaborative filtering: the memory-based and the model-based (JANNACH et al., 2010).

In the memory-based or neighborhood approach, the data is kept in memory and an algorithm is used to find out similarities, based on a metric, between users and items. This approach can still be divided into user-based and item-based, depending whether the used similarity is in relation of the user or the item.

In the model-based a model is built based on the ratings that represents the relevance of an item to a user. Many techniques from machine learning are used to build this kind of models. Once the model is built, it's possible to use it to predict the relevance of an unseen item to the user and recommend it or not.

The context-aware recommender systems are those in which the contextual information is used in the process of recommendation. Contextual information like space and time can influence a recommendation. There are three different paradigms to incorporate contextual information in the recommendation process: contextual pre-filtering, post-filtering, and modeling (JANNACH et al., 2010). Those paradigms describe the moment that the contextual information is used in the recommendation process. In the pre-filtering, the items are filtered based on the contextual information before the recommendation. Otherwise, in the post-filtering, the items are filtered based on the contextual information after the recommendation. In the modeling paradigm, the contextual information is used directly in the model.

Knowledge-based recommender systems are used in specific knowledge domains and are based on how some items satisfy users needs (JANNACH et al., 2010). One case of this kind of system are the cased-based, where a similarity function evaluates how much the user needs (problem) match the recommendations (solution). Another type is the constraint-based, where instead of using a similarity function the system explores a knowledge base with rules about how to relate user needs with recommendations.

In the hybrid context, two or more aforementioned approaches are combined. The goal of this combination is to use the advantages of some approach to fix the disadvantages of the other. Diverse techniques to make this combination have been proposed in the literature (JANNACH et al., 2010).

2.1.2 Machine Learning

Inductive reasoning is a kind of reasoning in which the premises are viewed as evidence of the truth of the conclusion. Differently from deductive reasoning, where the conclusion is ensured, in inductive reasoning the conclusion is probable, based on evidences. Generalization is a type of inductive reasoning that starts from a premise about a sample to a conclusion about the population. The inductive reasoning is inherently uncertain (MITCHELL, 1997).

Machine learning is the research area in artificial intelligence that tries to create or reproduce the human ability to learn in machines. To achieve this, algorithms that generalize from data are studied. These algorithms induce a model from data that is used in prediction or decision making (MITCHELL, 1997).

One informal definition of machine learning is that: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" (MITCHELL, 1997). In this definition, the experience E is viewed as the available data related to the task T learned by the algorithm and the performance measure P is the evaluation metric used to evaluate the machine learning algorithm. The main objective of a machine learning algorithm is to have a good performance when exposed to unseen sample of data related to task T. Overfitting is a concept in machine learning that describes when an algorithm have a good performance with available data but a poor performance with unseen data.

There are three main divisions in machine learning, depending of the feedback available to the system: supervised learning (labeled data), unsupervised learning (unlabeled data) and reinforcement learning (MOHRI; ROSTAMIZADEH; TALWALKAR, 2012).

In the supervised learning, during the training phase, the inputs and outputs are presented to the system. Using the inputs the system can generate a hypothesized output and compare this to the true output. This comparison generates a feedback that is used to learn the correct model.

Unlike the previous approach, in unsupervised learning only the inputs are available to the system. There is not the true output to the system compare and guide the learning process. Thereby, the main goal of the algorithm is find hidden structure in data.

Reinforcement learning is the paradigm of machine learning where the system interacts with the environment with some objective and without a teacher guiding the actions that system should take at each moment. In this case, the system needs to learn the set of best actions to achieve its objective.

A formal theory in the scope of supervised learning is the statistical learning theory (VAPNIK; VAPNIK, 1998). From the statistical learning theory, the principle of empirical risk minimization with regularization defines a form of finding a predictive function based on data as in Equation 2.1.

$$\arg\min_{h \in \mathscr{H}} \frac{1}{d} \sum_{k=1}^{d} \mathscr{L}(h(x_k), y_k) + \theta r(h) \tag{2.1}$$

The $\mathscr{H}$ is the hypothesis space of possible learnable functions h. The size of training set is defined by d. The loss function $\mathscr{L}$ measures the difference between the predictable value $h(x)$, where x is the input data, and the real output value y. The regularization term $r(h)$ controls overfitting and is configured by the parameter θ.

Classification and regression are two common tasks performed by the algorithms in the supervised learning paradigm (MOHRI; ROSTAMIZADEH; TALWALKAR, 2012). The main difference between both is that, in classification the output is discrete and in regression the output is continuous. Common tasks in unsupervised learning include grouping (clustering), summarization (dimensionality reduction) and association (association rules) (MOHRI; ROSTAMIZADEH; TALWALKAR, 2012). Clustering is the task where the objects are grouped according to their similarity. Dimensionality reduction is the task of reducing the number of considered variables generally before the use of another machine learning technique. In association rules, the objective is to discover valuable relationships between variables in the data.

Learning to rank is the application of techniques from machine learning in the construction of ranking models (LIU, 2011). The algorithm infers the ranking model based on training data (lists with some partial order specified between items). This order is commonly induced by giving a numerical or ordinal score or a binary judgment for each item. The purpose of the model is to produce a permutation of items in an unseen list, based on the generalization inferred of the seen lists in the training phase. There are three main approaches in the learning to rank area, depending in how the list is evaluated in the learning process: pointwise, pairwise and listwise (LIU, 2011).

The pointwise approach of learning to rank can be approximated by a regression problem. For each item in the list, an evaluation is predicted and the list is sorted in a decreasing order. Classical techniques of supervised learning can be applied for this purpose.

The pairwise approach approximates the learning to rank problem learning a binary classifier. The list is sorted based on the classification of pairs of items by relevance. The goal in this case is to minimize the average number of inversions during the ranking.

The main difference of the listwise approach to the others (pointwise and pairwise) is in the evaluation during the learning phase. In the time that the pointwise tries to predict the score of an item and the pairwise tries to predict the preference of an item in relation to another, the listwise approach optimize directly over the evaluation of the entire predicted list.

Machine learning algorithms can also be divided considering the moment that data is available, in incremental or online learning and batch or offline learning (BORODIN; EL-YANIV, 2005). Incremental machine learning is a method in which data become available in a sequential order and is used gradually to update the model. In the batch machine learning approach, the entire data is available in the learning phase and the model is inferred just once.

Another possible division of machine learning considers the underlying theory that support the algorithms. Knowledge from mathematical optimization, probability theory, information theory and, search techniques are common examples found in machine learning algorithms. One important example, related to the objective of this thesis, is the information theory.

From information theory (MACKAY, 2003), the concept of cross entropy $H(p,q)$ defines the measure of divergence between two discrete probability distributions $p(x)$ and $q(x)$, where x is a discrete random variable, as defined in Equation 2.2.

$$H(p,q) = -\sum_{x} p(x) \log q(x) \tag{2.2}$$

2.1.3 Mathematical Optimization

Mathematical optimization is an area of applied mathematics concerned with minimizing or maximizing the value of a function in a defined domain regarding some constraints. A general minimization optimization problem is represented by Equation 2.3.

$$\begin{aligned} &\arg\min_{x} && f(x) \\ &\text{subject to} && g_j(x) \leq 0, j \in \{0,\ldots,k\} \end{aligned} \tag{2.3}$$

The function $f(x)$ is known as objective function or loss function, in case of a minimization problems, or as utility function or fitness function, in case of maximization problems. The constraints, defined by inequalities $g(x)$, are restrictions on the domain and determine the search space composed by the candidate or feasible solutions. A feasible solution that minimizes or maximizes the loss function or the utility function, respectively, is known

as optimal solution. The area of convex programming studies the problems where the domain is a convex set (BOYD; VANDENBERGHE, 2004).

Problems in the area of mathematical optimization can be classified by the type or characteristics of the objective function and the inequalities or equalities constraints. For example, problems without constraints are known as unconstrained optimization problems. Problems with more than one objective function are known as multi-objective optimization problems. And the multi-modal optimization problems are the problems with more than one optimal solution (CHONG; ZAK, 2013).

Different techniques can be used to find the solution of an optimization problem. The direct methods that terminate in a finite number of steps can find an exact solution - in the absence of rounding errors. The iterative methods that approximate numerically the optimal solution, as the Newton's method or the gradient descent method. And the heuristics, that generate solutions without guarantee of convergence, as genetic algorithms (CHONG; ZAK, 2013).

Iterative methods are procedures that generate a sequence of improving approximate solutions for a problem, in which the actual solution is derived from the previous one (BURDEN; FAIRES, 2011). In the context of mathematical optimization, an iterative method is convergent if the corresponding sequence of generated solutions converges from the initial approximations to the optimal solution. Two important mathematical results are considered when analyzing iterative methods: proof of convergence and rate of convergence. The first, proof of convergence, guarantees that the iterative method converges to the optimal solution. The second, rate of converge, determines the speed of convergence.

Iterative methods in mathematical optimization differ according to whether they evaluate Hessians, gradient, or function values. The Hessian is a square matrix of second-order partial derivatives of a function, used in optimization problems within Newton-type methods. The gradient is the vector of first-order partial derivatives of a function, used in gradient based methods. Some methods do not require the gradient of the function and evaluate only the function values directly, as the pattern search methods (CHONG; ZAK, 2013).

