# Assessing and Improving Recommender Systems to Deal with User Cold-Start Problem

**Crícia Zilda Felício Paixão**

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2017

Crícia Zilda Felício Paixão

# Assessing and Improving Recommender Systems to Deal with User Cold-Start Problem

Tese de doutorado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Celia Aparecida Zorzo Barcelos
Coorientador: Philippe Preux

Uberlândia
2017

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

P149
2017

Paixão, Crícia Zilda Felício, 1977--
Assessing and improving recommender systems to deal with user cold-start problem / Crícia Zilda Felício Paixão. - 2017.
113 f. : il.

Orientadora: Célia Aparecida Zorzo Barcelos.
Coorientador: Preux, Philippe.
Tese (doutorado) - Universidade Federal de Uberlândia, Pós-Graduação em Ciência da Computação.
Inclui bibliografia.

1. Computação - Teses. 2. Percepção visual - Teses. 3. Usuários da Internet - Teses. 4. Sistemas de Recomendação Social - Teses. I. Barcelos, Célia Aparecida Zorzo. II. Preux, Philippe. III. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Ciência da Computação. IV. Título.

CDU: 681.3

*Aos meus filhos Pedro Paulo e Mateus e ao meu esposo Klérisson.*

# Agradecimentos

A Deus e a todo o apoio espiritual que recebi na minha jornada.

À professora Sandra de Amo, pela orientação de parte do meu trabalho e pelo exemplo de dedicação ao trabalho e competência como pesquisadora.

Ao professor Philippe Preux por ter me acolhido no grupo Sequel - INRIA, pela orientação e por ter colaborado ativamente nas publicações.

À professora Celia Barcelos por ter assumido o compromisso da minha orientação na reta final do meu doutorado, me dando a segurança necessária para que eu pudesse concluí-lo.

Ao Instituto Federal do Triângulo Mineiro - Campus Uberlândia Centro pela licença de 1 ano e 10 meses para que pudesse me dedicar exclusivamente ao doutoramento.

Aos colegas do LSI pelo auxílio com os artigos e discussões sobre o trabalho.

Ao meu filho Pedro Paulo pela graça de ser mãe, pelo sorriso que me revigora, pelo abraço que me conforta; ingredientes que fizeram a diferença na execução do meu trabalho. Ao meu filho Mateus que veio me acompanhar nos últimos meses dessa jornada enchendo meu coração de alegria.

Ao meu esposo Klérisson, por seu amor, companheirismo e compreensão.

*"We must let go of the life we have planned, so as to accept the one that is waiting for us."*

*(Joseph Campbell)*

# Resumo

Sistemas de recomendação fazem parte do nosso dia-a-dia. Os métodos usados nesses sistemas tem como objetivo principal predizer as preferências por novos itens baseado no perfil do usuário. As pesquisas relacionadas a esse tópico procuram entre outras coisas tratar o problema do *cold-start* do usuário, que é o desafio de recomendar itens para usuários que possuem poucos ou nenhum registro de preferências no sistema.

Uma forma de tratar o *cold-start* do usuário é buscar inferir as preferências dos usuários a partir de informações adicionais. Dessa forma, informações adicionais de diferentes tipos podem ser exploradas nas pesquisas. Alguns estudos usam informação social combinada com preferências dos usuários, outros se baseiam nos clicks ao navegar por sites Web, informação de localização geográfica, percepção visual, informação de contexto, etc. A abordagem típica desses sistemas é usar informação adicional para construir um modelo de predição para cada usuário. Além desse processo ser mais complexo, para usuários *full cold-start* (sem preferências identificadas pelo sistema) em particular, a maioria dos sistemas de recomendação apresentam um baixo desempenho. O trabalho aqui apresentado, por outro lado, propõe que novos usuários receberão recomendações mais acuradas de modelos de predição que já existem no sistema.

Nesta tese foram propostas 4 abordagens para lidar com o problema de *cold-start* do usuário usando modelos existentes nos sistemas de recomendação. As abordagens apresentadas trataram os seguintes aspectos:

- Inclusão de informação social em sistemas de recomendação tradicional: foram investigados os papéis de várias métricas sociais em um sistema de recomendação de preferências *pairwise* fornecendo subsidíos para a definição de um framework geral para incluir informação social em abordagens tradicionais.

- Uso de similaridade por percepção visual: usando a similaridade por percepção visual foram inferidas redes, conectando usuários similares, para serem usadas na seleção de modelos de predição para novos usuários.

- Análise dos benefícios de um framework geral para incluir informação de redes de usuários em sistemas de recomendação: representando diferentes tipos de informação adicional como uma rede de usuários, foi investigado como as redes de usuários podem ser incluídas nos sistemas de recomendação de maneira a beneficiar a recomendação para usuários *cold-start*.
- Análise do impacto da seleção de modelos de predição para usuários *cold-start*: a última abordagem proposta considerou que sem a informação adicional o sistema poderia recomendar para novos usuários fazendo a troca entre os modelos já existentes no sistema e procurando aprender qual seria o mais adequado para a recomendação.

As abordagens propostas foram avaliadas em termos da qualidade da predição e da qualidade do *ranking* em banco de dados reais e de diferentes domínios. Os resultados obtidos demonstraram que as abordagens propostas atingiram melhores resultados que os métodos do estado da arte.

**Palavras-chave:** Sistema de Recomendação; Preferências dos Usuários; Problema do *cold-start* do usuário; Sistemas de Recomendação Social; Percepção Visual; *Multi-armed bandits*.

**Crícia Zilda Felício Paixão**

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2017

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

The undersigned hereby certify they have read and recommend to the PPGCO for acceptance the thesis entitled "*Assessing and Improving Recommender Systems to Deal with User Cold-Start Problem*" submitted by **Crícia Zilda Felício Paixão** as part of requirements for obtain the **PhD degree in Computer Science**.

Uberlândia, ___ de _________ de _____

Supervisor: ___________________________________
Prof. Dra. Celia Aparecida Zorzo Barcelos
Universidade Federal de Uberlândia

CoSupervisor: ___________________________________
Prof. Dr. Philippe Preux
University of Lille

Examining Committee Members:

___________________________________
Prof. Dra. Maria Camila Nardini Barioni
Universidade Federal de Uberlândia

___________________________________
Prof. Dr. Bruno Augusto Nassif Travençolo
Universidade Federal de Uberlândia

___________________________________
Prof. Dr. João Gama
Universidade do Porto

___________________________________
Prof. Dr. Altigran Soares da Silva
Universidade Federal do Amazonas

# Abstract

Recommender systems are in our everyday life. The recommendation methods have as main purpose to predict preferences for new items based on user's past preferences. The research related to this topic seeks among other things to discuss user cold-start problem, which is the challenge of recommending to users with few or no preferences records.

One way to address cold-start issues is to infer the missing data relying on side information. Side information of different types has been explored in researches. Some studies use social information combined with users' preferences, others user click behavior, location-based information, user's visual perception, contextual information, etc. The typical approach is to use side information to build one prediction model for each cold user. Due to the inherent complexity of this prediction process, for full cold-start user in particular, the performance of most recommender systems falls a great deal. We, rather, propose that cold users are best served by models already built in system.

In this thesis we propose 4 approaches to deal with user cold-start problem using existing models available for analysis in the recommender systems. We cover the follow aspects:

- Embedding social information into traditional recommender systems: We investigate the role of several social metrics on pairwise preference recommendations and provide the first steps towards a general framework to incorporate social information in traditional approaches.

- Improving recommendation with visual perception similarities: We extract networks connecting users with similar visual perception and use them to come up with prediction models that maximize the information gained from cold users.

- Analyzing the benefits of general framework to incorporate networked information into recommender systems: Representing different types of side information as a user network, we investigated how to incorporate networked information into recommender systems to understand the benefits of it in the context of cold user recommendation.

- ❏ Analyzing the impact of prediction model selection for cold users: The last proposal consider that without side information the system will recommend to cold users based on the switch of models already built in system.

We evaluated the proposed approaches in terms of prediction quality and ranking quality in real-world datasets under different recommendation domains. The experiments showed that our approaches achieve better results than the comparison methods.

**Keywords:** Recommender system; User preferences; Cold-start User problem; Social Recommender Systems; Visual perception; Multi-armed bandits.

# List of Figures

# List of Tables

# Acronyms list

nDCG: *Normalized Discounted Cumulative Gain*

PdMS: *Prediction Model Selection*

RS: *Recommender Systems*

RMSE: *Root Mean Squared Error*

SRS: *Social Recommender Systems*

VP-Rec: *Visual Perception recommender*

VP-Similarity: *Visual Perception Similarity*

# Contents

I hereby certify that I have obtained all legal permissions from the owner(s) of each third-party copyrighted matter included in my thesis, and that their permissions allow availability such as being deposited in public digital libraries.

Crícia Zilda Felício Paixão

# Chapter 1

# Introduction

## 1.1 Context

Recommender systems popularization is direct related with the increasing amount of Web information. They can be seen as an information filtering mechanism, but different from search engines. Instead of having a user querying contents, recommender systems predict items/products users preferences. These systems try to predict which movies and songs people will enjoy, which clothes they should buy, which places they should visit, and even whom to date (RESNICK; VARIAN, 1997). The predictions are made based on user preferences information that can include items ratings or purchase histories.

Usually, recommender systems are classified based on how recommendation are made (ADOMAVICIUS; TUZHILIN, 2005): content-based (PAZZANI; BILLSUS, 2007; LOPS; GEMMIS; SEMERARO, 2011), collaborative filtering (HERLOCKER; KONSTAN; RIEDL, 2000; SU; KHOSHGOFTAAR, 2009) and hybrid (BURKE, 2002; GUNAWARDANA; MEEK, 2009). Content-based systems recommend items with similar content to the ones the user preferred in the past. Collaborative filtering recommended items that people with similar preferences liked in the past; and hybrid systems combine collaborative and content-based methods features.

Many of the research challenges that arise in studying collaborative filtering recommenders situate around data sparsity and cold-start problems (ADOMAVICIUS; TUZHILIN, 2005). Sparsity is related to the fact that there are a lot of items available in the system and few ratings per user. So, there are few ratings in common among users and that imposes more difficulties to the process of user preferences prediction. Item cold-start problem occur when an item is not rated by a substantial number of users, while user cold-start problem happen when the system has not enough users ratings to learn their preferences and make accurate recommendations.

Combining collaborative filtering and content-based methods in hybrid approaches is a way to address sparsity and item cold-start problem (MELVILLE; MOONEY; NAGARAJAN, 2002; GUNAWARDANA; MEEK, 2009; AMO; OLIVEIRA, 2014). Hybrid methods will recommend cold-start items based on its features that can be similar to others non-cold items.

However, to deal with user cold-start problem usually lies in two approaches: ask for user preference elicitation or use additional information to infer the preferences. We will not consider here the user preference elicitation to cope with the cold-start problem.

A user's preferences can be inferred from many kinds of information, such as social information (ALAHMADI; ZENG, 2015), click behavior (LIU; DOLAN; PEDERSEN, 2010), location-based information (CHENG et al., 2012a), visual attention (MELO; NOGUEIRA; GULIATO, 2015), and, more broadly speaking, contextual information (ADOMAVICIUS; TUZHILIN, 2015). With such information, the dominant trend of current studies has been towards designing new personalized prediction models for cold-start users. Henceforth, a recommender system cannot offer personalized recommendations to a user until it has collected enough preference information for that user.

Furthermore, Gomez-Uribe and Hunt show how it is important to provide recommendations that fit a user's needs (GOMEZ-URIBE; HUNT, 2015). They advocate that a higher level of personalization is fundamental to increase the chances of success when offering recommendations. According to them, Netflix, a major Internet TV enterprise, "saves up more than $1B per year" with the reduction of subscriber monthly churn, thanks to recommenders algorithms that better match users' expectations.

This raises one natural question: how to improve recommender systems to better deal with a cold user? This question of dealing with cold users is the main issue addressed in this thesis.

Regarding the vocabulary, new users are also qualified as "cold" users; dealing with them is the user "cold start" problem. Once enough information to provide accurate recommendations has been gathered, the user is said to be "warm" or "hot".

In short, the recommendation task is naturally a complex task. This may be aggravated at the beginning of a user experience when there is a lack of data to infer the users' preferences. As evidence, bad first recommendations correlate with less active users as shown in an analysis of MovieLens system (CHANG; HARPER; TERVEEN, 2015). Thus, recommender systems have to carefully deal with cold users.

## 1.2 Problem

There are mainly two practical ways to deal with cold users: (i) cold users are asked to complete a preference elicitation process; (ii) cold users are admitted directly into a system with no requirements, leading to an initial experience that is non-personalized.

In this thesis, we are primarily concerned with the second approach, as we focus our attention on the simplest scenario: cold users. For example, one of the most used datasets to evaluate recommender systems, the MovieLens (HARPER; KONSTAN, 2015), contains less than 1% of density, which represents the percentage of cells in the full user-item matrix that contain a known rating values. Alternative datasets also present similar behavior (DROR et al., 2012; MCAULEY; PANDEY; LESKOVEC, 2015; MCAULEY et al., 2015), showing that the cold user issue occurs very often.

Many problems may occur due to the little information given by users and the amount of cold users that we meet in real life. Below we list some concrete problems highlighted by the literature:

- Researchers have found that compelling a user to go through a lengthy bootstrap process often leads the user to leave the system. Also, bad initial recommendations decrease the chance that a user remains active (RASHID; KARYPIS; RIEDL, 2008; DRENNER; SEN; TERVEEN, 2008; CHANG; HARPER; TERVEEN, 2015).
- Recommendations from different algorithms have pros and cons no matter the user. On the one hand, a user can benefit from independent recommendation approaches. But on the other hand, users in general are not willing to switch algorithms by themselves (BELLOGÍN; SAID; VRIES, 2014; EKSTRAND et al., 2015).
- Multiple recommendation algorithms have the power to improve the user experience. However, keeping a large number of recommendation approaches is computationally not feasible (TÖSCHER; JAHRER; BELL, 2009; DOOMS; PESSEMIER; MARTENS, 2015).

These problems show the difficulty of a user recommendation, specially to offer high quality initial recommendations. It is important to ensure that the first user experience is correctly performed.

## 1.3 Improving Recommender Systems with Existing Prediction Models

To deal with user cold-start problems, approaches have been proposed to support recommender systems. One common approach consists in using side information (ZHOU et al., 2012; NING; KARYPIS, 2012). This information is provided (or crawled) to recommender systems, so that they can infer users' preferences (MA et al., 2011). Overall, the literature concentrates on solutions that are specific for each recommendation model. There is a lack for solutions that extend traditional recommender systems to consider side information, but take advantage of current recommendation algorithms, as described below.

PrefRec (AMO; OLIVEIRA, 2014), a model-based hybrid recommender system framework based on pairwise preference mining and preferences aggregation techniques, was devised in such a way that all users are compelled to provide ratings. Although, this system can handle cold items, it cannot deal with cold users. Another example is CRESA (MELO; NOGUEIRA; GULIATO, 2015), an image recommender. It combines textual attributes, visual features, and human visual attention to enhance items' profile. However, a user without preferences will not get any recommendation from this system.

A primary inhibitor to adopt many proposed solutions is that they are based on specific techniques. Therefore, a solution to provide personalized recommendation through side information can be a framework to extend recommender systems in such a way that leverages their own techniques and recommendation models.

A second common approach to alleviate cold user issues is based on switching mechanisms (BILLSUS; PAZZANI, 2000; ZANKER; JESSENITSCHNIG, 2009). Prior works draw on techniques to switch among recommenders (EKSTRAND; RIEDL, 2012; BRAUNHOFER; CODINA; RICCI, 2014; TANG et al., 2014). For instance, a cold user receives a recommendation from recommender that can perform without any ratings. Then, as more ratings become available, they switch to a collaborative recommender.

Apart from being costly, maintaining a stack of recommenders also requires a bigger range of side information. Moreover, some of this information may be noisy. In such cases, it can be risky to incorporate side-information into the recommendation process, because it can harm the efficacy of the process. Thus, a solution that better exploits the models generated by one recommender could mitigate such costs. We show an approach using sequential learning based on bandits' algorithms (AUER; CESA-BIANCHI; FISCHER, 2002a; SUTTON; BARTO, 1998) for this task (Chapter 6).

In summary, existing approaches lack of: (i) a deep understanding of the benefits provided by a general framework to incorporate side-information, (ii) a better use of prediction models already built in the system, and (iii) an analysis of the impact of prediction model selection from one recommender to deal with cold users.

## 1.4 Our Approach in a Nutshell

In this thesis, we propose methods and techniques to improve recommender systems to offer better recommendations to cold users. Such methods may ensure, for example, better use of information already existent in the recommender systems. Based on this principle we defined our hypothesis: **Hypothesis**: Cold users are best served by prediction models already available in the system instead of building new models or use baseline recommenders.

In order to validate the hypothesis we cover four aspects: the benefits of social information in traditional recommenders, the improvement of image recommenders, the benefits of a general framework to incorporate networked information in traditional recommenders, and the impact of prediction model selection to cold users.

**Embedding social information into traditional recommender systems.** While it is not new that social network information increase recommendation effectiveness (LIU; LEE, 2010; TANG; HU; LIU, 2013), incorporating such information into recommender systems is not trivial. We investigate the role of several social metrics on pairwise preference recommendations and provide the first step towards a general framework. The goal is to incorporate social information in traditional approaches to select between existent prediction models.

**Improving recommendation with visual perception similarities.** Current approaches related to image recommendation do not take advantage of visual attention to offer recommendation for cold users (MELO; NOGUEIRA; GULIATO, 2015). We propose two solutions, Matrix-factorization and Pairwise based methods, in order to benefit cold users. In both solutions, we extract networks of visual perceptions from users and use them to come up with prediction models that maximize the information gained from cold users. The matrix-factorization method made use of visual perception network to build new prediction models while pairwise method used visual perception to select a prediction model. Here we also analyze the benefits of select a prediction model instead of build a new one. However, we used visual perception similarities instead of online social network information.

**Analyzing the benefits of general framework to incorporate networked information into recommender systems.** Networked information proved to be valuable to many research fields (STROGATZ, 2001; BOCCALETTI et al., 2006). Many recommender systems are not able to handle such data structure or they are designed for specific tasks, for example, particular classes of social recommenders (GUY, 2015; SUN et al., 2015). We undergo an investigation on how to incorporate networked information into recommender systems. We wanted to understand whether such data structures are worthwhile enforcing, given the current state of the systems, in the context of cold user recommendation. Besides the use of a general user network information to select a prediction model for a cold user, we also change the way the system generate the prediction models. These changes make this aspect be different from the two others one.

**Analyzing the impact of prediction model selection for cold users.** Different prediction models are used to deal with distinct stages of a user experience. For example, a particular model works better in earlier stages when the recommender system does not know the user. However, in later stages, a different model should be more effective, and therefore, one switches to the more powerful model. Switching methods (BURKE, 2002) were designed to handle the cold-start problem. The idea is to switch from one model to another once the system has enough data about the user, so he is no longer cold. We empirically assess the efficacy of a model selection specifically within the cold-start stage. In the others aspects we take in consideration to select one single prediction model for cold user recommendation. The selection was based on information from social network,visual perception network and general user network. On the other hand, here the idea is switch between the different prediction models to learn the better one for a cold user recommendation. We based on user feedback and sequential learning to decide the better prediction model to use in a certain moment.

## 1.5 Contributions

The main contributions of this thesis are summarized as follows:

**Contribution 1.** We devised a method to incorporate social network information in pairwise recommender systems. We provide new experiments on a system that was not able to handle such data structures (FELÍCIO et al., 2015; FELÍCIO et al., 2016a).

**Contribution 2.** We provide two approaches to better support recommendation using visual perception similarities at user cold-start stage (FELÍCIO et al., 2016b; FELÍCIO et al., 2016c).

**Contribution 3.** We generalize our prior works to understand how a framework to incorporate network information can impact in typical recommender systems and help it deal with cold start problem (FELÍCIO et al., 2016e).

**Contribution 4.** We devised a method to take advantage of the number of prediction models generated by a recommender to learn which one might match a cold user expectation (FELÍCIO et al., 2016d; FELÍCIO et al., 2017).

## 1.6 Structure of the Thesis

### Chapter 2: Fundamentals

This chapter presents a background of traditional recommender systems and draws insight from other work in social recommender systems, image recommendation, and model selection in recommender systems. Note that we focus on user cold-start problem related works.

### Chapter 3: Pairwise Recommenders with Social Information

This chapter describes our first approach towards a general framework to extend recommender systems. We implement our method in a pairwise recommender system in order to be able to handle social networks information, and deal with cold users as consequence. We validate our approach on public datasets.

### Chapter 4: Improving Recommendation with Visual Perception Similarities

This chapter describes our approach to analyze the benefits provided by visual perception similarities under the specific context of recommendation based on item/product image at user cold-start stage. We validate our approach against two real datasets.

### Chapter 5: Benefits of a General Framework to Incorporate Networked Information

This chapter presents our analysis on the impact of a general framework to incorporate networked information into recommender systems. We include a large experimental results from 6 different datasets and show improvement in all experiments.

### Chapter 6: Learning to Select Prediction Model at Cold-Start Stage

This chapter describes our envisioned mechanism to learn the preferences of a cold user, in which an agent explores and exploits a given set of prediction models alternatives in the course of a sequential decision process. Finally, we provide experiments and discuss their results.

### Chapter 7: Conclusion

This chapter concludes the thesis and presents future work. Finally, it also presents the collaboration with the SequeL - INRIA Lille research group.

Chapter 2

# Fundamentals

## 2.1 Introduction

In this chapter, we present traditional recommender systems features and their current approaches. The focus is on the approaches that cope with cold user. Then, in related work section, we show that there are lacks in all approaches as to deal with cold users.

We argue that many related works are dedicated to propose new recommendation models and there is little focus on leveraging the available amount of existing prediction models already built in the recommender systems.

## 2.2 Traditional Recommender Systems

Generally, recommender systems information is composed by users attributes, items attributes and user feedback over items. User feedback represents preferences and is usually expressed as ratings (ratings range from 1 to 5, 1 to 10, etc.) or in a binary way (0 or 1) that will indicate if the user had purchase or not one product, *like* or *dislike*, etc.

According to Adomavicius and Tuzhilin (2005), the recommendation problem consists in predict the preferences for items not yet rated by a target user. A target user is whom the system will make recommendations. Once the predictions are done is possible to compute a ranking of items and recommend the ones with high score.

The recommendation problem formalism can be seen as: let $U = \{u_1, ..., u_m\}$ be a set of users and $I = \{i_1, ..., i_n\}$ be a set of items, $RU(A_1, ..., A_r)$ be a relational scheme related to users, and $RI(A_1, ..., A_t)$ be a relational scheme related to items. The user-item rating matrix in a system with $m$ users and $n$ items is represented by $R = [r_{u,i}]_{m \times n}$, where each entry $r_{u,i}$ represents the rating given by user $u$ on item $i$.

As example, consider a movie recommendation scenario where the users ratings is represented in Table 1 as a user-item rating matrix with ratings range from 1 to 5. For each movie we have items attributes, see Table 2. And each user has a set of demographic attributes as shown in Table 3.

| User | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ |
|---|---|---|---|---|---|
| Ted | 5 | - | 4 | - | - |
| Mary | 5 | - | 3 | 4 | - |
| Rose | 4 | - | 3 | 4 | - |
| Zoe | 5 | - | 4 | - | 3 |

Table 1 – Example of a user-item rating matrix. - means that the user has not rate the item

| ID | Title | Genre | Director | Year | Actor |
|---|---|---|---|---|---|
| $i_1$ | Seven | Suspense | Fincher | 1995 | Brad Pitt |
| $i_2$ | Babel | Drama | Inárritu | 2006 | Brad Pitt |
| $i_3$ | Titanic | Romance | Spielberg | 1997 | Di Caprio |
| $i_4$ | Lincoln | Drama | Spielberg | 2012 | Day-Lewis |
| $i_5$ | Amistad | Drama | Spielberg | 1997 | Anthony Hopkins |

Table 2 – Example of movies attributes.

| Name | Gender | Location | Birth date |
|---|---|---|---|
| *Ted* | M | Nova York | 1978-10-18 |
| *Mary* | F | Paris | 1980-04-07 |
| *Rose* | F | Paris | 1987-02-12 |
| *Zoe* | F | Los Angeles | 1985-06-05 |

Table 3 – Example of users demographic attributes.

