The Use of Machine Learning Algorithms in Recommender Systems: A Systematic Review

Ivens Portugal
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada
iportugal@uwaterloo.ca

Paulo Alencar
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada
palencar@uwaterloo.ca

Donald Cowan
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada
dcowan@csg.uwaterloo.ca

Abstract

Recommender systems use algorithms to provide users product recommendations. Recently, these systems started using machine learning algorithms because of the progress and popularity of the artificial intelligence research field. However, choosing the suitable machine learning algorithm is difficult because of the sheer number of algorithms available in the literature. Researchers and practitioners are left with little information about the best approaches or the trends in algorithms usage. Moreover, the development of a recommender system featuring a machine learning algorithm has problems and open questions that must be evaluated, so software engineers know where to focus research efforts. This work presents a systematic review of the literature that analyzes the use of machine learning algorithms in recommender systems and identifies research opportunities for the software engineering research field. The study concluded that Bayesian and decision tree algorithms are widely used in recommender systems because of their low complexity, and that requirements and design phases of recommender system development must be investigated for research opportunities.

Keywords: recommender system, machine learning, systematic review.

1 Introduction

Recommender systems (RS) are used to help users find new items, such as books, music, or even people, based on a set of information about the user, or the recommendation item [2]. These systems also play an important role in decision-making, helping users to maximize profits [15] or minimize risks [11]. Today, RSs are used in many computer science companies in such as Google [35], Twitter [4], LinkedIn [50], and Netflix [59]. The field of RS has its origins in the mid 1990s with the introduction of Tapestry [23], the first RS. As the RS field evolved, researchers studied the use of algorithms that belonged to an artificial intelligence (AI) field called machine learning (ML).

Machine learning, differently from RSs, has been under study since the late 1950s [40], with the appearance of the AI field. Today, there is a plethora of ML algorithms (k-nearest neighbor [48], clustering [28], Bayes network [22], to name a few), which are used in ways that range from vacuum cleaner robots [13] and disabled people assistance [30] to pattern recognition on images [63], or even self-driving vehicles [62]. The applications of ML algorithms are vast and the field is promising.

As previously mentioned, ML algorithms are being used in RSs to provide users with better recommendations. However, ML field does not have a clear classification of its algorithms, mainly because of the amount of approaches and the variations that researchers propose in the literature [37]. As a consequence, it becomes difficult and confusing to choose an ML algorithm that fits one's need when developing a RS. In addition, researchers may find it challenging to observe trends in the use of ML algorithms in RSs.

Software engineering (SE) is a field concerned into studying the design, development and maintenance of software [49] with tools for organization, structure, and use of computer science knowledge [27]. RS and ML fields can benefit from SE principles and definitions if studies in this direction are done. In the area of RSs with ML algorithms, researchers may not know which SE areas can have a great impact.

One possible way to help researchers and practitioners to decide which ML algorithm to use on a RS, and to identify SE areas that can be improved in the development of RSs that have a ML algorithm, is the study of the RS and ML fields. Research about RSs containing ML algorithms implemented in the literature can help observe trends and draw conclusions about the future of the fields involved.

Therefore, in this paper, it was decided to run a systematic review to investigate how real implemented RSs with ML algorithms are being used; which ML algorithm they use; and how SE field can impact on their development. It is expected that, with this systematic review, researchers and practitioners can obtain more information about RS field, and make better-informed implementation or research decisions.

This paper is organized as follows: section 2 describes the theoretical knowledge needed; Section 3 explains the systematic review plan, and section 4 explains its results. Section 5 finishes this study with conclusions and future work.

2 Theoretical Background

In this section, we give a brief overview of the fields of recommender systems and machine learning. This section is mostly concerned with introducing the fields with a quick historical description, and some field definitions and classifications.

2.1 Recommender System

Recommender systems (RS) use artificial intelligence (AI) concepts to provide users item recommendations. For example, an online bookshop may use a machine learning (ML) algorithm to classify books by genre and recommend other books to a user who wants to buy a book. RSs were introduced in 1992 when Tapestry, the first RS, appeared. Its authors used the term 'collaborative filtering' to refer to the recommendation activity. This term that is still used to classify RSs.