A classical gradient method is the gradient descent method, that takes steps proportionally to the negative of the gradient of the function at the current point, to find the minimum of the function (BOTTOU, 2010). In Equation 2.4, we observe that the $f(x)$ decreases in the direction of the negative gradient $\nabla f(x)$ considering a step size γ.

$$x_{k+1} = x_k - \gamma \nabla f(x_k) \tag{2.4}$$

The rate of convergence of the gradient descent method is $O(1/k)$, where k is the number of iterations.

Online optimization is a field of mathematical optimization that deal with optimization problems with incomplete information - approximated Hessians and approximated gradients or subgradient (SHALEV-SHWARTZ et al., 2012). Two online optimization methods are important in the context of this thesis: the stochastic gradient descent (SGD) method and the dual averaging (DA) method.

Firstly, stochastic gradient descent method, also known as incremental gradient descent, is a stochastic approximation of the gradient descent method (BOTTOU, 2010). In SGD, the true gradient $\nabla f(x)$ is approximated by a gradient at a single point. The rate of convergence of the stochastic gradient descent method is $O(1/\sqrt{k})$, where k is the number of iterations.

Secondly, the dual averaging method determines the next point in the series by solving a simple optimization problem that involves the average of all past subgradients (NESTEROV, 2009). The rate of convergence of the dual averaging method is $O(1/k^2)$, where k is the number of iterations.

2.2 PROBLEM DEFINITION

In order, to succinctly define the incremental collaborative filtering problem, consider the following definitions.

Definition 1 (Set of Users)
Consider the set of users $\mathscr{U} = \{u_1, \ldots, u_m\}$, where the element u is a user of the system.

Definition 2 (Set of Items)
Consider the set of items $\mathscr{I} = \{i_1, \ldots, i_n\}$, where the element i is an item of the system.

Definition 3 (Set of Evaluations)
Consider the set of evaluations $\mathscr{E} = \{e_1, \ldots, e_o\}$, where the element e is an evaluation and represents the relevance of an item to a user.

Definition 4 (Rating)
A rating is a 3-tuple (u, i, e) that corresponds to the evaluation $e \in \mathscr{E}$ given by a user $u \in \mathscr{U}$ to an item $i \in \mathscr{I}$.

Definition 5 (Set of Ratings)
Consider the set of all ratings $\mathscr{R} = \{(u_j, i_k, e) | j = 1, \ldots, m; k = 1, \ldots, n; e \in \mathscr{E}\}$, where the element is a rating.

Definition 6 (Set of Known Ratings)
Consider the set of known ratings $\mathscr{Q}$, where $\mathscr{Q} \subseteq \mathscr{R}$ and corresponds to the observed ratings.

Definition 7 (Set of Unknown Ratings)
Consider the set of unknown ratings $\mathscr{S}$, where $\mathscr{S} = \mathscr{R} \setminus \mathscr{Q}$ and corresponds to the unobserved ratings.

Definition 8 (Relevance Function)
Consider the relevance function $f : \mathscr{U} \times \mathscr{I} \to \mathscr{E}$, that relates the set of inputs of ordered pairs (u,i) given by the cartesian product $\mathscr{U} \times \mathscr{I}$ with the set of outputs $\mathscr{E}$.

Based on the above definitions, we can describe the incremental collaborative filtering problem.

Problem
The collaborative filtering problem can be viewed as a matrix decomposition problem. Consider that the set $\mathscr{R}$ can be represented as a matrix, as in Equation 2.5.

$$\mathscr{R} = \begin{array}{c} \\ u_1 \\ u_2 \\ \vdots \\ u_m \end{array} \begin{array}{c} \begin{array}{cccc} i_1 & i_2 & \cdots & i_n \end{array} \\ \begin{pmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1} & e_{m2} & \cdots & e_{mn} \end{pmatrix}_{m \times n} \end{array} \tag{2.5}$$

The objective is to find out two matrices U and I of latent features, based on known ratings $\mathscr{Q}$, that better approximate $\mathscr{R}$ as in Equation 2.6. A latent feature l is a feature that is not directly observed but is rather inferred from other variables that are observed. The latent features are used to describe a user and an item. The matrix U with dimensions $m \times p$ represents the user's latent features where the row U_u is the user's latent feature vector. The matrix I with dimensions $p \times n$ represents the item's latent features where the column I_i is the item's latent feature vector. The number of latent features is defined by p.

$$\mathscr{R} = \begin{array}{c} \\ u_1 \\ u_2 \\ \vdots \\ u_m \end{array} \begin{array}{c} \begin{array}{cccc} l_1 & l_2 & \cdots & l_p \end{array} \\ \begin{pmatrix} l_{11} & l_{12} & \cdots & l_{1p} \\ l_{21} & l_{22} & \cdots & l_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ l_{m1} & l_{m2} & \cdots & l_{mp} \end{pmatrix}_{m \times p} \end{array} \times \begin{array}{c} \\ l_1 \\ l_2 \\ \vdots \\ l_p \end{array} \begin{array}{c} \begin{array}{cccc} i_1 & i_2 & \cdots & i_n \end{array} \\ \begin{pmatrix} l_{11} & l_{12} & \cdots & l_{1n} \\ l_{21} & l_{22} & \cdots & l_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{p1} & l_{p2} & \cdots & l_{pn} \end{pmatrix}_{p \times n} \end{array} \tag{2.6}$$

Once we have matrices U and I, we can predict the unknown evaluations through the relevance function f with the user's feature vector and the item's feature vector. Consequently, we can predict the evaluation of all unknown ratings of the set $\mathscr{S}$.

The incremental version of the collaborative filtering problem determines that the elements of set $\mathscr{Q}$ become available gradually over time. In this case, the problem is extended by the requirement that existing model's knowledge needs to absorb new element without consuming all the available set $\mathscr{Q}$ again.

2.2.1 Problem Example

In order to clarify the problem definition, we use a simple and brief example. Consider that the set of all ratings is represented by the matrix $\mathscr{R}$ in Equation 2.7.

$$\mathscr{R} = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{array} \begin{array}{c} \begin{array}{cccc} i_1 & i_2 & i_3 & i_4 \end{array} \\ \begin{pmatrix} 2.5 & 2.0 & 2.5 & 2.5 \\ 4.0 & 3.0 & 3.5 & 3.0 \\ 3.5 & 3.0 & 4.0 & 4.5 \\ 5.0 & 4.0 & 5.0 & 5.0 \end{pmatrix}_{4\times4} \end{array} \tag{2.7}$$

In the learning process, we only have access to a subset of the matrix $\mathscr{R}$, as described in matrix $\mathscr{Q}$ by Equation 2.8.

$$\mathscr{Q} = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{array} \begin{array}{c} \begin{array}{cccc} i_1 & i_2 & i_3 & i_4 \end{array} \\ \begin{pmatrix} & 2.0 & & \\ 4.0 & 3.0 & & 3.0 \\ & & 4.0 & \\ & 4.0 & & 5.0 \end{pmatrix}_{4\times4} \end{array} \tag{2.8}$$

Decomposing the matrix $\mathscr{Q}$ in matrices U and I we can find the latent factors that describe the characteristics of both users and items, as in Equation 2.9 and Equation 2.10.

$$U = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{array} \begin{array}{c} \begin{array}{cc} l_1 & l_2 \end{array} \\ \begin{pmatrix} 1.0 & 1.0 \\ 2.0 & 1.0 \\ 1.0 & 2.0 \\ 2.0 & 2.0 \end{pmatrix}_{4\times2} \end{array} \tag{2.9}$$

$$I = \begin{matrix} \\ l_1 \\ l_2 \end{matrix} \begin{matrix} \begin{matrix} i_1 & i_2 & i_3 & i_4 \end{matrix} \\ \begin{pmatrix} 1.5 & 1.0 & 1.0 & 0.5 \\ 1.0 & 1.0 & 1.5 & 2.0 \end{pmatrix}_{2\times4} \end{matrix} \tag{2.10}$$

Based on the latent features of users and items, we can predict the unknow ratings. For example, suppose we want to discover the unknow rating of user u_3 for item i_1. To achieve that, we need to combine the user u_3 latent feature vector with the item i_1 latent feature vector as in Equation 2.11.

$$u_3 \begin{matrix} \begin{matrix} l_1 & l_2 \end{matrix} \\ \begin{pmatrix} 1.0 & 2.0 \end{pmatrix}_{1\times2} \end{matrix} \times \begin{matrix} \begin{matrix} i_1 \end{matrix} \\ \begin{matrix} l_1 \\ l_2 \end{matrix} \begin{pmatrix} 1.5 \\ 1.0 \end{pmatrix}_{2\times1} \end{matrix} = 3.5 \tag{2.11}$$

We can repeat this process with the other unknow ratings of user u_3, combining its latent feature vector with the i_2 and i_4 item latent feature vector. In the end, we have all the ratings of user u_3 as in Equation 2.12.

$$u_3 \begin{matrix} \begin{matrix} i_1 & i_2 & i_3 & i_4 \end{matrix} \\ \begin{pmatrix} 3.5 & 3.0 & 4.0 & 4.5 \end{pmatrix}_{1\times4} \end{matrix} \tag{2.12}$$

We can recommend items sorting the vector in descending order of evaluation. In this example, the recommendation would be i_4, i_3, i_1 and i_2.