Traditional recommender systems will make use of a function $predict(u_k, i_j)$ to predict the rating for a item $i_j \in I$ from a user $u_k \in U$. The way that function *predict* will compute the predicted score is different for each recommender system approach. The traditional recommender systems are based on collaborative filtering, content-based or hybrid methods.

In our example, the recommender system will use the user-item rating matrix in Table 1 to make predictions for a target user. Next section describes how the prediction is computed per method.

### 2.2.1 Collaborative Filtering

Collaborative filtering approach can be classified as memory-based and latent factors methods (KOREN; BELL; VOLINSKY, 2009). Memory-based collaborative filtering make the predictions based on the user or items neighborhood similarities. Latent factors methods will learn a prediction model, recognizing patterns in training set. Different similarity measures can be used in memory-based methods like Pearson correlation, Spearman correlation, cosseno similarity, etc. After computing the similarity, the prediction is made using the neighborhood set and the similarity score between them.

The $predict(u_k, i_j)$ equation computes the predict preference score of user $u_k$ on item $i_j$ according to the Eq. (1). In Eq. (1) $\bar{r}_{u_k}$ represents the $u_k$'s average rating score for items rated in common with users $u$, $sim(u_k, u)$ is the similarity between $u_k$ and $u$ calculated using one of similarity measures (Pearson, Spearman, Cosseno, etc.), $\bar{r}_u$ represents the $u$'s average rating score for items rated in common with user $u_k$ and $r_{u,i_j}$ is the rating of user $u$ for item $i_j$.

$$predict(u_k, i_j) = \bar{r}_{u_k} + \frac{\sum_{u \in U} sim(u_k, u)(r_{u,i_j} - \bar{r}_u)}{\sum_{u \in U} |sim(u_k, u)|} \tag{1}$$

*Example:* In Table 1, consider a cold user Paty was added and she gave film Lincoln a rating of 5. Suppose that a memory-based system will compute the prediction to Paty for Seven, and the method will compute the similarity between users. The similarity calculus between Paty and others users who rate the movie Lincoln in the past will show a high similarity score between Paty and users that also gave high ratings to Seven. There are a tendency of the predicted rating for Seven has a similar value of rating gave by users with a high similarity score.

The memory-based method disadvantages include the fact that the recommendation is not possible to users that had rated nothing in common with the others users and users with no ratings. Additionally, items with no ratings will not be recommended too.

Latent factors methods revolutionized recommender systems. Mainly because they present better accuracy than memory-based methods. Regarding the techniques, matrix factorization is the most popular. The idea behind matrix factorization (KOREN; BELL; VOLINSKY, 2009) is to profile both users and items in a user-item rating matrix. Then, through latent factor models, transform both items and users to the same latent factor space. Thus, making them directly comparable. Intuitively, the latent space tries to explain ratings by characterizing items and users on factors inferred from user ratings.

Figure 1 shows an illustration of matrix factorization method. **R** is the user-item rating matrix, **P** is the users latent factors and **Q** the items latent factors. Intuitively, **P** matrix and **Q** matrix are factorized from **R**. The product between **P** and **Q** contains the ratings predictions.

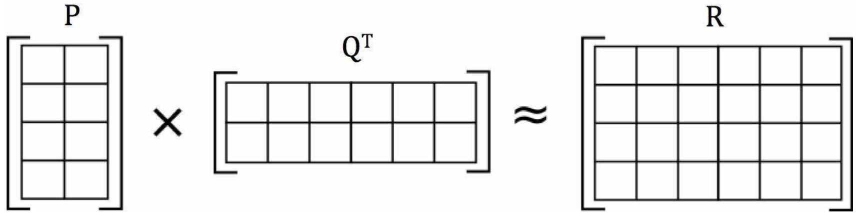


Figure 1 – Matrix Factorization method illustration

As example, when the items are paintings, factors could measure (i) dimensions such as impressionism vs. cubism or market value; (ii) less well defined dimensions such as expressiveness, sentimentalism. The high correspondence between items factors and users factors indicates one recommendation (KOREN; BELL; VOLINSKY, 2009).

We use a matrix factorization technique to get a matrix of predicted ratings $R'$ from the user-item rating matrix. $R'$ is expressed as a product of latent factors, $P$ and $Q$. The predicted rating of the item $i_k$ by user $u_j$ is $R'_{u_j,i_k} = predict(u_j, i_k, P, Q)$, the details of this function depending on the completion method being used. A simple method computes the prediction using the product of the user latent factor and item latent factor without any addition of terms or other latent factors. Then, using this type of method, the predicted rating of the item $i_k$ by user $u_j$ is given by $R'_{u_j,i_k} = P_{u_j} Q^T_{i_k}$.

Netflix competition (Netflix Prize[1]) had contributed to latent factors methods popularization. That company asked for solutions to improve its recommender system accuracy. The competition offer a million dollar prize, and motivate the increasing of collaborative filtering research. The winner group used matrix factorization technique. The proposed method obtained 10% of improvement over Netflix recommender system accuracy (see Netflix Prize website).

### 2.2.2 Content-based

Content-based methods compute $predict(u_k, i_j)$ using the ratings of items that are similar to a given item $i_j$. The items similarity can be defined based on items attributes. Then, the higher predicted ratings will be associated to items that have the same attributes contents from items that received high ratings from user $u_k$ in the past.

As example, consider the scenario in Section 2.2. Using content-based method, the selection of a film to recommend for user Mary is based on the movies attributes, see Table 2. Those attributes are used to compute the similarity among movies. Suppose we will consider similarity only by genre. A movie to recommend to Mary will be Babel, because it has the same genre as Lincoln, which received a rating of 4 from Mary.

Content-based methods are limited to recommend to target user only items that have similar attributes with items previous rated by the user. In our example, unless Mary give high ratings to movies from others genre, she will always receive recommendation of dramas. This drawback of content-based methods is called overspecialization (ADOMAVICIUS; TUZHILIN, 2005). Another crucial limitation is that the system will not recommend to users without ratings (*full cold start user*). Furthermore, the similarity between users preferences is not taken in consideration in content-based methods.

### 2.2.3 Hybrid

Hybrid method combines Collaborative Filtering and Content-based approaches. According to Adomavicius and Tuzhilin (2005), there are different ways to combine the two

[1] http://www.netflixprize.com/

approaches and the hybrid systems are classified in 4 types:

- Methods that combine the predictions of a collaborative filtering method with a content based method (CLAYPOOL et al., 1999; PAZZANI, 1999);
- Methods that incorporate content-based features in collaborative filtering approach (BALABANOVIĆ; SHOHAM, 1997; GOOD et al., 1999);
- Methods that incorporate collaborative filtering features in content-based methods (SOBOROFF; NICHOLAS, 1999);
- Unified methods with collaborative filtering features and content-based features (BASU et al., 1998; POPESCUL; PENNOCK; LAWRENCE, 2001).

Incorporating content-based features in collaborative approaches generates a system that is able to predict ratings to items without ratings based on similar items. Also allows recommendation to users using users similarities. In this way, even cold items can be recommended by the prediction method of content-based through similarity calculus. Moreover, similarity between users enables the recommendation of items that were not rated for the target user and is not similar to items previously rated.

In our example, consider Peter as a cold user in the system. He gives a rating of 4 to the movie Titanic. Analyzing Peter's preferences, Titanic is similar to Lincoln and Amistad if we take movies' director as an attribute to similarity calculus (content-based method). Beside that, Peter's preference for Titanic indicates the similarity with users that gave high rating to the film Seven (collaborative filtering method). Then, using a hybrid approach the three movies will be predicted as good recommendations to Peter.

One of the hybrid methods disadvantage is the need to rebuild the users prediction when a new item is add in the system.

## 2.3 Related Work

In this section we present the related work in social recommender systems, visual perception similarities and prediction model selection.

### 2.3.1 Social Recommender Systems

Enriching recommender systems with contextual social information has been studied widely (KAUTZ; SELMAN; SHAH, 1997; PENNOCK et al., 2000; BONHARD; SASSE, 2006; GROH; EHMIG, 2007; MA et al., 2011; MACEDO; MARINHO; SANTOS, 2015). A notable reference to social recommender systems is Guy (2015) and Bobadilla et al. (2013) presented a general survey.

Social attachments proved to be a valuable source of information to alleviate data sparsity issues. Furthermore, several social-based recommender systems have recently been proposed (CARRER-NETO et al., 2012; LI; WU; MAMOULIS, 2014; ALAHMADI; ZENG, 2015; REAFEE; SALIM; KHAN, 2016; BARJASTEH et al., 2016).

Ma et al. (2008) and Ma et al. (2011) proposed social enhanced algorithms to improve matrix factorization based recommenders. For instance, SoRec (MA et al., 2008) relies on probabilistic matrix factorization, to better deal with data sparsity and accuracy problems. Later, SoReg (MA et al., 2011) also relies on a matrix factorization framework, but incorporates social constraints into its built models. TrustMF (YANG et al., 2013), is an adaptation of matrix factorization technique to map users in terms of their trust relationship; and SocialMF (JAMALI; ESTER, 2010), explores the propagation of trust among users. Both systems present high scores dealing with cold-start users, so that we consider them in our experimental studies for the sake of comparison with our approaches.

SocialMF was rebuilt in (YANG; STECK; LIU, 2012) to take into consideration category-specific friends. The intuition is that a user may trust different subsets of users regarding different domains. In terms of assuming that links among users in a social network reflects actual similarities among users there are some works (MA, 2014; MA, 2013). In location-based social networks such as Foursquare, users with social relations are likely to do check-ins at the same locations (YE; LIU; LEE, 2012). Delporte et al. (2013) developed an improved matrix factorization based recommender combining social information with implicit feedback.

Cheng, Liu and Yu (2016) propose an online social trust model that explores public trusted users, that is users who have a large number of fans in the social network. The items it recommends are more likely accepted and trusted by the public. The authors construct a Bayesian network to model the trust relationship among these public and other users. Their recommendations are based on the similarity between the trust users and the target user in terms of items or reviews and on the proximity of time of accesses by two users (CHENG; LIU; YU, 2016).

**Summary**

Regarding social recommenders, the major focus of this thesis has been on the use of social networks metrics to select a consensual prediction model, more details on Chapter 3 and 5. Our study differentiates itself from all these existing studies since we aim to provide a method to select a more appropriate prediction models to deal with cold users.

### 2.3.2 Visual Perception Similarity

A pioneer study of Xu, Jiang and Lau (2008) uses similarity based on visual perception to build recommendation models. The experiments involved five users and compare their approach against Google and YouTube search queries results to recommend documents, images or videos.

Umemoto et al. (2012) proposed to relate users' eye movements with information seeking. Then, they rank search results to emphasize relevant parts on a Web page. The work (XU; JIANG; LAU, 2010) also used gaze positions of a user in conjunction with facial expressions as two types of implicit user feedback within the context of personalized web page recommendation. Although relied on visual perceptions, they did not handle images or videos elements, but text content that comes along with those elements in search queries.

Human visual perception data was adopted to build a gaze-based classifier for the image preference mining (SUGANO et al., 2014). The authors have shown that their approach had a higher level of accuracy than metadata-based baseline methods. This work is close to ours in the sense that user visual perception and preference data have been taken as a knowledge source to recommend using images (more details in Chapter 4).

There are many new others applications of visual perception similarity in academic research (MUTLU; VEAS; TRATTNER, 2016; TEO et al., 2016; WRÓBLEWSKA; RACZKOWSKI, 2016; SUGANO; ZHANG; BULLING, 2016). Our main motivation is to complement the work of Melo, Nogueira and Guliato (2015). They proposed a content-based filtering enhanced by human visual attention applied to clothing recommendation. Their approach is specific for the domain of clothes and relies on item visual perception similarity combined with the measures conventionally used in content-based image recommendation systems. However, their proposed method falls a great deal facing cold users.

**Summary**

Our work is innovative in the sense that we incorporate *visual perception data* as a contextual information for recommender systems. We use a clustering-based filtering approach that infers a visual perception network, mainly to tackle cold-start problem. The users visual perception similarities is represented by the way the users look at different images, such as eye fixation time and gaze position. Having the users with similar visual perception, we consider non-cold users preferences to provide recommendation to cold users.

### 2.3.3 Model Selection

Model selection methods have been applied to recommender systems as a switching mechanism between stages of a user experience (BURKE, 2002). However, we are interested in selecting better models within the cold-start stage.

There are some common themes between our approach and the work of Billsus and Pazzani (2000) and Mary, Gaudel and Preux (2015). The first work proposed a news recommender system that leverages explicit feedback from the user to learn and update the user model based on classification methods. The second one combines matrix factorization approach and bandits algorithms to address online recommendation problem where at each item recommendation the system receive a feedback, update the user prediction model and select the best item for the next recommendation. We also rely on user's feedback and bandits algorithms. However we use feedback to analyze which prediction model might be the best one to use in the next recommendation.

Our approach builds on the same understanding as Ekstrand and Riedl (2012): different prediction models unveil distinct results. While their focus is on switch hybrids systems, we proposed a solution to switch among consensual prediction models existing in the same system. Specially within cold-start stage, Braunhofer, Codina and Ricci (2014) also proposed a switching mechanism, but dependent on contextual information.

In Chapter 6, we aim to select prediction models without prior side information. Thus, we rely on special learning method called Multi-armed Bandits (AUER et al., 1995; AUER; CESA-BIANCHI; FISCHER, 2002b; CESA-BIANCHI, 2008; BUSA-FEKETE; HÜLLERMEIER, 2014). We believe that exploring a number of available existing models in a recommender system and we can better find which model might be the best one for a cold user.

Li et al. (2010) reported on personalized recommendation of news articles as a contextual bandit problem. They propose LinUCB, an extension to the UCB algorithm. It selects the news based on mean and standard deviation. It also has a factor $\alpha$ to control the exploration/exploitation trade-off.

Moreover, Caron and Bhagat (2013) incorporate social components into bandit algorithms to tackle the cold-start problem. They designed an improved bandit strategy to model the user's preference using multi-armed bandits.

Several works model the recommendation problem using a multi-armed bandits setting in which the items to be recommended are the arms (BOUNEFFOUF; BOUZEGHOUB; GANCARSKI, 2012; GIRGIN et al., 2012). In a different way, Lacerda, Veloso and Ziviani (2013) and Lacerda et al. (2015) model users as arms to recommend daily-deals. They consider strategies for splitting users into exploration and exploitation. Li, Karatzoglou and Gentile (2016) proposed to double cluster users and items using bandits. We also rely on clustering users, but to reach a prediction model (see Chapters 3 and 5) in that could be leverage by our model selection strategy.

**Summary**

In comparison, our goal is the selection of existing prediction models that might offer better recommendations for cold users. Our model selection setting is also different, since the arms are consensual prediction models, detail on Chapter 6. Besides that, our approach requires no prior effort from the user.

## 2.4 Summary

In this chapter we first presented the main concepts related with traditional recommender systems and then made a review of works related with our approach. We have shown that there are lacks in current approaches to support recommender systems, in particular to ensure better cold user recommendation. We considered three aspects: (i) social-based recommender systems, (ii) image recommendation, and (iii) prediction model selection.

We identified the following problems. First, social information have not yet been used to select a model already built in the system. Second, image recommenders do not properly handle cold user. Third, the real impact of networked information on recommendation at cold-start stage is not known. Finally, learning a model that fits a cold user without prior information is still challenging in recommender systems.

The next four chapters present our approaches to cover each of these problems.

CHAPTER 3

# Pairwise Recommenders with Social Information[1]

## 3.1 Introduction

Social recommender systems (SRS) are important to help users to find relevant content. This is in part because of social media contents now account for the majority of content published on web. Typical social recommender systems assume a social network among users and makes recommendations based on the ratings of the users that have direct or indirect social relations with the target user (JAMALI; ESTER, 2010).

However, explicit user's ratings suffer from two known drawbacks: (i) The problems of calibration (consistency), which consists in incompatible users ratings on same scale, for example, on 1 to 5 star ratings scale, a rating of 4 for user X might be comparable to a rating of 5 for user Y. (ii) Resolution (granularity), this problem states that any numeric scale for ratings, say 1 to 5 stars, may be insufficient to capture all the users interests without loss of information (BALAKRISHNAN; CHOPRA, 2012) (AMO; RAMOS, 2014).

Thus, building on PrefRec (AMO; OLIVEIRA, 2014), we propose SOCIAL PREFREC a social recommender that applies user preference mining and clustering techniques to incorporate social information on the pairwise preference recommender system. Besides the advantage of the pairwise preference model, our focus is mitigate the user cold-start problem and sparsity.

Researches related to collaborative social recommendation argue that social information can easily deal with cold users and data sparsity, because instead of relying only in user's preferences they use available ratings from users whose hold a relationship with the target user (MA et al., 2011; WANG et al., 2014). In this work, we propose an approach to incorporate social network to provide recommendations. To leverage social influence in our model, we exploit several well know social network metrics.

[1] This chapter was published in a modified form in (FELÍCIO et al., 2016a)

In addition, social recommender systems in general make use of social information to build prediction models (recommendation models). Thus, for each cold user a new model must be built. In comparison, our approach harnessing pre-existent models. Instead of building a new model from scratch for each cold user, we cluster existent users and generate prediction models for each group. Through social information we select among existent models the most appropriated for a cold user.

Different factors of social relationships have influence on cold users. Some of these factors contribute or even harm social recommender systems (YUAN et al., 2015). Understanding the extent to which these factors impact SR systems provides valuable insights for building recommenders. We investigated the role of several social metrics on pairwise preference recommendations. Given that user's preference is similar to or influenced by their connected friends (TANG; HU; LIU, 2013), we also studied how to apply social similarities in a pairwise preference recommender. Social PrefRec is evaluated on two datasets, named Facebook and Flixster, to verify the integrity of our results. Focusing on social pairwise preference recommendation, our study addresses six questions:

*Q1: How accurately does social information help on item recommendation?*

We assessed the accurateness of Social PrefRec by comparing it to PrefRec. This is the key to determine whether a pairwise preference recommender can benefit from social information.

*Q2: How relevant are the recommendations made by a social pairwise preference recommender?*

One of the main reasons for the relevance of Social PrefRec is to mitigate the cold start problem for users through social information. To further assess our model, we compare Social PrefRec to three state-of-art social recommenders.

*Q3: Which social metrics are the most important for item recommendation?*

The previous questions focus on understanding whether pairwise recommenders could benefit from contextual social information. Here, we want to evaluate the overall performance of each social metric: friendship, mutual friends, similarity, centrality and interaction.

*Q4: How effective is* Social PrefRec *to mitigate data sparsity problems?*

In social recommender systems there is a common assumption that contextual social information mitigates data sparsity problems. To assess our model in this context, we evaluate the effectiveness of Social PrefRec with regards to PrefRec against five data sparsity levels.

*Q5: Does social degree affect* Social PrefRec *as much as profile length affects* PrefRec*?*

To achieve high-quality personalization, recommender systems must maximize the information gained about users from item ratings. The more ratings a user's profile has, the merrier will be. We want to check whether increasing the number of friends impacts our approach.

*Q6: Are there major differences between recommendations quality of popular and unpopular users?*

Here we further investigate social popularity effects on recommender systems. This question complements Q5, offering valuable insights into when and which social metric impacts the predictions.

This chapter is organized as follows. Section 3.2 presents the background knowledge undertaking in this chapter. Section 3.3 describes our proposed framework the Social PrefRec, as well as the applied social metrics and recommender model selection strategies. Section 3.4 describes our experimental settings and Section 3.5 presents the results. Finally, Section 3.6 concludes the chapter.

## 3.2 Background

In this section, we introduce the main concept underlying this chapter: pairwise preference recommender systems.

### Pairwise Preference Recommender Systems

Let $U = \{u_1, ..., u_m\}$ be a set of users and $I = \{i_1, ..., i_n\}$ be a set of items, $RU(A_1, ..., A_r)$ be a relational scheme related to users, and $RI(A_1, ..., A_t)$ be a relational scheme related to items. The user-item rating matrix in a system with $m$ users and $n$ items is represented by $R = [r_{u,i}]_{m \times n}$, where each entry $r_{u,i}$ represents the rating given by user $u$ on item $i$. Table 4 shows a set of 8 items (movies) and their attributes. A user-item rating matrix with 7 users and movies ratings in the range $[1, 5]$ is illustrated in Table 5.

Table 4 – Movie attributes.

| Item | Title | Decade | Director | Star | Genre |
|---|---|---|---|---|---|
| $i_1$ | Gangs of New York | 2000 | Scorsese | Di Caprio | Drama |
| $i_2$ | Catch me If You Can | 2000 | Spielberg | Di Caprio | Drama |
| $i_3$ | The Terminal | 2000 | Spielberg | Tom Hanks | Drama |
| $i_4$ | The Departed | 2000 | Scorsese | Di Caprio | Thriller |
| $i_5$ | Shutter Island | 2010 | Scorsese | Di Caprio | Thriller |
| $i_6$ | Saving Private Ryan | 1990 | Spielberg | Tom Hanks | Drama |
| $i_7$ | Artificial Intelligence | 2000 | Spielberg | Haley J. Osment | Drama |
| $i_8$ | Bridge of Spies | 2010 | Spielberg | Tom Hanks | Drama |

In traditional recommender systems, the recommendation task is based on the predictions of the missing values in the user-item matrix. Pairwise preference recommender

Table 5 – User-item rating matrix.

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ | $i_6$ | $i_7$ | $i_8$ |
|---|---|---|---|---|---|---|---|---|
| Ted | 5 | 2 | - | 1 | - | 2 | 1 | - |
| Zoe | 5 | 2 | 4 | 1 | 5 | 1 | - | 3 |
| Fred | 4 | - | 5 | - | 5 | - | 1 | - |
| Mary | 2 | 5 | 3 | 5 | - | - | - | 5 |
| Rose | 1 | - | 2 | - | 2 | - | - | 4 |
| Paul | - | - | 3 | 4 | 1 | - | - | 5 |
| John | 2 | - | - | 5 | 2 | - | - | - |

systems predicts the preference between a pair of items with missing values in the user-item matrix. Both types of systems use the predictions to extract a ranking of items and recommend the top-k.

We focus on the PrefRec framework, a hybrid model-based approach to design pairwise preference recommender systems. Essentially, PrefRec works in two phases: (a) construction of the prediction models, and (b) recommendation.

*A) Construction of the prediction models.* The main activities of this phase are Preferences Clustering, Consensus Calculus and Preference Mining.

**Preferences Clustering**: First, PrefRec clusters users according to their preferences. This process applies a distance function and a clustering algorithm $\mathcal{C}$ over the rows of the user-item rating matrix. A preference vector of user $u_x$ is defined as $\theta_{u_x} = R_{u_x}$, where $R_{u_x}$ is a row of matrix $R$. The output of the clustering algorithm is a set of clusters $C$, where each cluster $C_s$ has a set of users with the most similar preference vectors.

**Consensus Calculus**: For each cluster $C_s$, a consensus operator $\mathcal{A}$ is applied to compute $\hat{\theta}_s$, the consensual preference vector of $C_s$. $\hat{\theta}_{s,j}$ is the average rating for item $j$ in cluster $C_s$. Please note that the $\hat{\theta}_{s,j}$ element is computed if and only if more than half of the users in $C_s$ rated the item. Otherwise, this position will be empty.

An example of clustering and consensus calculus can be seen in Table 6. The users from Table 5 were clustered in two groups according to their preference vectors, and a consensual preference vector for each cluster was computed using the group average rating per item.

In comparison with the original PrefRec proposed in (AMO; OLIVEIRA, 2014), one main enhancement done in these two activities, Preferences Clustering and Consensus Calculus, was the replacement of the preference matrix and the consensual matrix by vectors. This new representation not only reduces the algorithm complexity and execution time, but remarkably allows a clustering of a better quality.

