CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

ECT: AN OFFLINE METRIC TO EVALUATE CASE-BASED EXPLANATIONS FOR RECOMMENDER SYSTEMS

LUCAS GABRIEL LAGE COSTA

Supervisor: Anisio Mendes Lacerda
Universidade Federal de Minas Gerais

BELO HORIZONTE
DEZEMBRO DE 2020

LUCAS GABRIEL LAGE COSTA

ECT: AN OFFLINE METRIC TO EVALUATE CASE-BASED EXPLANATIONS FOR RECOMMENDER SYSTEMS

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Matemática e Computacional do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para a obtenção do título de Mestre em Modelagem Matemática e Computacional.

Área de concentração: Modelagem Matemática e Computacional.

Linha de pesquisa: Sistemas Inteligentes.

Supervisor: Anisio Mendes Lacerda
Universidade Federal de Minas Gerais

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL
BELO HORIZONTE
DEZEMBRO DE 2020

Costa, Lucas Gabriel Lage
C837e ECT: an offline metric to evaluate case-Based explanations for recommender systems / Lucas Gabriel Lage Costa. – 2020.
48 f.

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional.
Orientador: Anísio Mendes Lacerda.
Dissertação (mestrado) – Centro Federal de Educação Tecnológica de Minas Gerais.

1. Lógica matemática – Teses. 2. Sistemas de recomendação (Filtragem de informação) – Teses. 3. Engenharia matemática – Teses. 4. Inteligência computacional – Teses. I. Lacerda, Anísio Mendes. II. Centro Federal de Educação Tecnológica de Minas Gerais. III. Título.

CDD 519.6

Elaboração da ficha catalográfica pela bibliotecária Jane Marangon Duarte, CRB 6º 1592 / Cefet/MG

SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
COORDENAÇÃO DO PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

"ECT: AN OFFLINE METRIC TO EVALUATE CASE-BASED EXPLANATIONS FOR RECOMMENDATION SYSTEMS"

Dissertação de Mestrado apresentada por **Lucas Gabriel Lage Costa**, em 14 de dezembro de 2020, ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, e aprovada pela banca examinadora constituída pelos professores:

Prof. Dr. Anisio Mendes Lacerda (Orientador)
Universidade Federal de Minas Gerais

Prof. Dr. Daniel Hasan Dalip
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Adriano César Machado Pereira
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Flávio Luis Cardeal Pádua
Centro Federal de Educação Tecnológica de Minas Gerais

Visto e permitida a impressão,

Prof. Dr. Thiago de Souza Rodrigues
Coordenador do Programa de Pós-Graduação em
Modelagem Matemática e Computacional

Este trabalho é dedicado a você, familiar ou amigo que contribuiu muito na minha caminhada. Sem vocês eu nada seria.

Acknowledgements

Quero agradecer o meu professor orientador Anisio Mendes Lacerda, pelo empenho, apoio e paciência ao longo da elaboração do meu projeto de pesquisa. Também gostaria de deixar um agradecimento especial ao CEFET-MG por possibilitar a execução deste trabalho científico.

Agradeço também aos meus familiares que me ajudaram na realização do meu sonho. Aos meus amigos de trabalho e parceiros de pesquisa, por toda a ajuda e apoio durante este período tão importante da minha formação acadêmica. A todas as pessoas que direta ou indiretamente contribuíram para a realização da minha pesquisa.

“Querias ser livre. Para essa liberdade, só há um caminho: o desprezo das coisas que não dependem de nós.” (Epicteto)

Resumo

Sistemas de recomendação sugerem itens que melhor atendem aos gostos e interesses dos usuários. Eles são sistemas complexos e, frequentemente, são baseados em modelos do tipo caixa preta. No entanto, sabe-se que entender as recomendações é essencial para ajudar a construir a confiança e o envolvimento do usuário nesses sistemas. Portanto, as explicações das recomendações são uma forma de atingir esse objetivo. Uma explicação é uma informação exibida aos usuários, detalhando as motivações de porque um determinado item foi recomendado. A explicação baseada em caso é um dos estilos de explicação mais comumente usados em sistemas de recomendação. Essa abordagem é focada em fornecer explicações na forma de itens que foram avaliados previamente, como, por exemplo, em: "Porque você gostou de A, recomendamos B". Para medir a qualidade dos métodos que geram explicações, podemos fazer avaliações *online* e *offline*. As avaliações online requerem interação direta com os usuários e as avaliações offline requerem apenas dados históricos, ou seja, extraídos anteriormente à recomendação e explicação. A avaliação online pode tratar mais aspectos das explicações e produzir resultados mais completos, por isso é incentivada, mas nem sempre os testes envolvendo usuários estão acessíveis. A avaliação offline é frequentemente adotada pois elimina riscos de apresentar explicações que podem afetar negativamente a experiência dos usuários, além de ser mais fácil de implementar. Poucos trabalhos na literatura abordam métricas offline para medir a qualidade dos métodos de explicação baseados em casos. Assim, este trabalho propõe cobrir esta demanda por meio de uma nova métrica offline – *ECT* (*Ease-of-interpretation Coverage Triviality*) – para avaliar aspectos mais complexos dos métodos de explicação baseados em casos. É composta por três submétricas, uma delas já existente (*Coverage*), e duas novas (*Ease-of-interpretation* e *Triviality*). Experimentos foram conduzidos em cenários offline e online para validar o poder discriminativo da métrica proposta *ECT*, e mostram que ela apresenta uma correlação de 0,67 com o que os usuários consideram boas explicações para suas recomendações.

Palavras-chave: Interpretabilidade. Sistemas de recomendação. Explicações de caixa preta. Métrica de avaliação offline

Abstract

Recommender systems suggest items that best fit the users' interests and tastes. They are complex systems, which are often designed as black-box models. However, it is well-known that understanding recommendations is essential to help building users' trust and engagement in these systems, so explanations for recommendations are a way to achieve this goal. An explanation is a piece of information displayed to users, explaining why a particular item is recommended. The case-based explanation is one of the most commonly used explanation styles in recommender systems. This approach is focused on providing explanations in the form of previously liked items that are used to make the recommendation, such as: "Because you liked A, we recommend B". To measure the quality of methods that generate explanations, we can conduct online and/or offline evaluations. Online evaluations require direct interaction with users, and offline evaluations require just previously historic data. Online evaluation can handle more aspects of the explanations and produces more complete results, so it is encouraged, but tests involving users are not always accessible. In many scenarios, offline evaluation is preferred because it will not affect user's experience through bad explanations, and it is also easier to implement than the online counterpart. Few research projects approach offline metrics to measure the quality of case-based explanation methods, so it is an ill-defined problem. Thus to cover this gap in the literature, we propose a new offline metric – *ECT* (Ease-of-interpretation Coverage Triviality) – to evaluate more complex aspects of case-based explanation methods. It is based on three sub metrics, one of them has already been proposed: (*Coverage*), and two new metrics: (*Ease-of-interpretation* and *Triviality*). Experiments were conducted in offline and online scenarios to validate the discriminative power of the proposed metric *ECT*, and show that it presents a 0.67 correlation with what users consider good explanations to their recommendations.

Keywords: Interpretability. Recommender systems. Black-box explanations. Offline evaluation metric

List of Figures

List of Tables

List of Algorithms

List of abbreviations and acronyms

ECT	*Ease-of-interpretation Coverage Triviality*

List of symbols

U	Set of all users
I	Set of all items
R_u	Set of all recommendations for the user u. $R_u \subset I$
$\widehat{I}_u$	Set of all previously liked items by the user u. $\widehat{I}_u \subset I$
$\widehat{U}_r$	Set of users who previously liked item r. $\widehat{U}_r \subset U$
$E_{u,r}$	Set of explanations for the recommendation of the item r for user u $E_{u,r} \subset \widehat{I}_u$
$f(u, e, r)$	Score value of the previously liked item e as an explanation to the recommendation of item r for user u
η	Constant to represent how many items in an explanation set, the users are able to interpret
TE_t	Set of items considered as trivial explanations to the item t. $TE_t \subset I$
T	Set of items which have trivial explanations. $\{\forall t \in T \vert t \in I \wedge \vert TE_t\vert > 0\}$
$\ddot{U}_t$	Set of users who previously liked at least one item from TE_t, and received t as a recommendation. $\ddot{U}_t \subset U$
K	The number of neighbors used to get the transactions for Apriori
D	Size of the transactions for Apriori
LU_u	Latent factors of the user u
LI_i	Latent factors of the item i
S_u	Set of similar users to user u. $S_u \subset U$

Contents

1 Introduction

Recommender systems are tools that provide suggestions on items that are most likely to be of interest to a particular user. The goal is helping users to find relevant and personalized content among a large amount of information (RESNICK et al., 1994; RESNICK; VARIAN, 1997). Recommendations are related to various types of decision-making processes. For example in real world applications: which items to buy (AMAZON, (accessed October 19, 2020)), which music to listen (SPOTIFY, (accessed October 19, 2020)) or which news to read (THE. . . , (accessed October 19, 2020)).

Item is the generic term used to describe what the system recommends to users. Recommender systems typically focus on a specific item domain (for example, products, movies, music, or news) and, consequently, their design and graphic interface are customized to provide useful suggestions and effective for that specific domain. However, the most successful central recommendation techniques are domain free, such as the ones based on the matrix factorization method (KOREN; BELL; VOLINSKY, 2009).

Recommender systems are primarily aimed at individuals who do not have enough personal experience or competence to deal with information overload. Such systems have proven to be a valuable means of helping users make choices among a potentially overwhelming number of alternative items that a website can offer, for example. Due to its commercial application, there is an economic interest in developing research in the area.

The development of the full aspects of the recommender system is a multi-disciplinary task. Involving specialists from various areas, such as Artificial Intelligence, Human-Computer Interaction, Information Technology, Data Mining, Statistics, Adaptive User Interfaces, Information Systems Decision Support, Marketing, and Consumer Behavior (RICCI et al., 2010).

The state of the art in recommender systems are the models based on the collaborative filtering approach. The basic idea of this approach is that users who shared the same interest in the past would have similar preferences in the future (JANNACH et al., 2010). For instance, the user A and the user B highly rated the same movies in the past, then A positively rates a movie that user B has not seen yet, so this movie should also be recommended to the user B.

The collaborative filtering model that achieved the best results in recommending task was the model based on latent factors. It seeks to predict the evaluations, characterizing both users and items, in the form of numerical vectors in the same abstract space. These vectors are factors inferred from user rating patterns (KOREN; BELL; VOLINSKY, 2009).

The meaning of a latent factor is indirectly observed, instead of being provided by the model. In the movie domain, for example, the meaning of an item latent factor can be quantifications of well-defined aspects, such as the amount of action present in the movie, children orientation content, or comedy vs drama genres. Those aspects can also be less defined, such as depth in character development; exoticism of the film. Or even completely uninterpretable. The user latent factors can measure how much a given user likes the aspect represented by each one of the item latent factors.

The explanation is a piece of information displayed to users, detailing why a particular item is being recommended. Explanations for recommendations can be generated from different information sources and be presented in different display styles (TINTAREV; MASTHOFF, 2015), for example, a relevant user or item related to the recommendation, a sentence, an image, or a set of reasoning rules. Additionally, there could exist many different explanations for the same recommendation.

Explanations aid users make more precise decisions (BILGIC; MOONEY, 2005), improve user acceptance of recommendations (HERLOCKER; KONSTAN; RIEDL, 2000), and improve the trustability in the recommender system (SINHA; SWEARINGEN, 2002). In this project, we exploit the case-based explanation style, which means previously liked items are used to explain current recommendations.

In evaluating recommendation performance, the information about which items each user liked is available in the test set, allowing the application of efficient metrics in this evaluation. On the other hand, to evaluate the recommendation explanation performance, we do not have the information about what is the best explanation, so the best we can do are approximations by both online and offline evaluations (ZHANG; CHEN, 2018). Offline evaluation is preferred because it does not affect user experience and it requires only previously extracted data. However, a challenge here is evaluating the quality of case-based explanation methods. We emphasize that few previous research investigations tackle the problem of offline evaluation.

According to (TINTAREV; MASTHOFF, 2007a), the goals of the explanations are to improve the following attributes of the recommender system: transparency, scrutability, trustworthiness, effectiveness, persuasiveness, efficiency, and satisfaction. The increase in these attributes can be a metric to evaluate the quality of the explanations (TINTAREV; MASTHOFF, 2007b). The ways to measure this increase rely on online evaluations like: A/B test (ZHANG et al., 2014); *Movieexplain* (SYMEONIDIS; NANOPOULOS; MANOLOPOULOS, 2009), which is a prototype system where users can interact with the recommendations and explanations, while their satisfaction in the given explanation style is evaluated; and analysis of real users' behavior in a controlled environment test (CHEN et al., 2018; WANG et al., 2018).

The online approaches require participant recruiting and take a lot of time to be done.

Additionally, they may harm user experience by providing useless and irrelevant explanations. On contrary, they can handle more aspects of the explanation evaluation than the offline counterpart and can produce more complete results. Thus, they are encouraged but are not easily viable, particularly for research in academia (ZHANG; CHEN, 2018).

Usually, offline evaluation is easier to implement, but as mentioned before, few research projects approach it for the case-based explanations (ZHANG; CHEN, 2018). There are two offline methods to evaluate case-based explanations *Explainability* (ABDOLLAHI; NASRAOUI, 2017) and *Model Fidelity* (PEAKE; WANG, 2018). *Explainability* is a metric to evaluate recommendation explanations by calculating the probability of a recommended item be explainable for a user. *Model Fidelity* is a metric that measures the percentage of explainable items in the recommended items.

To cover the lack of offline metrics to measure the quality of case-based explanation methods, *ECT* is being proposed, as a new offline metric to evaluate more complex aspects of case-based explanation methods. It is composed of three sub-metrics, *Coverage*, *Ease-of-interpretation* and *Triviality*. This contribution can be promising for fastening the explanation evaluation process since it uses the widely available data instead of tests with users.

1.1 Project contributions

According to (TINTAREV; MASTHOFF, 2007b), if there is an increase of transparency, scrutability, trustworthiness, effectiveness, persuasiveness, efficiency, or satisfaction caused by explanations in the recommender system, this increase can be a metric to evaluate the quality of explanations. However, some aspects of these concepts are hard to quantify. In this work, we focus easy-of-interpretation, coverage, and triviality concepts to propose *ECT*, which is a new offline metric to evaluate case-based explanation methods.

This new metric is inspired by the concepts aforementioned and attempts to properly measure the quality of explanations in recommender systems. Specifically, each concept is translated into a metric, and *ECT* is a combination of these metrics, namely: *Triviality*, *Ease-of-interpretation* and *Coverage*.

Triviality attempts to quantify trustworthiness: if users perceive that there are good recommendations followed by obvious explanations (like a new sequel from an already seen movie franchise), they raise their confidence on the recommender system. Note that a trivial explanation is related to how the user understands the explanation and it is a desirable property by explanation algorithms. On contrary, a trivial recommendation refers to an obvious list of recommended items, and needs to be avoided by recommendation algorithms.

Ease-of-interpretation, in turn, works by penalizing large explanation sets, and han-

dles one aspect of persuasiveness: too much information has negative effects on how users can interpret the explanations (HERLOCKER; KONSTAN; RIEDL, 2000), and too much persuasion may have reverse reactions if the item recommended by the system is found to be unacceptable (TINTAREV; MASTHOFF, 2011). It also covers efficiency, once if users can interpret the explanations quickly, they can make decisions early.

Coverage is used as proposed in (PEAKE; WANG, 2018) and measures how many recommendations can be explained. It measures the effectiveness of the recommender system since explanations could lead to more acceptability of the recommendation.