2.3 PREVIOUS WORK

In this section, we present the previous work in the incremental collaborative filtering area. Considering the literature review, we organize the previous work based on how the relevance function is modeled, which is the incremental learning approach and how the proposed algorithm is evaluated.

Some concepts exposed are not directly in the scope of this thesis. Strictly, the relevance functions - graphical model, probabilistic reward and, knn; learning models - gaussian process and multi-armed bandit; online algorithm - greedy strategy, thopmson sampling and, temporal difference learning or temporal dynamics; and evaluation metrics - RMSE, Recall, MAE, CTR and, MAP. More information about this topics can be found directly in the cited paper.

In Table 1, we present previous work by the relevance function model and ranking approach.

In Table 2, we present the previous work by incremental learning model and online algorithm.

Table 1: Previous work by relevance function and ranking approach

Reference	Relevance Function	Learning to Ranking Approach
(LING et al., 2012)	Latent Variable Model Matrix Decomposition	Pointwise and Listwise
(DIAZ-AVILES et al., 2012a)	Latent Variable model Matrix Decomposition	Pairwise
(DIAZ-AVILES et al., 2012b)	Latent Variable Model Matrix Decomposition	Pairwise
(WANG et al., 2013)	Latent Variable Model Matrix Decomposition	Pointwise
(SILVA; CARIN, 2012)	Graphical Model	Pointwise
(HARIRI; MOBASHER; BURKE, 2014)	Probabilistic Reward	Pointwise
(LIU et al., 2010)	Knn	Pointwise

Table 2: Previous work by incremental learning model and online algorithm

Reference	Incremental Learning Model	Online Algorithm
(LING et al., 2012)	Online Optimization	Stochastic Gradient Descent and Dual Averaging
(DIAZ-AVILES et al., 2012a)	Online Optimization	Stochastic Gradient Descent
(DIAZ-AVILES et al., 2012b)	Online Optimization	Stochastic Gradient Descent
(WANG et al., 2013)	Online Optimization	Stochastic Gradient Descent
(SILVA; CARIN, 2012)	Active Learning (Gaussian Process)	Greedy Strategy
(HARIRI; MOBASHER; BURKE, 2014)	Multi-Armed Bandit	Thompson Sampling
(LIU et al., 2010)	Reinforcement Learning	Temporal Difference Learning Temporal Dynamics

In Table 3, we present the previous work by dataset and evaluation metric used to evaluate the proposed model.

2.3.1 Discussion

In this section we analyze the previous work about incremental collaborative filtering presented in Section 2.3. Most relevant to this thesis are the model-based collaborative filtering studies, where the relevance function is modeled as a latent variable model (DIAZ-AVILES et al., 2012a, 2012b; WANG et al., 2013; LING et al., 2012). All papers present the incremental learning model as an online optimization problem and they differ mainly in the learning to rank approach and in the evaluation process.

The work of (LING et al., 2012) proposes two approaches to incremental collaborative filtering problem. These approaches are the combination of two learning to rank algorithms with two incremental optimization techniques. A pointwise learning to rank approach from (SALAKHUTDINOV;

Table 3: Previous work by dataset and evaluation metric

Reference	Dataset	Evaluation Metric
(LING et al., 2012)	Movielens and Yahoo Music	RMSE and NDCG
(DIAZ-AVILES et al., 2012a)	Twitter	Recall
(DIAZ-AVILES et al., 2012b)	Twitter	Recall
(WANG et al., 2013)	Dating Agency, Jester Joke and Movielens	RMSE and MAE
(SILVA; CARIN, 2012)	Yahoo Music	RMSE
(HARIRI; MOBASHER; BURKE, 2014)	Yahoo Music and Cti Data	CTR
(LIU et al., 2010)	Netflix	RMSE and MAP

MNIH, 2007) and a listwise learning to rank approach from (SHI; LARSON; HANJALIC, 2010) compose the learning to rank algorithms. The two incremental optimization approaches used are the stochastic gradient method and the dual averaging method. An interesting point of this work is the application of the dual averaging optimization method. The advantage of the use of the dual averaging method is that it uses the integral form of the regularization term, providing greater stability to the learning algorithm. Other positive points are the evaluation metric and the datasets used in the experiments. In relation to the metric, the listwise learning to rank algorithm is evaluated by a metric that expresses objectively the quality of the recommendation. The same comment is direct to the datasets used, both are standard datasets in the recommender system research area. Another interesting point is the comparison of the incremental approaches with the batch counterpart algorithm, demonstrating how the performance of incremental approaches approximate the batch algorithm over time.

In the work of (DIAZ-AVILES et al., 2012a), an incremental pairwise learning to rank approach is presented. The relevance function is modeled based on the work of (JOACHIMS, 2002). The incremental optimization method applied is the stochastic gradient descent. A relevant point of this work is in the form with the received evaluations are used in the incremental algorithm. It is possible to use only the current evaluation in the relevance function update or to create a kind of evaluation pool and use these accumulated evaluations to perform the relevance function update. Another interesting point concerns with the dataset used, representing a stream of Twitter data. Although it is an algorithm classified as pairwise in the learning of rank context, the evaluation metric does not evaluate the quality of the recommendation ranking directly. Therefore, the negative point of this work is its evaluation metric.

In a later work on (DIAZ-AVILES et al., 2012b), a new proposal for

incremental pairwise learning is presented. The fundamental difference of the previous proposal is in the way the data is sampled from the data repository. Previously we had a fixed-size repository used to update the relevance function, and now a random sample searching for the most representative data is used. The data set and the evaluation metric are the same as the previous work.

Another approach, proposed by (WANG et al., 2013), presents an incremental pointwise learning proposal. The main advantage of this proposal is related to the multi-tasking approach. For each evaluation received from an item by a user, not only the user or item vectors are updated with this new information, but also the vectors of the other users who have already evaluated the same item. The advantage of this approach is a better use of the information received, unlike only the direct application of an incremental optimization method. On the other hand, the disadvantage of this model is related to the fact that it requires more processing, since a greater number of updates is necessary. The data set used in the experiments is standard in the area of recommender systems and the metric used, as it is a pointwise approach, correctly measures its performance.

2.3.2 Baseline

Based on the previous work about incremental collaborative filtering presented in Section 2.3 and analyzed in Section 2.3.1 the main consideration of this thesis is introduced.

One aspect observed was the lack of compromise between prediction performance and speed of convergence. All the previous work use the stochastic gradient descent method to compose the incremental version of the batch algorithm. Only one work, (LING et al., 2012), uses a different approach, the dual averaging method. The dual averaging method, despite having a lower convergence rate than the stochastic gradient descent method, presents a better consistency given that it uses the complete regularization terms during the optimization. The work of (LING et al., 2012) is also the only one that uses a listwise learning to ranking approach, which is more appropriate in the recommendation context and offers a better ranking performance.

Based on that, we delineate the general objective of this thesis, accelerate an incremental learning to rank algorithm in the context of the collaborative filtering problem. The work of (LING et al., 2012) is the baseline of our proposal, considering that it uses a superior approach in the learning to rank context and a more prominent method in the incremental optimization

context. Therefore, our objective can be interpreted as accelerate the chosen baseline algorithm.

3 PROPOSAL

As presented in Section 1.1 and justified in Section 2.3.2, the main objective of this thesis is propose an accelerated version of an incremental collaborative filtering algorithm. Firstly, we introduce how the problem of collaborative filtering was modeled in the baseline algorithm. After, we present how the baseline algorithm was adapted to the incremental learning problem, described in Section 2.2. Finally, we describe how the acceleration technique was applied to improve the speed of learning of the baseline algorithm in the incremental learning context.

3.1 OVERVIEW

The proposed algorithm is an accelerated version of an incremental algorithm based on an offline learning to rank for collaborative filtering algorithm. In the Figure 1 we present an overview of the algorithms hierarchy.

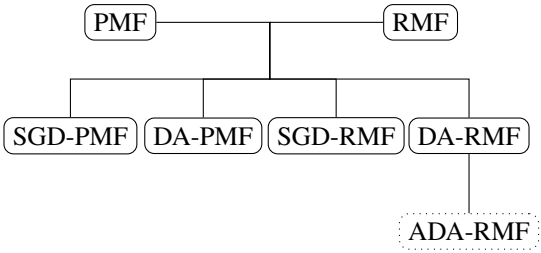

Figure 1: Algorithm Hierarchy

The Probabilistic Matrix Factorization (PMF) (MNIH; SALAKHUTDINOV, 2007) and the Top-one Probability Ranking Matrix Factorization (RMF) (SHI; LARSON; HANJALIC, 2010) are two offline learning to rank algorithms. In the work of (LING et al., 2012), the PMF and RMF were extended to an incremental version of the offline algorithms by the application of two online optimization methods, Stochastic Gradient Descent (SGD) and Dual-Averaging (DA) method, resulting in the SGD-PMF, DA-PMF, SGD-RMF and DA-RMF. Our proposal is an extension of the DA-RMF, the baseline algorithm described in Section 2.2, the Accelerated Dual Averaging-Ranking Matrix Factorization (ADA-RMF) algorithm.