Recommender systems are divided into three main categories, depending on the information that recommendations are based on: collaborative, content-based, and hybrid filtering [2]. RSs with a collaborative approach consider the user data when processing information for recommendation. For instance, by accessing user profiles on an online music store, the RS have access to data, such as the age, the country and city, and purchased songs of all users. With this information, the system can identify users that share the same music preference, and then suggest users songs that similar users bought. RSs with a content-based filtering approach bases their recommendations on the item data they have access to. As an example, consider a user who is looking for a new computer on an online store. When the user browses a particular computer (item), the RS gathers information about that computer and searchers in a database for computers that have similar attributes, such as price, CPU speed, and memory capacity. The result of this search is then returned to the user as recommendations. The third classification describes RSs that combines the two previous classifications into a hybrid filtering approach, recommending items based on the user and the item data. For example, on a social network, a RS may recommend profiles that are similar to the user (collaborative filtering), by comparing their interests. On a second step, the system may consider the recommended profiles as items and thus access their data to search for new similar profiles (content-based filtering). In the end, both sets of profiles are returned as recommendations.

When using a collaborative or a hybrid filtering approach, RSs must gather information about the user to base recommendations on. This activity can be done explicitly or implicitly. Explicitly user data gathering happens when users provides their information and are aware of it. For instance, when registering for a new online service, users usually fill in a form that asks their name, age, and email. Other forms of explicit user data gathering are when users express their preferences by rating items either using a numerical value or with a Facebook like. Conversely, implicitly user data gathering is the activity of accessing information about the user in an indirect way. For example, when visiting an online store, the server at the online store exchanges messages with the user's computer, and based on that, the store's RS may know the browser the user is using, as well as its country. More advanced applications monitor user click and keystroke log.

Besides the common recommendation process, in which users are presented with items that they might be interested in, recommendations can be done in different ways. Trust-based recommendations [45] take into consideration the trust relationship that users have between them. A trust relationship is a link in a social network either by friend or following connections. Recommendations based on trusted friends are worth more than those that do not have trust links. Context-aware recommendations [3] are done based on the context that the user is inserted. A context is a set of information about the current state of the user, such as the time at the user location (morning, afternoon, evening), or their activity (idle, running, sleeping). The amount of context information to be processed is high, making context-aware recommendations a challenging research field. Risk-aware recommendations [11] are a subset of context-aware recommendations and take into consideration a context in which critical information is available, such as user vital information. It is risk-aware because a wrong decision may threat the user life or cause real life damages. Some examples are recommending pills a user should take or which stocks the user should buy, sell, or invest on.

2.2 Machine Learning

Machine Learning (ML) is the AI field concerned with using computers to simulate human learning, which allows computers to identify and to acquire knowledge from the real world, and to improve its performance on some tasks based on the new knowledge. More formally, [43] defines ML as follows: “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”. Although the first concepts of ML were coined in the 1950s, ML was studied as a separate field in the 1990s [43]. Today, ML algorithms are used in several not limited to computer science, such as business [6], advertisement [16] and medicine [32].

Learning is the process of knowledge acquisition. Humans naturally learn from experience due to the ability of reasoning. In contrast, computers do not learn with reasoning, but instead, they learn with algorithms. Today, there are a huge amount of ML algorithms proposed in literature. They can be classified based on the approach used for the learning process. There are four main classifications: supervised, unsupervised, semi-supervised, and reinforcement learning. Supervised learning happens when algorithms are provided with training data and correct answers. The task of the ML algorithm is to learn based on the training data, and to apply the knowledge that was gained in real data without answers. As an example consider a ML learning algorithm being used in book classification in a bookstore. A training set (training data + answers) can be a table associating information about each book to a correct classification. Here, information about each book may be title, author, or even every word a book contains. The ML algorithm learns with the training set. When a new book arrives at the bookstore, the algorithm can classify it based on the knowledge about book classification it has acquired.

Differently, in unsupervised learning, ML algorithms do not have a training set. They are presented with some data about the real world and have to learn from that data on their own. Unsupervised learning algorithms are mostly concerned into finding hidden patterns in data. For example, suppose that a ML algorithm have access to user profile information in a social network. By using an unsupervised learning approach, the algorithm can separate users into personality categories, such as outgoing and reserved, allowing the social network company target advertisement for specific groups of users more efficiently.

Between supervised and unsupervised learning, ML algorithms can be classified as having a semi-supervised learning. It happens when algorithms are prepared to receive a training set with missing information, and still need to learn from it. An example is when a ML algorithm is provided with movie ratings. Not every user rated every movie and thus there shall be some missing information. Semi-supervised learning algorithms are able to learn and draw conclusions even with incomplete data.

Lastly, ML algorithms might have a reinforcement learning approach. It happens when algorithms learn based on an external feedback given either by a thinking entity, or the environment. This approach is analogous to teaching dogs to sit or jump. When the dog performs the action that the human asked it to do, it receives a small cookie (positive feedback). It does not receive any cookie (negative feedback) if it performs wrong or no action. As an example in the computer science field, consider an ML algorithm that plays games against an opponent. Moves that lead to victories (positive feedback) in the game should be learned and repeated, whereas moves that lead to losses (negative feedback) are avoided.