Experiments were conducted on real-world datasets by considering both offline and online settings. First, were evaluated the scores assigned by *ECT* considering 3 representative recommendation methods, and 7 explanation methods. Next, was performed an online evaluation with 104 users to calculate the correlation of the values assigned by *ECT* and human perception on the quality of explanations. Results show there is a correlation of 0.67 between *ECT* and human evaluations.

In summary, this project has the following contributions:

- The introduction of the concept of Triviality and respective metric, which measures the quality of explanation methods, considering it should be able to explain trivial items, such as sequel of movies or neighbor establishments within the same category;
- The definition of the concept of Easy of Interpretation and respective metric, which together with triviality and coverage compose a new combined metric, called *ECT*;
- The evaluation of *ECT* in both offline and online experiments, showing it presents a correlation of 0.67 with users' opinion on the best explanations provided by the recommender systems.

1.2 Text organization

This text is organized as follows. In Chapter 2, there is the background about recommender systems and explanations, and also previous related work. The formalization and explanation of the proposed evaluation metric are in Chapter 3. Chapter 4 shows both the offline and online experiments. Future works are discussed in Chapter 5. Finally, conclusions are presented in Chapter 6.

2 Background and related work

2.1 Recommender systems

Recommender systems are techniques to provide suggestions of items that a given user may be interested in. They are present in several domains of recommendation – for example, song (SPOTIFY, (accessed October 19, 2020)); products (AMAZON, (accessed October 19, 2020)); books (LIBRARYTHING, (accessed October 19, 2020)); movies (NETFLIX, (accessed October 19, 2020)). These systems are designed to be used, mainly, by people who do not have sufficient time, personal experience or knowledge to handle the information overload (RESNICK; VARIAN, 1997; RICCI et al., 2010).

A traditional approach to build recommendations is the collaborative filtering approach. Specifically, considering the scenario where user A and user B have positively rated the same movies in the past, and then A makes a positive rating for a movie that B has not yet seen. Collaborative filtering-based methods follows the logic of presenting this film to user B as well (JANNACH et al., 2010).

In the context of recommender systems, the state of the art refers to collaborative filtering approaches(KHASMAKHI; BALAFAR; DERAKHSHI, 2019). As aforementioned, the basic idea of this approach is that if users share the same interest in the past, they will also have similar tastes in the future. By their ratings, users implicitly collaborate to search the most promising items among a wide variety of items available.

Collaborative filtering-based methods have evolved from simple distance-based methods to more sophisticated model-based ones, including those based on matrix factorization (KOREN; BELL; VOLINSKY, 2009) and deep learning (ZHANG et al., 2017). The collaborative filtering models that reach the best results are based on latent factors (KOREN; BELL; VOLINSKY, 2009). The latent factors approach is a vectorized way to represent users and items in abstract vector spaces, and each dimension of those vectors is related to a concept, the meaning of this concept is not easy to guess, it could even be abstract or undecipherable. In item vectors, the dimensions represent how much that item has of these concepts, and in users vectors, the dimensions measure how much interest the user has in these concepts.

In contrast with the first distance-based methods, where justifying a decision was simply based on the closest neighbors, latent factor models represent users and items in abstract vector spaces, where each dimension of these vectors is related to a concept. The concepts represented in these models are not always easy to grasp, and hence these models end up acting as black-boxes, given no explanations about how the recommendation is made.

Figure 1 illustrates a simple example of this approach. The X-axis represents the men vs. women orientated content. And the Y-axis represents reality vs. fantasy genre. So, as more to the right the movie is, the greater the proportion of men within its fan base, and as further down the movie is, the greater its quantity of fantasy elements. As for users, the more to the right he is, the greater his preference for men orientated movies, and the further down he is, the greater his preference for films with more fantasy elements. And vice versa.

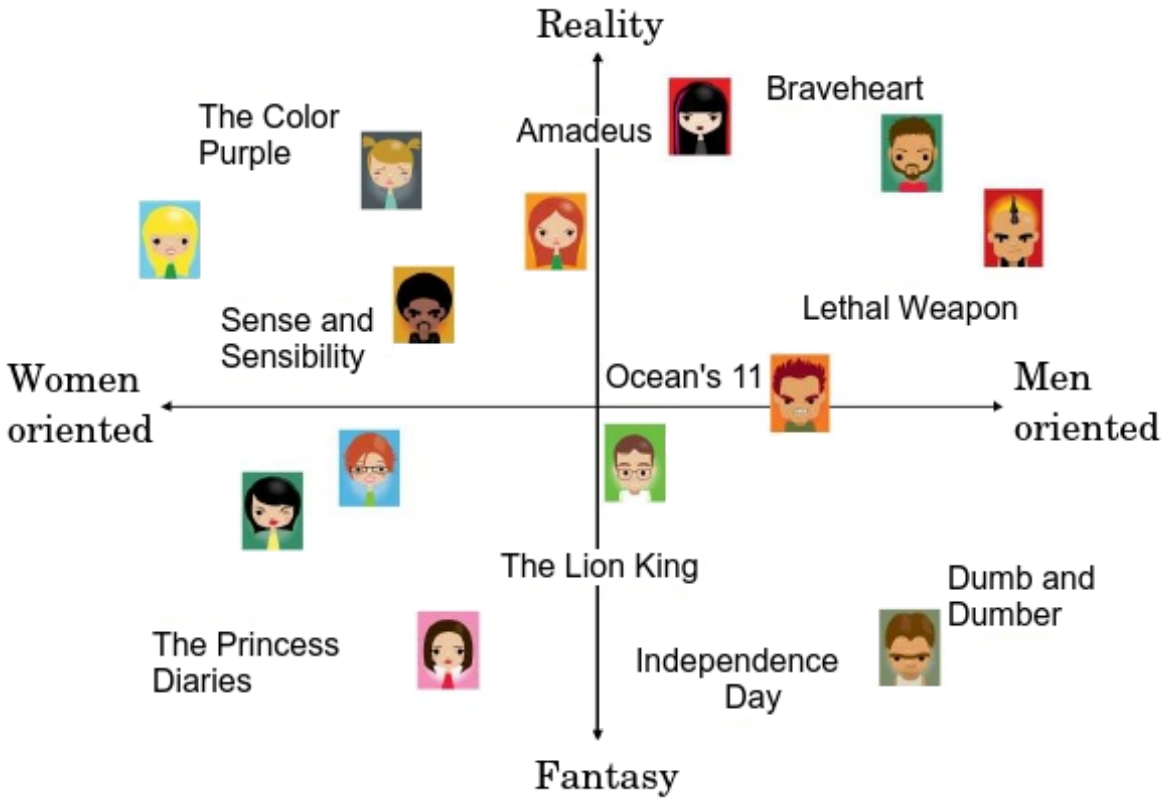


Figure 1 – Simplified example of the latent factors approach

The best technique for obtaining latent factors is the factoring of the rating matrix (KOREN; BELL; VOLINSKY, 2009). Given all users' ratings about the movies, this technique finds k factors for each user and each movie, and k is a parameter (usually between 20 to 100). This technique works by assigning random initial values to all factors. Then the iterative algorithm gradient descent is used to update these factors until the error converges to an acceptable value. Figure 2 exemplifies this technique, with matrix A representing the evaluations that users made about some movies. Each rating is a score from 1 to 5, and from them, the latent factors of users and movies are obtained. In this example $k = 3$.

Formalizing this process, we have that the Equation 1 represents the prediction of the evaluations that the users $u \in U$ would make on the item $i \in I$. The Equation 2 represents the set R_u of the N items that had the highest rating predictions, and that will be

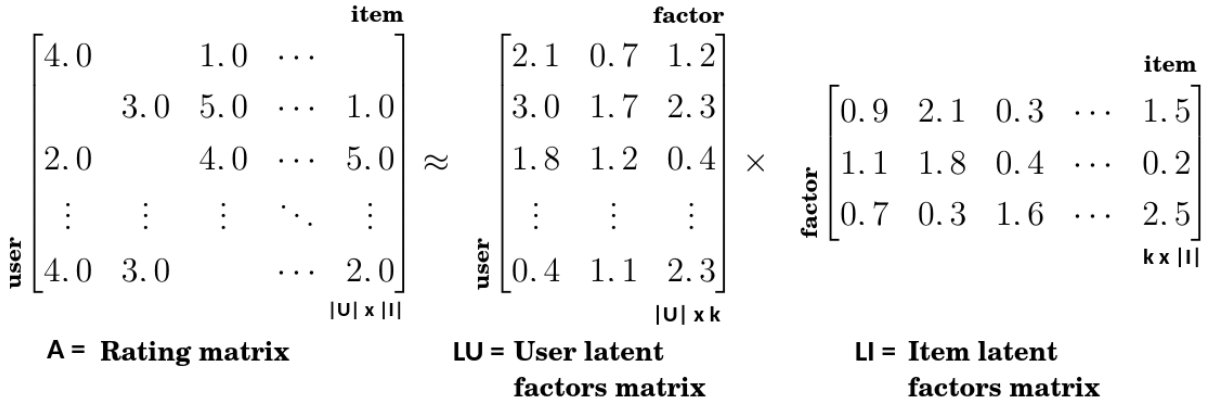


Figure 2 – Matrix factorization example

recommended to the user u.

$$p(u,i) = LI_i^T LU_u. \tag{1}$$

$$R_u = \{r_1,r_2,...,r_N \,|\forall r \in I \wedge p(u,r_n) \geq p(u,r_{n+1})\}. \tag{2}$$

Because both user and item factors are latent, the factoring technique does not provide the factor meaning. Discover the factors meaning are a difficult problem, which requires manual analysis of each value found. Some factors may be uninterpretable, and therefore impossible to label. Therefore, the labeling of latent factors is not part of the scope of the recommender system projects.

2.2 Explaining recommendations

The explanation is a piece of information presented to users, detailing why the decision model made a choice. In the recommender systems scenario, the explanation describes why a particular item is recommended to the user. Also, could exist many different explanations for the same recommendation.

Methods for machine learning explanation can either be model-intrinsic or model-agnostic (Post hoc) (MOLNAR, 2019). This criterion differs about who the interpretability is achieved: by restricting the complexity of the machine learning model (intrinsic) or by applying methods that analyze the model after training (agnostic).

The model-intrinsic approach refers to machine learning models that can be considered interpretable due to their simple structure, such as short decision trees or sparse linear models. Or some complex approaches that the explanation generation is part of the model training, for example, in recommender systems literature (LIU et al., 2019) propose a

recommendation method that generates recommendations and explanations via influence function at the same training process.

The model agnostic approach, or post hoc interpretability, allows the decision mechanism to be a black box, like in the state of the art algorithms in recommendations systems (KHASMAKHI; BALAFAR; DERAKHSHI, 2019). And the explanation model generates explanations after the decision mechanism's training when the decision has been made. Permutation feature importance is, for example, a post hoc interpretation method. In recommendations systems literature, (PEAKE; WANG, 2018) propose a post hoc interpretable approach that mine association rules among items in the recommender systems, and then provide the items in the rule as explanations.

The philosophy of these two approaches runs deep in our observation about human cognitive psychology. Sometimes we make decisions by careful, rational reasoning, so we can explain why we make those decisions. However, in other cases, we make decisions first and then seek explanations to support or justify ourselves.

Also, there are two others features related to explanations in recommender systems, the explanation attributes and explanation styles. The explanation attributes represent the benefits explanations provide to recommender systems, and they can be considered the criteria to measure which explanation is good. The explanation styles represents the form of the explanation and how it will be displayed. Those two features are describe in the next two subsections.

2.2.1 Explanation attributes

In (TAIE, 2013), there is a list of explanation attributes that represent benefits that explanations provide to the recommender system. Explaining the recommendations aid users make more precise decisions (BILGIC; MOONEY, 2005), improve user acceptance of recommendations (HERLOCKER; KONSTAN; RIEDL, 2000), and improve the trustability in the recommender system (SINHA; SWEARINGEN, 2002). So, some attributes can also be considered as criteria that measure if the quality of the explanation is good.

Some of these attributes may require certain overlapping and trade-offs. For example, personalized explanations may lead to greater satisfaction but not necessarily increase effectiveness, and also, transparency does not necessarily aid trust.

On the other hand, some attributes of explanations may lead to achieving the same goal. For example, transparency, trust, and satisfaction could be determined by measuring understandability.

These attributes are shown in Figure 3 and described bellow.

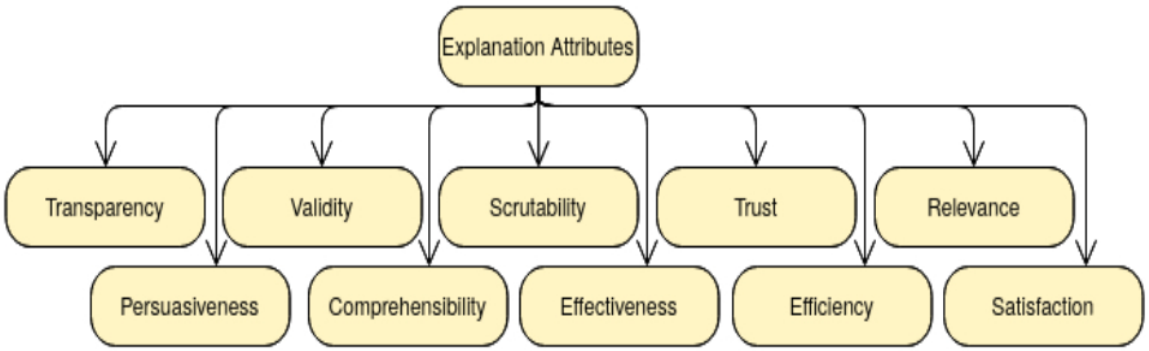


Figure 3 – Explanation attributes/benefits.

2.2.1.1 Transparency

Transparency intends to show how recommendations were made, giving an overview of how an item was recommended, or covering the broad concepts of the system. However, it may not describe how the logical and the calculation behind the recommender system really works.

In some cases, transparency can be challenging to make, such as: the difficulty of explaining the algorithms, the needy for protection of the business secrets, and more freedom to designing the explanations. If a recommendation does not show how the item was selected, then it is not transparent.

2.2.1.2 Validity

The explanations may be a way to allow users to validate the recommendations. It could happen by giving them a comparison of the requested and offered features used in the process. Even though validity is similar to transparency, it is not necessarily related.

2.2.1.3 Scrutability

Scrutability is the ability to enable users to tell if the system is right or wrong. It works together with transparency, allowing users to correct reasoning made by the system. Users will trust more if the system provides both explanations showing the transparency of its behavior and a mechanism to handle mistakes about the recommendations. It was not a task for the users to correct wrong recommendations on their own, but a guide to give extra help in the solution process.

2.2.1.4 Trust

In some aspects, trust-building can be handled as a way to simplify the decision-making process. The trust that users have in a recommender system can be increased using

explanations because it aids users to understand how the system works, it is usually related to transparency. Users intend to reuse trustworthy recommender systems, but it also relies on the accuracy of the recommendations.

2.2.1.5 Relevance

Explanations can be used to show the purpose of the additional information provided by users, and why is it important. This is true in the case of conversational recommender systems, where customers keep interacting with the system in order to refine their profiles.

2.2.1.6 Persuasiveness

Persuasiveness is the ability to convince users to accept the item. Explanations could improve users' acceptance of the recommendation and consecutively to the system. However, an overload of persuasion may have an inverse effect if the recommended item was considered unacceptable.

Persuasiveness can be calculated by the difference between a previous rating and a post rating using an explanation interface. Or, it can be measured as the difference between sales from a system with an explanation interface, and from a system without that facility.

2.2.1.7 Comprehensibility

Comprehensibility can be defined as the capacity of the explanations to relating user's knowledge to the knowledge used by the recommender systems. The systems can not naturally be sure about the knowledge of their users without the help of the other methods.