**Preferences Mining**: Having the consensual preference vector from each cluster, the system could establish the preference relation between pairs of items. Formally, a *preference relation* is a strict partial order over $I$, that is a binary relation $Pref \subseteq I \times I$

Table 6 – Clusters of users with consensual preferences.

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ | $i_6$ | $i_7$ | $i_8$ |
|---|---|---|---|---|---|---|---|---|
| Ted | 5 | 2 | - | 1 | - | 2 | 1 | - |
| Zoe | 5 | 2 | 4 | 1 | 5 | 1 | - | 3 |
| Fred | 4 | - | 5 | - | 5 | - | 1 | - |
| $\hat{\theta}_1$ | **4.7** | **2.0** | **4.5** | **1.0** | **5.0** | **1.5** | **1.0** | * |
| Mary | 2 | 5 | 3 | 5 | - | - | - | 5 |
| Rose | 1 | - | 2 | - | 2 | - | - | 4 |
| Paul | - | - | 3 | 4 | 1 | - | - | 5 |
| John | 2 | - | - | 5 | 2 | - | - | - |
| $\hat{\theta}_2$ | **1.3** | * | **2.7** | **4.7** | **1.7** | * | * | **4.7** |

Table 7 – $C_1$ pairwise relation.

| |
|---|
| $(i_1 > i_2)$ |
| $(i_1 > i_3)$ |
| $(i_3 > i_6)$ |
| $(i_5 > i_6)$ |
| $(i_2 > i_6)$ |
| $(i_5 > i_3)$ |
| $(i_2 > i_4)$ |
| $(i_6 > i_7)$ |

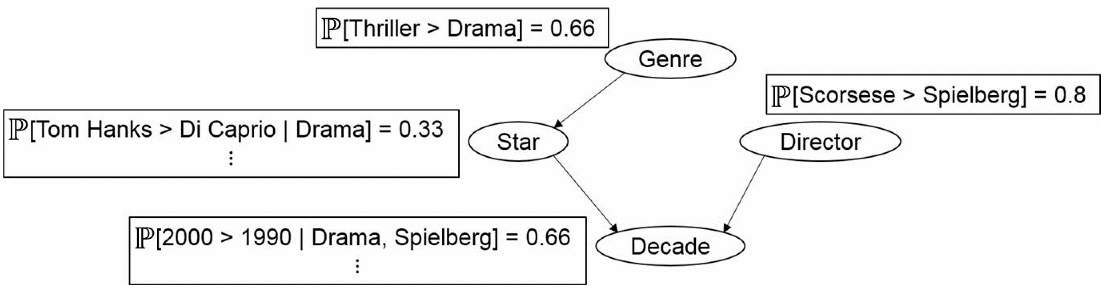


Figure 2 – Bayesian Preference Network $\mathbf{PNet}_1$ over $C_1$ preferences.

transitive and not reflexive. We denote by $i_1 > i_2$ the fact that $i_1$ is preferred to $i_2$. According to the previous example, a preference relation over consensual preference vector $\theta_1$ is presented in Table 7.

A preference miner $\mathcal{P}$ builds a prediction model for each group using item's features. The set of prediction models is $M = \{M_0 = (\hat{\theta}_1, P_1), \ldots, M_K = (\hat{\theta}_k, P_K)\}$, where $K$ is the number of clusters, $\hat{\theta}_s$ is the consensual preference vector, and $P_s$ is the preference model extracted from $\hat{\theta}_s$, for $1 \leq s \leq K$.

In this scenario, a prediction model is a contextual preference model. Thus, each model $P_s$ in $M$ is designed as a *Bayesian Preference Network* (BPN) over a relational schema $RI(A_1, ..., A_t)$. A BPN is a pair $(G, \varphi)$ where $G$ is a directed acyclic graph in which each node is an attribute, and edges represent attribute dependency; $\varphi$ is a mapping that associates to each node of $G$ a set of conditional probabilities $\mathbb{P}[E_2|E_1]$ of the form of probability's rules: $A_1 = a_1 \wedge \ldots \wedge A_v = a_v \rightarrow B = b_1 > B = b_2$ where $A_1, \ldots, A_v$ and $B$ are item attributes. The left side of the rule (condition event $E_1$ in conditional probability) is called the context and the right side (condition event $E_2$ in conditional probability) is the preference on the values of the attribute $B$. This rule reads: *if the values of the attributes $A_1, \ldots, A_v$ are respectively $a_1, \ldots, a_v$ then for the attribute $B$ the value $b_1$ is preferred to $b_2$*. Please note that the preferences on $B$ depend on the values of the context attributes. A contextual preference model is able to compare items: given two items $i_1$ and $i_2$, the model can predict which is preferred.

The construction of a BPN comprehends in: (1) the construction of a network structure represented by the graph $G$ and (2) the computation of a set of parameters $\varphi$ representing the conditional probabilities of the model. The preference miner used in this work, CPrefMiner (AMO et al., 2013; AMO et al., 2015), uses a genetic algorithm in the first phase to discover dependencies among attributes and then, compute conditional probabilities using the Maximum Likelihood Principle (NIELSEN; JENSEN, 2009).

*Example Overview.* Considering the relational schema of movie attributes in Table 4 and the user-item rating matrix in Table 5 PrefRec clusters users, extracts preference consensual vector from $\hat{\theta}_1$ (Table 6) and builds the pairwise preference relation (Table 7). Then, CPrefMiner can build the BPN depicted in Figure 2. $PNet_1$ represents the contextual preference model that is used to compare the set of pairs of items and make the predictions.

*B) Recommendation.* In its second phase, PrefRec aims at using a prediction model $M_s$ to recommend items for a **cold user**. It is executed online, in contrast to the first phase which is offline. The recommendation process is executed according to the following steps:

1. Given a target user $u_x$ and a (small) set of ratings provided by $u_x$ over some items of $I$, the first task consists in obtaining the consensual preference vector $\hat{\theta}_s$ more similar to $u_x$'s preferences. We compute the similarity between $\theta_u$ (the $u_x$'s preference vector) and each consensual preference vector. Let $\hat{\theta}_s$ be the consensual preference vector, related to cluster $C_s$, the most similar to $\theta_u$.

2. Consider the preference model $P_s$ corresponding to $\hat{\theta}_s$. $P_s$ is used to infer the preference between pairs of items in $I$ which have not been rated by the user $u_x$ in the past.

3. From the set of pairs of items $(i_j, i_k)$ indicating that user $u_x$ prefers item $i_j$ to item $i_k$, a ranking is built by applying a ranking algorithm adapted from the algorithm Order By Preferences (COHEN; SCHAPIRE; SINGER, 1999). Thus, the output is a ranking $(i_1, i_2, \ldots, i_n)$ where an item $i_j$ is preferred or indifferent to an item $i_k$, for $j < k$ and $j, k \in \{1, ..., n\}$.

*Example*: To illustrate how a preference model is used in a recommendation phase, suppose that the preference vector $\theta_u$ of a **cold user** $u_x$ is most similar to the consensual preference vector of group $C_1$, $\hat{\theta}_1$. Let us consider the BPN $\mathbf{PNet}_1$ built over $\hat{\theta}_1$ and depicted in Figure 2. This BPN allows to infer a preference ordering on items over relational schema $RI(Decade, Director, Star, Genre)$ of data movie setting. For example, according to this ordering, item $i_5$ = (2010, Scorsese, Di Caprio, Thriller) is preferred than item $i_8$ = (2010, Spielberg, Tom Hanks, Drama). To conclude that, we execute the following steps:

1. Let $\Delta : I \times I \rightarrow \{A_i, ..., A_l\}$ be the set of attributes for which two items differ. In this example, $\Delta(i_5, i_8) = \{Director, Star, Genre\}$.

2. Let $\min(\Delta(i_5, i_8)) \subseteq \Delta(i_5, i_8)$ such that the attributes in $\min(\Delta(i_5, i_8))$ have no ancestors in $\Delta(i_5, i_8)$. According to the $\mathbf{PNet}_1$ structure, directed edge linking *Genre* and *Star* implies remove *Star*, therefore, in this example, $\min(\Delta(i_5, i_8)) = \{Director, Genre\}$. To have $i_5$ preferred rather than $i_8$ is necessary and sufficient that $i_5[Director] > i_8[Director]$ and $i_5[Genre] > i_8[Genre]$.

3. Computing the probabilities: $p_1 = \textit{probability that } i_5 > i_8 = \mathbb{P}[Scorsese > Spielberg] * \mathbb{P}[Thriller > Drama] = 0.8 * 0.66 = 0.53$; $p_3 = \textit{probability that } i_5 > i_8 = \mathbb{P}[Spielberg > Scorsese] * \mathbb{P}[Drama > Thriller] = 0.2 * 0.33 = 0.06$; $p_2 = \textit{probability that } i_8 \textit{ and } i_5 \textit{ are incomparable} = 1 - (p_1 + p_3) = 0.41$.

To compare $i_5$ and $i_8$ we focus only on $p_1$ and $p_3$ and select the highest one. In this example, $p_1 > p_3$ so that we infer that $i_5$ is preferred to $i_8$. If $p_1 = p_3$ was true, we would conclude that $i_5$ and $i_8$ are incomparable.

We can note that is not always necessary to rebuild the prediction model when a new item is add to the system. This action only have to be made when the system include items with attributes values different from the ones existents in the system.

## 3.3 Social PrefRec

Social PrefRec proposes a new approach to address the cold user problem through social information. It is a PrefRec framework extension, incorporating social information at recommendation phase. There were no modification on how models are built, but at recommendation phase we propose an alternative based on social information to recommend items for cold users.

In a simple way, a recommendation for cold users using social information could recommend items well rated by his direct friends. Another option is to leverage the connection weight among friends to provide better recommendations. The challenge here is to determine how much influence or similarities exist among user's relationship. Connect weight among users is computed through similarities on profiles (profession, age bracket, location, etc.), interaction between users (messaging, photos, etc.) and degree of influence.

To support this feature, we extended PrefRec and devise Social PrefRec. Figure 3 presents its new structure. To better understand it, let us consider the set of users $U$ and the set of items $I$ aforementioned in Section 3.2. The weight function $w : U \times I \rightarrow \mathbb{R}$ computes a *user preference degree* for an item and is represented by a rating $r_{u,i}$ from a user-item rating matrix $R$. To represent a social network, let $G = (V, E)$ be a social graph, and $u_x$ and $u_y$ vertices of this graph (users of a social network). A set of friends

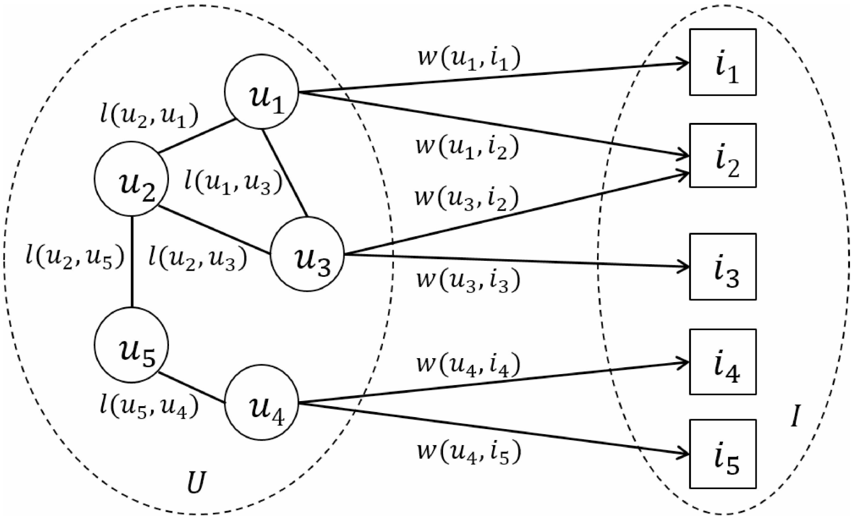


Figure 3 – SOCIAL PREFREC structure.

(neighbors) of a vertex $u_x$ is $F(u_x) = \{u_y | u_y \in V \wedge (u_x, u_y) \in E\}$ and a function $l : F \to \mathbb{R}$ defines *connection weight* between $u_x$ and $u_y$ in $[0, 1]$.

An illustrative example of a social graph in SOCIAL PREFREC is shown on Figure 4. Nodes represent users, and edges are friendship relations. Edges are labeled with the connection weights computed as explained in Section 3.3.2. Dashed groups are clusters of users, and each cluster is associated with a prediction model. Suppose that Paty is a cold user; so, there were no historical preferences associated to her. However, the system already clustered Paty's friends according to their preferences. As soon as Paty shows up, the connection weight is computed, and a suitable prediction model is selected.

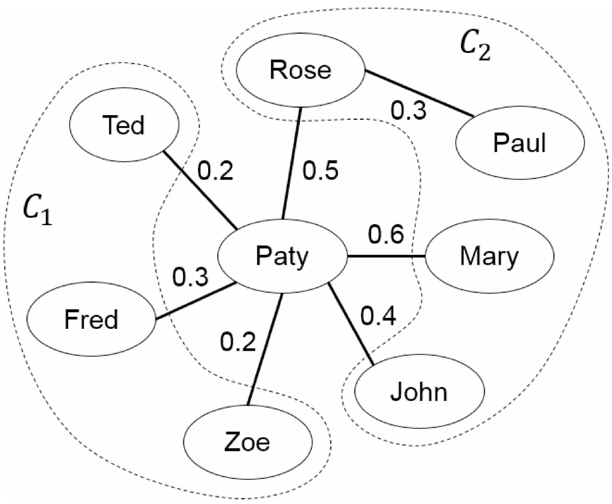


Figure 4 – Social network example.

### 3.3.1 The Framework

The general architecture of Social PrefRec, the interactions among the five modules, as well as their respective input and output are presented in Figure 5. Modules from 1 to 3 are from PrefRec, in module 4 we only add the cluster set changing the set of prediction model to $M = \{M_0 = (\hat{\theta}_1, C_1, P_1), \ldots, M_K = (\hat{\theta}_k, C_K, P_K)\}$. The cluster set is necessary to identify the set of users in each cluster in recommendation phase. At module 5 (Recommendation), Social PrefRec, unlike its predecessor, chooses proper prediction model using one social metric according to following steps.

1. Given a target user $u_x$ and a social metric, we will select $u_x$'s friends $F(u_x)$ and the related connection weight, previously computed as described in Section 3.3.2, between $u_x$ and each $u_y \in F(u_x)$.

2. Using one of the selection model methods (see Section 3.3.2), we will select the preference model $P_s$ corresponding to the cluster $C_s$ with more similar friends.

3. $P_s$ is used to infer the preference between pairs of items in $I$.

4. From the set of pairs of items $(i_j, i_k)$ indicating that user $u_x$ prefers item $i_j$ to item $i_k$, a ranking is built as mentioned in PrefRec approach.

Note that using this strategy, it is possible to recommend to a given user without taking into account any previous ratings, but relying on the user's relations in the cluster set.

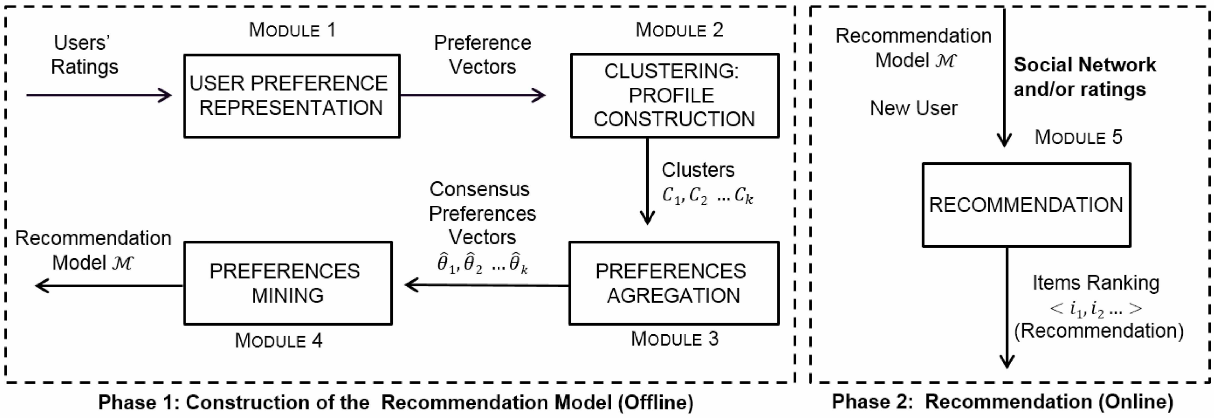


Figure 5 – Social PrefRec Framework.

### 3.3.2 Computation of connection weights and prediction model selection

Given the social graph $G$ for a target user $u_x$ and for each $u_y \in F(u_x)$, we compute user's connection weight according to the following metrics:

- *Friendship*: in this metric, connection weight is measured by $l(u_x, u_y) = 1$, where $1(\cdot)$ is the characteristic function (1 if argument is true, 0 otherwise).

- *Interaction level*: computed as $\frac{a(u_x,u_y)}{\widehat{a}(u_x)}$, where $a(u_x, u_y)$ is the number of times that user $u_y$ appears at $u_x$'s time-line, and $\widehat{a}(u_x)$ is the number of all occurrences of users $u_y$ at $u_x$'s time-line.

- *Mutual friends*: Represents the fraction of common friends or Jaccard similarity using $l(u_x, u_y) = \frac{F(u_x) \cap F(u_y)}{F(u_x) \cup F(u_y)}$.

- *Similarity score*: Given by demographic similarity between $u_x$ and $u_y$ according to function $l(u_x, u_y) = sims(u_x, u_y)$. We compute this value by the average of individual similarity in each demographic attribute (Age bracket, Sex, Religion, etc), using the binary function $similarity(u_x, u_y, A_i)$, wich returns 1 if attribute $A_i$ is similar for $u_x$ and $u_y$, 0 otherwise.

- *Centrality*: Calculated by average of closeness, betweenness and eigenvector centrality measures with $l(u_x, u_y) = centrality(u_y)$.

Social PrefRec allows the definition of any strategy to find a prediction model. To do so, Module 5 provides a function to select a prediction model. Let $select : U \to M$ be a function that selects the proper prediction model from $M$ for a target user $u_x$. In this work, Social PrefRec uses two strategies for prediction model selection based on connection weights: minimum threshold and average connection weight. Each strategy has a different type of implementation for function *select*, as explained in the following definitions:

- *Minimum threshold*: Let $\varepsilon \in [0, 1]$ be a minimum threshold for connection weight. The minimum threshold strategy selects the preference model $P_s$ (associated with model $M_s \in M$) which has more users who have a connection weight with the target user $u_x$ equal or above a minimum threshold according to Eq. (2).

$$select(u_x) = \arg\max_{M_s \in M} |\{u_y \in F(u_x) \wedge l(u_x, u_y) \geq \varepsilon\}| \tag{2}$$

- *Average*: The average strategy selects the preference model $P_s$ with users who have the highest average connection weight with the target user $u_x$ according to Eq. (3).

$$select(u_x) = \arg\max_{M_s \in M} \frac{1}{|F(u_x)|} \sum_{(u_x, u_y) \in F(u_x)} l(u_x, u_y) \tag{3}$$

## 3.4 Experimental Settings

### 3.4.1 Datasets

Table 8 summarizes our datasets. Recall that sparsity is the percent of empty ratings in user-item rating matrix and links are the number of users connections in the dataset. The particularities of each dataset is described next:

- *Facebook Dataset.* We surveyed this dataset through a Facebook web application we developed for this purpose. With volunteers permission, we crawled relationship status, age bracket, gender, born-in, lives-in, religion, study-in, last 25 posts in user's time-line, posts shared and posts' likes, and movies rated before on the Facebook platform. In addition, we asked each volunteer to rate 169 Oscar nominated movies on a 1 to 5 stars scale. We got data from 720 users and 1,454 movies, resulting in 56,903 ratings.

  In our experiments, we consider only ratings from the 169 Oscar nominated movies, which represent movies rated by most users. We split Facebook data into two datasets, FB50 and FB100, to represent the set of users who rated at least 50 and 100 movies, respectively. This was done to evaluate the overall system performance under datasets with different sparsity and social information levels. The movie's attributes are: genres, directors, actors, year, languages and countries. In FB50 and FB100, we compute user similarity metric using the attributes: relationship status, age bracket, gender, born-in, lives-in, religion and study-in. We also compute the interaction level considering the last 25 posts in the user time-line, posts shared and likes.

- *Flixster Dataset.* Jamali and Ester (2010) published this dataset. However, movie information was restricted to its title, then we improved it by adding genres, directors, actors, year, languages and countries information retrieved from IMDB.com public data. We also use two datasets from Flixster with different sparsity level, Flixster 175K and Flixster 811K. Flixster social information includes friend's relationships, mutual friends, friends centrality and users similarities. Similarity between users is computed only through three attributes: gender, age bracket and location. Interaction information is not available on Flixster dataset.

### 3.4.2 Comparison Methods

In our experiments we compare SOCIAL PREFREC with PREFREC and three social matrix factorization based recommender systems. The idea is to evaluate SOCIAL PREFREC recommendations compared to PREFREC. Note that the former chooses the prediction model using only social information whereas the latter needs user's first ratings to choose

Table 8 – Dataset features.

| *Dataset* | Users | Items | Ratings | Sparsity (%) | Rates/User (Average) | Links | Links/User (Average) |
|---|---|---|---|---|---|---|---|
| $FB50$ | 361 | 169 | 44.925 | 26.36 | 124.44 | 2,926 | 8.6 |
| $FB100$ | 230 | 169 | 35,459 | 8.77 | 154.16 | 1,330 | 6.4 |
| *Flixster* $175K$ | 357 | 625 | 175,523 | 26.36 | 491 | 706 | 2.8 |
| *Flixster* $811K$ | 1,323 | 1,175 | 811,726 | 47.78 | 613.54 | 6,526 | 5.34 |

a model. Further, the comparison with matrix factorization methods is used to evaluate SOCIAL PREFREC compared to other social approaches, which handle cold start users.

Social matrix factorization methods combine social information with rating data. They are distinct from SOCIAL PREFREC that uses social information only to choose a consensual prediction model between preference clusters. In addition, our method has its prediction model based on pairwise preferences. The three social matrix factorization methods do not make use of any clustering technique. We take these systems as comparison methods because they achieve high accuracy levels for cold start user as reported by the authors. The social matrix factorization particularities are reported next:

- *SoRec* (MA et al., 2008) is based on latent factors of items, users, and social network relationship. The influence of one neighbor on the prediction of a rating increases if he is trusted by a lot of users while it decreases if the target user has many connections.

- *SocialMF* (JAMALI; ESTER, 2010) applies a trust propagation mechanism. More distant users have less influence (weight) in rating prediction than the trust direct contacts.

- *TrustMF* (YANG et al., 2013) represents the influence of connections to target user preferences in two ways: truster and trustee. This approach provides recommendations to users that usually show influence on others and those who are typically influenced by others.

### 3.4.3 Experimental Protocol

Each experiment was performed on the datasets split into two parts: training set and test set.

PREFREC and SOCIAL PREFREC build clusters (K-Means clustering) of similar users using the training set. For each cluster $C_s$ the systems associate a prediction model $M_s$. Then, to recommend items for a given user $u_x$, it is necessary to select the most similar model (cluster) that fits $u_x$. This process is done during the test phase. However, those approaches take different directions. Since PrefRec is not able to deal with social information, it relies on previous ratings of $u_x$ to select its best prediction model. In contrast, as Social PrefRec requires social information to accomplish this task. We employ the leave-one-out protocol (SAMMUT; WEBB, 2010) to better validate our tests and simulate a realistic cold start scenario. Thus, for each test iteration, one user is taken for test purpose, and the training set is made of all other users. Each experiment is composed by $n$ iterations, where $n$ is the number of users. Importantly, because PrefRec cannot act in a full cold start scenario, we give PrefRec a few ratings to bootstrap the system.