3.2 RANKING MATRIX FACTORIZATION (RMF)

The top-one probability based ranking matrix factorization (SHI; LARSON; HANJALIC, 2010) is an offline listwise learning to rank for collaborative filtering algorithm. The idea is to model the probability of an item being ranked in the top of a user's recommendation list by a softmax or normalized exponential function. The softmax function normalizes a d-dimensional vector v of real values to a d-dimensional vector t of real values in the range $(0,1)$ that add up to 1 as in Equation 3.1.

$$s(v)_j = \frac{exp(v_j)}{\sum_{k=1}^{d} exp(v_k)}, \text{for } j = 1,\ldots,d \tag{3.1}$$

The $exp(x)$ represents the natural exponential function.

Based on the set $\mathscr{R}$ of all ratings, the top-one probability of the matrix representation of $\mathscr{R}$ associated with an item i in a ranking for a user u is defined as in Equation 3.2:

$$p_R(e_{ui}) = \frac{exp(e_{ui})}{\sum_{k=1}^{n} 1_{uk} exp(e_{uk})} \tag{3.2}$$

Through the decomposed matrices U and I, the top-one probability of the learned model is defined as in Equation 3.3.

$$p_{UI}(g_{ui}) = \frac{exp(g_{ui})}{\sum_{k=1}^{n} 1_{uk} exp(g_{uk})} \tag{3.3}$$

The e_{ui} is the evaluation e from a rating (u,i,e) of the set $\mathscr{R}$. The evaluation values have been mapped to interval $[0,1]$ by $(e - e_{min})/(e_{max} - e_{min})$, where e_{min} and e_{max} are the minimum and the maximum values of the set $\mathscr{E}$, respectively. The g_{ui} is the predicted evaluation mapped to the interval $[0,1]$ by the logistic function, defined as $1/(1+exp(-U_u I_i))$. The 1_{ui} is the indicator function which is equal to 1 if user u have evaluated the item i and 0 otherwise.

From information theory introduced in Section 2.1.2, cross entropy $H(p,q)$ measures the divergence between two discrete probability distributions $p(x)$ and $q(x)$. Assuming that $p(x) = p_R(e_{ui})$ and $q(x) = p_{UI}(g_{ui})$, we approximate $p_R(e_{ui})$ to $p_{UI}(g_{ui})$ as we minimize the divergence between the probability distributions by $H(p,q)$.

Based on the statistical learning theory, presented in Section 2.1.2, the

loss function defined by RMF is presented in Equation 3.4.

$$\mathscr{L} = \sum_{u=1}^{m} \left\{ -\sum_{i=1}^{n} 1_{ui} \frac{exp(e_{ui})}{\sum_{k=1}^{n} 1_{uk} exp(e_{uk})} \log \left\{ \frac{exp(g_{ui})}{\sum_{k=1}^{n} 1_{uk} exp(g_{uk})} \right\} \right\} + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_I}{2} \|I\|_F^2 \tag{3.4}$$

Both λ_U and λ_I are l_2-regularization parameters and the $\|.\|_F^2$ is the frobenius norm.

As presented in Section 2.2, the objective is to find out the matrices U and I, through the set of known ratings $\mathscr{Q}$, that approximate the unknown matrix R, composed by the elements of the set $\mathscr{R}$. Once the loss function $\mathscr{L}$ emphasizes the concept of measuring the difference between two ranks, represented by the two probability distributions p_R and p_{UI}, this is classified as a listwise learning to rank approach and comprises the offline the foundation for the baseline online algorithm.

3.3 DUAL AVERAGING RANKING MATRIX FACTORIZATION (DA-RMF)

The dual averaging ranking matrix factorization method (LING et al., 2012) is an online learning variation of the offline top-one probability based ranking matrix factorization (SHI; LARSON; HANJALIC, 2010). The online algorithm is based on the dual averaging optimization method, exposed in Section 2.1.3, applied for regularized stochastic learning and online optimization (XIAO, 2009). The dual averaging method absorbs previous rating information in an approximate average gradient of the loss function. Then it updates the model by solving an tractable analytically suboptimization problem based on the average gradient. The advantage is that it explicitly exploit the regularization structure in an online context, using the whole regularization term, not just its subgradient.

The cross entropy loss function $\mathscr{L}$ used in RMF is redefined to the objective function $\mathscr{O}$ without regularization terms, capturing only the cross entropy between p_R and p_{UI}, as in Equation 3.5. The l_2-regularization terms are applied later in the suboptimization problem.

$$\mathscr{O} = \mathscr{L} - \left(\frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_I}{2} \|I\|_F^2 \right) \tag{3.5}$$

We track the average gradients $\bar{G}_{U_u}$ and $\bar{G}_{I_i}$ as the observed rating

appears one by one, approximating the gradient of $\mathscr{O}$ with respect to U_u and I_i as defined by Equation 3.6 and Equation 3.7. The $\bar{G}_{U_u}^{t_u}$ denotes the gradient of $\mathscr{O}$ to U_u when the t_u-th evaluation is revealed by user u to item i. And $\bar{G}_{I_i}^{t_i}$ denotes the gradient of $\mathscr{O}$ to I_i when item i receives the t_i-th evaluation.

$$\begin{aligned}\bar{G}_{U_u}^{t_u+1} \leftarrow & \frac{\sum\limits_{k\in\mathscr{Q}_u^{t_u}} exp(e_{uk})}{\sum\limits_{k\in\mathscr{Q}_u^{t_u+1}} exp(e_{uk})}\bar{G}_{U_u}^{t_u} \\ & + \left\{ \frac{exp(g_{ui})}{\sum\limits_{k\in\mathscr{Q}_u^{t_u+1}} exp(g_{uk})} - \frac{exp(e_{ui})}{\sum\limits_{k\in\mathscr{Q}_u^{t_u+1}} exp(e_{uk})} \right\} g'_{ui} I_i \end{aligned} \tag{3.6}$$

$$\begin{aligned}\bar{G}_{I_i}^{t_i+1} \leftarrow & (1-\alpha^{\beta\times t_i})\bar{G}_{I_i}^{t_i} \\ & + \left\{ \frac{exp(g_{ui})}{\sum\limits_{k\in\mathscr{Q}_u^{t_u+1}} exp(g_{uk})} - \frac{exp(e_{ui})}{\sum\limits_{k\in\mathscr{Q}_u^{t_u+1}} exp(e_{uk})} \right\} g'_{ui} U_u \end{aligned} \tag{3.7}$$

The parameters α and β control the initial decay rate and how this rate decrease, respectively. The g'_{ui} is the derivative of the logistic function g_{ui}.

Once the average gradients $\bar{G}_{U_u}$ and $\bar{G}_{I_i}$ are calculated, the U_u and I_i are updated by solving a suboptimization problem:

$$U_u = \arg\min_w \left\{ \bar{G}_{U_u}^T w + \lambda_U \|w\|_2^2 \right\} \tag{3.8}$$

$$I_i = \arg\min_w \left\{ \bar{G}_{I_i}^T w + \lambda_I \|w\|_2^2 \right\} \tag{3.9}$$

Where w represents the vector we want to minimize and the $\lambda_U \|w\|_2^2$ and $\lambda_I \|w\|_2^2$ are the l_2-regularization terms.

Both Equation 3.8 and Equation 3.9 represent an optimization problem that is convex and differentiable and can be solved analytically. Taking the derivative equals to zero give us the update rules of U_u and I_i defined at Equation 3.10 and Equation 3.11.

$$U_u \leftarrow -\frac{1}{2\lambda_U}\bar{G}_{U_u} \tag{3.10}$$

$$I_i \leftarrow -\frac{1}{2\lambda_I}\bar{G}_{I_i} \tag{3.11}$$

The Algorithm 1 presents the DA-RMF proposed in (LING et al., 2012).

Algorithm 1: Dual averaging ranking matrix factorization (DA-RMF)

Data: Observation rating $(u,i,e) \in \mathscr{Q}$
Input: $n,m,l,\lambda_U,\lambda_I,\alpha,\beta$
Output: U,I

$U \in \mathbb{R}^{l \times n}$; // Randomly
$I \in \mathbb{R}^{l \times m}$; // Randomly
$\bar{G}_U \in \mathbb{R}^{l \times n} \leftarrow 0$; // Average gradient
$\bar{G}_I \in \mathbb{R}^{l \times m} \leftarrow 0$; // Average gradient
$S_Q \in \mathbb{R}^{n} \leftarrow 0$; // Sum vector
$S_{UI} \in \mathbb{R}^{n} \leftarrow 0$; // Sum vector
$T_I \in \mathbb{Z}^{m} \leftarrow 0$; // Index vector
for $(u,i,e) \in \mathscr{Q}$ **do**
  $t_i \leftarrow T_I(i)$;
  $s_r \leftarrow S_Q(u)$;
  $s_{ui} \leftarrow S_{UI}(u)$;
  $s'_r \leftarrow s_r + exp(e_{ui})$;
  $s'_{ui} \leftarrow s_{ui} + exp(g_{ui})$;
  $\bar{G}_{U_u} \leftarrow \frac{s_r}{s'_r}\bar{G}_{U_u} + \left\{\frac{exp(g_{ui})}{s'_{ui}} - \frac{exp(e_{ui})}{s'_r}\right\} g'_{ui} I_i$;
  $\bar{G}_{I_i} \leftarrow (1-\alpha^{\beta \times t_i})\bar{G}_{I_i} + \left\{\frac{exp(g_{ui})}{s'_{ui}} - \frac{exp(e_{ui})}{s'_r}\right\} g'_{ui} U_u$;
  $U_u \leftarrow -\frac{1}{2\lambda_U}\bar{G}_{U_u}$;
  $I_i \leftarrow -\frac{1}{2\lambda_I}\bar{G}_{I_i}$;
  $S_Q(u) \leftarrow s'_r$;
  $S_{UI}(u) \leftarrow s'_{ui}$;
  $T_I(i) \leftarrow t_i + 1$;

3.4 ACCELERATED DUAL AVERAGING RANKING MATRIX FACTORIZATION (ADA-RMF)

We propose a new accelerated version of a listwise learning to rank for collaborative filtering algorithm. The proposal is inspired in the acceleration technique proposed in (NESTEROV, 1983). The work of (NESTEROV, 1983) is a method of solving convex optimization problems with convergence rate of $O(1/k^2)$, where k is the number of iterations. Our motivation is that combining this method with the baseline algorithm will improve its speed of

learning.