2.2.1.8 Effectiveness

Effectiveness means using explanations to help users make good decisions about what item to select. It is closely related to the accuracy, precision, and recall of the recommender algorithm. One of the advantages of effectiveness is it can introduce a whole bunch of items to new users, and they can understand the full range of available options.

Effectiveness can be calculated as the number of users who still like an item they have bought after consuming it. Another way to measure it is to count the number of users who, after receiving explanations, chosen items more fitted to their expectations, then the number of users who were not using systems with explanation facilities.

2.2.1.9 Efficiency

Efficiency means how quickly a task can be performed. Explanations can improve the users' spending time in the choice of the right item. Efficiency, in the evaluation of conversational recommender systems, can be measured as the total interaction time or the

number of interactions needed to find useful items. In other systems, it is measured as the number of inspected explanations until good items are found.

2.2.1.10 Satisfaction

Satisfaction occurs when explanations can make users more satisfied with the overall system. It can be measured either directly by asking users if they enjoyed the system or indirectly by measuring their frequency of use.

2.2.1.11 Education

Education occurs when explanations are used to help users better know the product domain by providing them with deep knowledge about that specific domain.

2.2.2 Explanation styles

The explanation styles represents the form of the explanation and how it will be displayed, for example, it could be: textual sentences about the content of the recommended item; list of related items; demographic information about the user and so on. In (TINTAREV; MASTHOFF, 2011), there is a list of the most commonly used explanation styles; they are showed in Figure 4 and described bellow.

The style is influenced by the underlying algorithm of the recommender engine, even though those explanations do not necessarily reflect the underlying algorithm

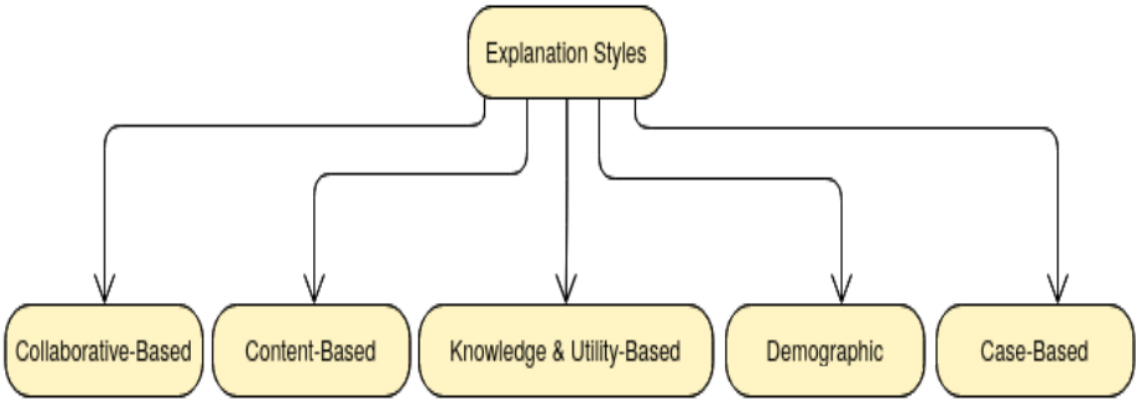


Figure 4 – Explanation styles.

2.2.2.1 Collaborative-based style explanations

In collaborative-based style explanations, the explanations are gathered using the user's ratings of an item to find users with similar preferences. This style is known as Neighbor Style Explanation (NSE) since similar users represent the neighborhood, and

the explanations are founded from neighbor's ratings. The most well-known usage of this explanation style is in the format of "Customers Who Bought This Item Also Bought...", like in Amazon.com.

2.2.2.2 Content-based style explanations

Content-based style the explanations are generated by levering the similarity between the items using their properties. For example: "Book B was recommended because you owned Book A with share the same genre and author". Other types of explanation from this style include Keyword Style Explanation (KSE) and Tag Style Explanation (TSE) with both properties: Tag preference, which is the user's sentiment toward a tag, and Tag relevance, which is the extent to which a tag describes an item.

2.2.2.3 Knowledge and utility-based style explanations

The input to the system, in this case, is the description of users' necessities and interests. Then the recommender algorithm recommends items based on these needs, and the explanations are the need levered. For example: "Beer was recommended because you said you are single and you have a high budget."

In some cases, there will be an overlap between three types of explanation styles: the knowledge-based style, the content-based style and the case-based style explanations.

2.2.2.4 Demographic style explanations

The input, to the recommender algorithm, is the user's demographic information. So, the algorithm identifies users that are demographically similar to the first one.

This type of explanation style is commonly used by recommender systems for commercial websites, due to the sensitivity of demographic information revealed. For example: "We recommend for you shaving cream because you are a male aged 18-32".

2.2.2.5 Case-based explanation

There is one of them that we are interested in, the case-based style of explanations. In this approach, the focus is to provide previous liked items as explanations. An example is: "Because you have selected or rated well A and B, we recommend C". For case-based explanation, there is no dataset with the exact information about which explanation is better than others. Hence, there are no metrics to evaluate the explanation methods as the ones that evaluate the recommendations. Also, few papers approach this problem, so offline evaluating case-based explanations is an ill-defined problem.

For a given recommendation, each previous liked item, by the user of the recommendation, has a score value as an explanation to the recommendation. So, the top-N scored previous liked items compose the explanation set associated with this recommendation.

Formalizing, let I be the set of items, U the set of users, $R \subset I$ the set of all recommendations, and $R_u \subset R$ the set of all recommended items provided to user u. Additionally, let $\widehat{I}_u \subset I$ refers to the set of items previously liked by user u, i.e., items in training data for user u. The case-based explanation for a recommendation $r \in R_u$ provided to user $u \in U$ is defined as $E_{u,r} \subset \widehat{I}_u$, it is a list of top-$N$ items sorted according to the scores assigned by a function $f(u,e,r) : \mathbb{R}^{|U|\times|I|\times|I|} \rightarrow \mathbb{R}$.

The score is assigned to all items previously liked by user u, and the items with highest scores are used as explanations assuming $|E_{u,r}| = N$. More formally, the case-based explanation is defined as:

$$E_{u,r} = \left\{ e_1,e_2,...,e_N \,\middle|\, \forall e \in \widehat{I}_u \wedge f(u,e_n,r) \geq f(u,e_{n+1},r) \right\}. \tag{3}$$

Further, different explanation methods instantiate different score functions $f(u,e,r)$, in Chapter 4 the score value calculated by several different explanation methods is explored and compared.

2.2.3 Evaluating explanation

The goals of explanations are to improve the following attributes of the recommender system: transparency, scrutability, trustworthiness, effectiveness, persuasiveness, efficiency, and satisfaction (TINTAREV; MASTHOFF, 2007a). Any increase in these attributes can be a metric to evaluate the quality of explanations (TINTAREV; MASTHOFF, 2007b), but these properties are subjective and identifying changes on them is difficult.

In evaluating recommendation performance, the information about which items each user liked is available in the test set, allowing the application of efficient metrics in this evaluation. On the other hand, to evaluate the explanation method performance, we do not always have this kind of information, what we can do are, both online and offline evaluations (ZHANG; CHEN, 2018).

Online evaluations require direct interaction with users, and offline evaluations require just previously extracted data. Online evaluation can handle more aspects of the explanations and produces more complete results, so it is encouraged, but tests involving users are not always accessible. Offline evaluation is preferred because it requires only previously extracted data, but the problem is, to evaluate the quality of case-based explanation methods, few research projects approach offline evaluation, so it is an ill-defined problem.

While online evaluations depending on real users are expensive and time consuming

(ZHANG et al., 2014). In addition, it is not always straightforward to map subjective concepts into objective metrics. For this reason, online evaluation approaches are highly encouraged (ZHANG; CHEN, 2018), as they can handle more aspects of the explanation evaluation and produce more complete results than objective offline evaluation metrics.

Concerning online evaluation approaches, (SYMEONIDIS; NANOPOULOS; MANOLOPOULOS, 2009) introduced *Movieexplain*, a prototype system where users can interact with the recommendations and explanations while their satisfaction in the given explanation style is evaluated. The system considers both case-based and content-based explanations. In another attempt, the authors in (CHEN et al., 2018; WANG et al., 2018) performed an analysis of real users' behavior in a controlled environment test.

The main drawback of online evaluations is that they may affect user experiences. Hence, offline evaluation is critical to assess the quality of explanations in recommender systems. However, offline evaluation is underexplored in the context of explanations for recommender systems, including case-based explanations (ZHANG; CHEN, 2018). There are two offline methods to evaluate case-based explanations: *Explainability* (ABDOLLAHI; NASRAOUI, 2017) and *Model FIdelity* (PEAKE; WANG, 2018).

Explainability (ABDOLLAHI; NASRAOUI, 2017) is a metric to evaluate recommendation explanations based on the probability of a recommended item be explainable for a user by using either user or item neighborhood. If this probability is greater than a threshold, the item is explainable for the user. Leveraging *Explainability* there are two other metrics: *Explainability precision* and *Explainability recall*. *Explainability precision* is the proportion of items in the recommendation set with probability to be explainable greater than the threshold relative to the number of recommended items for each user. *Explainability recall*, in turn, is the proportion of explainable items in recommended items relative to the number of all explainable items for a user.

The second metric, *Model Fidelity* (PEAKE; WANG, 2018), is a generic metric to evaluate explanation methods by measuring the percentage of explainable items among the recommended items, regardless of any quality aspect of the explanations.

There is not a study comparing these two metrics, but *Model Fidelity* is comparable to *Explainability precision*. The differences between them are:

- *Explainability precision* is not a generic metric, it works only with the probability of a recommended item be explainable for a user. But it has the threshold as a feature to ensure a certain level of explanation quality since the threshold can exclude bad explanations.
- *Model Fidelity* is a generic metric, it can be used with any explanation method. But it does not have a feature to ensure explanation quality since it does not have a threshold on itself. This metric needs a third party method to generate the explanations and then

just calculate the percentage of explainable items. Ensuring the explanation quality is the job of the third party explanation method.

The metric *ECT* has the advantages from both *Model Fidelity* and *Explainability*. *ECT* is a generic metric and has more complex features to measure explanation quality.

3 Proposed evaluation method

The *Explainability* and *Model Fidelity* offline metrics were previously proposed for case-based explanation, and focus on how many recommendations provided by the system can be explained, but they have limitations. *Explainability* is not a generic metric and *Model Fidelity* ignores the quality of the explanations. In order to fill this gap, this chapter introduces two new metrics, named *Ease of interpretation* and *Triviality*. These two metrics are then combined with model fidelity (PEAKE; WANG, 2018), which here was renamed to *Coverage* to better reflect its objective. The harmonic mean of these three metrics is *ECT*, which can simultaneously handle different aspects of case-based explanations.

3.1 Ease-of-interpretation

Ease-of-interpretation handles one aspect of persuasiveness but also efficiency in recommender systems. Interpretation is a complex and subjective concept, which varies from one user to another. Its complexity does not make it feasible to measure all of its aspects. However, according to the users' test results in (HERLOCKER; KONSTAN; RIEDL, 2000), too much information has negative effects on how users interpret explanations. Based on that, the only aspect we can measure about ease of interpretation is the amount of information provided to the user. In case-based, the provided explanations are previous liked items. As more items are provisioned in the explanation, more difficult it is to interpret it (harming the persuasiveness) and more time is necessary to interpret it (harming efficiency).

This behavior is captured by the following equation.

$$EI(x) = \frac{1}{1 + \exp(x - \eta - 1)}, \tag{4}$$

where x is the number of items provided in explanation ($|E_{u,r}|$), and η is a constant that represents the maximum number of items in an explanation users can easily interpret.

Users have different levels of interest and disponibility to interpret the recommendations, so η varies for each user. However, we need only one value of η for the whole evaluation, so we must generalize and find a universal value. The constant η should also vary for different recommendations domain. For example: In explanations of recommendations for a web store, or movie explanation recommendations, η could have a small value, because those users do not have enough time to ponder about the relation between the recommended item and several items presented in the explanation. On the other hand, for explanations in medical diagnosis recommender systems, η could have a high value, because users of this kind of systems are specialists and analyzing the relationship between recommendations and explanations is part of their jobs.

The η constant could also be interpreted as "how many items in the explanation, can my system present?". For example, in applications for web pages and cellphones, there is a limited space to display explanations. Then, we could set a limit on how many items are presented, and the value of η could be defined to penalize explanations that exceed this maximum capacity.

The rule is, as more items in the explanation, more the metric penalizes it. But it has light penalization if $x \leq \eta$ because the amount of items in the explanation doesn't exceed the threshold η of how many items users can easily interpret. And heavy penalization if $x > \eta$, because the threshold was trespassed. If $x \leq \eta$ then $EI(x)$ will follow a logarithmic decay, if $x > \eta$ $EI(x)$ will follow a exponential decay. Figure 5 shows that the sigmoid function shape in Equation 4 follow that rule, the first segment ($x \leq \eta$) looks like a logarithmic decay and the second segment ($x > \eta$) looks like an exponential decay. Figure 6 presents Equation 4 varying the value of x for several values of η.

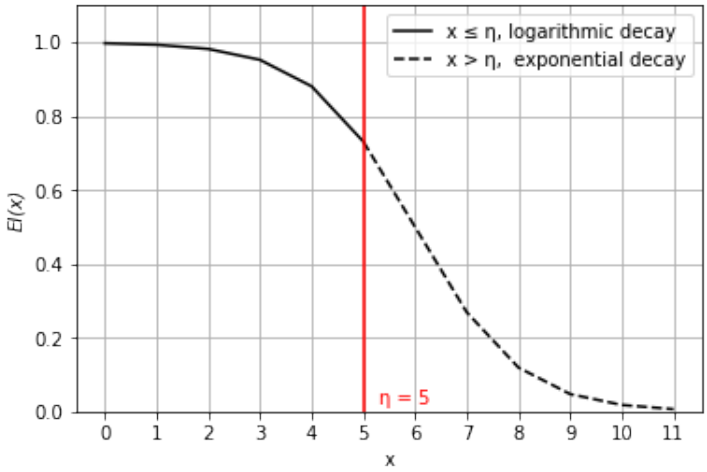


Figure 5 – Sigmoid function shape in Equation 4.

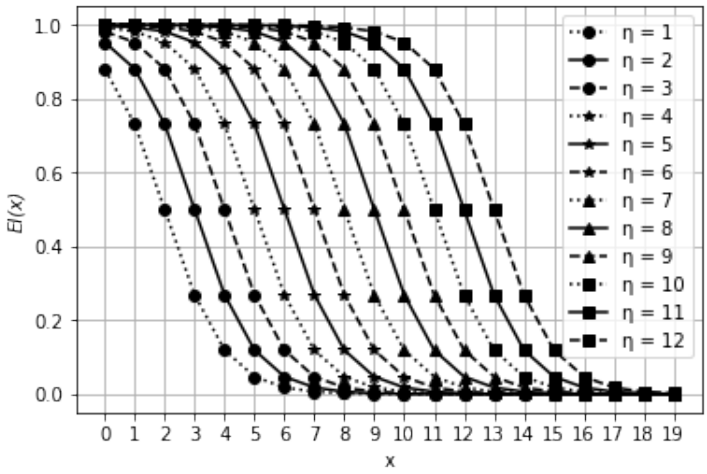


Figure 6 – $EI(x)$ varying the value of x for several values of η.

We emphasize that in Equation 4, if $x = 0$, $EI(x)$ will be higher than if $x = 1$, for each value of η. At first glance, it could be illogical that an explanation with no items has a higher score than an explanation with at least one. It happens because this metric measures how easy the explanation is to interpret, so an explanation with zero items demands zero effort to be interpreted. In this scenario, an explanation with zero items will score high in this sub-metric but will score nothing in the other two sub-metrics, measuring the score of all three sub-metrics the explanation with zero items will get a poor *ECT* value.