PrefRec *protocol.* The PrefRec prediction model is built offline. For the test phase $y$ ratings of the current test user $u_x$ chosen at random were considered for the choice of the most similar cluster $C_s$. Then, computing the similarity between the preference vector of $u_x$, $\theta_{u_x}$, and the consensual preference vector of $C_s$, $\hat{\theta}_s$ is a matter of computing the *Euclidian distance* between these two vectors weighted by the number of common ratings $(z)$, where $d_E(\theta_{u_x}, \hat{\theta}_i) = \frac{1}{z}\sqrt{\sum_{k=1}^{z} (\theta_{u_x,i_k} - \hat{\theta}_{i,i_k})^2}$. Please note that this similarity distance was used for preferences clustering (training) and selection models (test) phases. Finally, for validation purpose, the remaining ratings of the current test user $u_x$ were used.

Social PrefRec *protocol.* Building the prediction model is done as in PrefRec. However, during the test phase, we do not take any rating into account. Social PrefRec requires solely social information to find the most similar cluster, $C_s$, according to a given social metric and a model selection strategy.

*Matrix Factorization social approaches protocol.* For SoRec, SocialMF and TrustMF the experimental protocol builds a model $M_x$ for each user $u_x$ using friendship preferences information which includes all friend's item ratings. In contrast to previous protocols, the prediction model, $M_x$, is not a clustered preference model, but a specific preference model for each user.

### Parameter Settings

In our experiments, we use LibRec (GUO et al., 2015) which contains an implementation of SoRec, SocialMF and TrustMF methods with default parameters. We executed Matrix factorization approaches with 10 latent factors and the number of interactions set to 100. We use K-means as the clustering algorithm for PrefRec and Social PrefRec. In addition, we experimentally test several numbers of clusters. Then we set the optimal number of clusters for each dataset: 7 for FB50, 6 for FB100, 4 for Flixster 175K, and 2 for Flixter 811K. The minimum threshold $\epsilon$ has optimal values equal to 0.4 for FB50 and FB100, and 0.1 for Flixster 175K and Flixter 811K. However, we executed experiments related with Q5 and Q6, over FB50 and FB100 with $\epsilon = 0.1$ to have more users in the result set to evaluate these two questions.

### 3.4.4 Evaluation methods

Regarding our evaluation method, we present results using two metrics: (1) ***nDCG*** is a standard ranking quality metric to evaluate the ability of the recommender to rank the list of top-k items (SHANI; GUNAWARDANA, 2011). (2) We also compute the standard $F_1$ ***score***, based on precision and recall, to evaluate the prediction quality of pairwise preferences (AMO; OLIVEIRA, 2014).

In the ***nDCG*** Eq. (4), $r_{u,1}$ is the rating (according to the ground truth) of the item at the first ranking position. Accordingly, $r_{u,j}$ is the ground truth rating for the item ranked in position $j$. $M$ is the number of ranked items. $DCG(u)$ is the discounted cumulative gain of predict ranking for a target user $u$, $DCG^*(u)$ is the ground truth and N is the number of users in the result set.

$$DCG(u) = r_{u,1} + \sum_{j=2}^{M} \frac{r_{u,j}}{\log_2 j}, nDCG = \frac{1}{N} \sum_{u} \frac{DCG(u)}{DCG^*(u)} \tag{4}$$

Precision and recall were combined using $F_1$ score (Eq. (5)). The precision of a user $u$ is the percentage of good predictions among all the predictions made for user $u$. The recall is the percentage of good predictions among the amount of pairs of items in the current iteration. Final precision and recall of the test set are obtained by considering the harmonic mean of average precision and average recall of each user.

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{5}$$

Besides those metrics, we further analyze how the user ratings profile length and the number of friends impact the recommendation quality through two other metrics: (1) profile length factor and (2) social degree:

*Profile length Factor.* Let $\bar{R}$ be an average number of user ratings and an $\alpha$ coefficient, where $\alpha \in \mathbb{R}$. Eq. (6) represents the profile length factor calculus. In our experiments (Figure 8a) we compute the $F_1$ score for different profile length factors to determine the number of ratings necessary to better select a prediction model for a given dataset.

$$Pl_{factor} = \alpha \bar{R} \tag{6}$$

*Social Degree.* The social degree is given by the average degree of the social network ($\bar{S}$) and a $\beta$ coefficient where $\beta \in \mathbb{R}$. We compute the social degree according to Eq. (7). Using different number of friends to select a prediction model we evaluate the $F_1$ results.

$$S_d = \beta \bar{S} \tag{7}$$

## 3.5 Result and Discussion

In this section, we thoroughly assess the effectiveness of our proposed pairwise preference recommender approach, Social PrefRec. First, we analyze the quality of recommendations on the datasets (Q1). Then, we measure the relevance of recommendations (Q2), focusing on the ranking relevance of Social PrefRec compared to those provided by three social recommender systems, besides the original PrefRec. Furthermore, we measure the performance for each social metric (Q3) and under different sparsity levels (Q4). We close this section by analyzing how user's profile length versus its social degree (Q5) and popular versus unpopular users (Q6) influence the quality of the recommendations.

### 3.5.1 How accurately social information help on pairwise preference recommendation? (Q1)

$F_1$ scores are represented in Figure 6, for minimum threshold and average connection weight selection model strategies. Against all datasets with a profile length of 30-ratings for PrefRec versus 0-ratings for Social PrefRec, the social approach achieved better results using Minimum threshold strategy. Rate-15-items baseline is widely used to bootstrap traditional recommender systems (CHANG; HARPER; TERVEEN, 2015). Thus, to make a fair comparison we give 30-ratings for PrefRec, which means that all runs have a good safe margin and should not harm its performance. Nevertheless, our social approach performs at least equivalently to traditional one, as we further discuss on Q3. Note that those results are on cold start scenarios: under scenarios where a user provides enough ratings, a social approach does not add much value.

### 3.5.2 How relevant are the recommendations made by a social pairwise preference recommender? (Q2)

Tables 9, 10, 11, and 12 show the $nDCG$ results for rank size 5, 10, 15, 20, under minimum threshold strategy and 0-rating scenario. We apply each approach described in Section 3.4.3 on each dataset to assess the robustness of each. We observe that Social PrefRec obtains better results for cold users compared to the other social recommenders. One of the main reasons for the effective performance of our approach is that it chooses a suitable prediction model based on a consensual set of friends' preferences. The other approaches not only consider all friends' preferences, but SocialMF for example, also relies on trust propagation mechanism, which incorporates preferences from friends of friends. Thus, we argue that a specific set of friends (neighbors) might be a better source to give more relevant recommendations.

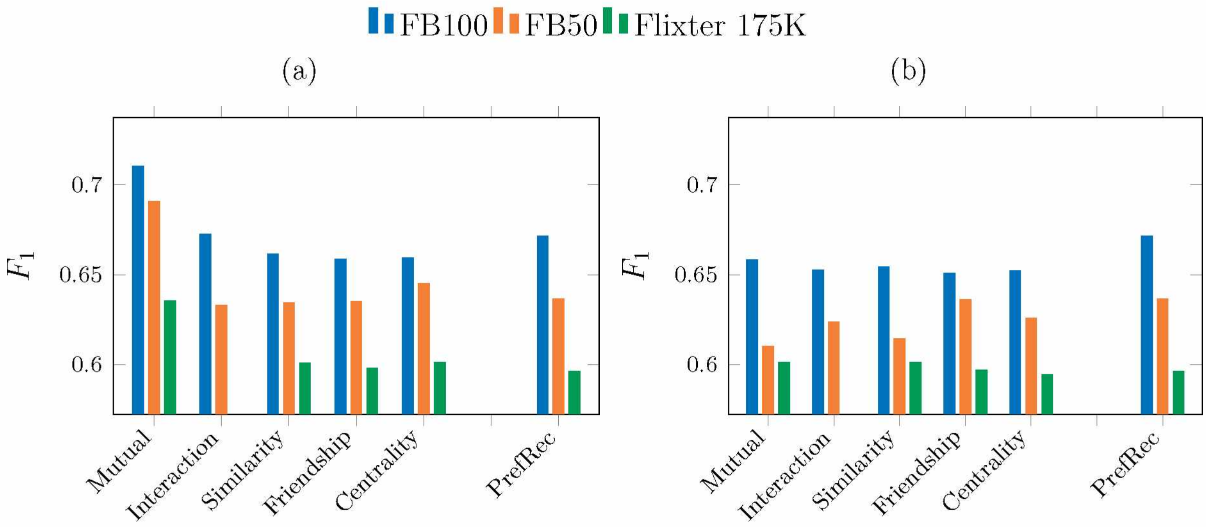


Figure 6 – $F_1$ scores for Social PrefRec and PrefRec for 2 model selection strategies: (a) Minimum threshold, (b) Average connection weight.

Another main difference is about how each approach deals with item attributes. Matrix factorization profiles both users and items in a user-item rating matrix and through latent factor models project items and users into the same latent space, thus making them comparable.

According to a Kruskal-Wallis test with 95% confidence, Social PrefRec performance is significantly better than social matrix factorization approaches. Mutual Friends is better than others Social PrefRec metrics in nDCG@5. For nDCG@10, there is no significant difference between Mutual Friends, Centrality, Friendship, Similarity and Interaction. The performance with Centrality achieves an equivalent score as Mutual Friends in nDCG@15 results. Finally, the nDCG@20 values show that Mutual Friends, Centrality, Friendship and Similarity are not significantly different.

Table 9 – Resulting nDCG@5, @10, @15, and @20 against FB50.

| Approach | Size of Rank | | | |
|---|---|---|---|---|
| | @5 | @10 | @15 | @20 |
| SoRec | 0.8515 ± .138 | 0.8412 ± .123 | 0.8340 ± .114 | 0.8297 ± .108 |
| SocialMF | 0.7469 ± .183 | 0.7536 ± .158 | 0.7550 ± .146 | 0.7576 ± .139 |
| TrustMF | 0.8373 ± .147 | 0.8296 ± .133 | 0.8259 ± .122 | 0.8250 ± .114 |
| Friendship | 0.9870 ± .035 | 0.9779 ± .039 | 0.9697 ± .040 | 0.9612 ± .042 |
| Similarity | 0.9860 ± .036 | 0.9770 ± .040 | 0.9683 ± .042 | 0.9601 ± .045 |
| Centrality | 0.9881 ± .033 | 0.9802 ± .038 | 0.9721 ± .039 | 0.9647 ± .041 |
| Mutual | **0.9934** ± .025 | **0.9890** ± .028 | **0.9752** ± .033 | **0.9665** ± .038 |
| Interaction | 0.9822 ± .043 | 0.9733 ± .046 | 0.9661 ± .047 | 0.9589 ± .046 |

Table 10 – Resulting nDCG@5, @10, @15, and @20 against FB100.

| Approach | Size of Rank | | | |
|---|---|---|---|---|
| | @5 | @10 | @15 | @20 |
| SoRec | 0.8358 ± .141 | 0.8251 ± .124 | 0.8180 ± .119 | 0.8114 ± .115 |
| SocialMF | 0.7124 ± .193 | 0.7100 ± .173 | 0.7111 ± .163 | 0.7166 ± .155 |
| TrustMF | 0.7742 ± .149 | 0.7819 ± .128 | 0.7835 ± .120 | 0.7804 ± .115 |
| Friendship | 0.9852± .036 | 0.9746 ± .042 | 0.9666 ± .044 | 0.9582 ± .046 |
| Similarity | 0.9850 ± .038 | 0.9746 ± .042 | 0.9667 ± .043 | 0.9587 ± .046 |
| Centrality | 0.9897 ± .028 | 0.9797 ± .037 | 0.9706 ± .041 | 0.9621 ± .044 |
| Mutual | **0.9933** ± .023 | **0.9836** ± .027 | **0.9715** ± .037 | **0.9636** ± .042 |
| Interaction | 0.9762 ± .053 | 0.9762 ± .060 | 0.9603 ± .061 | 0.9547 ± .061 |

Table 11 – Resulting nDCG@5, @10, @15, and @20 against Flixter 175K.

| Approach | Size of Rank | | | |
|---|---|---|---|---|
| | @5 | @10 | @15 | @20 |
| SoRec | 0.8209 ± .134 | 0.8236 ± .120 | 0.8224 ± .115 | 0.8214 ± .111 |
| SocialMF | 0.7715 ± .138 | 0.7753 ± .126 | 0.7755 ± .123 | 0.7751 ± .120 |
| TrustMF | 0.7603 ± .136 | 0.7521 ± .127 | 0.7494 ± .123 | 0.7485 ± .120 |
| Frienship | 0.9840 ± .039 | 0.9769 ± .038 | 0.9713 ± .038 | 0.9671 ± .039 |
| Similarity | 0.9852 ± .038 | 0.9779 ± .037 | 0.9726 ± .037 | 0.9675 ± .039 |
| Centrality | 0.9830 ± .039 | 0.9758 ± .039 | 0.9704 ± .038 | 0.9657 ± .040 |
| Mutual | **0.9916** ± .023 | **0.9810** ± .030 | **0.9772** ± .032 | **0.9766** ± .030 |

Table 12 – Resulting nDCG@5, @10, @15, and @20 against Flixter 811K.

| Approach | Size of Rank | | | |
|---|---|---|---|---|
| | @5 | @10 | @15 | @20 |
| SoRec | 0.8198 ± .131 | 0.8173 ± .118 | 0.8145 ± .113 | 0.8133 ± .109 |
| SocialMF | 0.7279 ± .139 | 0.7335 ± .122 | 0.7359 ± .116 | 0.7374 ± .113 |
| TrustMF | 0.7204 ± .135 | 0.7246 ± .122 | 0.7246 ± .117 | 0.7298 ± .113 |
| Frienship | 0.9810 ± .044 | 0.9748 ± .044 | 0.9699 ± .043 | 0.9662 ± .042 |
| Similarity | 0.9804 ± .045 | 0.9744 ± .044 | 0.9696 ± .044 | 0.9661 ± .043 |
| Centrality | 0.9809 ± .044 | 0.9742 ± .044 | 0.9699 ± .043 | 0.9667 ± .042 |
| Mutual | **0.9908** ± .029 | **0.9812** ± .036 | **0.9747** ± .038 | **0.9685** ± .041 |

### 3.5.3 Which social metrics are more important for item recommendation? (Q3)

We perform Kruskal-Wallis test to check statistical significance among Social PrefRec metrics results and PrefRec, see Figure 6. Mutual Friends, Interaction, Similarity are indicated as best performing. Furthermore, Friendship and Centrality results are not significantly different from PrefRec (profile length = 30-ratings) result. Thus, the test shows with 95% confidence, that with the first three metrics we can better recommend in social 0-rating profile scenario than 30-rating profile in a traditional recommender approach. Although the others social metrics achieved the same result as the traditional approach, they do not use previous rating from a user.

### 3.5.4 How effective is Social PrefRec to mitigate data sparsity problems? (Q4)

As sparsity is a big challenge faced by recommendation systems, we consider five subsets sampled from FB100. The basic idea is to simulate sparse scenarios where input datasets has many items to be rated with very few/sparse ratings per user. For instance, $FB100_{50}$ was obtained by eliminating around 50% of the ratings in FB100 in a stratified way, so we keep homogeneous subgroups of the original set. Table 13 shows the characteristics of the datasets extracted from FB100.

Table 13 – FB100 sparse subsets.

| $FB100$ ($Dataset$) | Ratings per user (Average) | Sparsity (%) |
|---|---|---|
| 10 | 137.9 | 18.4 |
| 20 | 122.6 | 27.4 |
| 30 | 107.28 | 36.6 |
| 40 | 91.8 | 45.7 |
| 50 | 76.3 | 54.8 |

Figure 7 shows that PrefRec is superior on less sparse datasets. However, the social approaches on sparser dataset, i.e. $FB100_{50}$ and $FB100_{40}$, exhibit better recommendations quality, particularly for Mutual connection weight metric. These results complement previous analyses of Social PrefRec.

### 3.5.5 Does social degree affect Social PrefRec as much as profile length affects PrefRec? (Q5)

Traditional recommender systems present better performance when they know more user's preferences. Figure 8a shows the prediction performance of PrefRec on two Face-

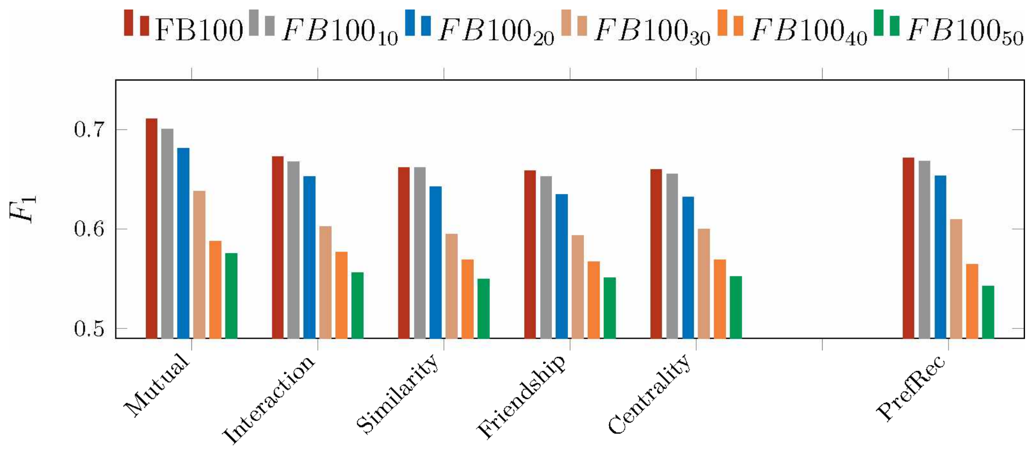


Figure 7 – Social PrefRec and PrefRec metrics across sparse scenarios with *minimum threshold* of 40%.

book datasets. We observe that the recommender predictions get better as the user's profile gets longer. For instance, PrefRec achieves $F_1$ equal to 71.18 on FB100 when we use 123 ratings for prediction model selection ($\alpha = 0.8$).

However, with Social PrefRec, we do not note a correlation between social degree and prediction performance. Figures 8b to 8f show the results for different social degrees. The overall picture is the same on all datasets and all social metrics. So, increasing the number of friends to select a prediction model do not increase the $F_1$ score. This leads us to the next question that further evaluates all social metrics for higher and lower social degrees.

### 3.5.6 Are there major differences between the quality of recommendations considering popular and unpopular users? (Q6)

To investigate the effects of social degree on Social PrefRec, we begin by recalling the definition of popular and unpopular users. First, we calculate the average number of friends on the subset $FB50$ and $FB100$. Popular users are those that have more than the average number of friends, whereas unpopular users have only half the average number of friends.

Figure 9 shows the ($F_1$) achieved by Social PrefRec for each social metric against each subset. Note that, the overall performance is similar between each subset. Regarding the major differences between popular and unpopular users, the mutual friends social metric achieves the worst results, which shows the need of larger amounts of friends to better select a prediction model. On the other hand, the centrality social metric performance shows that it is not affected by the number of friends.

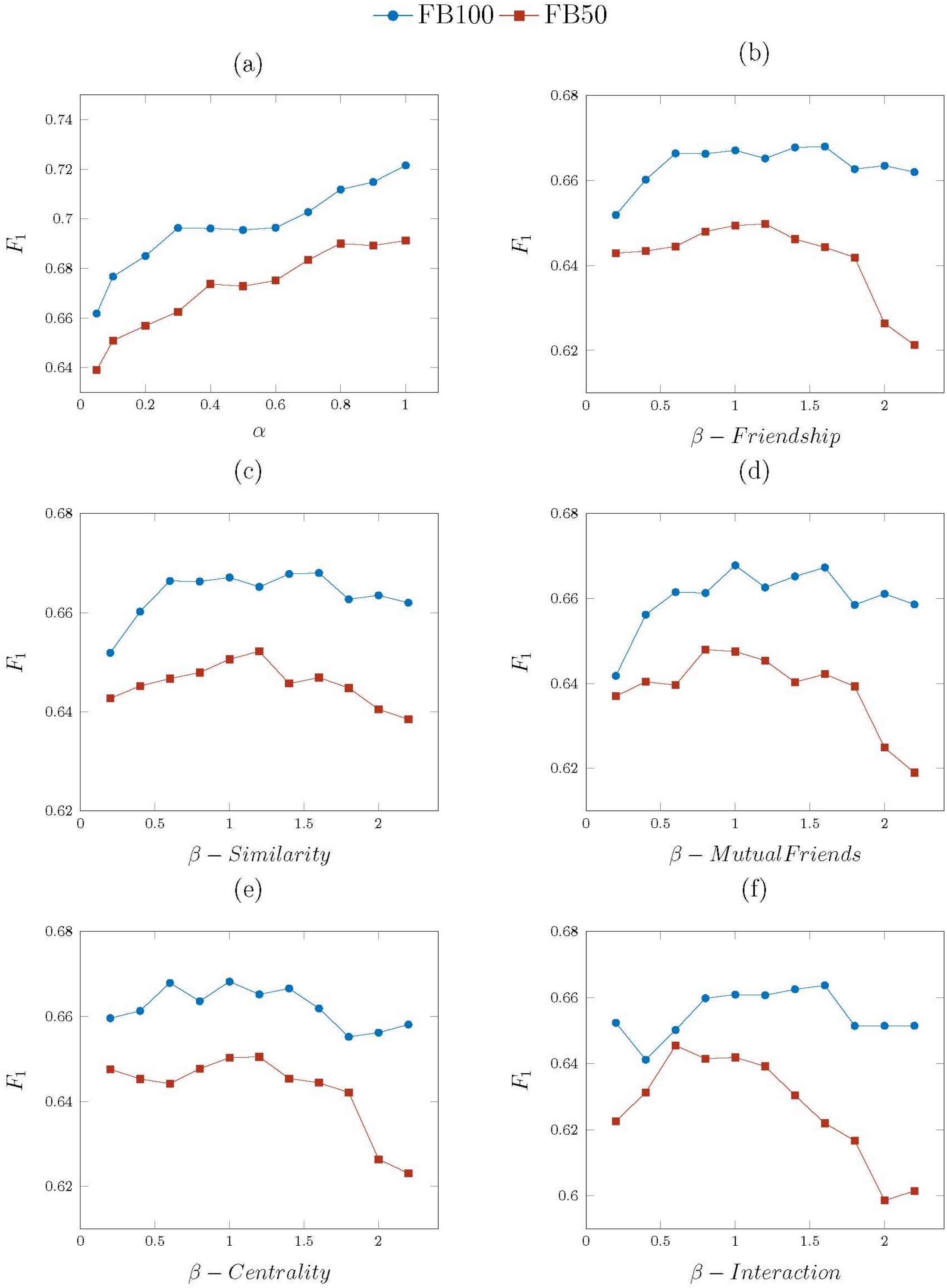


Figure 8 – Profile length factor effect ($\alpha$, see Eq. 6) over $F_1$ measure (PrefRec) in Figure 8a. Social degree effect variants ($\beta$, see Eq. 7) over $F_1$ measure in Figure 8a – 8f.