Basically the acceleration method proposes the addition of a new point y_k that is an extrapolation point of the original point x_k at each iteration. The gradient $\nabla f(y_k)$ is calculated in terms of the new extrapolation point y_k during the update of x_k, as in Equation 3.12. At each iteration, the extrapolation point y_{k+1} is updated considering the actual point x_k and some rate γ of the difference between the previous x_{k-1} and actual point x_k as defined in Equation 3.14. The rate γ is updated each iteration as proposed in the work (NESTEROV, 1983) and defined in Equation 3.13.

$$x_k = y_k - \gamma_k \nabla f(y_k) \tag{3.12}$$

$$\gamma_{k+1} = \frac{(1+\sqrt{4\gamma_k+1})}{2} \tag{3.13}$$

$$y_{k+1} = x_k + \frac{(\gamma_k - 1)(x_k - x_{k-1})}{\gamma_{k+1}} \tag{3.14}$$

Intuitively, the ideia is that the extrapolation preserves the direction of the minimum of the function. We use this information to improve the speed of convergence of the algorithm.

Based on the exposed, two extrapolation points X_u and Y_i are added and at each iteration of the tuple (u,i,e), the average gradients $\bar{G}_X$ and $\bar{G}_Y$ are calculated considering this two new points. The update step of U_u and I_i is made considering the average gradients $\bar{G}_{X_u}$ and $\bar{G}_{Y_i}$, and is executed by solving the minimization problems:

$$U_u = \arg\min_w \left\{ \bar{G}_{X_u}^T w + \lambda_U \|w\|_2^2 + \|w - X_u\|_2^2 \right\} \tag{3.15}$$

$$I_i = \arg\min_w \left\{ \bar{G}_{Y_i}^T w + \lambda_I \|w\|_2^2 + \|w - Y_i\|_2^2 \right\} \tag{3.16}$$

Where w represents the vector we want to minimize and the $\lambda_U \|w\|_2^2$ and $\lambda_I \|w\|_2^2$ are the l_2-regularization terms.

As both Equation 3.15 and Equation 3.16 are differentiable and convex, we can solve it analytically. Taking the derivative equals 0, gives us the update rules of U_u and I_i defined at Equation 3.17 and Equation 3.18.

$$U_u \leftarrow \frac{2X_u - \bar{G}_{X_u}}{2\lambda_U + 2} \tag{3.17}$$

$$I_i \leftarrow \frac{2Y_i - \bar{G}_{Y_i}}{2\lambda_I + 2} \tag{3.18}$$

The difference is that we are taking the next steps of U_u and I_i based on the average gradient of X_u and Y_i instead of directly the average gradient of U_u and I_i.

In the sequence, the objective is update the value of the extrapolation points X_u and Y_i. For that, we introduce two control variables, γ_u and γ_i. Based on these variables, we calculate how much we will extrapolate the values of U_u and I_i. The update of variables γ_u and γ_i are given by the equations 3.19 and 3.20, respectively. Both Equation 3.19 and Equation 3.20 are based on the work of (NESTEROV, 1983).

$$\gamma_u^{t+1} = (1 + \sqrt{4\gamma_u^t + 1})/2 \tag{3.19}$$

$$\gamma_i^{t+1} = (1 + \sqrt{4\gamma_i^t + 1})/2 \tag{3.20}$$

Next step is extrapolate the values of U_u and I_i and update the values of X_u and Y_i as defined by Equation 3.21 and Equation 3.22.

$$X_u^{t+1} \leftarrow U_u^t + (\frac{\gamma_u^t - 1}{\gamma_u^{t+1}})(U_u^t - U_u^{t-1}) \tag{3.21}$$

$$I_i^{t+1} \leftarrow I_i^t + (\frac{\gamma_i^t - 1}{\gamma_i^{t+1}})(I_i^t - I_i^{t-1}) \tag{3.22}$$

The input sequence of γ moves X and Y in direction of U and I.

The Algorithm 2 presents the ADA-RMF proposed algorithm.

Algorithm 2: Accelerated dual averaging ranking matrix factorization (ADA-RMF)

Data: Observation triplet $(u,i,e) \in \mathcal{Q}$
Input: $n,m,l,\lambda_U,\lambda_I,\alpha,\beta$
Output: U,I

$U \in \mathbb{R}^{l\times n}$; // Randomly
$I \in \mathbb{R}^{l\times m}$; // Randomly
$X \in \mathbb{R}^{l\times n}$; // Equals U
$Y \in \mathbb{R}^{l\times m}$; // Equals I
$\bar{G}_X \in \mathbb{R}^{l\times n} \leftarrow 0$; // Average gradient
$\bar{G}_Y \in \mathbb{R}^{l\times m} \leftarrow 0$; // Average gradient
$S_Q \in \mathbb{R}^{n} \leftarrow 0$; // Sum vector
$S_{UI} \in \mathbb{R}^{n} \leftarrow 0$; // Sum vector
$T_I \in \mathbb{Y}^{m} \leftarrow 0$; // Index vector
for $(u,i,e) \in \mathcal{Q}$ **do**
  $t_i \leftarrow T_I(i)$;
  $s_r \leftarrow S_Q(u)$;
  $s_{ui} \leftarrow S_{UI}(u)$;
  $s'_r \leftarrow s_r + exp(e)$;
  $s'_{ui} \leftarrow s_{ui} + exp(g_{ui})$;
  $\bar{G}_{X_u} \leftarrow \frac{s_r}{s'_r}\bar{G}_{X_u} + \left\{\frac{exp(g_{ui})}{s'_{ui}} - \frac{exp(e)}{s'_r}\right\} g'_{ui} Y_i$;
  $\bar{G}_{Y_i} \leftarrow (1-\alpha^{\beta\times t_i})\bar{G}_{Y_i} + \left\{\frac{exp(g_{ui})}{s'_{ui}} - \frac{exp(e)}{s'_r}\right\} g'_{ui} X_u$;
  $U_u \leftarrow \frac{X_u - \bar{G}_{X_u}}{\lambda_U + 1}$;
  $I_i \leftarrow \frac{Y_i - \bar{G}_{Y_i}}{\lambda_I + 1}$;
  $S_Q(u) \leftarrow s'_r$;
  $S_{UI}(u) \leftarrow s'_{ui}$;
  $T_I(i) \leftarrow t_i + 1$;
  $\gamma_u^{t+1} = (1+\sqrt{4\gamma_u^t + 1})/2$;
  $\gamma_i^{t+1} = (1+\sqrt{4\gamma_i^t + 1})/2$;
  $X_u^{t+1} \leftarrow U_u^t + (\frac{\gamma_u^t - 1}{\gamma_u^{t+1}})(U_u^t - U_u^{t-1})$;
  $I_i^{t+1} \leftarrow I_i^t + (\frac{\gamma_i^t - 1}{\gamma_i^{t+1}})(I_i^t - I_i^{t-1})$;

4 EXPERIMENTAL ANALYSIS

This chapter describes the experimental analysis used in this thesis. Our objective is that empirical results support the proposed algorithm. To accomplish that, we evaluate the proposal algorithm in some instances of the problem of incremental collaborative filtering. Comparisons against the baseline algorithm are statistically tested to determine their statistical significance.

4.1 EXPERIMENT

In this section we describe the datasets used in the experiment, the evaluation metric, the design of the experimental method, the parameters of the algorithms and the statistical analysis used to test the results.

4.1.1 Datasets

The instances of the problem of incremental collaborative filtering are represented by the datasets selected to evaluate the proposed algorithm. Two contexts are defined for the empirical experiments: movie and song recommendation.

From the Grouplens, a research group in recommender systems, we selected the Movielens[1] dataset in the context of movie recommendations. From Yahoo! Research, more specifically from Yahoo Webscope Program[2], we used two datasets, Yahoo R3 in the context of song recommendations and Yahoo R4 in the context of movie recommendations.

Information about the number of users, number of items, interval of evaluations and number of available ratings are presented in Table 4.

Table 4: Datasets description

Dataset	Movielens 100K	Yahoo R4	Yahoo R3
Users	943	7.642	15.400
Items	1.682	11.916	1.000
Evaluations	[1 . . . 5]	[1 . . . 13]	[1 . . . 5]
Ratings	100.000	221.367	365.704

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

[2] http://webscope.sandbox.yahoo.com

4.1.2 Evaluation Metric

In order to quantitatively evaluate the ranking recommended by the proposed algorithm, we use the evaluation metric known as Normalized Discounted Cumulative Gain (NDCG).