Before calculating $EI(x)$, the value of η must be defined, leveraging the recommendation domain, the user's profile, and the presentation constraints.

Equation 4 measures a single explanation, to calculate the *Ease-of-interpretation* (*EoI*) metric for the whole explanation method, we need to calculate $EI(x)$ for all users $u \in U$ and all recommendations made to that user R_u, as defined in Equation 5. $E_{u,r}$ is the case-based explanation for a recommendation $r \in R_u$ provided to user $u \in U$, as it was defined in Section 2.2.2.5.

$$Ease\_of\_interpretation(U, R, E) = \frac{1}{|U|} \sum_{u \in U} \left[\frac{1}{|R_u|} \sum_{r \in R_u} EI(|E_{u,r}|) \right], \tag{5}$$

$$\{Ease\_of\_interpretation(U,R,E) \in \mathbb{R} | 0 \leq Ease\_of\_interpretation(U,R,E) \leq 1\}$$

3.1.1 Example of Ease-of-interpretation

Figure 7 is a simple example to illustrate the functioning of *Ease-of-interpretation*. There are 3 different users, each of them liked different films. And they are all receiving the same item as a recommendation. Two explanations methods are generating explanations (A and B) for the recommendation.

In this case, method B has a higher *Ease-of-interpretation* value than method A, since Algorithm B provided fewer items among its explanations.

3.2 Coverage

This metric is the *Model Fidelity* proposed by (PEAKE; WANG, 2018), and was renamed to adapt to a broader context. In its original definition, *Model Fidelity* is a model-free metric, and it measures how many of the provided recommendations can be explainable. Hence, in this context this metric is the coverage of the explainable recommendations above all recommendations.

Coverage (*Cvr*) is a generic metric and works for any explanation method. It measures how many recommendations given to a user u can be explained, regardless of how good the explanation is.

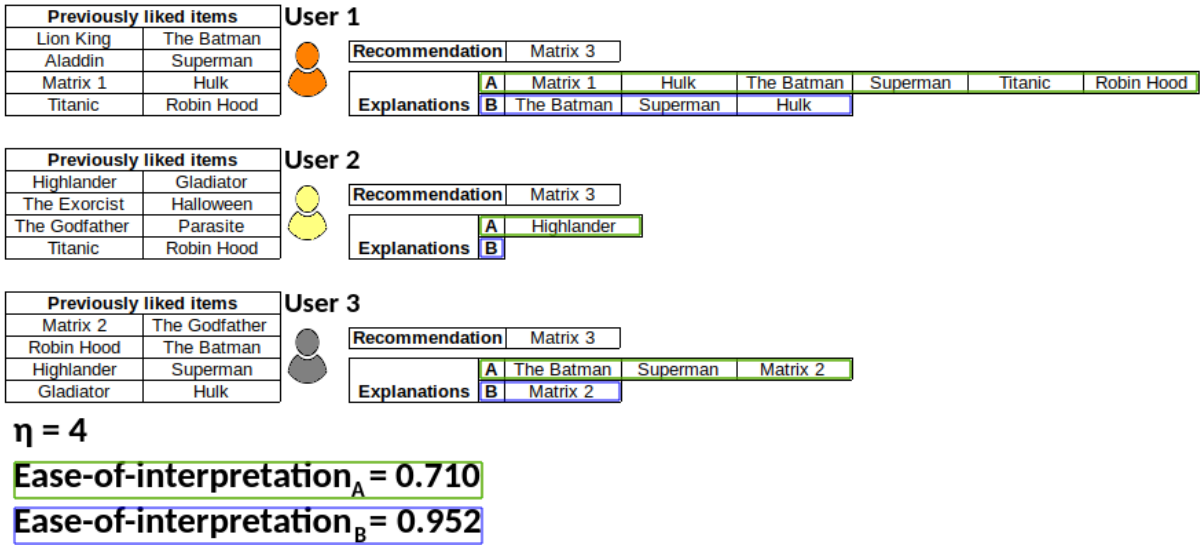


Figure 7 – Example of Ease-of-interpretation

Coverage is defined in Equation 6 where, for each user in $u \in U$, it calculates the proportion of recommendations that can be explained averaged over all users. $E_{u,r}$ is the case-based explanation for a recommendation $r \in R_u$ provided to user $u \in U$, as it was defined in Section 2.2.2.5.

$$Coverage(U, R, E) = \frac{1}{|U|} \sum_{u \in U} \left(\frac{1}{|R_u|} \sum_{r \in R_u} \begin{cases} 1, & \text{if } |E_{u,r}| > 0 \\ 0, & \text{otherwise} \end{cases} \right), \tag{6}$$

$$\{Coverage(U, R, E) \in \mathbb{R} | 0 \leq Coverage(U, R, E) \leq 1\}$$

3.2.1 Example of Coverage

Figure 8 is a simple example to illustrate the functioning of *Coverage*. There are 3 different users, each of them liked different films. And they are all receiving the same item as a recommendation. Two explanations methods are generating explanations (A and B) for the recommendation.

Method A received the highest score in *Coverage* because it generates an explanation for all recommendations. Method B, on the other hand, failed to generate an explanation to the recommendation of user 2, so it was penalized.

3.3 Triviality

Triviality is proposed to quantify a piece of trustworthiness in the recommender system. If the users perceive that there are good recommendations followed by obvious explanations (like a new sequel from an already seen movie franchise), they can get more confidence about the recommender system.

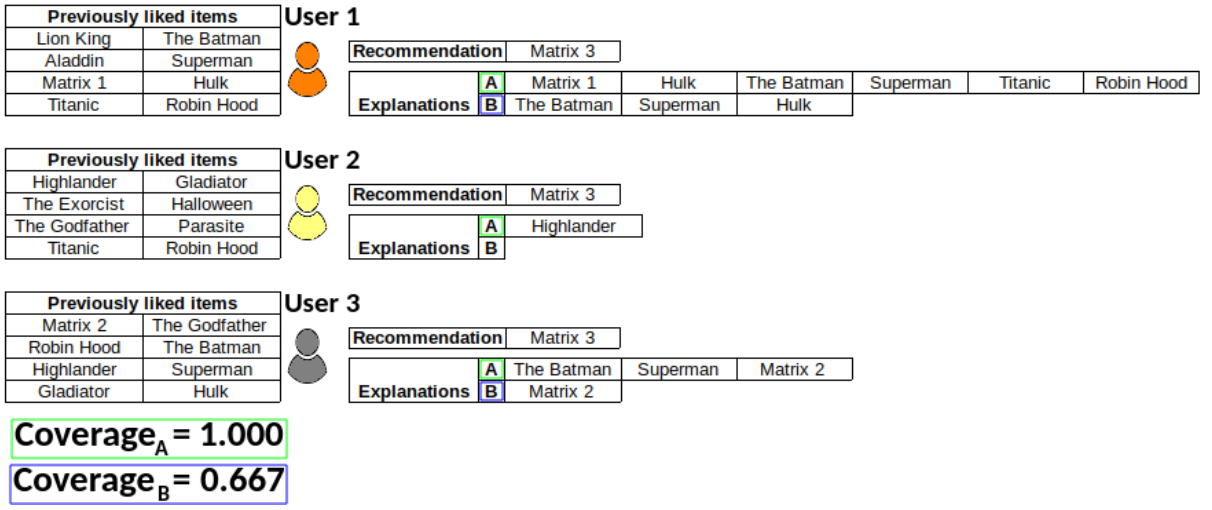


Figure 8 – Example of Coverage

In order to introduce triviality, we first define the concept of trivial set. A trivial set TE_i contains a set of trivial explanations for an item i, where $TE_i \subset I$. For example, in movie recommendation, we know that the movies Matrix 1 and Matrix 2 are trivial explanations for a Matrix 3 recommendation. Hence, the trivial set TE_i for item i = Matrix 3 is TE_i ={Matrix 1, Matrix 2}. We define $T \subset I$ as the set of all items i that have a trivial explanation set TE_i, where $|TE_i| > 0$.

Triviality (*Trv*) measures if the explanation method can provide items from the trivial set as explanations for the recommendation of items in T and is defined in Equation 7. $\ddot{U}_i$ represents the set of users who liked at least one item in TE_i and were recommended i. The idea of this metric is to penalize explanation methods that do not provide the trivial explanations to the trivial items.

An important observation about the set up of the trivial set, it must be hidden for the explanation method. Otherwise, *ECT* could be easily overfitted. For instance, an algorithm could adopt the following mechanism: it explains with trivial items when there exists (through memorization or simple heuristics) and explains with a random item when there is no trivial explanation. This kind of problem already exists in the machine learning world; if the training algorithm knows the test set, it could overfit the training process just to score high in any chosen metric. In any of those cases, it will be subject to Goodhart's Law: "When a measure becomes a target, it ceases to be a good measure." (STRATHERN, 1997)

Finally, notice that, for case-based explanation, there is no dataset with the exact information about which explanation is better than others. Hence, again, *Triviality* is an approximation metric that checks whether the explanation methods are providing correct explanations to the simplest cases.

$$Triviality(U,T,E) = \frac{1}{|T|}\sum_{t\in T}\left(\frac{1}{|\ddot{U}_t|}\sum_{u\in\ddot{U}_t}\begin{cases}1, & \text{if } |E_{u,t}\cap TE_t| > 0\\ 0, & \text{otherwise}\end{cases}\right), \tag{7}$$

$$\{Triviality(U,T,E) \in \mathbb{R} | 0 \leq Triviality(U,T,E) \leq 1\}$$

To get the trivial sets is the most challenging aspect of applying this metric. In Session 4.6, we describe the trivial sets that we got in two distinct recommendation domains.

3.3.1 Example of Triviality

Figure 9 is a simple example to illustrate the functioning of *Triviality*. There are 3 different users, each of them liked different films. And they are all receiving the same item as a recommendation. Two explanations methods are generating explanations (A and B) for the recommendation.

Triviality is evaluated for each item that has a trivial set. Here we are evaluating only the item Matrix 3, which has a trivial set. So the first step is to select all users who received Matrix 3 as a recommendation and who have previously liked Matrix 2 or Matrix 1. Although all users received Matrix 3 as a recommendation, only users 1 and 3 have already liked Matrix 1 or 2. Therefore, only these two users have been selected.

Method A reaches the maximum value in *Triviality* because in all cases it provided Matrix 1 or 2 in the explanation set. Method B was penalized because it did not provide Matrix 1 in the explanation of the recommendation for User 1.

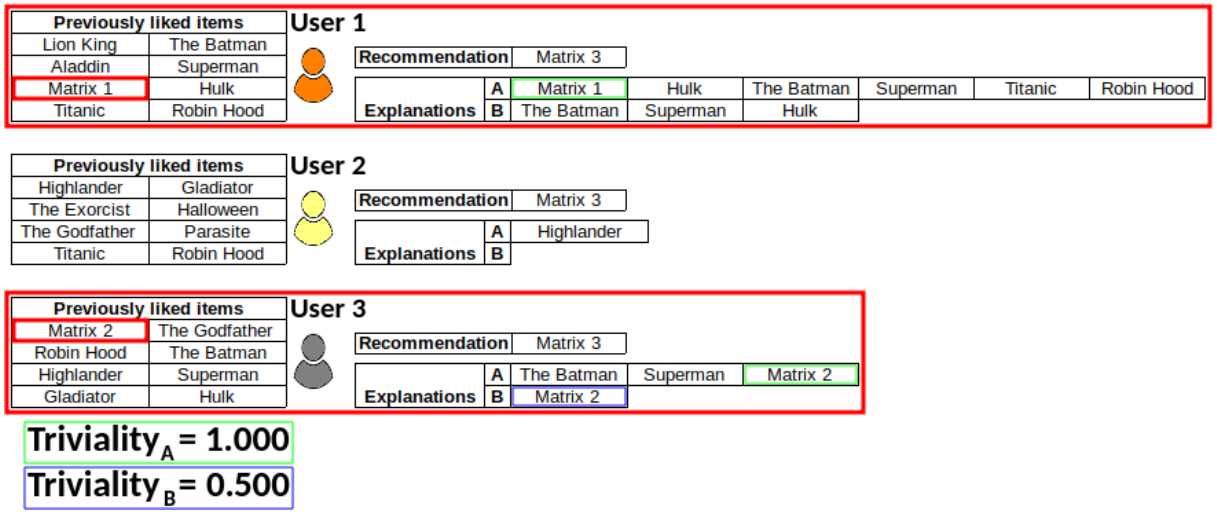
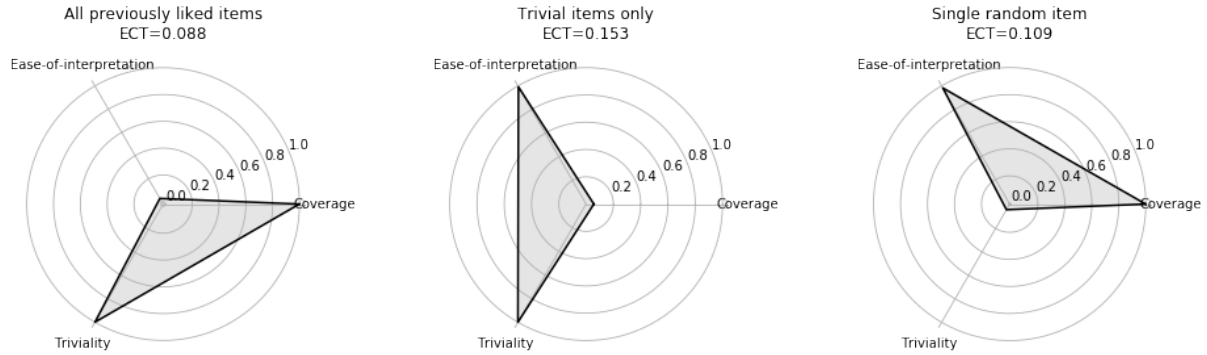


Figure 9 – Example of Triviality

3.4 ECT

ECT is the combination of the *Ease-of-interpretation*, *Coverage*, and *Triviality* metrics, and is being proposed to evaluate the quality of an explanation method, rather than the quality of a single explanation.



(a) All previously liked items are provided as explanations ($E_{u,r} = \widehat{I}_u$). The higher is $|E_{u,r}|$, the lower is *Ease-of-interpretation*.

(b) Only the items in the set of trivial explanations are provisioned as explanations ($E_{u,r} = TE_r$). *Coverage* decreases because just a few items have a set of trivial explanations.

(c) Only one item, chosen randomly, is provided as the explanation($|E_{u,r}| = 1$). The lower is $|E_{u,r}|$, the lower is the chance of a trivial explanation be provisioned. Hence the lower is *Triviality*.

Figure 10 – The three components of *ECT* are conflicting and require a level of balance between them. Each image represents a case of an extreme explanation method. In all of these cases, one sub-metric is completely injured.

Each one of the three presented sub-metrics can handle an aspect of evaluating case-based explanations. *Ease-of-interpretation* measures the effort require to interpret the explanations; *Coverage* measures how many recommendations can be explained; and *Triviality* measures if the explanation methods are solving, at least, the most simple cases.

The three components of *ECT* works as counterweights to each other, similar to the relation between *precision* and *recall* in evaluating recommendations (BERRY, 1955). If we aim to maximize just one sub-metric, the two others could be decreased, so a balance between *Ease-of-interpretation*, *Coverage*, and *Triviality* is desirable. Situations like that are illustrated in Figure 10, where each image represents an example of a weird explanation method that one sub-metric is exploited and the other two are neglected.