ML became popular in recent years with the evolution of technology. As a consequence, the field now has a large number of algorithms that uses mathematical or statistical analysis to learn, draw conclusions or infer data. This number is ever increasing, because of the amount of scientific publications that propose variations or combinations of these algorithms. For that reason, academia has tried to categorize ML algorithms based on the mathematical intuitions or purposes that they present. Some results can be found on [56], and [34], although the field still do not have a standard classification.

3 Systematic Review

When developing RSs, software engineers must decide which recommender algorithm to be used among several available ML algorithms. This choice has great impacts in the rationale of the RS, in the data that will be needed from users and recommendation items, and in performance issues. The number of algorithm variations and combinations in the literature turns this choice into a challenging task.

In addition, software engineering research field is concerned with studying software development with the goal of improving the knowledge about the field and facilitating developer tasks with tools. As ML research field is constantly changing, new studies must be done to observe new open problems and trends, and further enrich the knowledge base about the field.

The author then decided to run a systematic review to analyze the development of RS containing ML algorithms and to draw conclusions. The systematic review has two main goals: to identify which ML algorithms are used in RSs and to discover open questions in RS development that SE field can assist. The study though had one restriction. The author decided to limit the set of scientific publications to those who describe a case study, or any form of implementation. A case study is a scientific method that describes real world phenomena, either by observing or applying a new technology. In the scope of this work, the author is concerned with finding publications that were implemented and validated with real data. The restriction is formally defined as follows:

- R1: The study investigates RS approaches that contain ML algorithms and that were implemented and tested with real or simulation data.

The main reason for this restriction is that several publications in the literature propose new algorithms that are never tested or validated. These results represent noise in the conclusions, as they might never be used in real settings.

The systematic review has the following two research questions (RQ):

- RQ1: What are the most used machine learning algorithms in recommender systems?
- RQ2: What are the software engineering areas that can help answering open questions in the development of recommender systems containing machine learning algorithms?

To answer the first question, the author must read each publication returned by a search query and list the algorithms that are proposed. The strategy to answer the second question is different. First, the author limits the scope of SE areas to the five stages of the waterfall model for software lifecycle: requirements, design, implementation, verification, and maintenance [49]. The waterfall model describes high-level stages that any software development ideally follows. Next, the author reads the conclusion and future work sections of every publication and identifies problems or research opportunities that can be investigated. Lastly, the author classifies the problem or research opportunity into one of the five software lifecycle stages. By doing it, the author expect to have a perspective of which SE area future research can be focused on.

To have a set of publications with good quality and increase the strength of the conclusions, this study have some exclusion criteria (EC) that are applied to the retrieved publications prior to the reading process. They are presented below and are explained in the next paragraph.

- EC1: Publications must have been peer-reviewed and published in a conference, journal, press, etc. Publications that were not peer-reviewed or published are excluded.
- EC2: Publications must describe the recommender system and the machine learning algorithm sufficiently well that it is possible to identify the algorithm. Publications that do not clearly state the machine learning algorithm being used are excluded.
- EC3: Publications must contain the description of a case study or the implementation of the approach that is proposed. Publications that do not describe a real life implementation are excluded.
- EC4: Publications must be primarily in English. Publications in languages other than English are excluded.
- EC5: Books, letters, notes, and patents are excluded.
- EC6: Publications must be unique. If a publication is repeated, other copies of that publication are excluded.

There are several terms to refer to RSs. Based on [29] this systematic review considers synonyms of RS terms that replace “recommender” by “recommendation”, and terms that replace “system” by “platform”, or “engine”. This systematic review does not consider any “machine learning” synonyms.

The following research query (RQ) was created for this systematic review:

- RQ: TITLE-ABS-KEY(((recommender OR recommendation) PRE/0 (system OR engine OR platform)) AND "machine learning" AND (implementation OR "case study"))

This research query inspects publication title, abstract and keywords and attempts to find terms that relates to the field of RS, ML, and some indication that the proposed approach was implemented. PRE/0 is a search engine-specific syntax to express the relationship between two terms. In this search query, PRE\0 restricts the results to publications that have “recommender” (or its synonyms) followed by “system” (or its synonyms). Publications must also contain the term “machine learning” in its title, abstract, or keywords. To retrieve implemented approaches, the search query also looks for the terms “implementation” or “case study”.

The search query was used on the popular academic search engine Scopus [54] on July 24th. It returned 35 publication entries that had to be reviewed for quality. Some publication entries were excluded based on the exclusion criteria previously explained, and they are summarized on Table 3.1.

Table 3.1 - Publications that were included and excluded from the systematic review.