NDCG is a standard evaluation measure of learning to rank systems (JÄRVELIN; KEKÄLÄINEN, 2002). The NDCG is defined in Equation 4.1, where $e_{\pi(i)}$ is the evaluation e of the item at position i on the recommended ranking π, $e_{\pi^*(i)}$ is the evaluation e of the item at position i on the ideal ranking π^* and n is the number of recommended items.

$$NDCG@n = \sum_{i=1}^{n} \frac{2^{e_{\pi(i)}} - 1}{\log(1+i)} \Big/ \sum_{i=1}^{n} \frac{2^{e_{\pi^*(i)}} - 1}{\log(1+i)} \tag{4.1}$$

The fundamental aspect of the NDCG is that items ranked higher receive more weight than items ranked lower.

4.1.3 Design of Experiment

To evaluate the performance of the proposed algorithm we conduct experiments using the same design of experiment defined at the baseline algorithm (LING et al., 2012). The objective is to better represent the incremental environment where training data is gradually available and training phase and test phase are interleaved.

Firstly, the dataset is randomly divided in training data and test data. This division is guided by a *rate* of how much data is reserved for each phase - training phase and test phase. Then the training data is divided in *partitions*. The *assessment loop* defines the interleaved aspect of the experiment, where after each partition of the training data that is fed in the algorithm, an *assessment* is realized. The assessment of the algorithm is made by the evaluation of its recommendation, through the NDCG metric, considering each user in the complete test data - this generates as many NDCG evaluations as the number of users in the test data. For each assessment, a *NDCG assessment mean* is calculated considering the individual NDCG evaluations.

The number of repetitions of the assessment loop represents the number of *measures*. The *algorithm evaluation* of the algorithm is given by the mean of the NDCG assessment mean obtained at each assessment - the number of NDCG assessment mean available must be equal to the number of measures times the number of assessments - number of partitions.

A complete execution of the steps described above is considered a

block of the experiment. At each block execution the dataset is randomly divided, what guarantee the independence of each algorithm evaluation, and results in an algorithm evaluation. In the end, the *overall evaluation* of the algorithm is given by the mean of the algorithm evaluations for all block executions.

4.1.4 Parameters

A group of parameters control the internal execution of both proposed algorithm and baseline algorithm. The parameters m and n determine the number of users and number of items respectively - both are implicit, obtained by the number of users and items from the datasets. The latent feature dimension l determines the size of the latent feature vector, for both users and items. The parameters λ_U and λ_I control the regularization of the model. The parameters α and β control the decay and drop-rate of decay of items respectively. Exclusively for the proposed algorithm, the parameters γ_u and γ_i are used to control the degree of extrapolation.

4.1.5 Statistical Analysis

We test for statistically significant differences between proposed algorithm and baseline algorithm results using an one-sided student's test. Based on the statistical test, we can guarantee the performance of the proposed algorithm in comparison with the baseline algorithm with a statistical level of certainty. We consider differences between runs statistically significant if the obtained p-value is less then 0.01 and significance level of 1.00%.

4.2 EXECUTION

In this section we conduct the execution of the experiments defined in the section 4.1. Firstly we describe the general configuration of the experiments and subsequently the configuration of the parameters used to configure the internal execution of the algorithms. In the end, the results are presented.

4.2.1 Experiment Configuration

Based on the design of experiment presented at Section 4.1.3, we define three different settings of rates for the division of the dataset in training data and test data, T20, T50 and, T80. The setting T20 defines 20% of data for training and 80% of data for test, the setting T50, 50% of data for training and 50% of data for test and the setting T80, 80% of data for training and 20% of data for test.

The number of partitions (divisions of the training data) is configured to 5 and the the number of measures (number of repetitions of the assessment loop) is configured to 20. Based on the number of partitions and the number of measures, we have 100 NDCG assessment means to calculate the algorithm evaluation.

We execute 20 blocks of the experiment for each dataset. The overall evaluation of the algorithm is calculated considering the algorithm evaluation resulted from each block execution.

This configuration generates the data used to evaluate the algorithms stability on different settings of the datasets.

4.2.2 Parameters Configuration

The number of users m and number of items n are both determined by datasets used in the experiment. For the latent feature dimension we employed $l = 10$ for all settings in the experiment. The better performance was obtained with λ_U and λ_I with the value of 0.014. The values were set as 0.8 for α and 0.2 for β. The values used to control the degree of extrapolation by γ_u and γ_i were set to 0.001.

4.2.3 Results

Results are organized by dataset used in the experiment. For each dataset, the NDCG mean and the standard deviation is presented. Results per experiment with statistically significant differences are highlighted in **bold**.

In Table 5, we present NDCG mean and standard deviation for the Movielens 100k, considering all the experimental settings.

Table 5: Mean and standard deviation in the Movielens 100k dataset

	T20		T50		T80	
Algorithm	Mean	Std	Mean	Std	Mean	Std
DA-RMF	0.5923	0.0092	0.6288	0.0130	0.7220	0.0059
ADA-RMF	**0.6225**	0.0056	**0.6614**	0.0064	**0.7378**	0.0033

The ADA-RMF has been consistently better than the DA-RMF in all settings in the Movielens 100k dataset. The stability of the proposed algorithm is confirmed by values of standard deviation. In all settings, the ADA-RMF presented a lower standard deviation when compared with de baseline algorithm.

In Figure 3, Figure 5 and, Figure 7, we visualize the relative frequency and the normal distribution of the mean of NDCG of the DA-RMF algorithm and ADA-RMF algorithm for the Movielens 100k dataset.

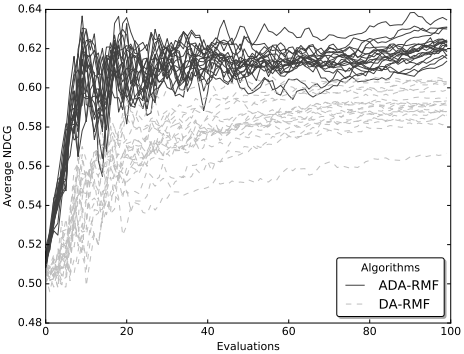

Figure 2: All evaluations in T20 setting for MovieLens 100k dataset

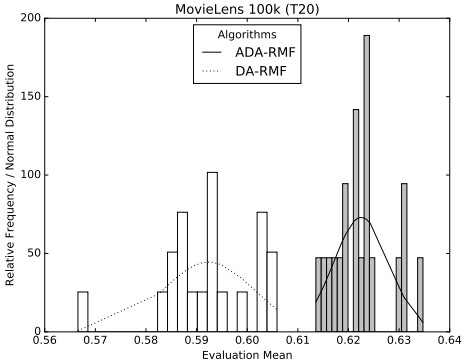

Figure 3: Results in T20 setting for Movielens 100K dataset

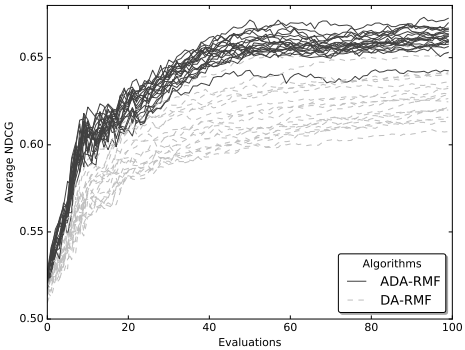

Figure 4: All evaluations in T50 setting for MovieLens 100k dataset

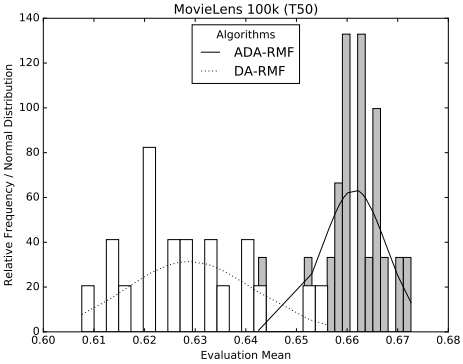

Figure 5: Results in T50 setting for Movielens 100K dataset

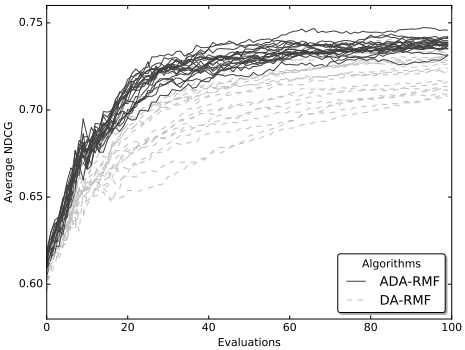

Figure 6: All evaluations in T80 setting for MovieLens 100k dataset

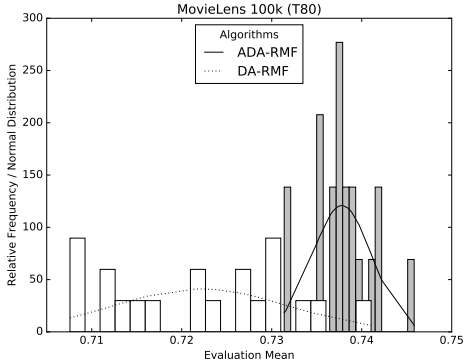

Figure 7: Results in T80 setting for Movielens 100K dataset

In Table 6, we present NDCG mean and standard deviation for the Yahoo R4 dataset, considering all the experimental settings.