By these counterweights, the *ECT* metric can cover several aspects of evaluating the quality of case-based explanations and can provide a most complex evaluation.

A balance between *Ease-of-interpretation*, *Coverage*, and *Triviality* is desirable, so the value of *ECT* is the harmonic mean between *Ease-of-interpretation*, *Coverage*, and *Triviality*. Formally:

$$\begin{aligned} EoI &\leftarrow Ease\_of\_interpretation(U,R,E), \\ Cvr &\leftarrow Coverage(U,R,E), \\ Trv &\leftarrow Triviality(U,T,E), \\ ECT(U,R,E,T) &= \frac{3 \times EoI \times Cvr \times Trv}{EoI \times Cvr + EoI \times Trv + Cvr \times Trv}, \end{aligned} \tag{8}$$

$$\{ECT(U, R, E, T) \in \mathbb{R} | 0 \leq ECT(U, R, E, T) \leq 1\}$$

3.5 ECT relation with explanation attributes

According to (TINTAREV; MASTHOFF, 2007b), if there is an increase of transparency, scrutability, trustworthiness, effectiveness, persuasiveness, efficiency, or satisfaction caused by explanations in the recommender system, this increase can be a metric to evaluate the quality of explanations. However, some aspects of these concepts are hard to quantify. This work focus in easy-of-interpretation, coverage, and triviality concepts to propose *ECT*, which is a new offline metric to evaluate case-based explanation methods. This new metric is inspired by the concepts aforementioned and attempts to properly measure the quality of explanations in recommender systems. Specifically, each concept is translated in a metric, and *ECT* is a combination of these metrics, namely: *Triviality*, *Ease-of-interpretation* and *Coverage* [1].

Ease-of-interpretation works by penalizing large explanation sets, and handles one aspect of persuasiveness: too much information has negative effects on how users can interpret the explanations (HERLOCKER; KONSTAN; RIEDL, 2000), and too much persuasion may have reverse reactions if the item recommended by the system is found to be unacceptable (TINTAREV; MASTHOFF, 2011). It also covers efficiency, once if users can interpret the explanations quickly, they can make decisions early. *Coverage*, in turn, is used as proposed in (PEAKE; WANG, 2018) and measures how many recommendations can be explained. It measures the effectiveness of the recommender system since explanations could lead to more acceptability of the recommendation. Finally, *Triviality* attempts to quantify trustworthiness: if the users perceive that there are good recommendations followed by obvious explanations (like a new sequel from an already seen movie franchise), they raise their confidence on the recommender system. Note that a trivial explanation is related to how the user understands the explanation and it is a desirable property by explanation algorithms. On contrary, a trivial recommendation refers to an obvious list of recommended items, and needs to be avoided by recommendation algorithms.

To summarize, Table 1 shows the relation between the three metrics that compose *ECT* and the explanation attributes presented in Section 2.2.1. Each row of this table has the explanation attributes that inspired the metric in the columns.

[1] The same names for concepts and metrics were interchangeably used to avoid confusion.

Table 1 – Relation between the metrics and the explanation attributes

	Ease-of-interpretation	**Coverage**	**Triviality**
Transparency			
Validity			
Scrutability			
Trust			✔
Relevance			
Persuasiveness	✔		
Comprehensibility			
Effectiveness		✔	
Efficiency	✔		
Satisfaction			
Education			

4 Experiments

In order to evaluate the proposed metrics, an offline and an online evaluation were performed. Offline experiments were proposed to show how *ECT* works, using it to compare several case-based explanation methods, and these results were used further in online evaluation. Online experiments were proposed to show the correlation between *ECT* and real users' preferences about the quality of the explanations methods.

Basically, 3 datasets were used, which were given as input to 3 recommenders (see Section 4.2). Each of the selected recommenders returned the top-5 items for each user in the dataset. The number 5 was chosen to not overflow the users in the experiments with real users.

Next, 7 explanation methods were used to explain the results of the top-5 items, then *ECT* was used to compare the quality of these methods. Two types of experiments were used: fixed explanation set size and thresholding score.

The *ECT* has two parameters: the value of η for *Ease-of-interpretation* and the trivial sets. How to chosen those parameters and all other details about the experiments are presented in this section.

4.1 Datasets

To avoid any dataset bias, the data from two distinct domains of recommendation were used, MovieLens (movie recommendation) and Yelp (business recommendation). Both MovieLens and Yelp datasets contain only explicit feedback, a rating between 1 and 5. However, the recommender models, used in the experiments, work only for implicit feedbacks, so we converted the explicit to implicit data by thresholding the ratings such that, ratings greater than 3 are considered a view or a like. This preprocessing is in line with the standard procedures in the literature (PEAKE; WANG, 2018; Heckel et al., 2017; KOREN, 2009).

The statistics of the datasets are presented in Table 2.

Table 2 – Dataset Summary

Dataset	# users	# items	# ratings
ML 100K	943	1,682	100,000
ML 1M	6,040	3,706	1,000,209
Yelp Madison	32,662	3,494	104,637

4.1.1 MovieLens

The MovielLens dataset (HARPER; KONSTAN, 2015) is a common movie dataset, used in several recommender systems projects, which contains the users' ratings about the movies. The data are available on the website [1].

In this research, two versions of MovieLens dataset were used: 100K and 1M. The difference between them is the number of ratings, the label of the dataset is the number of ratings that it has. Increasing the number of ratings, consequently increases the number of users and items.

4.1.2 Yelp

Yelp is an enterprise that owns software to connect people with local businesses. Besides the connections, the company also provides a way to users rate the businesses. Those ratings are a subset of the Yelp dataset [2], which has a lot of other kinds of data. In this research, only the ratings from the city of Madison were used, because the amount of ratings in this city is equivalent to MovieLens 100k.

Yelp is a sparse dataset, in Table 2 shows that this dataset has the same number of rates as ML100K, but also has far more users. This sparsity had an impact on the experiments on public datasets in Section 4.7.

4.2 Recommender models

Recommender systems are based on the relationship between users and items, with no information about both, the only needed data are the interactions between them. There are two types of data to represent these interactions: explicit and implicit feedback. Explicit is a rating score. Implicit information is not trivial to define, such as a click, view, or purchase. Regardless of this disadvantage, our recommender models should be designed to work, mainly, with implicit feedback data, because they are more abundant than explicit data. More data usually means a better model. So implicit feedback is a way to go.

In the experiments, three implicit feedback based recommenders were used: Bayesian Personalized Ranking from Implicit Feedback (BPR) (RENDLE et al., 2009); Probabilistic Matrix Factorization (PMF) (SALAKHUTDINOV; MNIH, 2007) and Collaborative Metric Learning (CML) (HSIEH et al., 2017).

The data was split in these proportions for all datasets: $train = 72\%$; $validation = 8\%$; $test = 20\%$. Notice that a cross-validation procedure was not considered due to its posterior evaluation by real users. All methods were executed using OpenRec (YANG

[1] https://grouplens.org/datasets/movielens/
[2] https://www.yelp.com/dataset/

et al., 2018) with the following parameters: 50 latent factors for both users and items($dim\_user\_embed = 50$; $dim\_item\_embed = 50$); $num\_negatives = 500$; $num\_process = 5$; $pos\_ratio = 0.2$.

4.3 Explanation methods

The function $f(u, e, r)$ in Equation 3 defines the score value for a previous liked item as an explanation to the recommendation, and it is a generic function. As was mentioned in Chapter 2.2.2.5, explanation methods differ on the way they calculate this score value, and this section describes seven ways to instantiate it, and these instances were run among all datasets and all recommender models.

4.3.1 Apriori

The Apriori is a well-known algorithm (AGRAWAL; SRIKANT, 1994) to mine association rules between items over transactions. Each rule is composed of two different sets of items A and B, $\{A \rightarrow B\}$, where A,$B \subseteq I$. A is called antecedent or left-hand-side and B consequent or right-hand-side. There are three indicators to measure the association rule: confidence, lift, and support.

The associated rules mined by the Apriori algorithm was used to get case-based recommendations explanations in (PEAKE; WANG, 2018). In this approach, the transactions are the top-D scored items of each user. The set of items A represents the previous liked items, and B represents the recommendations. In each rule, these sets have only one element, so the rules are in the form of $explanation \rightarrow recommendation$.

In (PEAKE; WANG, 2018), Apriori was executed for all users, and the transactions are the top-D scored items for each one of the top-K similar users to a given user. So the transactions are a matrix with dimension $K \times D$. In our experiments, we used two values for K: $K = 10$ and $K = |U|$, in the second case, the transaction of all users are considered, and there are no neighbors.

The output rules were ranked using a combination of the three indicators usually considered to measure the quality of an association rule: confidence, lift, and support. A simple arithmetic mean of these three indicators was used to generate the function $f(u, e, r)$ for an association rule of the form $AR : e \rightarrow r$ (see Equation 9), which when ordered was used as an explanation method. In (PEAKE; WANG, 2018) the authors use these metrics separately and produce different explanation scores for each. As here our goal is not to propose a good explanation method but to learn how to evaluate them, so the three metrics were averaged. The average could be replaced by many different weighted combinations of

the three.

$$Apriori(K, u, e, r) = \frac{confidence_{u,K}(AR) + lift_{u,K}(AR) + support_{u,K}(AR)}{3}. \tag{9}$$

$$\begin{aligned} f(u, e, r) &= Apriori(K = 10, u, e, r), \\ f(u, e, r) &= Apriori(K = |U|, u, e, r). \end{aligned} \tag{10}$$

To run this method was used Apyori [3], a Python API which implements the Apriori algorithm, and it was used $D = 30$.

4.3.2 Distance metrics

All recommender model in Section 4.2 uses the latent factors approach (KOREN; BELL; VOLINSKY, 2009). It is a vectorized way to represent entities in abstract vector spaces. Due to this kind of representation, some mathematical operations can be done in real-world objects. In recommender systems, vectors represent users and items, and each dimension of these vectors is related to a concept. In item vectors, the dimensions represent how much that item has of these concepts. Thus, the distance between two item latent factor vectors can be used to calculate the similarity between the items. As close as the latent factors vectors get, more similar are the items. So, in this case, the similarity between two items can be defined as the inverse of the distance between their respective latent factors vector.

$$f(u, e, r) = \frac{1}{Distance\left(LI_e, LI_r\right))}. \tag{11}$$

Many different distance metrics can be used to generate the score. Four of them were used, namely the Euclidean, Manhattan, cosine and Chebyshev distances, defined in Equation 12 to Equation 15.

4.3.2.1 Euclidean distance

$$Distance(a, b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}. \tag{12}$$

4.3.2.2 Manhattan distance

$$Distance(a, b) = \sum_{i=1}^{n} |a_i - b_i|. \tag{13}$$

[3] https://pypi.org/project/apyori/

4.3.2.3 Cosine distance

$$Distance(a,b) = 1 - \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}}. \tag{14}$$

4.3.2.4 Chebyshev distance

$$Distance(a,b) = max(|a_i - b_i|). \tag{15}$$

4.3.3 Explainability

Abdollahi and Nasraoui (ABDOLLAHI; NASRAOUI, 2017) proposed a metric to evaluate recommendation explanations called *Explainability*. Although it evaluates explaniability it could be easily used to offer explanations to users. It uses either the user or item neighborhood to calculate the probability that a recommended item is explainable to a user. The probability $Pr_{user}(u,e)$ of a previous liked item e be used as an explanation for user u, is defined in Equation 16. S_u is the set of neighbors of a user u, and $\widehat{U}_e \subset U$ represents the set of users who liked item e.

$$Pr_{user}(u,e) = \frac{|S_u \cap \widehat{U}_e|}{|S_u|}. \tag{16}$$

This approach was extended to calculate, using the user's neighbors, the probability $Pr_{item}(e,r)$ of a previous liked item e be used as an explanation for the recommended item r.

$$Pr_{item}(e,r) = \frac{|\widehat{U}_r \cap \widehat{U}_e|}{|\widehat{U}_r|}. \tag{17}$$

In the experiments was used $|S_u| = 30$.

The mean of these two equations was used to be $f(u,e,r)$. So the metric to evaluate recommendation explanations in (ABDOLLAHI; NASRAOUI, 2017) was turned to an explanation method.

$$f(u,e,r) = \frac{Pr_{user}(u,e) + Pr_{item}(e,r)}{2}. \tag{18}$$

4.4 Experiment types

Two types of experiments was made, and the difference between them is in the size of the explanation set($|E_{u,r}|$) from Equation 3.

4.4.1 Fixing explanation set size

In this type of experiment, 4 explanations set $E_{u,r}$ were generated, fixing its size, using $|E_{u,r}| = [1,2,3,4]$. The maximum value is chosen to be 4 because of the value of η in Section 4.5.

4.4.2 Thresholding score

In this type of experiment, instead of fixing the value for $|E_{u,r}|$ a threshold was set for $f(u,e,r)$ and then $E_{ur} = \{\forall e \in E_{u,r} | f(u,e,r) > threshold\}$. The threshold was defined as the mean plus standard deviation from all values of $f(u,e,r)$. This approach has the advantage of automatically setting the size of the explanation.

4.5 Choosing the value of η for Ease-of-interpretation

As described in Section 3.1, choosing a value for the constant η is essential to calculate the *Ease-of-interpretation*. The constant η represents how many items provided as explanations the users can interpret. It is an uncertain concept and, as explained in Section 3.1, it varies by recommendation domain and for each user. This a complex problem and demands a complete study about it, all we can do is search for benchmarks.

Hingston reported some experiments with users (HINGSTON, 2006), in two of them, there are how many items were provided as explanations, 7 in one experiment, and 3 in another. In *LibraryThing* web site (LIBRARYTHING, (accessed October 19, 2020)), 4 items are provided as explanations in the book recommendation domain. So, to simplify, $\eta = 4$ were used for all experiments, because 4 is close to the mean of these values, and it is the number of items used in *LibraryThing*, a real application, but further experimentation with higher values are left to future work.

4.6 Trivial sets

As was mentioned in Section 3.3, to be able to use *Triviality*, it's needed to get some set of items that are previously known as trivial explanations for a specific item, and this is the most challenging aspect of implementing this metric. Two distinct domains of recommendations were used in the experiments, so a way to get trivial items among both scenarios was founded. Table 3 shows statistics of the datasets, including their number of trivial sets and their mean size.

Table 3 – Trivial sets dimensions.

Dataset	# items	# trivial sets	mean trivial sets size
ML 100K	1,682	109	1.91
ML 1M	3,706	2,691	3.26
Yelp Madison	3,494	933	6.18

4.6.1 Movie recommendation

In the movie domain, were considered movies from the same franchise as trivial explanations for all other movies from that franchise. For example, the trivial set for Star Wars 1 is a set containing all other Star Wars.

The Movie Database (TMDb) [4] is a popular, user-editable database for content information about movies and TV shows. Using TMDb is possible to get an aggregation of movies by franchises. The TMDb data is available to get via API, but there is a Kaggle page where a file containing the data for more than 45.000 movies can be downloaded. [5]

4.6.2 Business recommendation

In addition to the ratings, the Yelp dataset contains information about the business latitude, longitude, and categories. So, one establishment is in the trivial set of another one if they are closer than half a mile from each other and they have, at least, four categories in common.

4.7 Offline evaluation

To show how the proposed evaluation method works, it was used to compare several case-based explanation methods. To achieve this, the two types of experiments were run among three public datasets. In Algorithm 1, there is detailed information about each step followed in these experiments.

4.7.1 Fixing explanation set size

Figure 11 shows the result of *ECT* when fixing the explanation set size. Notice that, for all datasets, the value of *ECT* tends to increase as we make the size of the explanation set higher. For all datasets and explanation methods, there is a point where the values of the metric stabilize and from its values start to decrease.