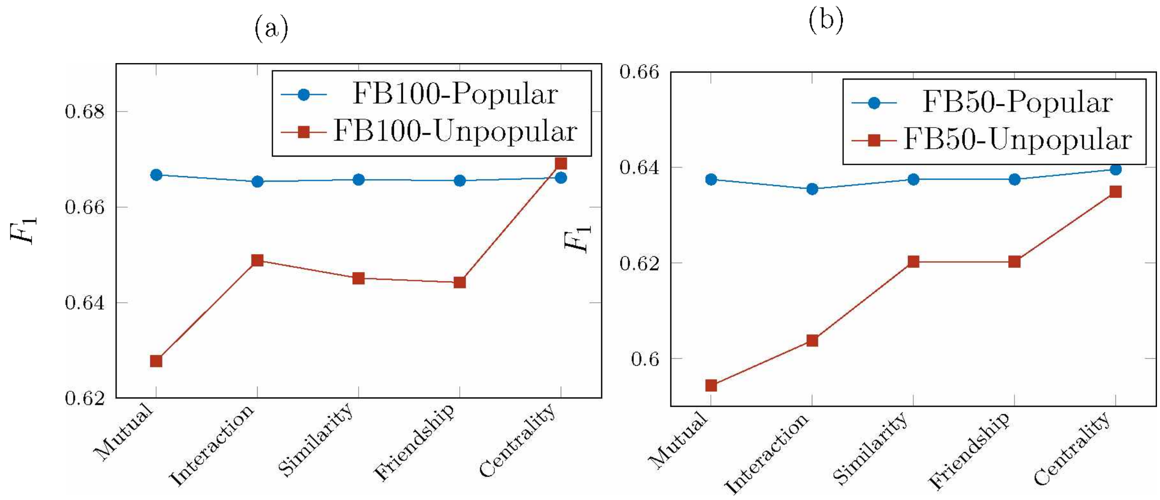


Figure 9 – (a) $F_1$ metric for Popular users and Unpopular users in FB100 and (b) FB50 with *minimum threshold* of 10%.

## 3.6 Summary

We have devised and evaluated Social PrefRec, an approach whose goal is to help pairwise preferences recommender systems to deal with 0-rating user's profile. Driven by six research questions, we expand earlier work by analyzing and demonstrating the effectiveness of our proposed social preference learning approach. Our analyses were performed on four real datasets. We also carefully investigate the role of five well-known social metrics in pairwise preference recommendation and proposed a clustering based approach to incorporate social networks into recommender systems. With Social PrefRec approach, we can bring novel ways to extend traditional recommenders.

Finally, although focused on social networks, our work could be extended to tackle other networks (graphs) where we can compute similarity scores between nodes, such as scientific networks or inferred networks (that will be discuss in the next chapter). Another interesting direction for future work is the study of how to choose more influential nodes, *e.g.* find out the friends who have a stronger influence on a user and apply their preferences to tackle cold start recommendations.

Chapter 4

# Improving Recommendation with Visual Perception Similarities[1]

## 4.1 Introduction

Recommender systems are in our everyday life. We are usually asked to make choices without enough personal experience of the alternatives. So, we rely on others' recommendations and that is why RS have become ubiquitous nowadays. To do recommendations, those systems exploit users' previous choices and predict new products that would fulfill users' expectations. In the same way as Chapter 3, we are also interested in dealing with low accuracy levels of recommendation and poor cold users experiences.

Reliable user cold-start solutions do exist. The standard path is to infer additional information of the cold user to work around cold-start problem. As additional information we can mention social information (MA et al., 2008), user click behavior (LIU; DOLAN; PEDERSEN, 2010), location-based information (CHENG et al., 2012b) and, more recently, user visual perception (MELO; NOGUEIRA; GULIATO, 2015). In fact, tracking users eyes movements to capture their attention became an important source of knowledge with the accessibility to emerging technologies like smartphones cameras or eye tracking devices (SUGANO; ZHANG; BULLING, 2016).

Melo et al. (MELO; NOGUEIRA; GULIATO, 2015) proposed a content-based image recommendation approach applied to clothing shopping. Their approach uses items' ratings combined with users' visual attention. The goal is to recommend clothes similar to clothes already well rated by a user. Similarity among clothes is given by a measure calculated from visual attention similarity between them. Such approach achieves reasonable accuracy levels, but it does not deal with user cold-start problem.

The intuition for the approach that will be present in this Chapter is that users with similar visual perceptions have similar tastes. For instance, Figure 10 shows a painting

[1] This chapter was published in a modified form in (FELÍCIO et al., 2016c)

containing two main scenes: a cat and a dog [2]. Some people looking at the painting might focus their attention to the cat. Others, to the dog. We can have two distinct groups of users. Thus, we explore users similarities within a single group to recommend items.

Figure 10 – Painting of a Dog and Cat. Some people might focus their attention to the cat, but others to the dog.

In this chapter, we combining *user visual perception* with prediction models of *pairwise preferences*. In Chapter 3 we presented a pairwise preference recommender systems, where pairwise preference is a specific type of opinion that establishes an order relation between two objects. For example, when a user says: "I prefer surrealism than cubism", we clearly identify his preference to paintings of the surrealism movement over cubism.

Our new approach, called VP-Rec, uses visual perception to recommend images in a pairwise preference fashion. Therefore, it takes the advantages aforementioned, besides been a hybrid recommender systems. Instead of using only historical ratings, items features are applied to create the prediction model and visual perception is used to define the items recommendation. The hypothesis is that *matching new people with existing people that present similar visual perceptions might help on providing accurate recommendations for cold-start users*. We address this by investigating three research questions:

**RQ1:** How effective is VP-Rec for cold-start user?

**RQ2:** How is the performance of VP-Rec under data sparsity?

**RQ3:** What is the performance comparison of matrix factorization approaches on users with observed ratings versus VP-Rec?

We compare our approach with four state-of-art social recommender system in terms of *nDCG* metric. Our results show that VP-Rec increases up to 90% the ranking quality compared to those systems.

This chapter is organized as follows. Section 4.2 presents the background knowledge undertaking in this approach. Section 4.3 describes our proposed framework the VP Framework. Section 4.4 describes our experimental settings and Section 4.5 presents the results. Finally, Section 4.6 concludes the chapter.

[2] Oil Painting of a Dog and Cat, available at http://www.dailypainters.com/paintings/138359/Oil-Painting-of-a-Dog-and-Cat/Nancy-Spielman

## 4.2 Background

In this section, we introduce the main concepts underlying VP-Rec. To enhance readability, we give an illustrative example along with the problem formalism. The focus is how is computed the visual perception similarity between users.

**Input and Output.** Let $\mathcal{I} = \{I_1, ..., I_m\}$ be a set of images, and $\mathcal{U} = \{u_1, ..., u_n\}$ be a set of users. Let $RI(A_1, ..., A_p)$ be a relational scheme related to images. The user-item rating matrix is represented by $\mathcal{R} = [r_{u,I}]_{m\times n}$, where each entry $r_{u,I}$ represents the rating given by user $u$ on item image $I \in \mathcal{I}$.

As we see on Section 3.2, pairwise preference recommender systems predict the preference between a pair of items with missing values in the user-item rating matrix. On the other hand, in traditional recommender systems, the recommendation task is based on the predictions of the missing values in the user-item rating matrix.

To adopt visual perception as additional information for recommendation systems, first, we rely on the *VP-Similarity Method* (FELÍCIO et al., 2016b; ALMEIDA, 2016). This method infers visual perception similarities among users.

We consider the use of Eye tracker devices, that capture information over user's visualization behavior (gaze positions, duration, sequence). We concentrate our definitions on gaze position and fixation length (length of time that visual attention lasts).

**Definition 1** (Visual Fixation)**.** *A visual fixation of a user $u_t$ over an image $\mathcal{I}_k$ is a pair $(p, f)$ where p is the position, represented by the pixels cluster centroid of that fixation, and $f$ is the duration. We denominate $\mathcal{F}_{tk} = \{(p_1, f_1), ..., (p_z, f_z)\}$ the set of visual fixations of $u_t$ over $\mathcal{I}_k$ (Fig. 11a).*

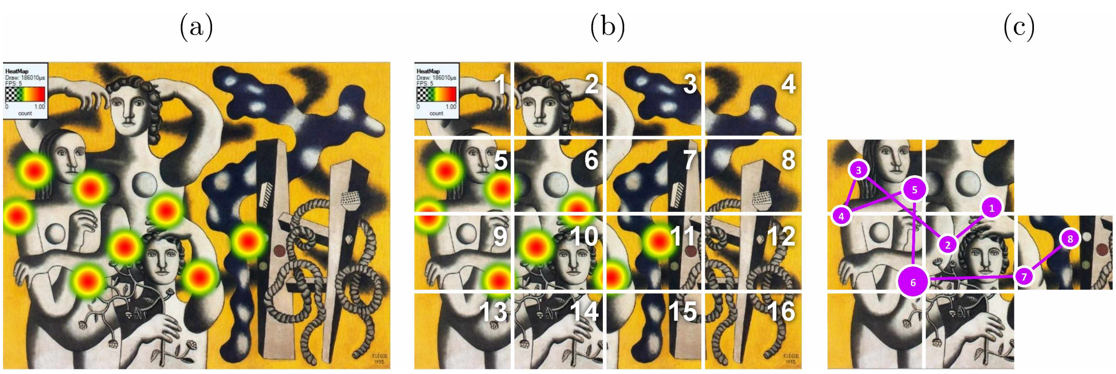


Figure 11 – 11a Gaze positions and fixation length captured. 11b Painting splits in sixteen equal parts. 11c Image parts with nonzero fixation length.

**Definition 2** (Visual Perception)**.** *Let the images in $\mathcal{I}$ be divided in $r$ equal parts $Q = \{q_1, ..., q_r\}$ as illustrated in Fig. 11b. From the positions and durations described in the set of visual fixations $\mathcal{F}_{tk}$, we call $v_s$ the percentage of time that $u_t$ fixed to $\mathcal{I}_k$ in each* *part $q_s$, for $1 \leq s \leq r$ (Fig. 11c). The visual perception of a user $u_t$ over an image $\mathcal{I}_k$ is defined as the vector $\mathcal{P}_{tk} = (v_1, ..., v_r)$. Finally, the visual perception of $u_t$ over all images $\mathcal{I}$ is represented by the concatenation of all visual perceptions vectors from $u_t$: $\mathcal{P}_t = \mathcal{P}_{t1} \parallel ... \parallel \mathcal{P}_{tx}$. We denote by $\mathcal{P}$ the set of all users' visual perception vectors.*

An example of visual perception can be seen in Table 14. There are visual perceptions from 6 users over 2 images. Images are divided in 4 equal parts. For each user and each image, we have the percentage of time a given user fixed his visual attention in a corresponding part.

Table 14 – Users' visual perception over two images of paintings dataset.

| | $\mathcal{I}_1$ | | | | $\mathcal{I}_2$ | | | |
|---|---|---|---|---|---|---|---|---|
| | $q_1$ | $q_2$ | $q_3$ | $q_4$ | $q_1$ | $q_2$ | $q_3$ | $q_4$ |
| $u_1$ | 0.50 | 0.10 | 0.40 | 0.00 | * | * | * | * |
| $u_2$ | 0.60 | 0.20 | 0.10 | 0.10 | 0.10 | 0.70 | 0.10 | 0.10 |
| $u_3$ | 0.40 | 0.40 | 0.20 | 0.00 | 0.00 | 0.90 | 0.10 | 0.00 |
| $\hat{\mathcal{P}}_1$ | **0.50** | **0.23** | **0.23** | **0.03** | **0.05** | **0.80** | **0.10** | **0.05** |
| $u_4$ | * | * | * | * | 0.75 | 0.08 | 0.05 | 0.12 |
| $u_5$ | 0.05 | 0.25 | 0.20 | 0.50 | 0.70 | 0.05 | 0.10 | 0.15 |
| $u_6$ | 0.15 | 0.20 | 0.25 | 0.40 | 0.82 | 0.02 | 0.10 | 0.06 |
| $\hat{\mathcal{P}}_2$ | **0.10** | **0.23** | **0.23** | **0.45** | **0.76** | **0.05** | **0.08** | **0.11** |

*VP-similarity score* is computed between two users $u_1$ and $u_2$ as the distance between their respective visual perceptions vectors $\mathcal{P}_1$ and $\mathcal{P}_2$. This distance is defined by the function $l(u_1, u_2)$, where $l : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}$ and $l(u_1, u_2)$ can assume any classic similarity function like Euclidean distance, cosine similarity or Pearson distance correlation. By abuse of notation, we will write $l(u_1, u_2)$ as $l_{1,2}$. For example on Table 14, we have that the VP-similarity score between $u_4$ and $u_5$, considering $l$ as cosine similarity is 0.76 (* has been assumed as 0).

As we hypothesize that users with similar visual perceptions are a good source for cold user recommendation, we propose to cluster users according to their VP-similarity scores. In this approach, we use *K-means* as classical clustering algorithm, and refer to visual perception clusters as VP-clusters. This process is shown in the left side of Fig. 12.

We define as *cluster consensual vector* the vector containing the averages of all visual perceptions from users inside the same VP-cluster. Table 14 illustrates two VP-clusters and their respective consensual vectors $\hat{\mathcal{P}}_1$ and $\hat{\mathcal{P}}_2$ . This notion is specially important on recommendation phase: when a target user $u_t$ is added to the system, some visual perception of him is collected. Our VP-Similarity method generates the visual perception vector $\mathcal{P}_t$ of $u_t$, and a VP-similarity score between $u_t$ and each VP-cluster $C_j$ is computed. We denote $\delta_{t,k}$ as the VP-similarity score between a user $u_t$ and a VP-cluster $C_j$ (right side of Fig. 12). This notation is similar to $l$, previously defined. The goal is to find the most similar VP-cluster concerning the target user and associate him to the group. With the VP-clusters information the system will infer and update the Visual Perception Network and use it in the recommendation process (see Section 4.3).

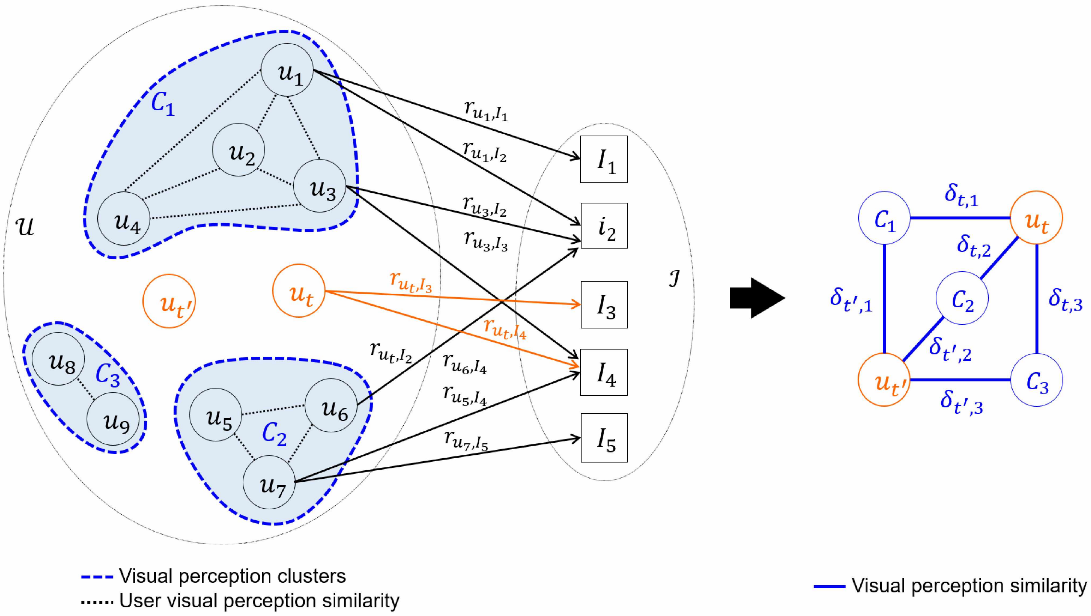


Figure 12 – Visual perception clusters and users' ratings (left), selection of visual perception cluster for $u_{t'}$ (cold start) and $u_t$ (right).

## 4.3 VP Framework

In this work, we propose an approach to incorporate VP-similarity in matrix factorization and pairwise recommender systems to deal with cold-start problem. Figure 13 shows an overview of VP framework.

**Building Visual Perception Network**: Given the users' visual perception over the set of images $\mathcal{I}$, the users are clustered (as described in Section 4.2) according to the visual perception (Module 1), generating a set of VP-clusters. Each VP-cluster $C_j$ comprises a set of users and one consensual vector. Let $G = (V, E)$ be the visual perception network (VP-network) and $u_t$ and $u_v$ vertices of this graph. The VP-Network is build connecting all users in the same VP-cluster. Then, a set of neighbors of a user $u_t \in C_j$ is $N(u_t) = \{u_v | u_v \in V \wedge (u_t, u_v) \in E \wedge (u_v \in C_j)\}$.

**Updating Visual Perception Network**: Update in VP-Network have to be made when a user is added to a VP-cluster or a user is take out from one. When a user $u_t$ is added to a VP-cluster $C_j$, we will insert edges on the VP-Network connecting $u_t$ with each $u_v$ in the same cluster. On the other hand, if a user $u_t$ is take out from one VP-cluster $C_j$ we will drop from the VP-Network all $u_t$'s connections with users in $C_j$. These situations can happen when a cold user is added to the system or an old user move to another cluster.

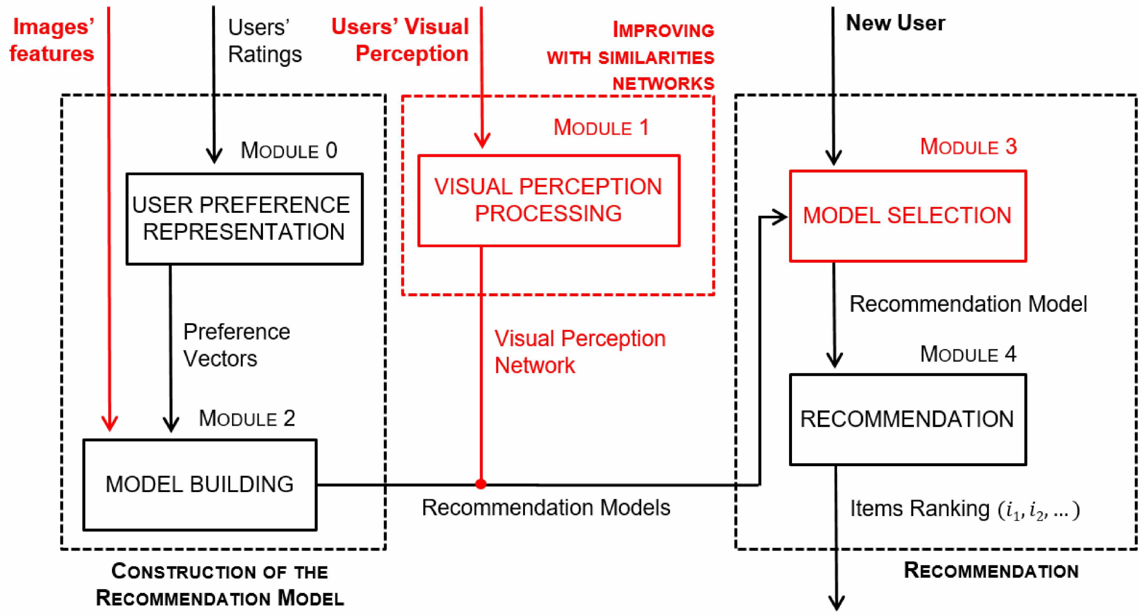


Figure 13 – VP Framework general representation.

**Building Prediction Models with VP-Rec**: To build the prediction models VP-Rec, as Pref-Rec does, computes the clustering of $\mathcal{R}$ matrix and mining the preferences. Clustering the rows of user-item rating matrix $\mathcal{R}$ results in a set of clusters (called here as Pref-clusters $C^r$). For each Pref-cluster $C^r_j$ we apply a consensus operator to get a consensual preference vector $V_j$, where each position has the average ratings per item (independent of the number of users who rated the item). From each $V_j$ and the images features, we apply CPrefMiner algorithm (AMO et al., 2013) (Module 2) and has as output a preference model $Pm_j$. After building recommendations models we have a set of prediction models $M_{vp} = \{M_{vp_0} = (C^r_1, V_1, Pm_1), \ldots, M_K = (C^r_K, V_K, Pm_K)\}$, where $K$ is the number of Pref-clusters and each $C^r_j$ represent the set of users in the Pref-cluster. Note that the set of users in a cluster was not used by PrefRec, but is necessary to VP-Rec locate the prediction models of the target user's neighbors.

**VP-Rec Recommendation**: VP-Rec method chooses between consensual prediction models the most suitable for a cold user. To recommend for a user is necessary to have visual perception information from him due the neighborhood is given by the VP-Network. In VP-Rec, given a target user $u_t$ and his neighbors $(N(u_t))$, the first task is select the prediction model $Pm_j$ corresponding to the Pref-cluster $C^r_j$ with more visual perception neighbors. $Pm_j$ is used to infer the preference between pairs of images in $\mathcal{I}$. We build a ranking using the set of predicted preferences between image pairs (Module 4) and evaluate the ranking quality over the top-$k$ images.

**Example**: Table 15 shows an example of relational schema with attributes of 8 paintings images. A user-item rating matrix with the same 8 images and 6 users is exemplified in Table 16. An example of clustering and consensus calculus can be seen in Table 17. We cluster the users from Table 16 in three Pref-clusters, and compute a consensual preference vector for each cluster using the group average rating per item. In a different way from 3.2, we compute the average rating even less than 50% of users rated the item. To build the prediction model for the first group, VP-Rec compute a preference relation over consensual preference vector $V_1$ as showed in Table 18. Then, the Bayesian preference network PNet$_1$ is computed (Fig. 12). Consider a cold user $u_1$ that is part of the VP-cluster $C_1$ (Table 14). So, the set of $u_1$'s neighbors is $N(u_1) = \{u_2, u_3\}$. At Table 17 we can see that $u_2$ and $u_3$ is on Pref-cluster $C_1^r$. How $C_1^r$ is the Pref-cluster with more neighbors, we will apply the prediction model $Pm_1$, represent by the $PNet1$, to make predictions to user $u_1$.

Table 15 – Relational schema of paintings images.

| | Title | Decade | Artist | Type | Art Movement |
|---|---|---|---|---|---|
| $I_1$ | Dora Maar | 1930 | Picasso | Portrait | Surrealism |
| $I_2$ | Portrait of Gala | 1930 | Dali | Portrait | Surrealism |
| $I_3$ | Shades of Night | 1930 | Dali | Landscape | Surrealism |
| $I_4$ | Nusch Eluard | 1930 | Picasso | Portrait | Cubism |
| $I_5$ | Bust of a woman | 1940 | Picasso | Portrait | Cubism |
| $I_6$ | Summer night | 1920 | Dali | Landscape | Surrealism |
| $I_7$ | The Bleeding Roses | 1930 | Dali | Nudism | Surrealism |
| $I_8$ | The Persistence of Memory | 1930 | Dali | Landscape | Surrealism |

**Building Prediction Models and Recommendation as an extension of Social**

Table 16 – Users ratings over painting images.

| | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ | $I_6$ | $I_7$ | $I_8$ |
|---|---|---|---|---|---|---|---|---|
| $u_2$ | 5 | 2 | 4 | 1 | 5 | 2 | - | - |
| $u_3$ | 4 | 1 | 4 | 1 | 5 | 2 | 5 | 5 |
| $u_4$ | 2 | 5 | 3 | 5 | - | - | - | - |
| $u_7$ | 2 | - | - | 5 | 2 | - | - | - |
| $u_5$ | 1 | - | 2 | 4 | 2 | 4 | - | - |
| $u_6$ | - | - | 2 | 4 | 1 | - | 5 | - |

Table 17 – Three Pref-clusters from user-item rating matrix in Table 16.

| | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ | $I_6$ | $I_7$ | $I_8$ |
|---|---|---|---|---|---|---|---|---|
| $u_2$ | 5 | 2 | 4 | 1 | 5 | 2 | - | - |
| $u_3$ | 4 | 1 | 4 | 1 | 5 | 2 | 5 | 5 |
| $V_1$ | 4.5 | 1.5 | 4 | 1.0 | 5.0 | 2.0 | 5.0 | 5.0 |
| $u_4$ | 2 | 5 | 3 | 5 | - | - | - | - |
| $u_7$ | 2 | - | - | 5 | 2 | - | - | - |
| $V_2$ | 2.0 | 5.0 | 3.0 | 5.0 | 2.0 | - | - | - |
| $u_5$ | 1 | - | 2 | 4 | 2 | 4 | - | - |
| $u_6$ | - | - | 2 | 4 | 1 | - | 5 | - |
| $V_3$ | 1.0 | - | 2.0 | 4.0 | 1.5 | 4 | 5 | - |

Table 18 – $V_1$ pairwise preference relation

| |
|---|
| $(I_1 > I_3)$ |
| $(I_3 > I_6)$ |
| $(I_5 > I_6)$ |
| $(I_6 > I_2)$ |
| $(I_5 > I_3)$ |
| $(I_2 > I_4)$ |
| $(I_7 > I_6)$ |
| $(I_7 > I_1)$ |
| $(I_8 > I_1)$ |

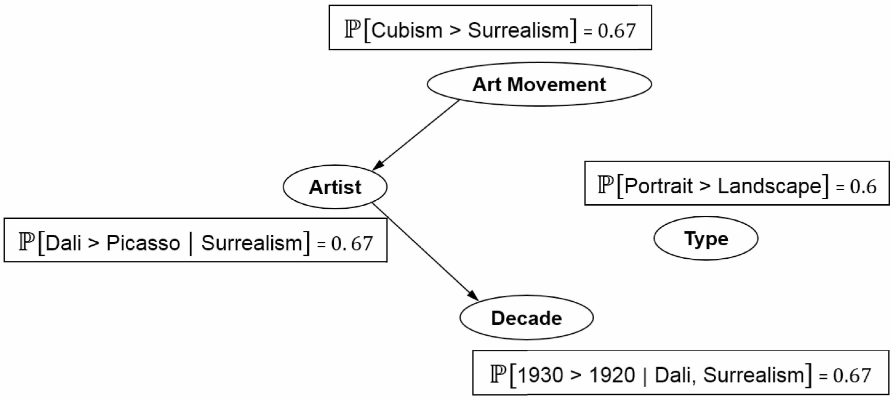


Figure 14 – Bayesian Preference Network $\mathbf{PNet}_1$ over $V_1$ preferences.