Table 6: Mean and standard deviation in the Yahoo R4 dataset

	T20		T50		T80	
Algorithm	Mean	Std	Mean	Std	Mean	Std
DA-RMF	0.5187	0.0054	0.6164	0.0059	0.7675	0.0047
ADA-RMF	**0.6252**	0.0069	**0.7014**	0.0033	**0.8121**	0.0016

Expressively, in the Yahoo R4 dataset, the ADA-RMF is consistently better than the DA-RMF in all experimental settings. Once more, as in the Movielens 100k dataset, the ADA-RMF presented a lower standard deviation when compared with de baseline algorithm, confirming the stability of the proposed algorithm.

In Figure 9, Figure 11 and, Figure 13, we visualize the relative frequency and the normal distribution of the mean of NDCG of the DA-RMF algorithm and ADA-RMF algorithm for the Yahoo R4 dataset.

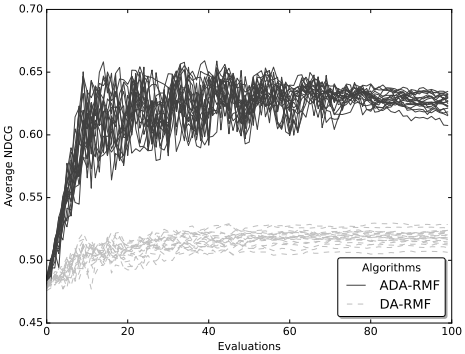

Figure 8: All evaluations in T20 setting for Yahoo R4 dataset

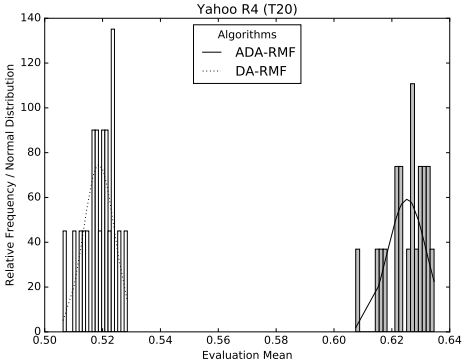

Figure 9: Results in T20 setting for Yahoo R4 dataset

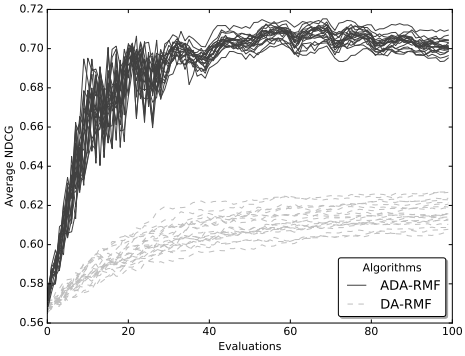

Figure 10: All evaluations in T50 setting for Yahoo R4 dataset

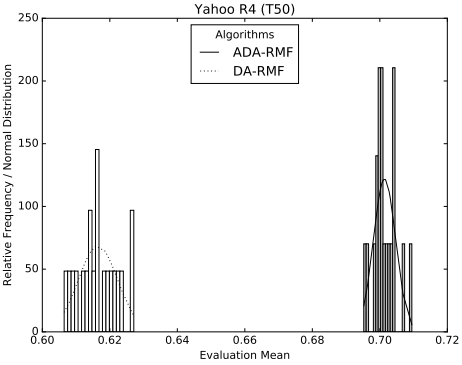

Figure 11: Results in T50 setting for Yahoo R4 dataset

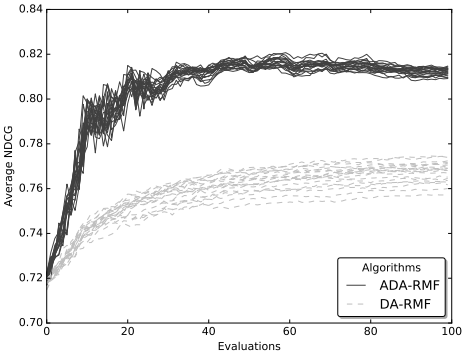

Figure 12: All evaluations in T80 setting for Yahoo R4 dataset

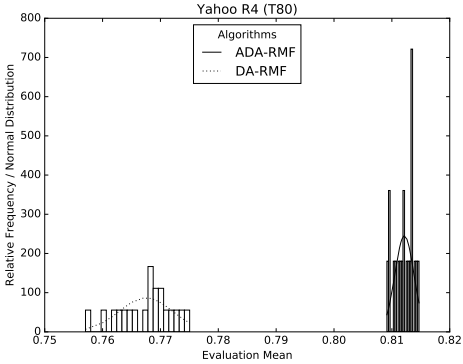

Figure 13: Results in T80 setting for Yahoo R4 dataset

In Table 7, we present NDCG mean and standard deviation for the Yahoo R3 dataset, considering all the experimental settings.

Table 7: Mean and standard deviation in the Yahoo R3 dataset

	T20		T50		T80	
Algorithm	Mean	Std	Mean	Std	Mean	Std
DA-RMF	0.5245	0.0021	0.6171	0.0023	0.7768	0.0017
ADA-RMF	0.5230	0.0077	**0.6462**	0.0020	**0.8053**	0.0017

Notably, at the T20 setting on the Yahoo R3 dataset, we don't observe a statistically difference between the proposed algorithm and the baseline algorithm. We also observe that the proposed algorithm is more unstable in the T20 setting, presenting a greater standard deviation. In the settings T50 and T80, the proposed algorithm presents statistically significant better results.

In Figure 15, Figure 17 and, Figure 19, we visualize the relative frequency and the normal distribution of the mean of NDCG of the DA-RMF algorithm and ADA-RMF algorithm for the Yahoo R3 dataset.

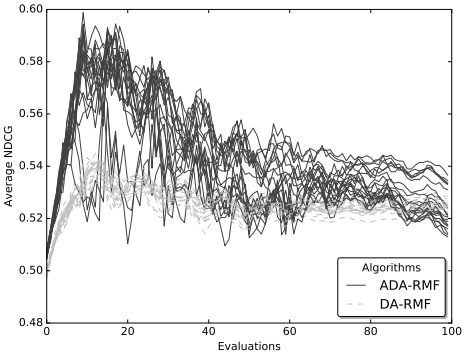

Figure 14: All evaluations in T20 setting for Yahoo R3 dataset

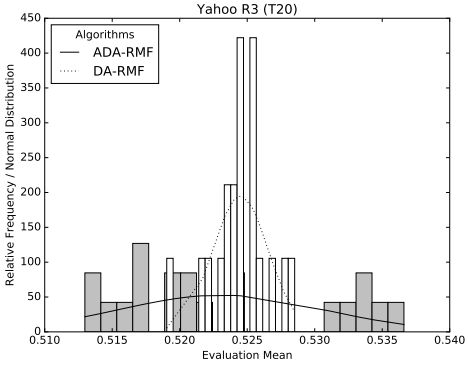

Figure 15: Results in T20 setting for Yahoo R3 dataset

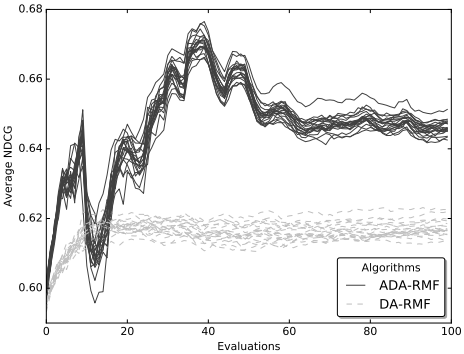

Figure 16: All evaluations in T50 setting for Yahoo R3 dataset

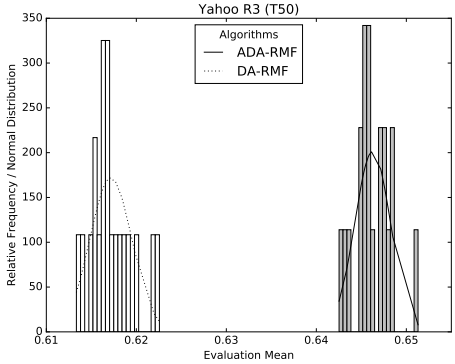

Figure 17: Results in T50 setting for Yahoo R3 dataset

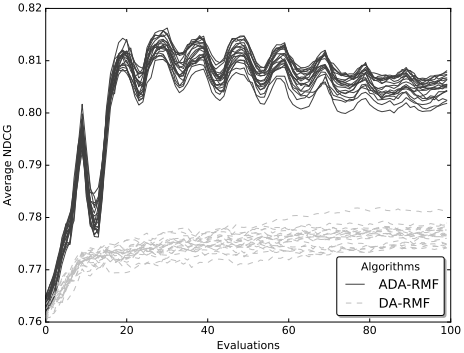

Figure 18: All evaluations in T80 setting for Yahoo R3 dataset

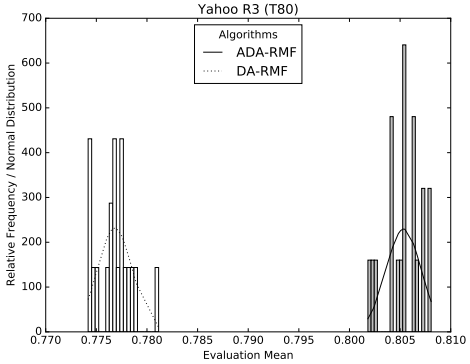

Figure 19: Results in T80 setting for Yahoo R3 dataset

4.3 DISCUSSION

Even with statistically better results presented by the proposed algorithm when compared with the baseline algorithm in terms of NDCG evaluation, this does not represents a better performance in absolute terms. The main advantage is that we reach a better performance early, but extrapolating the number of block executions in the experiment should lead us to the same performance at all.