According to *ECT*, the score calculated by the cosine distance was the best explanation method. For ML100K (Figure 11(a)) and ML1M (Figure 11(b)), the results obtained by

[4] https://www.themoviedb.org/
[5] https://www.kaggle.com/rounakbanik/the-movies-dataset

Algorithm 1 Offline evaluation steps

Generating recommendations and explanations
1: **for each** *dataset* in Section 4.1 **do**
2: **for each** *recommender model* in Section 4.2 **do**
3: train the *recommender model*
4: generate 5 recommendations for each user
5: **for each** *explanation method* in Section 4.3 **do**
6: **for each** *experiment type* in Section 4.4 **do**
7: generate $E_{u,r}$ for all recommendations, using the instance of function $f(u, e, r)$ from the *explanation method*, and the size of $E_{u,r}$ from the *experiment type*
8: calculate $ECT(U, R, E, T)$, using the value of η from Section 4.5 and the trivial sets from Section 4.6

Presenting the results
9: **for each** *experiment type* in Section 4.4 **do**
10: **for each** *dataset* in Section 4.1 **do**
11: **for each** *explanation method* in Section 4.3 **do**
12: show $mean(ECT(U, R, E, T))$ among all recommender models in Section 4.2

the Euclidean and Manhattan distances are very close to the ones obtained by the cosine. However, Chebyshev was significantly worse, probably because it is much simpler than the others.

For Yelp (Figure 11(c)), except for Apriori_K@|U|, all explanation methods got similar results. This dataset is sparser than MovieLens, and the sparsity could be a reason for this result. Regarding the Apriori explanation method, in the ML100K and ML1M Apriori_K@10 got a better result than Apriori_K@|U|, but in Yelp the opposite happened. The Apriori_K@|U| run among the transactions of all users instead of using a set of neighbors like Apriori_K@10. In the sparse Yelp dataset, limiting the number of neighbors increases the value of *ECT*.

Figure 12 shows the value of the three metrics that compose *ECT* when $|E_{u,r}|$ varies for the Cosine explanation method. Notice that in both versions of MovieLens *Coverage* is always 1. This happens because, for these datasets, all users have at least 20 ratings, so an explanation is always available. This does not occur in Yelp, which has a *Coverage* value of 0.613. The Yelp dataset is sparse and has several users in a cold start condition. However, notice that *Coverage* impacts *Ease-of-interpretation*, as many cases of recommendation with explanation size 0 may appear. For the Movielens datasets, as $|E_{u,r}|$ increases, *Ease-of-interpretation* decreases, as expected. *Triviality* increases as the size of the explanation increases.

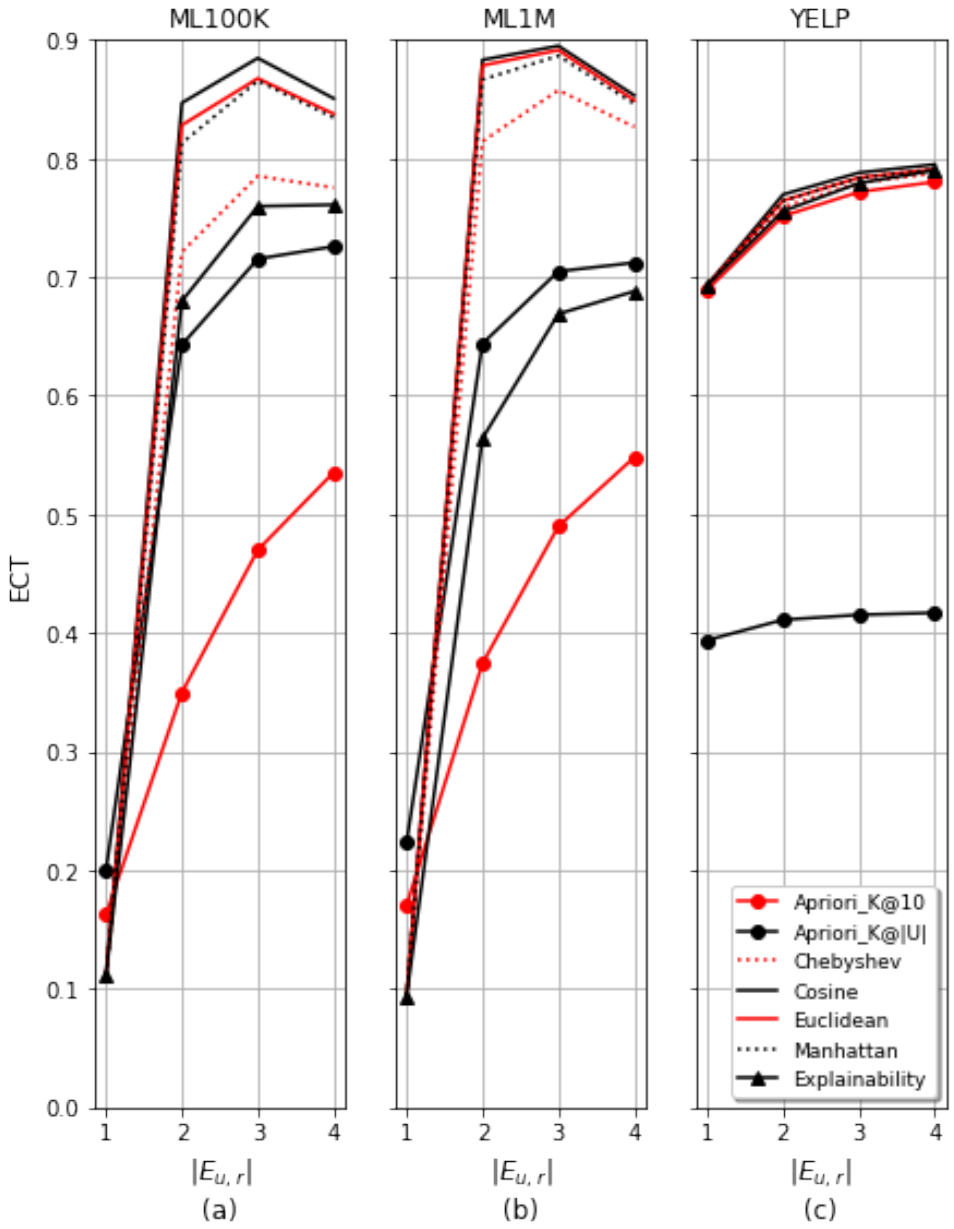


Figure 11 – Comparison of all evaluated explanations methods using *ECT* while fixing explanation set size.

4.7.2 Thresholding score

Figure 13 shows these same results when we replaced a fixed set size by a threshold applied to the function $f(u, e, r)$, and Table 4 shows the threshold and the mean size among all explanation sets. Notice that for the Movilens datasets the average explanation size is, in general, much higher than 4. For Yelp we have the opposite: the average number of explanation size is smaller than 1 (recall this is related to the sparsity of the dataset).

Different from the experiments with fixed explanation sizes, for Yelp (Figure 13(c)), as users have few ratings and a small set of previously liked items (mean of 2.14 and median of 1.00), the threshold leads to several recommendations without any explanation,

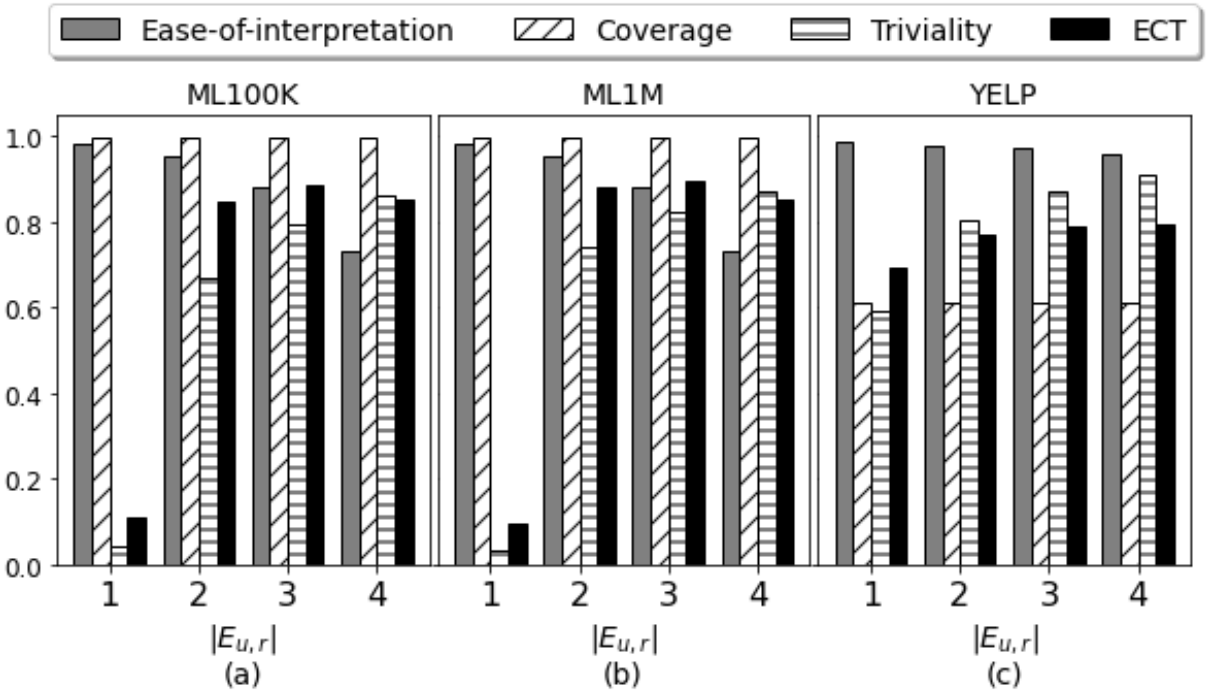


Figure 12 – Values of all metrics reached by Cosine in fixing explanation set size experiment.

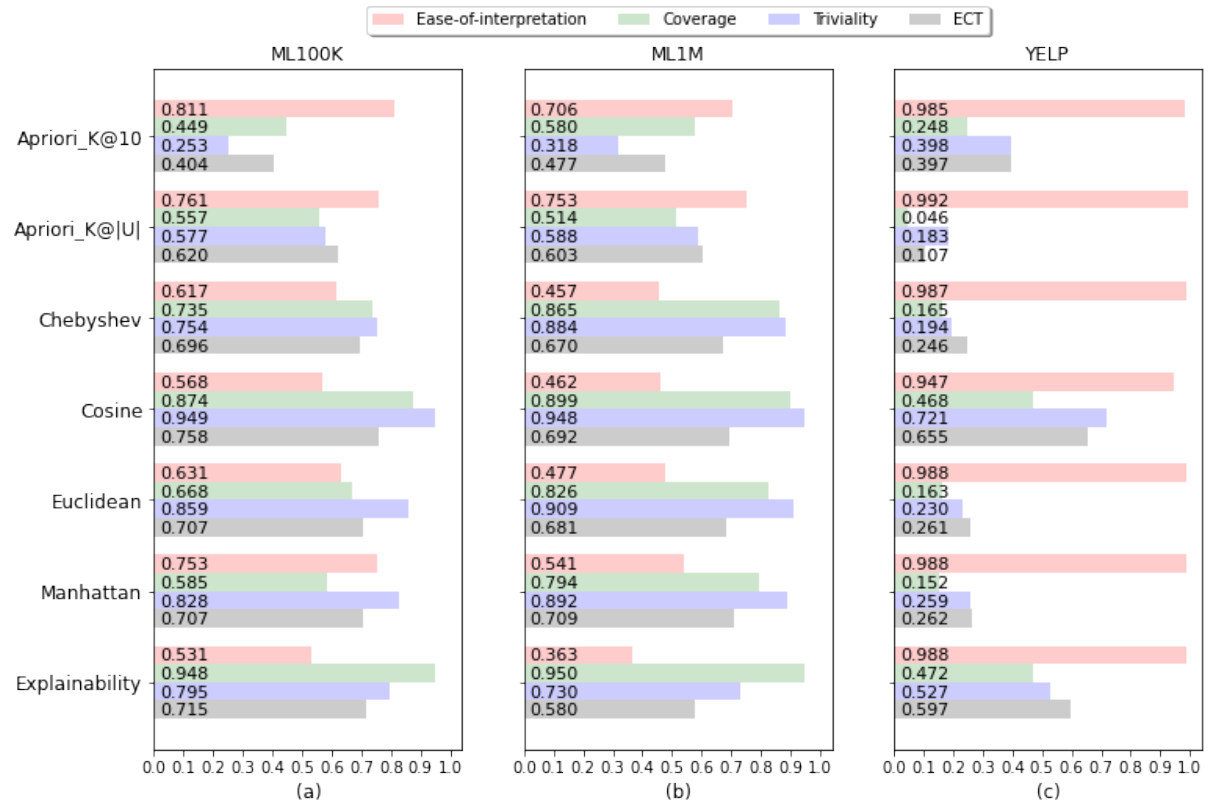


Figure 13 – Comparison of all evaluated explanations methods using *ECT*, in thresholding score experiment

generating extremely high values for *Ease-of-interpretation* for all explanation methods, but low value for *Triviality* and *Coverage* (consequently *ECT*) for Apriori_K@10, Apriori_K@|U|,

Table 4 – The threshold for $f(u, e, r)$ and the mean size among all explanation sets for each explanation method in thresholding score experiment

	ML100K		ML1M		YELP	
Explanation method	**threshold**	$\overline{\lvert E_{u,r} \rvert}$	**threshold**	$\overline{\lvert E_{u,r} \rvert}$	**threshold**	$\overline{\lvert E_{u,r} \rvert}$
Apriori_K@10	3.446	1.890	4.426	3.000	2.996	0.281
Apriori_K@|U|	1.626	2.979	0.766	3.549	1.189	0.063
Chebyshev	1.608	6.160	1.979	9.689	2.127	0.221
Cosine	1.441	6.047	1.713	8.693	1.032	1.092
Euclidean	0.521	6.196	0.651	9.766	0.683	0.214
Manhattan	0.103	4.487	0.119	8.481	0.123	0.163
Explainability	0.340	6.519	0.308	10.315	0.314	0.486

Chebyshev, Euclidean, and Manhattan. Only Cosine and Explainability reached average values for *Triviality* and *Coverage* in Yelp.

4.7.3 Comparing experiment types

Two types of experiments were done according to Section 4.4, and comparing the results generated when using a fixed explanation size (Figure 11) against those where a threshold was used (Figure 13), the latter has lower value of *ECT*. Table 5 illustrates why this behavior occurs using the cosine distance as the explanation method in the ML100K and Yelp datasets.

4.7.3.1 Fixed explanation set size

Table 5 (a) and (c) show images about the behavior of the fixed explanation set size experiment in ML100K and Yelp.

By fixing the size of $E_{u,r}$ with non zero values, *Coverage* is always 1 in ML100K, because all users have at least 20 ratings, so an explanation is always available. This does not occur in Yelp, which has a *Coverage* value of 0.613, because it is sparse and has several users in a cold start condition, so it is not guaranteed that at least one explanation will be available for every user.

Increasing $|E_{u,r}|$, *Ease-of-interpretation* decreases, as the more items are provisioned as explanations, the more work will be required to interpret them. On the other hand, *Triviality* increases, as the more items are provisioned as explanations, the more are the chances that a trivial explanation is among them.