**Matrix Factorization Recommenders**: These methods will build one personalized model for each user $u_x$ based on latent factors of items and latent factor of users in the same VP-Cluster as $u_x$. The recommendation for the user $u_x$ will be made apply the personalized prediction model that will predict the ratings for the items that will be recommended.

## 4.4 Experimental Settings

### 4.4.1 Dataset

There are several visual perception datasets, but for evaluating our prediction model a suitable dataset must have item's attributes and ratings. Given the various factors that may influence recommendation systems, we analyze two different sets:

- **Paintings Dataset.** In the work presented in (ALMEIDA, 2016) ,the author recruited 193 volunteers for rating 200 paintings, which were randomly chosen between 605 paintings public available at <http://pintura.aut.org/>. For each volunteer, an eye tracker device captures eye movements on each painting displayed on the 22' monitor with image resolution of 500 x 700 pixels. The paintings are composed by epoch, art movement, country, artist, type, color intensity and hue (image attributes). The volunteer should rate each painting in a 1-5 scale according to its preference.

- **Clothing Dataset.** Melo et al. (MELO; NOGUEIRA; GULIATO, 2015) also recruited volunteers to rate a clothing dataset. Hence, the full set is composed by two subsets of ratings over female and masculine clothing. In addition, they also collected visual attention through an eye tracker device. Clothing specific attributes are composed by class body, category, predominant color, color intensity, pattern, shape, size and sleeve. From original dataset we got only items rated in common among all users because we want to test networked information.

Table 19 summarizes datasets statistics.

Table 19 – Paintings and Clothing dataset features.

| Features | Paintings | Female-Clothing | Male-Clothing |
|---|---|---|---|
| # of users | 194 | 121 | 120 |
| # of items | 605 | 210 | 210 |
| # of ratings | 38,753 | 25,396 | 25,193 |
| Sparsity (%) | 67.00 | 0.05 | 0.03 |
| Links | 28,992 | 7,204 | 9,531 |
| Average # of ratings | 199.88 | 209.88 | 209.94 |

### 4.4.2 Comparison Methods and Parameter Settings

To assess the effectiveness of VP-Rec, we compare it with four renowned recommenders:

**PMF:** A probabilistic matrix factorization approach (SALAKHUTDINOV; MNIH, 2008). This is the unique comparison method that does not use VP-similarity information. This method can be seen as a general baseline algorithm.

**SoRec**, **SocialMF** and **TrustMF** were described in Section 3.4.2.

*Parameter Settings.* VP-Similarity scores were computed splitting images in 4 equal parts. All methods make use of the visual perception generated by Module 1 of VP-Rec. We use LibRec (GUO et al., 2015) library implementation of SoRec, SocialMF, TrustMF and PMF methods with default parameters. For matrix factorization approaches the experiments were executed with 10 latent factors and number of interactions equal to 100. VP-Rec cluster algorithm is K-means and the distance measure is Euclidean. We test several cluster size for preference and visual perception. Then for Pref-clusters the optimal numbers are 9 clusters for Painting dataset, 9 for Female-Clothing and 6 for Male-Clothing. To VP-clusters the optimal number is 2 clusters for all datasets.

### 4.4.3 Evaluation Protocols

We performed two classes of experiments reflecting differing numbers of ratings available to train each method. The first protocol, called **0-ratings protocol**, is basically the standard leave-one-out cross-validation, where the number of folds is equals to the number of instances in the dataset. Thus, each recommender system is applied once for each instance, using all other instances as a training set, but one selected as a single-user test.

We train the system with all users but one, which is the one selected for testing purpose. Note that none item ratings from the testing user is given to the system. Thus, we simulate a realistic cold-start scenario. In the second set of experiments, we apply the standard **five-fold cross-validation**.

With social approaches, we replace the required social network information by our visual perception network. Although our network is not a real social network, it is build based on the homophily assumption (MCPHERSON; SMITH-LOVIN; COOK, 2001), which states that users linked with each other in social networks tend to have similar tastes, hence we linked users based on their visual perceptions similarities. Furthermore, we aim to investigate human visual attention to bootstrap recommender systems, mainly to handle cold-start problem. Because social recommenders is well known for dealing with cold users, we chose them to compare to our approach.

## 4.5 Result and Discussion

Here, we assess the effectiveness of VP-Rec approach for item recommendation. In particular, we aim to answer our three research questions:

### 4.5.1 How effective is VP-Rec for cold-start user? (RQ1)

We assess the prediction quality of visual perception approaches among the state-of-art recommenders presented in Section 4.4.2. Table 20 shows the result of this comparison in terms of ***nDCG*** rank size of 5, 10, 15, and 20 for items recommended in our three datasets (Paintings, Female-Clothing, and Male-Clothing).

The experimental results, for 0-ratings protocol, show the superiority of VP-Rec over all datasets. In particular, its performance might be explained because it needs none rating to build its prediction model, which is the situation met in real applications. The recommendation for a 0-rating user $u_k$ is then made selecting the consensual model according to $u_k$'s visual perception network. Inside $u_k$ VP-Network we can have distinct Pref-clusters, and VP-Rec chooses the one that contains more users. Recalling RQ1, this attests the effectiveness of apply visual perception for 0-ratings user in contrast to others social approaches.

We checked the normality and homogeneity of the nDCG results for each method using Shapiro and Bartlett test. We observed that the results values are not normally distributed and not homogeneous. Therefore, we performed the global comparisons with Kruskal-Wallis test. Our approach, with 95% of confidence, produced significant higher-quality results.

Table 20 – nDCG for cold-start scenario (0-rating) against our three datasets.

(a) Paintings

| Approach | Size of Rank @5 | @10 | @15 | @20 |
|---|---|---|---|---|
| SoRec | 0.8332 ± .126 | 0.8301 ± .110 | 0.8258 ± .101 | 0.8219 ± .098 |
| SocialMF | 0.8086 ± .123 | 0.8051 ± .103 | 0.8015 ± .097 | 0.8028 ± .091 |
| TrustMF | 0.6337 ± .145 | 0.6325 ± .127 | 0.6348 ± .122 | 0.6406 ± .118 |
| PMF | 0.6263 ± .157 | 0.6348 ± .135 | 0.6394 ± .128 | 0.6441 ± .118 |
| VP-Rec | **0.9707** ± .053 | **0.9616** ± .048 | **0.9530** ± .101 | **0.9457** ± .049 |

(b) Female-Clothing

| Approach | Size of Rank @5 | @10 | @15 | @20 |
|---|---|---|---|---|
| SoRec | 0.7662 ± .157 | 0.7559 ± .137 | 0.7572± .128 | 0.7632 ± .119 |
| SocialMF | 0.7569 ± .155 | 0.7559 ± .135 | 0.7572 ± .127 | 0.7632 ± .122 |
| TrustMF | 0.6062 ± .139 | 0.6139 ± .122 | 0.6154 ± .118 | 0.6221 ± .113 |
| PMF | 0.5987 ± .162 | 0.5977 ± .134 | 0.6050 ± .122 | 0.6098 ± .114 |
| VP-Rec | **0.9352**± .079 | **0.9202** ± .078 | **0.9107**± .073 | **0.9130** ± .073 |

(c) Male-Clothing

| Approach | Size of Rank @5 | @10 | @15 | @20 |
|---|---|---|---|---|
| SoRec | 0.7842 ± .129 | 0.7752 ± .115 | 0.7691± .105 | 0.7785 ± .098 |
| SocialMF | 0.7708 ± .132 | 0.7655 ± .118 | 0.7645 ± .111 | 0.7698 ± .099 |
| TrustMF | 0.5941 ± .167 | 0.5955 ± .146 | 0.5993 ± .134 | 0.6045 ± .126 |
| PMF | 0.5759± .145 | 0.5794 ± .130 | 0.5852 ± .121 | 0.5919 ± .115 |
| VP-Rec | **0.9314** ± .077 | **0.9231** ± .069 | **0.9154** ± .068 | **0.9122** ± .067 |

### 4.5.2 How is the performance of VP-Rec under data sparsity? (RQ2)

Sparsity is the percent of empty ratings in user-item rating matrix. We investigate RQ2 using eight subsets obtained from Male-Clothing by eliminating a certain amount of ratings, see Table 21. The reason for these experiments is the fact that sparsity is a big challenge faced by recommendation systems in general (ADOMAVICIUS; TUZHILIN, 2005). The idea is to simulate sparse scenarios where input datasets contains too many item to be rated and few items rated per user. For instance, Male-Clothing$_{80}$ was obtained by eliminating around 80% of the ratings in a stratified manner (COHEN, 2011), so that we keep homogeneous subgroups of the original set.

Because VP-Rec and SoRec were the methods that achieved better results under cold-start scenario, we choose them to test and compare their results under sparse subsets. Figure 16 shows the performance of each method per subset.

We note that VP-Rec is substantially affected by data sparsity. Its performance decreases as the data sparsity increases. On the hand, SoRec presents better results under sparser subset, enough to overcome VP-Rec performance against the most sparse scenario

Table 21 – Male-Clothing sparser subsets.

| Male-Clothing (Dataset) | # of Ratings (Average) | Ratings per user (Average) | Sparsity (%) |
|---|---|---|---|
| 10 | 22,720 | 189.33 | 9.84 |
| 20 | 20,186 | 168.21 | 19.89 |
| 30 | 17,672 | 147.26 | 29.87 |
| 40 | 15,150 | 126.25 | 39.88 |
| 50 | 12,617 | 105.14 | 49.93 |
| 60 | 10,116 | 84.3 | 59.85 |
| 70 | 7,579 | 63.15 | 69.92 |
| 80 | 5,083 | 42.35 | 79.82 |

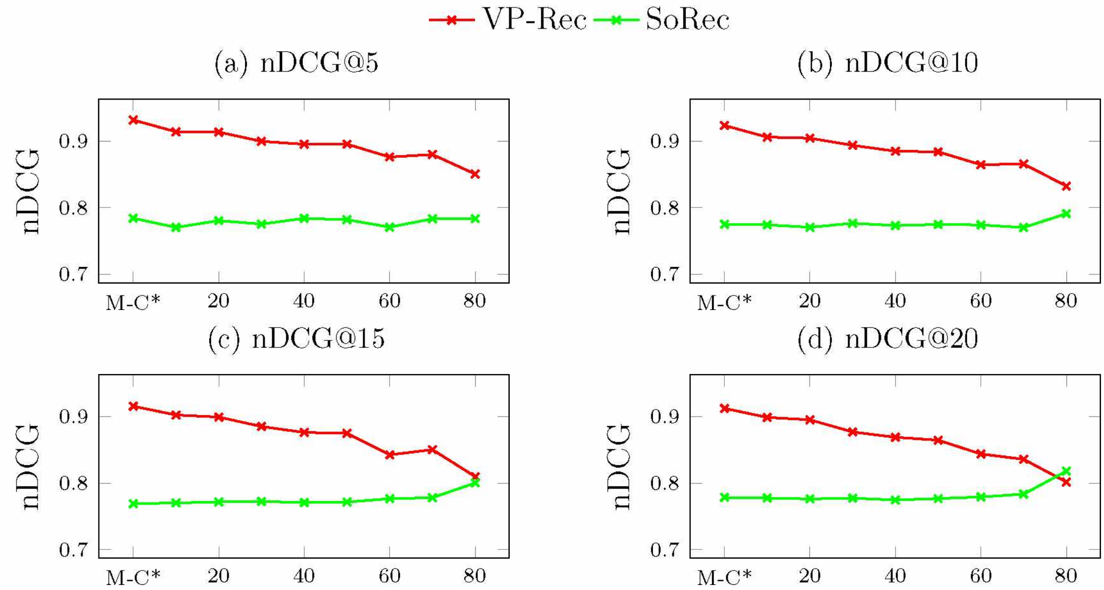


Figure 16 – nDCG scores across Male-Clothing sparser subsets.

(80% of sparsity). Overall, the results suggest that VP-Rec effectiveness might be related with dataset density. However, its results was only surpassed for rank size of 20 items.

### 4.5.3 What is the performance comparison of matrix factorization approaches on users with observed ratings versus VP-Rec? (RQ3)

The last experiment investigates the performance of VP-Rec, with no ratings, against traditional approaches with certain amount of ratings. The idea is to analyze to what extent visual perception data suffice to offer accurate recommendation in the image data.

We test using 5-fold-cross validation technique, providing 20% of items ratings from each test user to bootstrap each matrix factorization system recommender. In these experiments we have PMF using 80% of ratings to build the target user prediction model. SoRec, SocialMF and TrustMF combine 80% of ratings with visual perception information for the same task. On the other hand, VP-Rec select a consensual prediction model using only visual perception information. All methods make predictions over the same 20% of ratings.

The overall result was the same under 0-rating protocol, see Table 22. Again, we performed Kruskal-Walis statistical test and it shows that VP-Rec is superior with 95% of confidence. Using only visual perception to select a consensual prediction model, instead of build a personalized one, our approach is a good alternative to recommend images.

Table 22 – nDCG for 5-fold-cross-validation protocol against our three datasets.

(a) Paintings

| Approach | Size of Rank | | | |
|---|---|---|---|---|
| | @5 | @10 | @15 | @20 |
| SoRec | 0.8287 ± .093 | 0.8210 ± .071 | 0.8181 ± .060 | 0.8132 ± .054 |
| SocialMF | 0.6713 ± .108 | 0.6766 ± .083 | 0.6791 ± .071 | 0.6804 ± .064 |
| TrustMF | 0.7389 ± .117 | 0.7360 ± .090 | 0.7334 ± .079 | 0.7314 ± .072 |
| PMF | 0.6292 ± .129 | 0.6281 ± .099 | 0.6258 ± .084 | 0.6247 ± .075 |
| VP-Rec | **0.9284** ± .082 | **0.9144** ± .080 | **0.9029** ± .082 | **0.8938** ± .083 |

(b) Female-Clothing

| Approach | Size of Rank | | | |
|---|---|---|---|---|
| | @5 | @10 | @15 | @20 |
| SoRec | 0.7367 ± .113 | 0.7316 ± .087 | 0.7298 ± .073 | 0.7322 ± .065 |
| SocialMF | 0.5785 ± .129 | 0.5719 ± .099 | 0.5529 ± .082 | 0.5511 ± .074 |
| TrustMF | 0.6710 ± .121 | 0.6636 ± .093 | 0.6616 ± .082 | 0.6626 ± .075 |
| PMF | 0.5688 ± .123 | 0.5689 ± .094 | 0.5706 ± .081 | 0.5747 ± .074 |
| VP-Rec | **0.9044** ± .086 | **0.8886** ± .079 | **0.8741** ± .080 | **0.8607** ± .080 |

(c) Male-Clothing

| Approach | Size of Rank | | | |
|---|---|---|---|---|
| | @5 | @10 | @15 | @20 |
| SoRec | 0.7300 ± .117 | 0.7253 ± .093 | 0.7282± .081 | 0.7321 ± .074 |
| SocialMF | 0.6121 ± .115 | 0.6086 ± .088 | 0.6023 ± .075 | 0.6049 ± .068 |
| TrustMF | 0.6527 ± .146 | 0.6538 ± .119 | 0.6590 ± .107 | 0.6660 ± .098 |
| PMF | 0.5491 ± .124 | 0.5548 ± .096 | 0.5611 ± .084 | 0.5676± .077 |
| VP-Rec | **0.9118**± .093 | **0.9008** ± .087 | **0.8924** ± .084 | **0.8844** ± .082 |

## 4.6 Summary

In this chapter we introduced VP-Rec, an approach to handle user cold-start problem in image recommendation. We proposed to combine *user's visual perception*, as a valuable source of additional information, with prediction models based on *pairwise preferences* and social matrix factorization approaches. We thorough evaluated VP-Rec against two images dataset and showed that our approach beat state-of-art recommender systems that handle contextual networks, reaching up to 90% of ranking quality.

The ability to handle visual perception networks introduced by VP-Rec opens several avenues for future research. First, we will experiment other kinds of model build in VP framework, mainly because matrix factorization approach showed better results under high data sparsity. We will exploit other ways to measure visual similarities among users and apply filters during the recommendation phase according to a visual perception similarity score. In Chapter 5 we will present one approach that attend these two innovations. We also intend to experiment other visual contexts domains such as online dating services in the future.

Chapter 5

# Benefits of a General Framework to Incorporate Networked Information[1]

## 5.1 Introduction

In Chapter 2 we talk about the increase of social information and how it is of particular interest for a cold user, because he is considered initially by the recommendation system though he has not yet provided any information about his preferences.

There has been substantial research interest in improving certain aspects of the user cold-start problem using social information (BARJASTEH et al., 2016; ALAHMADI; ZENG, 2015; PEREIRA; HRUSCHKA, 2015; MACEDO; MARINHO; SANTOS, 2015; QUIJANO-SáNCHEZ; RECIO-GARCíA; DíAZ-AGUDO, 2013; DELPORTE et al., 2013). However, the dominant trend of these studies have been towards designing new prediction models. The typical approach is to use social information to build a recommendation model for each cold user.

Due to the inherent complexity of this modeling process, the performance of most SRS decreases for cold users: those systems cannot offer personalized recommendations until they collect enough preference information from users.

This approach is our reaction to these experiences. Figure 17 illustrates the method we propose. ToSocialRec takes advantage of all prediction models already built in the system. It chooses the most suitable one and creates a consensual model for a cold user based on how strong he is similar to the other users. The selection process is based on the *homophily* principle, which is the tendency of individuals to associate and bond with similar others.

Earlier social approaches investigated the homophily assumption to build new prediction models. Herein, we focus on inspecting the existing models, looking for those that might maximize information gained about the cold user. The differences between Social

[1] This chapter was published in a modified form in (FELÍCIO et al., 2016e)

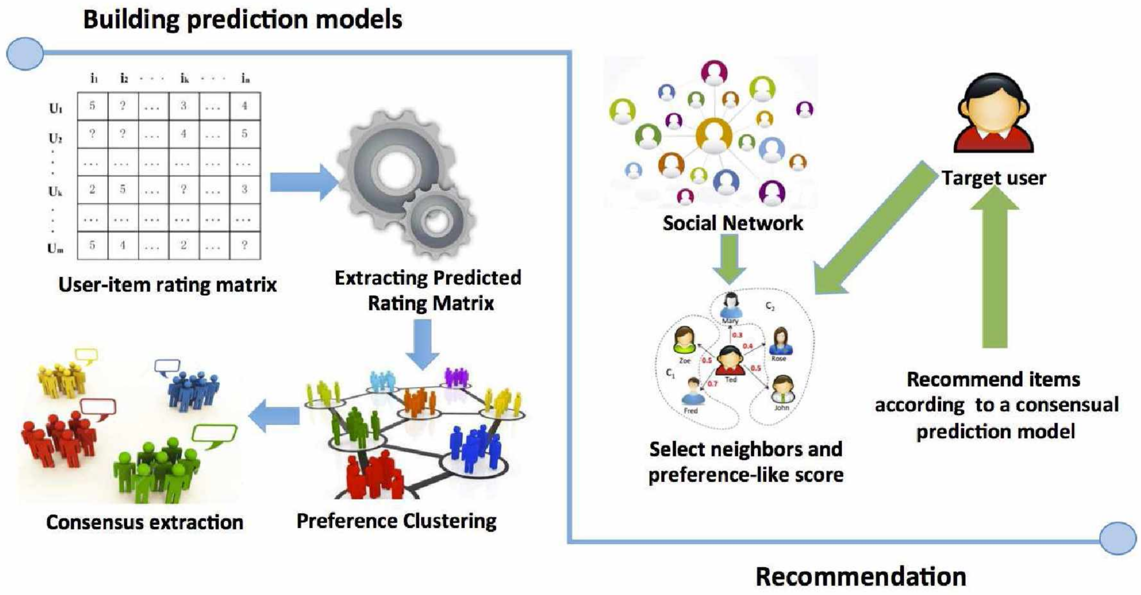


Figure 17 – ToSocialRec selects a prediction model from a set of consensual ones previously built for other users.

PrefRec and ToSocialRec include the use of another type of prediction model and the prediction model selection based on a users general network. The users do not need to be part of an online social network and the users network can be inferred using some information that indicates the similarity between them.

We structure our work around the following research question:

**RQ:** Does a recommender system already hold suitable prediction models to deal with a cold user?

Model selection is a broad subject. We will look at two distinct sub-questions to examine ToSocialRec:

***RQ#1*:** How well can a selected model predict the ratings of cold users?

***RQ#2*:** How well can a selected model rank items to cold users?

Our main contributions of this chapter are threefold:

1. We compare and contrast several matrix factorization based social recommender systems in a cold-start scenario;

2. We propose a general model selection approach that leverages existing models to recommend relevant items to cold users;

3. We thoroughly evaluate ToSocialRec against six distinct data sets and we show its effectiveness in contrast to state-of-the-art matrix factorization based recommenders.

This chapter is organized as follows. First, we further motivate the need for a new SRS and its main concept (Section 5.2). Next we present the framework and how it works (Section 5.3). Section 5.4 describes our experimental settings and Section 5.5 discusses the results. Finally, Section 5.6 concludes the chapter.

## 5.2 Background

In this section, we present an illustrative example of ToSocialRec and introduce the main concepts underlying this approach.

### 5.2.1 Motivating example

As a motivating example, let us consider the domain of movie recommendation. Let us assume that the recommender system has only two different genres of movies: romance and thriller. The system has yet dealt with people who prefer romances and others fond of thrillers. We can find these groups based on the ratings already given. The first group will have users that have given high ratings to romantic movies and the second group will be made of people who give high ratings to thrillers.

Now, let us assume a cold user $u$ looking for help to find movies he would like to watch. A common situation is when the cold user has yet given no rating; he is admitted into the system in exchange of his social information. Once the system has collected enough preference data from the user, it can build an initial prediction model. Later, as the user provides ratings, the system improves the model. For now, the recommender system may associate the cold user with other users. For instance, it might associate $u$ along with users from the same gender, age or present similar affinities. Assuming that $u$ has a lot in common with people who like romances rather than thrillers, we might further inspect prediction models in the romance group to select one for $u$.

We hypothesize that it is reasonable to use people's preferences from one of identified groups to offer a recommendation to a cold user. To select one group, we use features that characterize the connection strength between connected users.

### 5.2.2 Preference-like Score

*Preference-like score* is the information that is correlated with preference similarity among users. Through this score, we can filter a set of users whose prediction models may be a good enough for initial recommendations.