Total retrieved		35
Reason	**Publication**	**Total**
Conference / Proceedings entries		3
Not describing a case study or implementation	[46]	1
Not sufficiently describing the approach	[38] [51] [52]	3
Not able to access	[47]	1
Other language		0
Duplicated	[52]	1
Books, letters, notes, and patents		0
Total retained	[5] [7] [8] [9] [10] [12] [14] [19] [20] [21] [24] [25] [31] [36] [39] [41] [53] [55] [57] [58] [60] [61] [64] [64] [67] [68]	26

The number of publications to be read in the systematic review decreased from 35 to 26 when filtered by the exclusion criteria. Three of the publication entries were conference or proceeding descriptions and are excluded because they are not written scientific work. The author did not have access to one publication, even after asking help for colleagues and visiting libraries. This publication then was excluded from the systematic review. Scopus seems to have a minor issue because it listed the same publication twice. However, one copy of this publication was excluded according to EC6. The remaining 30 publications were downloaded and assessed for quality. The author read the abstract and the approach description and, if present, the case study description. Based on that, one publication was excluded for not describing a case study (EC3) and three others were excluded due to poor description of the approach and the ML algorithm (EC2). In the end, 26 publications were retained and ready to be analyzed.

4 Systematic Review Results

The reading process was focused into finding three pieces of information: the ML algorithm, the domain of the case study or implementation, and the problems or open questions associated with the proposed work. Therefore, after reading the abstract and the introduction, the author read the approach and case study description, and finally the conclusion or future work of each publication included in the systematic review.

The author has a spreadsheet with an identification of each publication alongside three spaces for noting the pieces of information previously described. After reading all publications, the author processed the information contained in the spreadsheet and organized in a presentable manner. The results and conclusions are presented in the following paragraphs.

Processing ML algorithm names means that the author separated algorithms into categories based on [56]. Some algorithms had obvious classifications because they were small variations of well-established algorithms (e.g. incremental matrix factorization is a variant of the matrix factorization algorithm). However, some algorithms do not intuitively fit on any category. For these cases, the algorithm was listed under a new category with its own name (e.g. Topic Independent Scoring Algorithm is on category TISA). Table 4.1 shows the results.

Table 4.1 - Types of machine learning algorithms used in recommender systems.

Category	Publications	Total
Bayesian	[20] [21] [24] [36] [39] [55] [66]	7
Decision Tree	[21] [36] [39] [55] [57]	5
Matrix factorization-based	[7] [53] [60] [61]	4
Neighbor-based	[25] [31] [39] [60]	4
Neural Network	[5] [21] [39] [41]	4
Rule Learning	[24] [25] [36] [58]	4
Ensemble	[10] [21] [57]	3
Gradient descent-based	[8] [14] [60]	3
Kernel methods	[14] [21] [66]	3
Clustering	[19] [20]	2
Associative classification	[36]	1
Bandit	[12]	1
Lazy learning	[21]	1
Regularization methods	[14]	1
Topic Independent Scoring Algorithm	[41]	1

When inspecting Table 4.1, one may notice that Bayesian approaches are more used as the algorithm for RSs in 7 out of the 26 publications read. The author theorizes that this result is due to the reduced complexity of Bayesian calculations. The same explanation may justify the use of Decision Trees as the second most approach for ML algorithms in RS. Both Bayesian approaches and Decision Trees have similar underlying calculations and became popular in recent years.

Neural networks are also popular and their usage seems promising, with applications in self-driving cars and image inspection and classification. Although, neural network approaches were found in four publications, much more work can be done in using algorithms of this type for recommending items to users. One reason that explains a lower adoption of neural networks in RS field is that researchers still do not fully understand why some neural networks take a particular decision. The knowledge about this method is still under study.

Two other ML areas that became popular in the last two decades are Bandit algorithms and Ensemble approaches, but both had a low number of publications in this systematic review. Bandit algorithms are used to decide how many times and in which order decisions are made to maximize a given value [65]. Its high complexity may be a decisive factor that prevents the adoption of this algorithm in RSs. Ensemble learning [33] became increasingly popular especially after the Netflix Prize [42] in 2009. The intuition is to combine several results from different algorithms into a large knowledge base and draw conclusions from it. Similar to the Bandit algorithm, the complexity of implementing and generating results impedes its growth in the RS field.

While reading the 26 publications of this systematic review, the author noted that some of them described the ML algorithm with some mathematical or statistical method. For example, [12] describes a multi-armed Bandit algorithm with cosine measure. Other works described the ML algorithm in a distributed environment with Map Reduce. Table 4.2 depicts the publications that had a mathematical or statistical method, and those who used MapReduce algorithm.