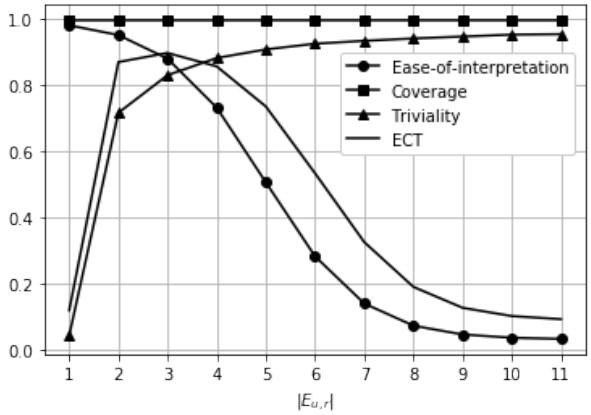
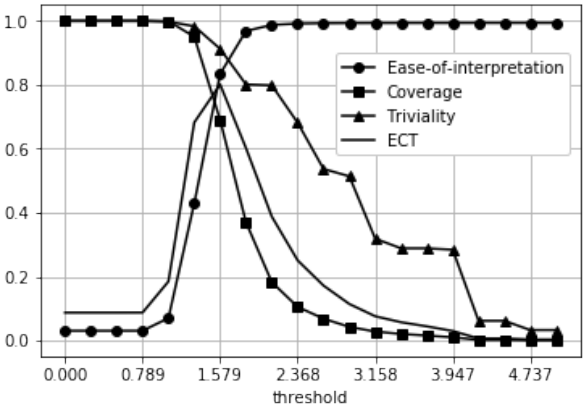
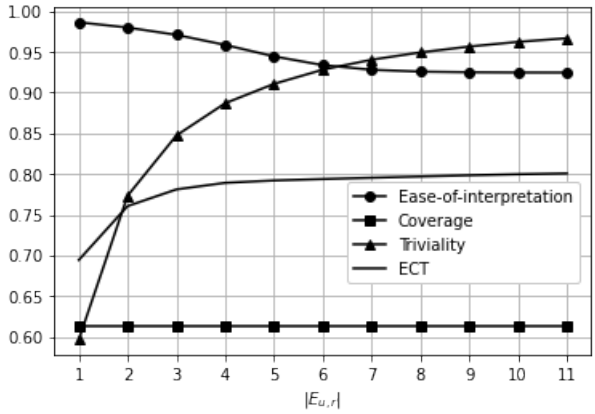
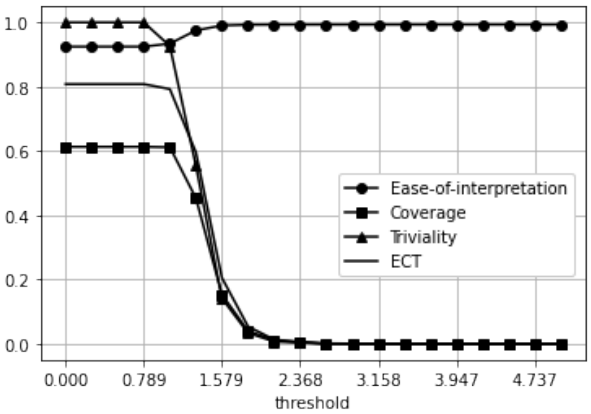
In Yelp, the decrease of *Ease-of-interpretation* has a non-zero limit, different to ML100K where it tends to zero. This happens because Yelp has a mean of 2.14 and a median of 1.00 ratings per user (recall that was used $\eta = 4$ for all experiments). So, the mean amount of ratings per user is lower than η, causing the non-zero bottom limit.

4.7.3.2 Thresholding score

Table 5 (b) and (d) show images about the behavior of the Thresholding score experiment in ML100K and Yelp. The higher the threshold, the fewer liked items e will have $f(u, e, r) > threshold$, and the value of $|E_{u,r}|$ will be lower, and vice versa.

Increasing the threshold, *Ease-of-interpretation* increases because $|E_{u,r}|$ decreases; *Coverage* decreases because the number recommendations which have at least one explanation reduces; *Triviality* decreases because the smaller $|E_{u,r}|$ the lower the chances of a trivial explanation be provisioned.

Table 5 – Comparing the experiment types and datasets, using Cosine as explanation method.

Fixed explanation set size	Thresholding score
ML100K	
 (a)	 (b)
Yelp Madison	
 (c)	 (d)

4.8 Online Evaluation

4.8.1 Experiment setup

Experiments were proposed to show the correlation between ECT and real users' preferences about the quality of the explanations methods. The same two experiments did in offline evaluations were done with real users: fixing explanation set size and thresholding score.

The first step is to create the participants' profiles in the MovieLens 1M dataset by asking them to mark some movies they liked on the item list, then generate 5 personalized recommendations for them. We varied the recommendation models among the participants; the used models were the same described in Section 4.2. Then, for each recommendation, 4 sets of case-based explanations were provided, each one generated by one of the following explanation methods: Apriori_K@10, Apriori_K@|U|, Cosine, and Explainability. The participants were asked to rank the explanations. If the participants haven't seen the recommended movie or one of the movies in the explanation set, they were instructed to read the synopses of the movie.

From all explanations methods in Section 4.3, were chosen to be used just four of them: Apriori_K@10, Apriori_K@|U|, Cosine, and Explainability. All distance metrics perform closely in the public dataset experiment, as we can see in Figure 11. So to not overload the participants with too many explanations to be evaluated, the only explanation method based on distance metrics used was Cosine.

In the experiments in offline evaluations, were used two different types of recommendation domains: movie and business. It had been decided that the experiment with real users were run only for movie recommendation because it is easier to find people who watch movies than people who attend certain business establishments.

The MovieLens 1M dataset was used because it has a good amount of items and ratings, as we can see in Table 2, and because this dataset has a particularity; the newest movies are from the year 2000. Our goal is to evaluate the explanations, not the recommendations, to achieve that, the participant must have minimum knowledge about those movies, so it is a great advantage the movies being old, because there is a high chance that the participant has already seen them. As mentioned, the participants were instructed to read the movie synopses if they haven't seen it before.

All kind of experiment involving real users is challenging to be done, but those two, in particular, are toughest, because they demand two interactions with the participants: the first one to get their previous liked items to build their profiles in the dataset, and the second to ask them to rank the explanations. For this cause, we were not able to broadcast them on the internet to reach a vast amount of people, so we had to post on personal social media asking

Table 6 – How the participants were distributed among the experiments and the recommendation models

Experiment	Recommender Model	# participants
Fixed size explanations	PMF	24
	BPR	22
	UCML	23
Thresholding score	BPR	15
	UCML	20

for participants, and people approach us willing to participate, and all communication was made via social media. The participation of each person was made individually, including the presentation of the project, clarification of doubts, and the running of the experiment.

A total of 104 participants took part of the experiment, being 47 men and 57 women between 19 and 47 years-old, with experience with the recommendation domain (movies). They had already used at least one movie recommender system, and had no experience with machine learning. This part of the experiment lasted two months, as it took a long time to deal with the 104 people individually.

Table 6 shows how the participants were distributed among the experiments and the recommendation models. Note that the PMF algorithm was not used for the experiment with thresholding score, as it demanded more than three days to run between the first and the second human interaction. We feared we would lose participant in this meantime, and for that reason only BRR and UCML were considered.

For every participant, was made 5 recommendations, and 4 explanations were generated for each recommendation. The participants must provide one rank of the most suitable explanations for each recommendation. So, there was a necessity to aggregate the participants' ranks, to compare them with the rank generated by ECT. A total of 5 different methods to aggregate the ranks were used. There is a short description of them in Section 4.8.2.

So, 5 aggregated ranks were compared with the rank generated by ECT, and for those comparations was used the Kendall tau-b correlation coefficient (KENDALL, 1962) (DANIEL, 1990), it is a statistic metric used to measure the similarity of the orderings between two ranks. The coefficient must be in the range [-1, 1], and if the two rankings are the same, the coefficient has value 1. If one ranking is the opposite of the other, the coefficient has value -1. If the two rankings are independent, then we would expect the coefficient to be approximately zero. The final result about the performance of the ECT metric is the mean of all 5 Kendall tau-b correlation coefficient.

In Algorithm 2, there is detailed information about each step followed in these experiments.

Algorithm 2 Online evaluation steps

Generating recommendations and explanations

Create the participants' profiles in the MovieLens 1M dataset by asking them to mark some movies they liked on the item list.
for each *recommender model* in Section 4.2 **do**
    train the *recommender model*
    for each *participant* acconrding to Table 6 **do**
        generate 5 personalized recommendations to the *participant*
        for each *recommendation* made for the the *participant* **do**
            for each *experiment type* in Section 4.4 **do**
                for each *explanation method* used in this experiment **do**
                    generate $E_{u,r}$ for the *recommendation*, using the instance of function $f(u, e, r)$ from the *explanation method*, and the size of $E_{u,r}$ from the *experiment type*
                ask the *participant* to rank each explanation group $E_{u,r}$ for the *recommendation*

Presenting the results

for each *experiment type* in Section 4.4 **do**
    for each *aggregation method* in Section 4.8.2 **do**
        using all ranks provided by the participants, compute the aggregated rank using the *aggregation method*
        calculate the Kendall tau-b correlation coefficient between the aggregated rank and the rank generated by ECT

4.8.2 Rank aggregation methods

4.8.2.1 Borda count

Borda count (EMERSON, 2013) is a system in which works assigning a point value to each position in the rank. The explanation methods are scored by summing how many points it has in each rank provided by the participants. The aggregated rank is given by ordering, decreasingly, the explanation methods by their scores.

4.8.2.2 Condorcet method

In the condorcet method (VALOGNES; GEHRLEIN, 2001), an explanation method won another competitor if it is ranked more often in a higher position than the competitor. The aggregated rank is given by ordering, decreasingly, the explanation methods by their numbers of wins.

4.8.2.3 Instant-runoff voting

Instant-runoff voting or Ranked Choice Voting (FARRELL, 2011), (FAIRVOTE. . . ,), is an election system, it simulates a series of runoff elections, and can be used to aggregate ranks. If no explanation method is the first choice of more than half of the participants, then all

votes cast for the explanation method with the lowest number of first choices are redistributed to the remaining, based on who is ranked next on each rank. When an explanation method receives a majority, it is removed from the system, and it is inserted in the aggregated rank. The process is repeated until the aggregated rank is fulfilled. If this does not result in any explanation method receiving a majority, further rounds of redistribution occur.

4.8.2.4 Schulze method

The Schulze method (SCHULZE, 2018) is an electoral system and can also be used to create a sorted list of winners; in other words, to aggregate ranks. For every pair of two explanation methods, they are compared, ignoring all others, generating a full set of pairwise preferences. Then, each explanation method is represented as a node in a directed graph, and then the strongest paths among each other have to be identified. The strength of the path between two nodes is the strength of its weakest link. An explanation method wins another if its strongest path is greater than its opponents' strongest path. The aggregated rank is given by ordering, decreasingly, the explanation methods by their numbers of wins.

4.8.2.5 Single transferable vote

Single transferable vote (ROWE, 1975) is a multiple winners voting system that can be used to aggregate ranks. Votes are totaled, counting only the first position in each ranked provided by the participants, and a quota (the minimum number of votes required to win) is derived. If the top voted explanation method achieves the quota, it is inserted in the aggregated rank. Any surplus vote is transferred to other explanation methods, in proportion to its occurrence in the next position in the ranks. The quota is recalculated. And this process continues until enough explanation methods exceed the quota to fill the aggregated rank.

4.8.3 Results

Table 7 shows, the ranking of the explanation methods according to the offline experiments for all metrics that compose *ECT* and *ECT* itself. And the ranks according to online evaluation of real users aggregated by the five different aggregated methods.

The ranks from the offline experiments were compared with the ranks generated by the online evaluation using the Kendall tau-b correlation coefficient (DANIEL, 1990). Kendall tau measures the similarity of the orderings between two ranks. The coefficient is in the range [-1, 1], and returns 1 if two rankings are identical or -1 if they are in reverse order. If the two rankings are independent, we expect its value to be approximately zero. The results are in Table 8.

ECT and all aggregated ranks generated from the users opinions agree that the overall winner among all explanations methods is the cosine distance when fixed-size

Table 7 – Comparison of the ranks generated by offline and online evaluation.

Fixed-size explanation experiment				
Position / Rank	**1st**	**2nd**	**3rd**	**4th**
Offline Evaluation				
Ease-of-interpretation	Apriori_K@|U|	Apriori_K@10	Cosine	Explainability
Coverage	Cosine	Explainability	Apriori_K@10	Apriori_K@|U|
Triviality	Cosine	Apriori_K@|U|	Explainability	Apriori_K@10
ECT	Cosine	Apriori_K@|U|	Explainability	Apriori_K@10
Online Evaluation				
Borda	Cosine	Explainability	Apriori_K@|U|	Apriori_K@10
IRV	Cosine	Explainability	Apriori_K@10	Apriori_K@|U|
Condorcet	Cosine	Apriori_K@|U|	Explainability	Apriori_K@10
STV	Cosine	Explainability	Apriori_K@10	Apriori_K@|U|
Schulze	Cosine	Apriori_K@|U|	Explainability	Apriori_K@10
Thresholding score experiment				
Position / Rank	**1st**	**2nd**	**3rd**	**4th**
Offline Evaluation				
Ease-of-interpretation	Apriori_K@|U|	Apriori_K@10	Cosine	Explainability
Coverage	Explainability	Cosine	Apriori_K@10	Apriori_K@|U|
Triviality	Cosine	Explainability	Apriori_K@|U|	Apriori_K@10
ECT	Cosine	Apriori_K@|U|	Explainability	Apriori_K@10
Online Evaluation				
Borda	Cosine	Apriori_K@|U|	Apriori_K@10	Explainability
IRV	Cosine	Apriori_K@|U|	Explainability	Apriori_K@10
Condorcet	Apriori_K@|U|	Cosine	Apriori_K@10	Explainability
STV	Cosine	Apriori_K@|U|	Explainability	Apriori_K@10
Schulze	Apriori_K@|U|	Cosine	Apriori_K@10	Explainability

explanation is used. For the thresholding score experiment, the cosine distance also got the first position according to *ECT* and it has, undoubtedly, the best performance among the aggregated ranks. In these two experiments, the rank generated by *ECT* is the same and the mean of the Kendall coefficient is 0.67. The major differences are on the aggregated ranks, where they have pairs of identical ranks: IRV / STV and Condorcet / Schulze; it is caused by some inherent properties of the rank aggregation methods.

Recall that the *Explainability metric* (ABDOLLAHI; NASRAOUI, 2017) was previously proposed in the literature with the same purpose of *ECT*. It is divided into two parts: *Explainability precision* and *Explainability recall*, as discussed earlier. *Model fidelity* (PEAKE; WANG, 2018) (or, in our case, *Coverage*) is a generalization of *Explainability precision*. Hence, its correlation to the users' online evaluation can also be comparable, which is on average 0.53 for fixed-size explanation and 0.33 for the thresholding score scenario, while *ECT* got 0.67.

Table 8 – Kendall tau-b correlation coefficient between the ranks generated by offline and online evaluation.

Kendall tau-b correlation coefficient				
Fixed-size explanation experiment				
	Ease-of-interpretation	**Coverage**	**Triviality**	**ECT**
Borda	0.33	0.00	0.67	0.67
IRV	0.00	1.00	0.33	0.33
Condorcet	0.00	0.33	1.00	1.00
STV	0.00	1.00	0.33	0.33
Schulze	0.00	0.33	1.00	1.00
Mean	**0.07**	**0.53**	**0.67**	**0.67**
Thresholding score experiment				
	Ease-of-interpretation	**Coverage**	**Triviality**	**ECT**
Borda	-0.33	0.33	0.33	0.67
IRV	0.00	0.00	0.67	1.00
Condorcet	-0.67	0.67	0.00	0.33
STV	0.00	0.00	0.67	1.00
Schulze	-0.67	0.67	0.00	0.33
Mean	**-0.33**	**0.33**	**0.33**	**0.67**

5 Future works

An important characteristic to be considered in recommender systems is recency; it is the preference by the users' recent ratings to make predictions more time accurate. In some recommendation domains, the recent ratings reflect users' future preferences more than older ones, for example, news recommendation, where items typically have short shelf lives. After all, few sports fans will be interested in the past breaking scores of one week ago.