We assume the existence of a network among users to compute the connection weight between them. For instance, preference-like score could be the demographic similarity in a friendship network or the centrality degree.

Formally, we can represent the users network as a graph $G = (V, E)$, in which users are the vertices of this graph. A set of friends (neighbors) of a vertex $u$ is $F(u) = \{v|v \in V \wedge (u, v) \in E\}$ and a function $l : F \to \mathbb{R}$ defines *Preference-like Score* between $u$ and $v$ in $[0, 1]$.

Social network presents several ways to compute the preference-like score. In this approach, we exploit the following well-known network metrics (MISLOVE et al., 2007; ZAFARANI; ABBASI; LIU, 2014) as defined in (FELÍCIO et al., 2015; FELÍCIO et al., 2016a) and in Section 3.3.2:

- **Friendship:** This score is equal to 1 for each connection between the target user and a neighbor.
- **Mutual Friends:** This is given by the mutual neighbors score computed by Jaccard coefficient;
- **Similarity:** This is the demographic similarity between a target user and his neighborhood;
- **Centrality:** We can set the connection weight according to centrality of the target user neighbors in the social network.

We advocate that ToSocialRec is not restricted to the commonly used social network metrics. Therefore, we also analyze the performance of the preference-like score given by non-traditional networks similarities, using visual perception networks. Based on the VP-Similarity method defined in Section 4.2, we define the following metrics:

- **VP-similarity:** Similarity score based on visual perception's similarities among users;
- **VP-friendship:** A specialized friendship connection represented by visual perception network where users in the same visual perception cluster are connected and have preference-like score equal to 1.

In summary, we generalize the concept of network similarity and that is why we call it ***Preference-like score***. We argue that if we can define a function $l$ that determines the connection strength between users in a recommender system, we can use this score to help a recommender system dealing with cold users.

## 5.3 ToSocialRec

In this section, we describe ToSocialRec highlighting how to incorporate the preference-like score in Matrix Factorization based recommender systems.

Let $U$ be a set of users and $I$ be a set of items. Each user $u \in U$ and each item $i \in I$ has a unique identifier. The user-item rating matrix is $R = [r_{u,i}]_{m \times n}$, where each entry $r_{u,i}$ is the rating given by user $u$ on item $i$, and $m$ is the number of users, and $n$ is the number of items. An example of a user-item rating matrix with 6 users and 7 items, and ratings in the range $\{1, 2, 3, 4, 5\}$ is provided in table 23.

Table 23 – Example of a user-item rating matrix. - means that the user has not rate the item.

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ | $i_6$ | $i_7$ |
|---|---|---|---|---|---|---|---|
| Zoe | 5 | 2 | 4 | - | 5 | 1 | - |
| Fred | 4 | - | 5 | - | 5 | - | 1 |
| Mary | 2 | 5 | 3 | 5 | - | - | - |
| Rose | 1 | - | 2 | - | 2 | - | - |
| Paul | - | - | 3 | 4 | 1 | - | - |
| John | 2 | - | - | 5 | 2 | - | - |

In traditional recommender systems, the recommendation task is based on the prediction of the missing values of the user-item rating matrix. Then, predictions are used to rank items and recommend the k top-ranked.

In our work, we present ToSocialRec, an approach to extend traditional recommender systems to incorporate preference-like scores. The goal is to deal with cold-start users. ToSocialPrec alternates two phases, (a) construction and update of the prediction models, and (b) making recommendations. These two phases are described in the next two subsections.

### 5.3.1 Construction and Update of the Prediction Models

The main steps of the prediction model construction are: (i) Ratings prediction, (ii) Preference clustering, and (iii) Consensus computation. Each step is detailed below. To keep track of the evolving nature of the environment (such as the set of available ratings), the model has to be updated; though essential in a live system, this step is not considered in this work and only briefly described below.

**Rating prediction**: from the user-item rating matrix, we use a matrix factorization technique to get a matrix of predicted ratings $R'$. As mentioned in Section 2.2.1 the predicted rating of the item $i_k$ by user $u_j$ is $R'_{u_j,i_k} = predict(u_j, i_k, P, Q)$ and the details of this function depending on the completion method being used.

As an example, Table 24 shows the predicted rating matrix $R'$ obtained from the user-item matrix of Table 23, as completed using the BiasedMF[2] algorithm (KOREN; BELL; VOLINSKY, 2009).

[2] The name BiasedMF comes from the LibRec library that we use in the experiments.

Table 24 – Predicted rating matrix.

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ | $i_6$ | $i_7$ |
|---|---|---|---|---|---|---|---|
| Zoe | 4.6 | 2.09 | 4.23 | 4.24 | 4.84 | 1.07 | 1.0 |
| Fred | 4.2 | 3.8 | 4.42 | 5.0 | 4.86 | 2.28 | 1.2 |
| Mary | 1.97 | 4.84 | 3.22 | 4.87 | 2.68 | 2.68 | 1.61 |
| Rose | 1.19 | 3.24 | 2.17 | 3.56 | 1.92 | 1.23 | 1.0 |
| Paul | 1.77 | 3.16 | 2.81 | 4.07 | 1.14 | 1.6 | 1.56 |
| John | 2.09 | 4.32 | 3.29 | 4.77 | 2.09 | 2.46 | 1.98 |

With BiasedMF, the prediction function is $predict(u_j, i_k, P, Q) = \mu + b_{u_j} + b_{i_k} + P_{u_j} Q_{i_k}^T$, where $\mu$ is the overall average rating, $b_{u_j}$ is the deviation from $\mu$ of user $u_j$ ratings, $b_{i_k}$ is the deviation from $\mu$ of item $i_k$ ratings, $P_{u_j}$ is the $u_j^{\text{th}}$ row of matrix $P$, which are the latent factors for user $u_j$, and $Q_{i_k}^{\text{th}}$ row of matrix $Q$ which are the latent factors for item $i_k$. Finally, given the predicted rating matrix $R'$, the preference vector for a user $u_j$ is defined as the predicted ratings for user $u_j$, $\theta_j = R'_{u_j}$.

**Preference clustering**: Given a predicted rating matrix $R'$, we can cluster users according to their preference vectors, that is the rows of $R'$. A distance function and a clustering algorithm $\mathcal{C}$ are used. After clustering, we have a set of cluster $C$, where each cluster $C_s$ contains a set of users with the similar preferences.

**Consensus computation**: for each cluster $C_s$, we apply a consensus operator $\mathcal{A}$ to get the consensual preference vector $\hat{\theta}_s$ of cluster $C_s$. In this approach, the operator is the average, that is $\hat{\theta}_{s,k}$ is the average predicted rating for item $k$. We obtain $M = \{M_1 = (C_1, \hat{\theta}_1), \ldots, M_K = (C_K, \hat{\theta}_K)\}$, the set of prediction models where each $M_s$ is composed of a cluster of users $C_s$ and its consensual preference vector $\hat{\theta}_s$.

Table 25 – Consensual preference vectors.

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ | $i_6$ | $i_7$ |
|---|---|---|---|---|---|---|---|
| Zoe | 4.6 | 2.09 | 4.23 | 4.24 | 4.84 | 1.07 | 1.0 |
| Fred | 4.2 | 3.8 | 4.42 | 5.0 | 4.86 | 2.28 | 1.2 |
| $\hat{\theta}_1$ | 4.4 | 2.94 | 4.32 | 4.62 | 4.85 | 1.67 | 1.1 |
| Mary | 1.97 | 4.84 | 3.22 | 4.87 | 2.68 | 2.68 | 1.61 |
| Rose | 1.19 | 3.24 | 2.17 | 3.56 | 1.92 | 1.23 | 1.0 |
| Paul | 1.77 | 3.16 | 2.81 | 4.07 | 1.14 | 1.6 | 1.56 |
| John | 2.09 | 4.32 | 3.29 | 4.77 | 2.09 | 2.46 | 1.98 |
| $\hat{\theta}_2$ | 1.76 | 3.89 | 1.98 | 4.32 | 2.87 | 1.99 | 1.53 |

Table 25 continues the example and exemplifies the clustering process, and the consensus computation: the users from Table 24 were clustered in two groups according to their preference vectors, and the consensual preference vector for each cluster was computed.

**Model Update**: in a live recommendation system, the set of prediction models $M$ must be rebuilt when the insertion of new ratings in the rating matrix $R$ increases the difference between $R$ and prediction rating matrix $R'$. Let $D(t) = [d_{u,i}]_{m \times n}$ be the absolute difference matrix between $R$ and $R'$ at time t, where each $d_{u,i} = |r_{u,i} - r'_{u,i}|$. Let us define a difference score $diff(t) = \sum_{j=1}^{m} \sum_{k=1}^{n} d_{u_j,i_k}$. After each update on $R$ at any time $t' > t$, it is straightforward to update $diff(t')$ incrementally. Then, we decide on updating $M$ once $diff(t')$ reaches a certain threshold.

### 5.3.2 Making Recommendations

In its second phase, ToSocialRec makes use of a prediction model $M_s$ to recommend items for a cold user. The recommendation process is executed online, differently from the previous phase which is offline.

The selection of a prediction model uses preference-like scores. Let $select : U \to M$ be a function that selects the suitable prediction model from $M$ for a target user $u$ defined by the minimum threshold strategy, adapted from Chapter 3, Section 3.3.2, as follows.

*Minimum threshold*: let $\varepsilon \in [0, 1]$ be a preference-like minimum threshold. The minimum threshold strategy selects the prediction model $M_s \in M$ which associated cluster of users $C_s$ that has more friends of $u$ satisfying a threshold, according to the Eq. (8).

$$select(u) = \arg\max_{M_S \in M} |\{v \in F(u) \wedge l(u,v) \geq \varepsilon\}| \tag{8}$$

The recommendation process for a cold-start user is executed as follows:

1. Given a target user $u$ and a Preference-like metric, the system will select the neighbors $F(u)$ of $u$, and the Preference-like score, previously computed, between $u$ and each $v \in F(u)$.

2. Using minimum threshold, select the prediction model $M_s$ according to Eq. (8).

3. The consensual preference vector of $M_s$ $\hat{\theta}_s$, is used to rank the items.

4. k top-ranked items are recommended to $u$.

*Example.* To explain how our recommendation phase works, we consider the preference-like network in Figure 18. The user Ted (our cold-start user) is connected with users of the two computed groups (see Table 25). Preference-like score between Ted and his connections stands for the level of similarity between them. Given a minimum preference-like score of 0.5, we identify that $C_1$ is the group with more users satisfying this threshold. We will use the consensual predictions of group $C_1$, $\hat{\theta}_1$ to offer recommendations to Ted. Then, the item ranking is: $\{i_5, i_4, i_1, i_3, i_2, i_6, i_7\}$.

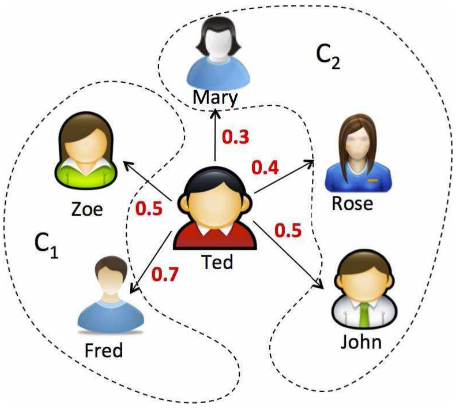


Figure 18 – Example of a preference-like network with a cold-start user (Ted) and his neighbors. Dashed contours identify the two preference clusters.

Table 26 – Dataset features.

| *Dataset* | Users | Items | Ratings | Sparsity (%) | Ratings/User (Average) | Links | Links/User (Average) |
|---|---|---|---|---|---|---|---|
| *Facebook* | 498 | 169 | 49,729 | 40.9 | 99.85 | 5,468 | 10.9 |
| *Flixster* | 1,323 | 1,175 | 811,726 | 47.78 | 613.54 | 6,526 | 5.34 |
| *FilmTrust* | 1,508 | 2,071 | 35,494 | 98.86 | 23.53 | 1,853 | 3.0 |
| *Epinions* | 1,161 | 529 | 25,781 | 95.8 | 22.2 | 62,903 | 55.03 |
| *Paintings* | 194 | 605 | 38,753 | 67 | 200 | 28,992 | 149.44 |
| *Clothing* | 121 | 210 | 25,396 | 0.05 | 209.88 | 7,204 | 59.53 |

## 5.4 Experiment Setting

In this section, we conduct extensive experiments on several datasets. We also compare ToSocialRec against a set of state of the art algorithms. Then, we evaluate how statistically significant our results are.

### 5.4.1 Datasets and Configuration

We evaluate ToSocialRec on six datasets: 3 movie rating datasets, 1 product review dataset, 1 painting dataset, and 1 clothing dataset. Table 26 summarizes their main descriptive features. It is noteworthy that those datasets present a variety of features among themselves. We briefly describe each dataset:

***Facebook Dataset*** (FELÍCIO et al., 2015) the general features of this dataset was described in Section 3.4.1. However we use a more sparse sample of this dataset where the users rated unless 20 movies.

***Flixster Dataset*** (JAMALI; ESTER, 2010) also was described in Section 3.4.1 as Flixster 811K.

***Filmtrust Dataset*** (GUO; ZHANG; YORKE-SMITH, 2013; GUO; ZHANG; YORKE-

SMITH, 2016) is about movie sharing and ratings. The preference-like network is based on users trust network. From a trust network, we compute a mutual friend score, and a user centrality score.

***Epinions Dataset*** (MASSA; AVESANI, 2007) contains data from product reviews. Epinions' preference-like network is also computed with users trust network. From trust network we compute the mutual friend score, and the user centrality score. Due to computational restrictions, we only consider a fraction of the original dataset. We further discuss this point in Section 5.5.

***Paintings Dataset*** (FELÍCIO et al., 2016b) and ***Clothing Dataset*** (MELO; NOGUEIRA; GULIATO, 2015) was previous described in Section 4.4.1. Here we got only the female clothing information from original Clothing dataset.

### 5.4.2 Evaluation Metrics

We adopt the same **0-ratings protocol** as the protocol described in Section 4.4.3.

The goal is to measure the performance of each algorithm to predict item ratings. We use the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE), Eq. (9) and Eq. (10), as evaluation criterion, where $r_{u,i}$ is the rating for an item $i$ from a target user $u$, $\hat{r}_{u,i}$ is the predicted rating for $i$ and $X$ is the total number of ratings.

$$MAE = \frac{\sum_{u,i} |r_{u,i} - \hat{r}_{u,i}|}{X}, \tag{9}$$

$$RMSE = \sqrt{\frac{\sum_{u,i} (r_{u,i} - \hat{r}_{u,i})^2}{X}} \tag{10}$$

Though widely used, MAE and RMSE do not characterize the quality of the recommendation. Ranking quality is measured computing the Normalized Discounted Cumulative Gain (nDCG) metric, Eq. (4) in Section 3.4.4.

### 5.4.3 Other methods

To assess the effectiveness of ToSocialRec, we compare it with the same other social matrix factorization based recommender systems we compare with Social PrefRec and VP-Rec. *SoRec*, *SocialMF* and *TrustMF* were described in Section 3.4.2. These methods were designed to combine social information with rating data. They are distinct from ToSocialRec that uses social information only to select a consensual prediction model between preference clusters. The weight of social information in the building model process is determined by a parameter in the three approaches. None of them makes use of any clustering technique.

### 5.4.4 Parameter Settings

We use LibRec (GUO et al., 2015), which provides an implementation of SoRec, SocialMF and TrustMF. The implementation of ToSocialRec was done on top of BiasedMF algorithm also in LibRec library. Therefore, we cluster BiasedMF prediction models and include Preference-like score in the recommendation process.

Experiments were executed with 10 latent factors and 100 iterations for the model building phase. The social information weight are measured by $\lambda_c$, $\beta$ and $\lambda_t$ in SoRec, SocialMF and TrustMF respectively, we were varying them between 0.1 and 100. Optimal experimental settings for $\lambda_c$ is equal to 20 in Facebook, 50 in Flixster, 100 in Filmtrust and 1 in Epinions, Paintings and Clothing. SocialMF achieves better results with $\beta$ equal to 100 in Facebook and FilmTrust, 1 in Flixster, 50 in Epinions, 0.5 in Paintings and 0.1 in Clothing. Finally, $\lambda_t$ has optimal values equal to 100 in Facebook, Filmtrust, Epinions and Paintings, 0.9 in Flixster and 20 in Clothing.

With ToSocialRec, we also experimentally test several cluster sizes. Then we set the optimal number of clusters to 6 clusters for FilmTrust, 7 clusters for Epinions, 4 clusters for Facebook, 9 clusters for Flixster, 3 clusters for Paintings and 5 clusters for Clothing dataset. Beside this, we apply K-means (using the Euclidean distance measure) as the clustering algorithm. Minimum threshold $\epsilon$ has optimal values for Similarity equal to 0.4 and 0.2 in Facebook and Flixster. VP-Sim. achieves better results with $\epsilon = 0.7$ for Paintings and $\epsilon = 0.5$ to Clothing. Centrality has optimal $\epsilon$ value equal to 0.1 for Facebook, Flixster, FilmTrust and Epinions. While Mutual has best results to $\epsilon = 0.4$ for Facebook, 0.2 for Flixster and Epinions and 0.1 to FilmTrust.

## 5.5 Result and Discussion

Figure 19 presents the histograms of MAE (upper part) and RMSE (lower part) for each approach, and for all datasets. Recall that those metrics are negatively-oriented scores: lower values are better.

We can see that ToSocialRec methods perform better than the other methods it is compared to: SoRec, SocialMF, and TrustMF. Specifically, comparing the best result among of these three state of the art algorithms against ours, we note the following improvements in MAE per each dataset: 10.78% on Facebook, 27.22% on Flixster, 17.61% on FilmTrust, 26.43% on Epinions, 2.74% on Paintings, and 6.72% on Clothing.

Results in terms of RMSE were similar to MAE. The improvements in RMSE are: 8.73% on Facebook, 22.29% on Flixster, 11.42% on FilmTrust, 19.69% on Epinions, 4.07% on Paintings, and 4.85% on Clothing.

Tables 27 to 32 present the $nDCG$ at rank positions 5, 10, 15, and 20. For each approach and for each dataset, the largest values is indicated by boldface in each column . The lower part of each table shows the results of ToSocialRec using different Preference-

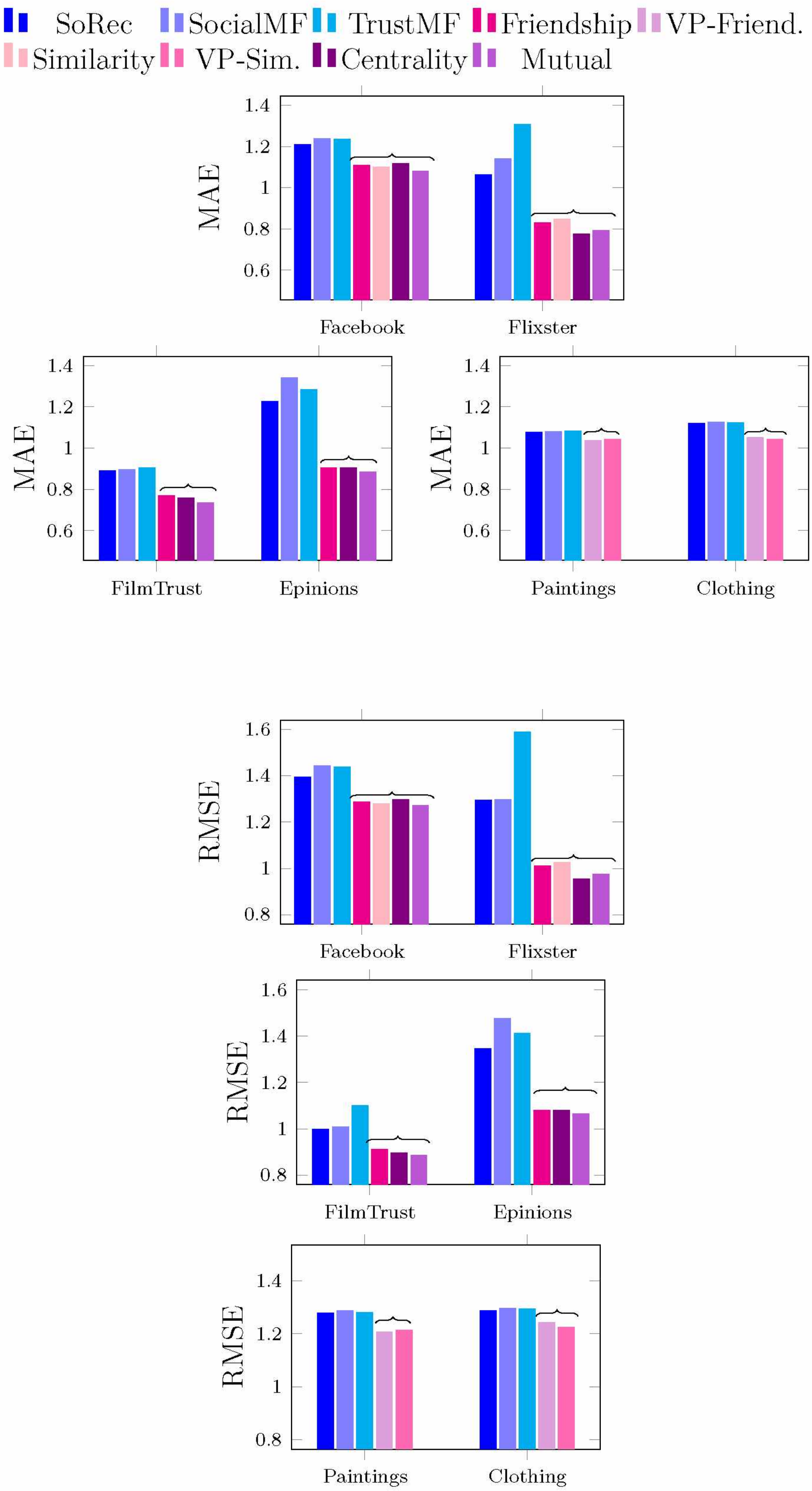


Figure 19 – MAE and RMSE histograms for each approach per dataset under cold-start scenario (0-rating protocol). Please note that in each pane, the 3 leftmost blueish bars are algorithms we compare ToSocialRec to, while the pinkisk rightmost bars (over-braced) are variant of ToSocialRec.

like score methods. Although we can see that ToSocialRec is performing better than the other methods, the difference is quite small on some datasets. For instance, Table 27 presents SoRec achieving 0.8537 for *nDCG*@5 while ToSocialRec Centrality score is 0.8541.

#### Statistical Analysis

We checked the normality and homogeneity of the results for each method for each metric (MAE, RMSE, and nDCG) using Shapiro and Bartlett test. We observed that the results are not normally distributed and not homogeneous. Therefore, we performed the global comparisons with Kruskal-Wallis test.

As measured in terms of MAE and RMSE, ToSocialRec produces better results with 95% of confidence. Remarkably, between *Mutual and Centrality* methods (our best results), there is no statistically significant difference. The analysis of nDCG brings slightly different results. Although ToSocialRec, using *Mutual, Centrality, and Friendship*, produced overall significantly better nDCG results, *SoRec* scores the same as *VP-Friendship* and *VP-Similarity*, again with a 95% confidence level.