Table 4.2 - Publications that described a machine learning algorithm with a mathematical or statistical method, and publications that did it with MapReduce algorithm.

Method	Publications	Total
Mathematical / Statistical	[7] [8] [9] [10] [12] [14] [20] [25] [39] [53] [64] [67] [68]	13
MapReduce	[19] [20] [53] [61]	4

Mathematical or statistical methods are calculations such as cosine measure [18], least squares [1] or Pearson correlation [18]. They were listed separately, because ML algorithms use them to generate parameters that help in recommendations. It is expected that the great majority of publications describe a mathematical or statistical method because ML research field has its roots in these areas. MapReduce algorithm [17], which became popular in mid-1990s, is a programming model suited for distributed and parallel computation. Four of the publications described their approach in an environment which MapReduce algorithm was used. Although it was not the main algorithm, MapReduce presence shows the importance of the field of Big Data in the future of RSs.
The author, when reading the case study or implementation description, noted the domains in which the publications validate their proposals. The identification of these domains enriches the knowledge of the field and help future researchers when validating their work. Table 4.3 shows the results. Note that some publications appear more than once, either in the same or in different rows. It happens because some publications reported more than one case study or implementation.

Table 4.3 - Domains of case study and implementations described in the publications.

Domain	Publications	Total
Movie	[5] [5] [5] [14] [36] [39] [53] [53] [60] [60]	10
Documents	[9] [19] [20] [20]	4
Product review	[25] [51] [57] [58]	4
Jokes	[14] [60]	2
Music	[14] [53]	2
TV	[58] [68]	2
Academic	[61]	1
Books	[36]	1
Elections	[64]	1
E-mail	[24]	1
E-shop	[21]	1
Mobile navigation	[12]	1

Safety-critical process automation	[10]	1
Social	[31]	1
Stocks	[67]	1
Telecommunications	[10]	1
Tourism	[66]	1
Traffic	[55]	1
Webpages	[8]	1

The domain of Movies is the most used one with 10 occurrences in the 26 publications. One reason to this result is the ease of access to data in the movie domain. The University of Minnesota maintains a dataset with several movie ratings, named MovieLens [44], which is widely used in several RS case studies. Another source of user ratings in the movie domain is the Internet movie database (IMDb) [26], which contains millions of titles and ratings that can be gathered to build a testing dataset. The domains of Documents and Product reviews obtained expressive results with four occurrences each. Text processing is one of the most popular forms of data gathering. Several algorithms are aimed at extracting information from text sources, and this is the main factor behind the position of both domain areas in this ranking.

The second research questions (RQ2) is concerned with the software engineering areas that research effort can be improved in RS development with a ML algorithm. After listing five SE high-level areas, taken from the Waterfall model for software development, the author read both the conclusion and the future work sections of publications looking for the problems and future work described. Later, the author separated the results into the five areas. The final result is shown in Table 4.4.

Table 4.4 - Number of publications that reported problems in each of the five software engineering areas considered in this study.

SE area	Publications	Total
Requirements	[5]	1
Design	[25] [24] [55]	3
Implementation	[10] [12] [19] [24] [25] [39] [61] [64] [66]	9
Verification	[8] [10] [12] [20] [24] [25] [39] [51] [53] [61] [64] [67]	12
Maintenance		0

Roughly 80% of the publications describe problems or future work that focus on the implementation or verification areas. This result highlights the importance of these areas in RS development. However, limiting the search query results to publications that have a case study or an implementation description had an impact on the number of problems and future work reported. Contrarily, only four publications were concerned with problems or future work in requirements, design or maintenance of RSs. This implies that little effort is being spent on these areas and several open questions may exist.

5 Conclusion & Future Work

Nowadays, recommender systems (RS) are widely used in e-commerce, social network, and several other domains. Since its introduction in mid 1990s, this research field has been evolving. One progress step in RS history is the adoption of machine learning (ML) algorithms, which allow computers to learn based on user information and to further personalize recommendations. Machine learning is an Artificial Intelligence (AI) research field that encompasses algorithms whose goal is to predict the outcome of data processing. ML has major breakthroughs in the fields of image recognition, search engines, and security. However, ML field has several algorithms described in the literature, with characteristics that make them suitable for an application or not. The literature lacks a good classification of the algorithms showing the environment they are suitable to be used. Therefore, choosing a ML algorithm to be used in RSs is a difficult task. In addition, researchers do not have a clear view of the trends in ML algorithm usage for RSs to focus studies on. Software engineering (SE) is a computer science field that studies the development of computer software from conception to implementation, and maintenance, from different perspectives. The field has tools that can assist the development of RSs that contain ML algorithms. However, researchers need to know which SE area RS development lacks resources. This study then proposes a systematic review to observe the ML algorithms that are used in RSs and what SE areas can assist the development of such RSs.