Some works incorporate recency in recommender systems (DING; LI; ORLOWSKA, 2006), (CAMPOS; RUBIO; CANTADOR, 2014). So, it could be helpful if, somehow, *ECT* could leverage recency to evaluate a case-based explanation. But the toughest challenge in this proposal is to elaborate experiments with real users to validate it. To conduct this kind of experiment is necessary previous access to the participants' historical of rated or liked items along time, but it is nearly impossible to ask them, for example, to remember exactly when they saw all movies that they liked. The only viable way to get this information is by an already running system, which has already had the historical data and can interact with users via A/B testing. A system like that isn't available yet for us.

Also, *ECT* has two dependencies, the choosing for the value of η used in *Ease-of-interpretation*, and the trivial sets used in *Triviality*. So, other future works are: to research for methods to get the value of η in different contexts; to develop methods to define trivial sets in other recommendation domains; and to explore new ways to combine the three metrics that compose *ECT*, giving them different weights according to the context they are inserted.

6 Conclusion

In this project, we propose a new metric to evaluate the quality of explanations in recommender systems. This metric is a combination of a previous proposed metric (i.e., Coverage), and two new metrics (i.e., Ease-to-interpretation and Triviality). We conduct extensive evaluation to address our goal, i.e., we conduct online and offline experiments with real recommender systems and datasets to validate our proposed metric.

There are few offline evaluation approaches to measure the quality of case-based explanation methods, and they mainly focus on how many recommendations given by a system can be explained, leaving the explanation quality aside. In this direction, we proposed *ECT*, an offline evaluation metric that accounts for both the quality and the coverage of the explanation method. *ECT* is composed of three metrics, two of them also proposed in this research project: *Easy-of-interpretation* and *Triviality*.

In Section 4.7 shows some experiments to exemplify how *ECT* can be used, involving 3 public datasets, 3 recommendation methods and 7 explanation methods. In these experiment, according to ECT, the inverse of the cosine distance is the best explanation method among all evaluated methods described in Section 4.3. This result is confirmed by the experiments with real users in Section 4.8, the majority of the participants also ranked the inverse of the cosine distance as the best option. This agreement between *ECT* and real users' choice is one positive qualification for this proposed metric.

One more substantial proof of the effectiveness of *ECT*, is the correlation between it and the real users' choices about the best explanation method. The experiments with real users in Section 4.8 results in a correlation of 0.667, in the range [-1, 1], measured by Kendall tau-b correlation coefficient. *ECT* is an offline metric, which means, easy to implement, and cheap to calculate. So, considering the tradeoff between calculating cost and total reproduction of users' real choices, a correlation like that is desirable. We consider *ECT* one step forward better offline evaluation methods.

Bibliography

ABDOLLAHI, B.; NASRAOUI, O. Using explainability for constrained matrix factorization. In: **Proceedings of the Eleventh ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2017. (RecSys '17), p. 79–83. ISBN 978-1-4503-4652-8. Disponível em: <http://doi.acm.org/10.1145/3109859.3109913>. Citado 4 vezes nas páginas 3, 14, 29 e 41.

AGRAWAL, R.; SRIKANT, R. Fast algorithms for mining association rules in large databases. In: **Proceedings of the 20th International Conference on Very Large Data Bases**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994. (VLDB '94), p. 487–499. ISBN 1-55860-153-8. Disponível em: <http://dl.acm.org/citation.cfm?id=645920.672836>. Citado na página 27.

AMAZON. (accessed October 19, 2020). Disponível em: <https://www.amazon.com>. Citado 2 vezes nas páginas 1 e 5.

BERRY, J. W. P. A. K. M. M. Machine literature searching x. machine language; factors underlying its design and development. **American Documentation**, 1955. Disponível em: <https://doi.org/10.1002/asi.5090060411>. Citado na página 22.

BILGIC, M.; MOONEY, R. J. Explaining recommendations: Satisfaction vs. promotion. In: **Proceedings of Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research at the 2005 International Conference on Intelligent User Interfaces**. San Diego, CA: [s.n.], 2005. Disponível em: <http://www.cs.utexas.edu/users/ai-lab/?bilgic:iui-bp05>. Citado 2 vezes nas páginas 2 e 8.

CAMPOS, P.; RUBIO, F.; CANTADOR, I. Time-aware recommender systems: A comprehensive survey and analysis of existing evaluation protocols. **User Modeling and User-Adapted Interaction**, v. 24, 02 2014. Citado na página 43.

CHEN, X. et al. Visually explainable recommendation. **CoRR**, abs/1801.10288, 2018. Disponível em: <http://arxiv.org/abs/1801.10288>. Citado 2 vezes nas páginas 2 e 14.

DANIEL, W. **Applied nonparametric statistics**. PWS-Kent Publ., 1990. (The Duxbury advanced series in statistics and decision sciences). ISBN 9780534919764. Disponível em: <https://books.google.com.br/books?id=0hPvAAAAMAAJ>. Citado 2 vezes nas páginas 38 e 40.

DING, Y.; LI, X.; ORLOWSKA, M. E. Recency-based collaborative filtering. In: **ADC**. [S.l.: s.n.], 2006. Citado na página 43.

EMERSON, P. The original borda count and partial voting. **Social Choice and Welfare**, v. 40, 02 2013. Citado na página 39.

FAIRVOTE. Ranked Choice Voting / Instant Runoff. <https://www.fairvote.org/rcv>. Accessed April 12, 2020. Citado na página 39.

FARRELL, D. M. Book. **Electoral systems : a comparative introduction / David M. Farrell**. 2nd ed.. ed. [S.l.]: Palgrave Macmillan Houndmills, Basingstoke, Hampshire, UK ; New York, 2011. xiii, 279 p. : p. ISBN 9781403912312 1403912319 9780230546783 0230546781. Citado na página 39.

HARPER, F. M.; KONSTAN, J. A. The movielens datasets: History and context. **ACM Trans. Interact. Intell. Syst.**, ACM, New York, NY, USA, v. 5, n. 4, p. 19:1–19:19, dez. 2015. ISSN 2160-6455. Disponível em: <http://doi.acm.org/10.1145/2827872>. Citado na página 26.

Heckel, R. et al. Scalable and interpretable product recommendations via overlapping co-clustering. In: **2017 IEEE 33rd International Conference on Data Engineering (ICDE)**. [S.l.: s.n.], 2017. p. 1033–1044. ISSN 2375-026X. Citado na página 25.

HERLOCKER, J. L.; KONSTAN, J. A.; RIEDL, J. Explaining collaborative filtering recommendations. In: **Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work**. New York, NY, USA: ACM, 2000. (CSCW '00), p. 241–250. ISBN 1-58113-222-0. Disponível em: <http://doi.acm.org/10.1145/358916.358995>. Citado 5 vezes nas páginas 2, 4, 8, 16 e 23.

HINGSTON, M. **User Friendly Recommender Systems**. Tese (Doutorado) — The University of Sydney, 2006. Honours Thesis. Citado na página 30.

HSIEH, C.-K. et al. Collaborative metric learning. In: **Proceedings of the 26th International Conference on World Wide Web**. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2017. (WWW '17), p. 193–201. ISBN 978-1-4503-4913-0. Disponível em: <https://doi.org/10.1145/3038912.3052639>. Citado na página 26.

JANNACH, D. et al. **Recommender Systems: An Introduction**. 1st. ed. New York, NY, USA: Cambridge University Press, 2010. ISBN 0521493366, 9780521493369. Citado 2 vezes nas páginas 1 e 5.

KENDALL, M. **Rank correlation methods**. Hafner Pub. Co., 1962. (Theory and applications of rank order-statistics). Disponível em: <https://books.google.com.br/books?id=1whKAAAAMAAJ>. Citado na página 38.

KHASMAKHI, N. N.; BALAFAR, M.; DERAKHSHI, M. R. F. The state-of-the-art in expert recommendation systems. **Engineering Applications of Artificial Intelligence**, v. 82, p. 126–147, 06 2019. Citado 2 vezes nas páginas 5 e 8.

KOREN, Y. Collaborative filtering with temporal dynamics. In: **Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2009. (KDD '09), p. 447–456. ISBN 978-1-60558-495-9. Disponível em: <http://doi.acm.org/10.1145/1557019.1557072>. Citado na página 25.

KOREN, Y.; BELL, R.; VOLINSKY, C. Matrix factorization techniques for recommender systems. **Computer**, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 42, n. 8, p. 30–37, ago. 2009. ISSN 0018-9162. Disponível em: <http://dx.doi.org/10.1109/MC.2009.263>. Citado 4 vezes nas páginas 1, 5, 6 e 28.

LIBRARYTHING. (accessed October 19, 2020). Disponível em: <https://www.librarything.com/>. Citado 2 vezes nas páginas 5 e 30.

LIU, H. et al. In2rec: Influence-based interpretable recommendation. In: . [S.l.: s.n.], 2019. p. 1803–1812. ISBN 978-1-4503-6976-3. Citado na página 7.

MOLNAR, C. **Interpretable Machine Learning**: A guide for making black box models explainable. [S.l.: s.n.], 2019. <https://christophm.github.io/interpretable-ml-book/>. Citado na página 7.

NETFLIX. (accessed October 19, 2020). Disponível em: <https://www.netflix.com>. Citado na página 5.

PEAKE, G.; WANG, J. Explanation mining: Post hoc interpretability of latent factor models for recommendation systems. In: **Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining**. New York, NY, USA: ACM, 2018. (KDD '18), p. 2060–2069. ISBN 978-1-4503-5552-0. Disponível em: <http://doi.acm.org/10.1145/3219819.3220072>. Citado 10 vezes nas páginas 3, 4, 8, 14, 16, 18, 23, 25, 27 e 41.

RENDLE, S. et al. Bpr: Bayesian personalized ranking from implicit feedback. In: **Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence**. Arlington, Virginia, United States: AUAI Press, 2009. (UAI '09), p. 452–461. ISBN 978-0-9749039-5-8. Disponível em: <http://dl.acm.org/citation.cfm?id=1795114.1795167>. Citado na página 26.

RESNICK, P. et al. Grouplens: An open architecture for collaborative filtering of netnews. In: . [S.l.]: ACM Press, 1994. p. 175–186. Citado na página 1.

RESNICK, P.; VARIAN, H. R. Recommender systems. **Commun. ACM**, ACM, New York, NY, USA, v. 40, n. 3, p. 56–58, mar. 1997. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/245108.245121>. Citado 2 vezes nas páginas 1 e 5.

RICCI, F. et al. **Recommender Systems Handbook**. 1st. ed. Berlin, Heidelberg: Springer-Verlag, 2010. ISBN 0387858199, 9780387858197. Citado 2 vezes nas páginas 1 e 5.

ROWE, E. How democracies vote: A study of electoral systems. **Representation**, Routledge, v. 15, n. 58, p. 11–12, 1975. Disponível em: <https://doi.org/10.1080/03115517508619380>. Citado na página 40.

SALAKHUTDINOV, R.; MNIH, A. Probabilistic matrix factorization. In: **Proceedings of the 20th International Conference on Neural Information Processing Systems**. USA: Curran Associates Inc., 2007. (NIPS'07), p. 1257–1264. ISBN 978-1-60560-352-0. Disponível em: <http://dl.acm.org/citation.cfm?id=2981562.2981720>. Citado na página 26.

SCHULZE, M. The schulze method of voting. **CoRR**, abs/1804.02973, 2018. Disponível em: <http://arxiv.org/abs/1804.02973>. Citado na página 40.

SINHA, R.; SWEARINGEN, K. The role of transparency in recommender systems. In: **CHI '02 Extended Abstracts on Human Factors in Computing Systems**. New York, NY, USA: ACM, 2002. (CHI EA '02), p. 830–831. ISBN 1-58113-454-1. Disponível em: <http://doi.acm.org/10.1145/506443.506619>. Citado 2 vezes nas páginas 2 e 8.

SPOTIFY. (accessed October 19, 2020). Disponível em: <https://www.spotify.com>. Citado 2 vezes nas páginas 1 e 5.

STRATHERN, M. 'improving ratings': audit in the british university system. In: . [S.l.: s.n.], 1997. Citado na página 20.

SYMEONIDIS, P.; NANOPOULOS, A.; MANOLOPOULOS, Y. Moviexplain: A recommender system with explanations. In: **Proceedings of the Third ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2009. (RecSys '09), p. 317–320. ISBN 978-1-60558-435-5. Disponível em: <http://doi.acm.org/10.1145/1639714.1639777>. Citado 2 vezes nas páginas 2 e 14.

TAIE, M. Explanations in recommender systems overview and research approaches. **International Arab Journal of Information Technology**, 09 2013. Citado na página 8.

THE New York Times. (accessed October 19, 2020). Disponível em: <https://www.nytimes.com/>. Citado na página 1.

TINTAREV, N.; MASTHOFF, J. Effective explanations of recommendations: User-centered design. In: **Proceedings of the 2007 ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2007. (RecSys '07), p. 153–156. ISBN 978-1-59593-730–8. Disponível em: <http://doi.acm.org/10.1145/1297231.1297259>. Citado 2 vezes nas páginas 2 e 13.

TINTAREV, N.; MASTHOFF, J. A survey of explanations in recommender systems. In: UCHYIGIT, G. (Ed.). **Data Engineering Workshop**. [S.l.]: IEEE Computer Society, 2007. p. 801–810. ISBN 9781424408320. Citado 4 vezes nas páginas 2, 3, 13 e 23.

TINTAREV, N.; MASTHOFF, J. Designing and evaluating explanations for recommender systems. In: ____. **Recommender Systems Handbook**. Boston, MA: Springer US, 2011. p. 479–510. ISBN 978-0-387-85820-3. Disponível em: <https://doi.org/10.1007/978-0-387-85820-3_15>. Citado 3 vezes nas páginas 4, 11 e 23.

TINTAREV, N.; MASTHOFF, J. Explaining recommendations: Design and evaluation. In: ____. **Recommender Systems Handbook**. 2. ed. [S.l.]: Springer US, 2015. p. 353–382. ISBN 978-1-4899-7636-9. Citado na página 2.

VALOGNES, F.; GEHRLEIN, W. Condorcet efficiency: A preference for indifference. **Social Choice and Welfare**, v. 18, p. 193–205, 02 2001. Citado na página 39.

WANG, N. et al. Explainable recommendation via multi-task learning in opinionated text data. **CoRR**, abs/1806.03568, 2018. Disponível em: <http://arxiv.org/abs/1806.03568>. Citado 2 vezes nas páginas 2 e 14.

YANG, L. et al. Openrec: A modular framework for extensible and adaptable recommendation algorithms. In: **Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining**. New York, NY, USA: ACM, 2018. (WSDM '18), p. 664–672. ISBN 978-1-4503-5581-0. Disponível em: <http://doi.acm.org/10.1145/3159652.3159681>. Citado na página 27.

ZHANG, S. et al. **Deep Learning based Recommender System: A Survey and New Perspectives**. 2017. Citado na página 5.

ZHANG, Y.; CHEN, X. Explainable recommendation: A survey and new perspectives. **CoRR**, abs/1804.11192, 2018. Disponível em: <http://arxiv.org/abs/1804.11192>. Citado 4 vezes nas páginas 2, 3, 13 e 14.

ZHANG, Y. et al. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In: **Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval**. New York, NY, USA: ACM, 2014. (SIGIR '14), p. 83–92. ISBN 978-1-4503-2257-7. Disponível em: <http://doi.acm.org/10.1145/2600428.2609579>. Citado 2 vezes nas páginas 2 e 14.