5 CONCLUSION

Collaborative filtering technique plays an active role overcoming the information overload problem. In the incremental learning paradigm, data is continuously received by the system and is used to extend the existing model's knowledge. The main advantage of this approach is related to its low memory complexity which provides scalability, since that only information been processed needs to be in memory. The drawback is that with less available information, the speed of learning is compromised and reflects on the accuracy of the model. The trade-off between low memory complexity and speed of learning in incremental learning to rank models applied in collaborative filtering problem was the focus of this research.

This investigation was guided by the assumption that the convergence rate of the choosen incremental learning to rank baseline algorithm, DA-RMF, could be improved keeping its low memory complexity. Specifically, we applied a technique known as Nesterov's acceleration in the dual-averaging optimization method in the context of matrix decomposition. The acceleration technique is basically an extrapolation of the gradient information used during the optimization in the learning process. Through the combination of the Nesterov's acceleration technique with the baseline algorithm based on the Dual Averaging optimization method we proposed the ADA-RMF algorithm.

To evaluate the proposed algorithm we conducted experiments with real world instances of the problem. In the context of movie recommendations we used the Movielens and the Yahoo R4 datasets and in the context of song recommendations we used the Yahoo R3 dataset. In order to quantitatively evaluate the ranking recommended by the proposed algorithm, we use the NDCG evaluation metric. The comparisons with the baseline algorithm were statistically tested to determine their statistical significance. Our objective with the empirical experiments was to obtain results that could support the proposed algorithm.

Although we do not provide theoretical results, demonstrating the effectiveness of the application of the acceleration technique, empirical results showed that the proposed algorithm learn the relevance function faster than the baseline algorithm. Statistically comproved results on all real word datasets showed that our proposal algorithm accelerates the learning process and keeps the accuracy of the model. Altough, it is important to note that even with better results obtained in the experiments by the proposed algorithm in comparison with the baseline algorithm, this does not represent a better performance in absolute terms. The main advantage is that the pro-

posed algorithm reachs a better performance early, but extending the experiments should lead us to the same performance at all. In other words, the best performance of the algorithm is achieved early, what is a great advantage in scenarios where incremental learning is applied.

5.1 FUTURE WORK

Some directions could guide the future development of this research. Considering these directions, we highlight the development of theoretical results, the investigation of other acceleration techniques and the hyper parameter optimization of the proposed algorithm as the most important.

Theoretical results should be developed in order to prove the convergence of the proposed algorithm and determine its convergence rate. This also could enable the analysis of the impact of using another type of regularization term during the learning phase.

The investigation of the use of other acceleration techniques should be carried out. The impact of this could be better measured with the theoretical results previously cited.

The parameters used in the configuration of the proposed algorithm during the experiments were defined by empirical examination. A hyper parameter optimization should be executed in order to select the best set of parameters and improve the performance of the proposed algorithm.

BIBLIOGRAPHY

BOBADILLA, J. et al. Recommender systems survey. **Knowledge-Based Systems**, Elsevier, v. 46, p. 109–132, 2013.

BORODIN, A.; EL-YANIV, R. **Online Computation and Competitive Analysis**. [S.l.]: Cambridge University Press, 2005.

BOTTOU, L. Large-scale machine learning with stochastic gradient descent. In: **Proceedings of Conference on Computational Statistics**. [S.l.]: Springer, 2010. p. 177–186.

BOYD, S.; VANDENBERGHE, L. **Convex Optimization**. [S.l.]: Cambridge university press, 2004.

BURDEN, R.; FAIRES, J. **Numerical Analysis**. [S.l.: s.n.], 2011.

CHONG, E. K.; ZAK, S. H. **An Introduction to Optimization**. [S.l.]: John Wiley & Sons, 2013.

DIAZ-AVILES, E. et al. What is happening right now... that interests me? In: ACM. **Proceedings of the 21st ACM International Conference on Information and Knowledge Management**. [S.l.], 2012. p. 1592–1596.

DIAZ-AVILES, E. et al. Real-time top-n recommendation in social streams. In: ACM. **Proceedings of the sixth ACM Conference on Recommender Systems**. [S.l.], 2012. p. 59–66.

EPPLER, M. J.; MENGIS, J. The concept of information overload: A review of literature from organization science, accounting, marketing, mis, and related disciplines. **The Information Society**, Taylor & Francis, v. 20, n. 5, p. 325–344, 2004.

FIAT, A. Online algorithms: The state of the art. **Lecture Notes in Computer Science**, Springer, 1998.

HANANI, U.; SHAPIRA, B.; SHOVAL, P. Information filtering: Overview of issues, research and systems. **User Modeling and User-Adapted Interaction**, Kluwer Academic Publishers, v. 11, n. 3, p. 203–259, 2001.

HARIRI, N.; MOBASHER, B.; BURKE, R. Context adaptation in interactive recommender systems. In: ACM. **Proceedings of the 8th ACM Conference on Recommender Systems**. [S.l.], 2014. p. 41–48.

HIMMA, K. E. The concept of information overload: A preliminary step in understanding the nature of a harmful information-related condition. **Ethics and Information Technology**, Springer, v. 9, n. 4, p. 259–272, 2007.

JANNACH, D. et al. **Recommender Systems: An Introduction**. [S.l.]: Cambridge University Press, 2010.

JÄRVELIN, K.; KEKÄLÄINEN, J. Cumulated gain-based evaluation of IR techniques. **ACM Transactions on Information Systems**, ACM, v. 20, n. 4, p. 422–446, 2002.

JOACHIMS, T. Optimizing search engines using clickthrough data. In: ACM. **Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S.l.], 2002. p. 133–142.

LING, G. et al. Online learning for collaborative filtering. In: IEEE. **The 2012 International Joint Conference on Neural Networks**. [S.l.], 2012. p. 1–8.

LIU, N. N. et al. Online evolutionary collaborative filtering. In: ACM. **Proceedings of the Fourth ACM Conference on Recommender Systems**. [S.l.], 2010. p. 95–102.

LIU, T.-Y. Learning to rank for information retrieval. **Foundations and Trends in Information Retrieval**, Now Publishers Inc., v. 3, n. 3, p. 225–331, 2009.

LIU, T.-Y. **Learning to Rank for Information Retrieval**. [S.l.]: Springer Science & Business Media, 2011.

MACKAY, D. J. **Information Theory, Inference and Learning Algorithms**. [S.l.]: Cambridge university press, 2003.

MILLER, G. A. The cognitive revolution: A historical perspective. **Trends in Cognitive Sciences**, Elsevier, v. 7, n. 3, p. 141–144, 2003.

MITCHELL, T. M. **Machine Learning**. [S.l.]: McGraw-Hill Science, 1997.

MNIH, A.; SALAKHUTDINOV, R. Probabilistic matrix factorization. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2007. p. 1257–1264.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. **Foundations of Machine Learning**. [S.l.]: MIT press, 2012.

NESTEROV, Y. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. In: **Soviet Mathematics Doklady**. [S.l.: s.n.], 1983. v. 27, n. 2, p. 372–376.

NESTEROV, Y. Primal-dual subgradient methods for convex problems. **Mathematical Programming**, Springer, v. 120, n. 1, p. 221–259, 2009.

RICCI, F. et al. **Recommender Systems Handbook**. [S.l.]: Springer, 2011.

SALAKHUTDINOV, R.; MNIH, A. Probabilistic matrix factorization. In: **Neural Information Processing Systems**. [S.l.: s.n.], 2007. v. 1, n. 1, p. 2–1.

SHALEV-SHWARTZ, S. et al. Online learning and online convex optimization. **Foundations and Trends in Machine Learning**, Now Publishers, Inc., v. 4, n. 2, p. 107–194, 2012.

SHI, Y.; LARSON, M.; HANJALIC, A. List-wise learning to rank with matrix factorization for collaborative filtering. In: ACM. **Proceedings of the Fourth ACM Conference on Recommender Systems**. [S.l.], 2010. p. 269–272.

SILVA, J.; CARIN, L. Active learning for online bayesian matrix factorization. In: ACM. **Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S.l.], 2012. p. 325–333.

VAPNIK, V. N.; VAPNIK, V. **Statistical Learning Theory**. [S.l.]: Wiley New York, 1998.

WANG, J. et al. Online multi-task collaborative filtering for on-the-fly recommender systems. In: ACM. **Proceedings of the 7th ACM Conference on Recommender Systems**. [S.l.], 2013. p. 237–244.

XIAO, L. Dual averaging method for regularized stochastic learning and online optimization. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2009. p. 2116–2124.