The systematic review collected 26 publications, after filtering out noise according to exclusion criteria. All publications were read and the conclusions are as follows. In RS development, the ML algorithm used is a Bayesian or a decision tree approach in most of the cases. Their recent popularity and the low complexity in calculations and implementation contribute for the result. Moreover, most of the approaches have a mathematical or statistical description of algorithms, since the ML field has origins in mathematics and statistics. Four of the publications have proposed RSs that work with the MapReduce algorithm. This result shows that the field of Big Data is relevant in RS studies, allowing even more personalized recommendations.

During the systematic review, the author investigated the domains in which proposals of RSs with a ML algorithm were validated. The three most used domains are movies, documents, and product review, mainly because of the ease in accessing data. For the movie domain, MovieLens and IMDb are two online datasets of movie ratings. For the documents and product review domains, the author notes that text is one of the most common forms of processing data, and several ML algorithms were created for text processing.

The second goal of this study concerns analyzing the development of RSs that have an ML algorithm and discover problems or open questions that SE areas can assist. The problems or open questions were found by reading the conclusion and future work sections of publications, and they were classified in one of the five areas of the Waterfall model for software lifecycle. The results show that researchers report more problems and open questions in implementation and verification areas. However, as the search question looks for study case or implementation description in the publications, this affected the results. The lack of problems or open questions reported in the other SE areas (requirements, design, and maintenance) may indicate that there are research opportunities to be investigated.

This study serves as a basis for investigating RS development. In the future, more studies on the use of Bayesian algorithms in RSs can be done to observe the implications of their use, performance, and utility. Moreover, RS development lacks studies analyzing early stages, such as requirements and design, and late stages, such as maintenance. Open questions in these stages must be investigated to further improve the knowledge about the field.

References

1. Abdi, H. (2007). The method of least squares. *Encyclopedia of Measurement and Statistics. CA, USA: Thousand Oaks*.
2. Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, *17*(6), 734-749.
3. Adomavicius, G., & Tuzhilin, A. (2011). Context-aware recommender systems. In *Recommender systems handbook* (pp. 217-253). Springer US.
4. Ahmed, A., Kanagal, B., Pandey, S., Josifovski, V., Pueyo, L. G., & Yuan, J. (2013, February). Latent factor models with additive and hierarchically-smoothed user preferences. In *Proceedings of the sixth ACM international conference on Web search and data mining* (pp. 385-394). ACM.
5. Alvarez, S. A., Ruiz, C., Kawato, T., & Kogel, W. (2011). Neural expert networks for faster combined collaborative and content-based recommendation. *Journal of Computational Methods in Sciences and Engineering*, *11*(4), 161-172.
6. Apte, C. (2010). The role of machine learning in business optimization. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (pp. 1-2).
7. Arenas-García, J., Meng, A., Petersen, K. B., Lehn-Schioler, T., Hansen, L. K., & Larsen, J. (2007, August). Unveiling music structure via plsa similarity fusion. In *Machine Learning for Signal Processing, 2007 IEEE Workshop on* (pp. 419-424). IEEE.
8. Balabanović, M. (1998). Exploring versus exploiting when learning user models for text recommendation. *User Modeling and User-Adapted Interaction*, *8*(1-2), 71-102.
9. Bhatia, L., & Prasad, S. S. (2015, February). Building a Distributed Generic Recommender Using Scalable Data Mining Library. In *Computational Intelligence & Communication Technology (CICT), 2015 IEEE International Conference on* (pp. 98-102). IEEE.
10. Borg, M. (2014, September). Embrace your issues: compassing the software engineering landscape using bug reports. In *Proceedings of the 29th ACM/IEEE international conference on Automated software engineering* (pp. 891-894). ACM.
11. Bouneffouf, D., Bouzeghoub, A., & Ganarski, A. L. (2013, January). Risk-aware recommender systems. In *Neural Information Processing* (pp. 57-65). Springer Berlin Heidelberg.

12. Bouneffouf, D., Bouzeghoub, A., & Gançarski, A. L. (2012). Hybrid-ε-greedy for mobile context-aware recommender system. In *Advances in Knowledge Discovery and Data Mining* (pp. 468-479). Springer Berlin Heidelberg.
13. Burhams, D., & Kandefer, M. (2004). Dustbot: Bringing Vacuum-Cleaner Agent to Life. *Accessible Hands-on Artificial Intelligence and Robotics Education*, 22-24.
14. Chen, H., Tang, Y., Li, L., Yuan, Y., Li, X., & Tang, Y. (2013). Error analysis of stochastic gradient descent ranking. *Cybernetics, IEEE Transactions on*, *43*(3), 898-909.
15. Chen, L. S., Hsu, F. H., Chen, M. C., & Hsu, Y. C. (2008). Developing recommender systems with the consideration of product profitability for sellers. *Information Sciences*, *178*(4), 1032-1048.
16. Cui, Q., Bai, F. S., Gao, B., & Liu, T. Y. (2015). Global Optimization for Advertisement Selection in Sponsored Search. *Journal of Computer Science and Technology*, *30*(2), 295-310.
17. Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, *51*(1), 107-113.
18. Egghe, L., & Leydesdorff, L. (2009). The relation between Pearson's correlation coefficient r and Salton's cosine measure. *Journal of the American Society for information Science and Technology*, *60*(5), 1027-1036.
19. Ericson, K., & Pallickara, S. (2011, December). On the performance of distributed clustering algorithms in file and streaming processing systems. In *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on* (pp. 33-40). IEEE.
20. Ericson, K., & Pallickara, S. (2013). On the performance of high dimensional data clustering and classification algorithms. *Future Generation Computer Systems*, *29*(4), 1024-1034.
21. Felden, C., & Chamoni, P. (2007, January). Recommender systems based on an active data warehouse with text documents. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on* (pp. 168a-168a). IEEE.
22. Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, *29*(2-3), 131-163.
23. Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, *35*(12), 61-70.
24. Gorodetsky, V., Samoylov, V., & Serebryakov, S. (2010, August). Ontology–based context–dependent personalization technology. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on* (Vol. 3, pp. 278-283). IEEE.
25. Hariri, N., Castro-Herrera, C., Mirakhorli, M., Cleland-Huang, J., & Mobasher, B. (2013). Supporting domain analysis through mining and recommending features from online product listings. *Software Engineering, IEEE Transactions on*, *39*(12), 1736-1752.
26. IMDb. Internet Movie Database. Retrieved August 8, 2015, from http://www.imdb.com

27. Isazadeh, A. (2004). Software engineering: The trend. *INFORMATICA-LJUBLJANA-*, *28*(2), 129-138.
28. Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*(3), 264-323.
29. Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). *Recommender systems: an introduction*. Cambridge University Press.
30. Karimanzira, D., Otto, P., & Wernstedt, J. (2006, February). Application of machine learning methods to route planning and navigation for disabled people. In *MIC'06: Proceedings of the 25th IASTED international conference on Modeling, indentification, and control* (pp. 366-371).
31. Kelley, P. G., Hankes Drielsma, P., Sadeh, N., & Cranor, L. F. (2008, October). User-controllable learning of security and privacy policies. In *Proceedings of the 1st ACM workshop on Workshop on AISec* (pp. 11-18). ACM.
32. Kononenko, I. (2001). Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine*, *23*(1), 89-109.
33. Krogh, A., & Sollich, P. (1997). Statistical mechanics of ensemble learning. *Physical Review E*, *55*(1), 811.
34. Kulkarni, S. (Ed.). (2012). *Machine Learning Algorithms for Problem Solving in Computational Applications: Intelligent Techniques: Intelligent Techniques*. IGI Global.
35. Liu, J., Dolan, P., & Pedersen, E. R. (2010, February). Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces* (pp. 31-40). ACM.
36. Lucas, J. P., Segrera, S., & Moreno, M. N. (2012). Making use of associative classifiers in order to alleviate typical drawbacks in recommender systems. *Expert Systems with Applications*, *39*(1), 1273-1283.
37. Lv, H., & Tang, H. (2011, October). Machine learning methods and their application research. In *2011 International Symposium on Intelligence Information Processing and Trusted Computing* (pp. 108-110). IEEE.
38. Ma, L., Haihong, E., & Xu, K. (2011, October). The design and implementation of distributed mobile points of interest (POI) based on Mahout. In *Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on* (pp. 99-104). IEEE.
39. Marović, M., Mihoković, M., Mikša, M., Pribil, S., & Tus, A. (2011, May). Automatic movie ratings prediction using machine learning. In *MIPRO, 2011 Proceedings of the 34th International Convention* (pp. 1640-1645). IEEE.
40. Martens, H. H. (1959). Two notes on machine "Learning". *Information and Control*, *2*(4), 364-379.
41. Martineau, J. C., Cheng, D., & Finin, T. (2013). TISA: topic independence scoring algorithm. In *Machine Learning and Data Mining in Pattern Recognition* (pp. 555-570). Springer Berlin Heidelberg.
42. Meuth, R. J., Robinette, P., & Wunsch, D. C. (2008, June). Computational intelligence meets the netflix prize. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on* (pp. 686-691). IEEE.

43. Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (Eds.). (2013). *Machine learning: An artificial intelligence approach*. Springer Science & Business Media.
44. MovieLens. Non-commercial, personalized movie recommendations. Retrieved August 8, 2015, from https://movielens.org
45. O'Donovan, J., & Smyth, B. (2005, January). Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces* (pp. 167-174). ACM.
46. Oard, D. W. (1997). The state of the art in text filtering. *User Modeling and User-Adapted Interaction*, *7*(3), 141-178.
47. Padmanabhan, V., Ganapavarapu, P., & Sattar, A. (2014). Group recommender systems: Some experimental results.
48. Patrick, E. A., & Fischer, F. P. (1970). A generalized k-nearest neighbor rule. *Information and control*, *16*(2), 128-152.
49. Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave Macmillan.
50. Rodriguez, M., Posse, C., & Zhang, E. (2012, September). Multiple objective optimization in recommender systems. In *Proceedings of the sixth ACM conference on Recommender systems* (pp. 11-18). ACM.
51. Russell, I., & Markov, Z. (2009, October). A multi-institutional project-centric framework for teaching AI concepts. In *Frontiers in Education Conference, 2009. FIE'09. 39th IEEE* (pp. 1-6). IEEE.
52. Santos, O. C. (2013). PhD Dissertation Summary: Contributions to the Design, Implementation and Evaluation of Adaptive Learning Management Systems based on standards, which integrate instructional Design with User Modelling based on Machine Learning. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, *16*(51), 1-4.
53. Schelter, S., Boden, C., Schenck, M., Alexandrov, A., & Markl, V. (2013, October). Distributed matrix factorization with mapreduce using a series of broadcast-joins. In *Proceedings of the 7th ACM Conference on Recommender Systems* (pp. 281-284). ACM.
54. Scopus. Retrieved August 8, 2015, from http://www.scopus.com
55. Seric, L., Jukic, M., & Braovic, M. (2013, May). Intelligent traffic recommender system. In *Information & Communication Technology Electronics & Microelectronics (MIPRO), 2013 36th International Convention on* (pp. 1064-1068). IEEE.
56. Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.
57. Shinde, A., Haghnevis, M., Janssen, M. A., Runger, G. C., & Janakiram, M. (2013). Scenario Analysis Of Technology Products With An Agent-Based Simulation And Data Mining Framework. *International Journal of Innovation and Technology Management*, *10*(05), 1340019.
58. Smyth, B., McCarthy, K., Reilly, J., O'Sullivan, D., McGinty, L., & Wilson, D. C. (2005). Case Studies in Association Rule Mining for Recommender Systems. In *IC-AI* (pp. 809-815).

59. Steck, H. (2013, October). Evaluation of recommendations: rating-prediction and ranking. In *Proceedings of the 7th ACM Conference on Recommender Systems* (pp. 213-220). ACM.
60. Takács, G., Pilászy, I., Németh, B., & Tikk, D. (2009). Scalable collaborative filtering approaches for large recommender systems. *The Journal of Machine Learning Research*, *10*, 623-656.
61. Tewari, N. C., Koduvely, H. M., Guha, S., Yadav, A., & David, G. (2013, October). MapReduce implementation of Variational Bayesian Probabilistic Matrix Factorization algorithm. In *Big Data, 2013 IEEE International Conference on* (pp. 145-152). IEEE.
62. Thrun, S. (2007). Self-Driving Cars-An AI-Robotics Challenge. In *FLAIRS Conference* (p. 12).
63. Torralba, A., Fergus, R., & Weiss, Y. (2008, June). Small codes and large image databases for recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (pp. 1-8). IEEE.
64. Tsapatsoulis, N., Agathokleous, M., Djouvas, C., & Mendez, F. (2015). On the Design of Social Voting Recommendation Applications. *International Journal on Artificial Intelligence Tools*, *24*(03), 1550009.
65. Vermorel, J., & Mohri, M. (2005). Multi-armed bandit algorithms and empirical evaluation. In *Machine Learning: ECML 2005* (pp. 437-448). Springer Berlin Heidelberg.
66. Wang, Y., Chan, S. C. F., & Ngai, G. (2012, December). Applicability of demographic recommender system to tourist attractions: a case study on trip advisor. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 03* (pp. 97-101). IEEE Computer Society.
67. Yoo, J., Gervasio, M., & Langley, P. (2003, January). An adaptive stock tracker for personalized trading advice. In *Proceedings of the 8th international conference on Intelligent user interfaces* (pp. 197-203). ACM.
68. Zibriczky, D., Petres, Z., Waszlavik, M., & Tikk, D. (2013, December). EPG content recommendation in large scale: a case study on interactive TV platform. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on* (Vol. 2, pp. 315-320). IEEE.