Table 27 – nDCG for cold-start scenario (0-rating protocol) against Facebook dataset.

| Approach | Rank size @5 | @10 | @15 | @20 |
|---|---|---|---|---|
| SoRec | 0.8537 ± .132 | 0.8457 ± .117 | 0.8441 ± .107 | 0.8424 ± .109 |
| SocialMF | 0.8226 ± .136 | 0.8205 ± .122 | 0.8202 ± .114 | 0.8240 ± .115 |
| TrustMF | 0.8509 ± .136 | 0.8445 ± .118 | 0.8427 ± .109 | 0.8428 ± .109 |
| Friendship | 0.8549 ± .130 | 0.8475 ± .114 | 0.8451 ± .106 | 0.8437 ± .107 |
| Similarity | 0.8562 ± .131 | 0.8490 ± .114 | 0.8469 ± .106 | 0.8454 ± .107 |
| Centrality | 0.8541 ± .130 | 0.8447 ± .115 | 0.8433 ± .106 | 0.8461 ± .108 |
| Mutual | **0.9054** ± .076 | **0.8869** ± .064 | **0.8840** ± .055 | **0.8716** ± .055 |

Table 28 – nDCG for cold-start scenario (0-rating protocol) against Flixter dataset.

| Approach | Rank size @5 | @10 | @15 | @20 |
|---|---|---|---|---|
| SoRec | 0.8226 ± .127 | 0.8197 ± .115 | 0.8171 ± .110 | 0.8156 ± .107 |
| SocialMF | 0.7416 ± .138 | 0.7415 ± .125 | 0.7437 ± .119 | 0.7461 ± .115 |
| TrustMF | 0.7204 ± .135 | 0.7246 ± .122 | 0.7270 ± .117 | 0.7298 ± .113 |
| Friendship | 0.8344 ± .125 | 0.8291 ± .113 | 0.8259 ± .108 | 0.8242 ± .105 |
| Similarity | 0.8376 ± .122 | 0.8331± .112 | 0.8306 ± .107 | 0.8292 ± .104 |
| Centrality | 0.8388 ± .117 | 0.8343 ± .106 | 0.8315 ± .102 | 0.8298 ± .099 |
| Mutual | **0.8530** ± .094 | **0.8428** ± .093 | **0.8382** ± .088 | **0.8384** ± .087 |

Table 29 – nDCG for cold-start scenario (0-rating protocol) against Filmtrust dataset.

| Approach | Rank size | | | |
|---|---|---|---|---|
| | @5 | @10 | @15 | @20 |
| SoRec | 0.8432 ± .129 | 0.8447 ± .111 | 0.8458 ± .103 | 0.8476 ± .097 |
| SocialMF | 0.8444 ± .129 | 0.8460 ± .110 | 0.8476 ± .102 | 0.8489 ± .096 |
| TrustMF | 0.8146 ± .129 | 0.8203 ± .111 | 0.8277 ± .101 | 0.8346 ± .092 |
| Friendship | 0.8456 ± .124 | 0.8500 ± .110 | 0.8517 ± .100 | 0.8547 ± .094 |
| Centrality | 0.8489 ± .117 | 0.8527 ± .102 | 0.8549 ± .094 | 0.8605 ± .089 |
| Mutual | **0.8633** ± .108 | **0.8586** ± .096 | **0.8605** ± .089 | **0.8614** ± .084 |

Table 30 – nDCG for cold-start scenario (0-rating protocol) against Epinions dataset.

| Approach | Rank size | | | |
|---|---|---|---|---|
| | @5 | @10 | @15 | @20 |
| SoRec | 0.9093 ± .089 | 0.9108 ± .069 | 0.9141 ± .059 | 0.9181 ± .054 |
| SocialMF | 0.8983 ± .095 | 0.9021 ± .073 | 0.9063 ± .062 | 0.9109 ± .057 |
| TrustMF | 0.8222 ± .130 | 0.8217 ± .109 | 0.8262 ± .097 | 0.8332 ± .089 |
| Friendship | 0.9141 ± .085 | 0.9144 ± .065 | 0.9172 ± .055 | 0.9224 ± .050 |
| Centrality | 0.9146 ± .085 | 0.9147 ± .065 | 0.9173 ± .055 | 0.9226 ± .050 |
| Mutual | **0.9189** ± .084 | **0.9170** ± .068 | **0.9196** ± .056 | **0.9236** ± .052 |

Table 31 – nDCG for cold-start scenario (0-rating protocol) against Paintings dataset.

| Approach | Rank size | | | |
|---|---|---|---|---|
| | @5 | @10 | @15 | @20 |
| SoRec | 0.8332 ± .126 | 0.8301 ± .110 | 0.8258 ± .101 | 0.8219 ± .098 |
| SocialMF | 0.7187 ± .153 | 0.6961 ± .130 | 0.6818 ± .117 | 0.6663 ± .113 |
| TrustMF | 0.6524 ± .145 | 0.6576 ± .130 | 0.6668 ± .121 | 0.6736 ± .115 |
| VP-Friend. | 0.8403 ± .124 | 0.8307 ± .112 | 0.8289 ± .101 | 0.8232 ± .098 |
| VP-Sim. | **0.8434** ± .127 | **0.8329** ± .113 | **0.8313** ± .101 | **0.8257** ± .098 |

Table 32 – nDCG for cold-start scenario (0-rating protocol) against Clothing dataset.

| Approach | Rank size | | | |
|---|---|---|---|---|
| | @5 | @10 | @15 | @20 |
| SoRec | 0.7662 ± .157 | 0.7559 ± .137 | 0.7572 ± .128 | 0.7632 ± .119 |
| SocialMF | 0.7715 ± .153 | 0.7638 ± .134 | 0.7628 ± .125 | 0.7715 ± .120 |
| TrustMF | 0.7684 ± .147 | 0.7676 ± .129 | 0.7677 ± .123 | 0.7726 ± .118 |
| VP-Friend. | 0.7769 ± .152 | 0.7703 ± .130 | 0.7731 ± .122 | 0.7744 ± .113 |
| VP-Sim. | **0.7785** ± .151 | **0.7709** ± .134 | **0.7732** ± .124 | **0.7757** ± .116 |

### Discussion

We asked whether a recommender system, based on its current set of prediction models that have been built considering current users, might offer accurate recommendations for cold users. To this question, it seems the answer is positive.

**RQ#1: How well can a selected model predicts the ratings of cold users?**
The use of existing prediction models pays off: by exploiting them, ToSocialRec can predict item rating better than a personalized new one built with just social information.

**RQ#2: How well can a selected model rank items for a cold user?**
Two of our Preference-like functions did not present better results. However, selecting a model instead of building a new one can still lead to high quality items ranking.

Our experimental results on real datasets indicates that ToSocialRec performs better or at least equivalently as all methods we compare to for the cold-start user.

### Limitations

The insights from this work are limited by the methodology and the dataset that have been used. The main limitation is that we conducted our experiments comparing Matrix Factorization approaches. While we test on real, diverse and well studied datasets, ToSocialRec might not yet perform better than other recommender methods. Future work could compare a larger set of state of the art systems.

Another threat arises from the randomized sampling of the original Flixter and Epinions datasets. Due to computational resource limitations we deliberately reduce the size of those datasets. The problem is about parameter selection. For example, setting the number of clusters requires running the model with multiple parameter values and then selecting the best one. However, we feel rather confident about our results because we test our hypothesis on four other datasets. In addition, ToSocialRec achieves better results on sparser datasets, which is the situation met in real applications. Future work could investigate not only how to set the parameters in a more scalable way, but might evaluate the approach against very large datasets.

## 5.6 Summary

Social information is often massive, and social recommender system are already taking advantage of this source of information. In this work, we proposed a novel approach to exploit existing prediction models instead of creating new ones, which allows improving the recommendations for cold-start users. We studied the network metrics in social and non-social contexts. We found that using a similarity score, dubbed Preference-like, among users of a recommender system is capable to accurately recommend items for cold users.

The dominant trend in SRS has been towards designing new prediction models using social data. However, in many real-life situations, integrating new models into legacy systems may not be possible. Furthermore, the results of this work suggest that it is fruitful to explore the predictions and users already using the recommendation system. The experiments provided statistical evidences that these existing models and users hold enough information to lead to more accurate item recommendations for cold users.

Overall, ToSocialRec makes the following main contributions:

1. We compare and contrast several matrix factorization based social recommender systems in a cold user scenario;

2. We propose a general model selection approach that leverages existing prediction models to offer items for cold users;

3. We thoroughly evaluate our approach on six distinct datasets and show its effectiveness in contrast to state-of-the-art matrix factorization recommenders.

Finally, we can observe that Social PrefRec and VP-Rec, presented in Chapters 3 and 4 respectively presented better results than ToSocialRec in terms of nDCG over the same datasets. These results indicated a better performance of hybrid systems over collaborative approaches, mainly in dense datasets. On the other hand, our experiments showed good performance of matrix factorization approach (including ToSocialRec) over more sparse datasets. Based on these findings we can think that combining the two approaches (matrix factorization and pairwise preferences) might be direction to future work.

Chapter 6

# Learning to Select Prediction Model at Cold-Start Stage[1]

## 6.1 Introduction

Given a cold user, say, a new Netflix' client, how can we effectively recommend movies to him? In most existing works, a typical recommender system will request initial ratings (CHANG; HARPER; TERVEEN, 2015) and/or it will harvest the World Wide Web looking for the user's tastes (MISLOVE et al., 2010) to bootstrap the system. But apart from that, what happens if we do not have the right information to build the user's profile? So, how can we estimate the tastes of a cold user without prior side information? The focus of this chapter is also on offering better recommendations for cold users.

Different prediction models are used to deal with distinct stages of a user experience. For example, a particular model works better in earlier stages when the recommender system does not know the user. However, in later stages, a different model should be more effective, and therefore, one switches to the more powerful model. Switching methods (BURKE, 2002) were designed to handle the cold-start problem. The idea is to switch from one model to another once the system has enough data about the user, so he is no longer cold.

While the concept of switching models (BILLSUS; PAZZANI, 2000) is not new, the availability of several cold-start methods provides enriched resources to *model selection*. Applied to the cold-start stage, a model selection method may be seen as a framework to alternate among prediction models to find the most suitable one at each time, while the user warms-up. Few works have sought to empirically assess the efficacy of a model selection, specifically at the cold-start stage (BRAUNHOFER; CODINA; RICCI, 2014; TANG et al., 2014).

In this chapter, we propose PdMS (Fig. 20), an effective *Prediction Model Selection*

[1] This chapter was published in a modified form in (FELÍCIO et al., 2017)

method to deal with individuals without prior side information. A primary adoption inhibitor to many proposed solutions is that they rely on side information that might not be available. Such side information may be of different kinds, such as social information (ALAHMADI; ZENG, 2015), user click behavior (LIU; DOLAN; PEDERSEN, 2010), location-based information (CHENG et al., 2012a), user's visual perception (MELO; NOGUEIRA; GULIATO, 2015), and, broadly, contextual information (ADOMAVICIUS; TUZHILIN, 2015).

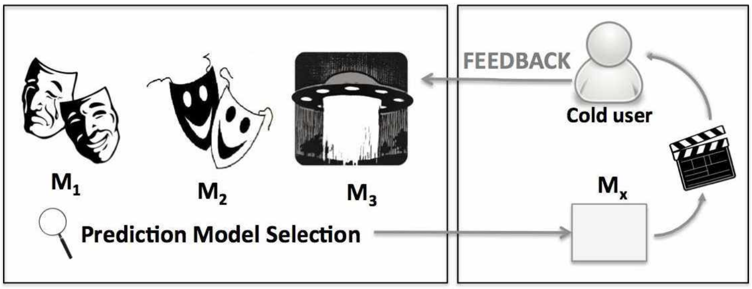


Figure 20 – PdMS relies on feedback to learn how to appropriately select a consensual prediction model to deliver high performance for cold users.

The first insight is to explore how a model selection is useful to provide better recommendations to cold users. The reason we choose to select prediction models instead of building new ones is that cold users might be best served by models already built in the system (FELÍCIO et al., 2016e).

The second insight is to design our goal as a multi-armed bandit (MAB) problem (AUER; CESA-BIANCHI; FISCHER, 2002a). Recommender systems dealing with a cold user repeatedly offer items yielding uncertain returns. In this situation of sequential decision-making under uncertainty, actions should be selected to find a suitable prediction model and to investigate other ones (explore vs. exploit). The idea is to be benefited from the relative performance of various prediction models (EKSTRAND; RIEDL, 2012). Therefore, a model selection that maximizes the recommendation gain might be more precise.

Our primary contributions are:

- We show how to formalize the model selection problem as a multi-armed bandit problem.
- We design PdMS, an effective approach to deal with cold users, i.e. users without prior side information.
- We empirically test PdMS against four real, public datasets.

This chapter is organized as follows. First, we further motivate the need for a new cold-start method and its main concept (Section 6.2). Next, we present the approach and how it works (Section 6.3). Section 6.4 describes our experimental settings and results. Then, Section 6.5 discusses our results, and Section 6.6 concludes the Chapter.

## 6.2 Background

As a motivating example, let us consider the domain of movie recommendations. Based on the ratings already given we can apply a recommendation algorithm and make predictions for the movies not already rated using the personalized preferences from each user. User's preferences might indicate a preferred movie genre, director, actors, etc., and be represented by a prediction model.

Now, let us assume a new (cold) user $u$ looking for help to find movies he would like to watch. A common situation is the cold user gives no initial rating. However, he is admitted into the system, without offering information about his tastes. The cold user probably has similar tastes to other users in the system. Then, one way to recommend to him is switches between the prediction model of others users for the initial recommendations. Later, as the user provides ratings, the system improves the model.

Considering for each recommended movie the cold user will give a feedback that will be used to make the next recommendation. So, we will first explore the different prediction models and then use feedback to define the best one.

To understand our approach, called PdMS, we introduce some terminology and notations on recommender systems, and on multi-armed bandits.

Let $U$ be a set of $m$ users and $I$ be a set of $n$ items. Each user $u \in U$ and each item $i \in I$ has a unique identifier. The user-item rating matrix is $R = [r_{u,i}]_{m \times n}$, where each entry $r_{u,i}$ is either the rating given by user $u$ on item $i$, or unknown. The recommendation task is based on the prediction of the missing values of the user-item rating matrix. Then, prediction models are used to rank items and recommend the $k$ top-ranked.

A Multi-Armed Bandit problem is a sequential decision problem where an algorithm continually chooses among a set of possible actions (arms) which we assume to be finite in this work. At each time step $t$, an arm $a$ is selected and pulled which leads to a reward $X_a(t)$. This reward is distributed according to a certain unknown law. Here, we consider that the goal is to learn, as fast as possible, through repeated arm pulls, the arm that returns the maximum expected reward.

In this work, we assume a set of prediction models as the arms from Multi-Armed Bandit problem. Then, our bandit algorithm is sequentially applied to choose among the prediction models either the best performing one at the moment (exploitation), or an other arm to better learn how it performs (exploration). We rely on the $UCB1$ algorithm (AUER; CESA-BIANCHI; FISCHER, 2002a) and $\epsilon$-Greedy algorithm (SUTTON; BARTO, 1998) to implement our model selection. $UCB1$ maintains the mean reward of each arm (prediction model) $a$, denoted by $\bar{X}_a$. Each time arm $a$ is played, the mean reward $\bar{X}_a$ is updated. The number of pulls of arm $a$ is denoted by $n_a$. In both notations, $t$ is implicit.

In UCB1 algorithm, each arm is initially played a couple of times to estimate its mean reward. Then, at turn $t$, the algorithm selects arm $a(t)$ according to Eq. (11).

$$a(t) = \arg\max_{s=1\ldots k}\left(\bar{X}_{a_s} + \sqrt{\frac{2\ln t}{n_{a_s}}}\right) \tag{11}$$

The chosen arm is the one maximizing the sum of the mean reward $\bar{X}_a$, and a confidence term $ba(t) := \sqrt{\frac{2\ln t}{n_a}}$. At each time $t$, the arm to be played is selected as the one maximizing the balance between immediate profit and the gathering of useful information.

$\epsilon$-Greedy also maintains the mean reward of each arm (prediction model) $a$, denoted by $\bar{X}_a$. At each round, the $\epsilon$-Greedy algorithm selects the arm with the highest mean reward with probability $1-\epsilon$, and selects an arm uniformly at random with probability $\epsilon$. The $\epsilon$ value is given as input parameter to $\epsilon$-Greedy algorithm.

## 6.3 PdMS Approach

Exemplifying our problem, a user-item matrix is represented in Table 33a. Applying the BiasedMF algorithm (KOREN; BELL; VOLINSKY, 2009) we obtained the predicted ratings for the five users in Table 33b. We consider that each row of Table 33b is a prediction model.

When a cold user arrives at the system, we can switch between the 5 prediction models to make recommendations for him. However this is not reasonable in a real-world recommendation system, where we have million of users. Then, our first challenge is reduce the number of prediction models available to the switch process. Observing the prediction models is possible to identify similar values between some users, what lead us to think that clustering users according to their predict ratings is a good way to decrease the number of prediction models and still preserving the users preferences.

Follow this intuition we define PdMS as an algorithm made of two phases: (i) computing and updating of prediction models and (ii) recommendation. Computing and updating of prediction models is based on the proposal presented in Chapter 5.

### 6.3.1 Model Computing and Updating

To define the set of prediction models, we apply the same four steps: Rating prediction, Preference clustering, Consensus computation and Model Update that were defined in Section 5.3.1. We summarized their main features as follow:

- *Rating prediction*: obtained the matrix of predicted ratings $R'$ applying a matrix factorization technique. Given the predicted rating matrix $R'$, the preference vector for a user $u_j$ is defined as the predicted ratings for user $u_j$, $\theta_j = R'_{u_j}$.

Table 33 – (a) Example of a user-item rating matrix. "-" means that the user has not rated the item. (b) Predicted rating matrix. (c) Consensual preference vector.

(a)

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ | $i_6$ | $i_7$ |
|---|---|---|---|---|---|---|---|
| $u_1$ | 5 | 2 | 4 | - | 5 | 1 | - |
| $u_2$ | 4 | - | 5 | - | 5 | - | 1 |
| $u_3$ | 2 | 5 | 3 | 5 | - | - | - |
| $u_4$ | 1 | - | 2 | - | 2 | - | - |
| $u_5$ | - | - | 3 | 4 | 1 | - | - |

(b)

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ | $i_6$ | $i_7$ |
|---|---|---|---|---|---|---|---|
| $u_1$ | 4.6 | 2.09 | 4.23 | 4.24 | 4.84 | 1.07 | 1.0 |
| $u_2$ | 4.2 | 3.8 | 4.42 | 5.0 | 4.86 | 2.28 | 1.2 |
| $u_3$ | 1.97 | 4.84 | 3.22 | 4.87 | 2.68 | 2.68 | 1.61 |
| $u_4$ | 1.19 | 3.24 | 2.17 | 3.56 | 1.92 | 1.23 | 1.0 |
| $u_5$ | 1.77 | 3.16 | 2.81 | 4.07 | 1.14 | 1.6 | 1.56 |

(c)

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ | $i_6$ | $i_7$ |
|---|---|---|---|---|---|---|---|
| $u_1$ | 4.6 | 2.09 | 4.23 | 4.24 | 4.84 | 1.07 | 1.0 |
| $u_2$ | 4.2 | 3.8 | 4.42 | 5.0 | 4.86 | 2.28 | 1.2 |
| $\hat{\theta}_1$ | 4.4 | 2.94 | 4.32 | 4.62 | 4.85 | 1.67 | 1.1 |
| $u_3$ | 1.97 | 4.84 | 3.22 | 4.87 | 2.68 | 2.68 | 1.61 |
| $u_4$ | 1.19 | 3.24 | 2.17 | 3.56 | 1.92 | 1.23 | 1.0 |
| $u_5$ | 1.77 | 3.16 | 2.81 | 4.07 | 1.14 | 1.6 | 1.56 |
| $\hat{\theta}_2$ | 1.64 | 3.74 | 2,73 | 4.16 | 1,91 | 1.83 | 1.39 |

- *Preference clustering*: will cluster users according to their preference vectors, that are the rows of $R'$. After clustering, we have a set of cluster $C$, where each cluster $C_s$ contains a set of users with the similar preferences.

- *Consensus computation*: will applies a consensus operator to get the consensual preference vector $\hat{\theta}_s$ for each cluster $C_s$. Here the operator is also the average, that is $\hat{\theta}_{s,k}$ is the average predicted rating for item $k$. The output is the set of prediction models $M = \{M_0 = (C_1, \hat{\theta}_1), \ldots, M_K = (C_K, \hat{\theta}_K)\}$, where each $M_s$ is composed of a cluster of users $C_s$ and its consensual preference vector $\hat{\theta}_s$. Table 33c shows the result of clustering the predicted rating matrix rows (Table 33b) in 2 clusters and presents their consensual preference vectors $\hat{\theta}_1$ and $\hat{\theta}_2$.

- *Model update*: defined that the set of prediction models $M$ must be rebuilt when the insertion of new ratings in the user-item rating matrix $R$ increases the difference between $R$ and prediction rating matrix $R'$ and reaches a certain threshold.

### 6.3.2 Recommendation

We propose two methods to select a prediction model to determine the items to recommend to a user, Random-MS and PdMS. The Random-MS is a baseline that selects at random a prediction model and PdMS which applies a multi-armed bandit algorithm to that end.

**Random-MS**: receives as input the set of prediction models $M$ and makes a recommendation for a user $u$ at each time $t$ according to the steps:

1. Randomly select a prediction model $M_s$;
2. Randomly select an item $i$ not recommended yet from $\hat{\theta}_s$;
3. Recommend item $i$ to user $u$;
4. Receive a feedback from $u$.

**PdMS**: After getting the prediction models, we sort the consensual preference vectors according to their ratings. So, for each $\hat{\theta}_s$ we have a $\hat{\theta}'_s$ that represents the consensual preference vector in a sorted order. The idea is to recommend the items with high ratings in each model first. We hypothesize that this strategy can contribute to learn users preference faster. For instance, the correspondent $\hat{\theta}'_1$ to $\hat{\theta}_1$ in Table 33c will have the sorted list of items equal to $\{i_5, i_4, i_1, i_3, i_2, i_6, i_7\}$, while for $\hat{\theta}'_2$ we will have $\{i_4, i_2, i_3, i_5, i_6, i_1, i_7\}$. At each time $t$ a recommendation for a user $u$ is made according to a bandit algorithm $\mathcal{B}$ as following:

1. Select a prediction model $M_s$ using the bandit algorithm $\mathcal{B}$;
2. Select the next item $i$ not yet recommended from $\hat{\theta}'_s$ (consensual preference vector of $M_s$ sorted by ratings);
3. Recommend item $i$ to user $u$;
4. Receive a feedback from $u$;
5. Compute the reward;
6. Update the prediction model statistics in $\mathcal{B}$.

#### Computing the Reward

The goal of step 5 in PdMS is compute the reward using the feedback of the recommendation. It is computed applying the following method:

**Feedback-based reward:** We define the reward measure in Eq. (12) based on the feedback of the recommendation. In this way, when a user gives a higher feedback for a recommended item, we will have a high reward. This method is easily adapted to implicit feedback, such as a click, view, purchase, etc.

$$X_{M_s}(t) = \frac{r_{u,i}}{r_{max}} \tag{12}$$

Where $r_{max}$ represents the largest rating in the dataset and $r_{u,i}$ is the feedback of user $u$ for the recommended item $i$ according to the selected model $M_s$. For a implicit feedback we can consider that the value of $r_{u,i}$ is binary. For example, in a music recommender system we will have $r_{u,i} = 1$ if the user listen a recommended song and $r_{u,i} = 0$ otherwise.

*Example*: Consider that the system will make recommendations for a user $u$ based on the consensual prediction models in 33c. It first recommends the item $i_5$, according to $\hat{\theta}'_1$ and receive a feedback equal to 4. Suppose that the $r_{max} = 5$, we will have a reward equal to 0.8.

## 6.4 Experiment Setting

### 6.4.1 Datasets

We evaluate PdMS on 4 real movies datasets. Table 34 summarizes their main features.

Table 34 – Dataset features.

| *Dataset* | Users | Items | Ratings | Sparsity (%) |
|---|---|---|---|---|
| Facebook | 498 | 169 | 49,729 | 40.9 |
| FilmTrust | 1,508 | 2,071 | 35,494 | 98.86 |
| Movielens | 943 | 1,682 | 100,000 | 93.7 |
| *Flixster* | 1,323 | 1,175 | 811,726 | 47.78 |

***Facebook Dataset*** was described in Section 3.4.1 and is composed by users who rated unless 20 movies. Ratings range from 1 to 5.

***Filmtrust Dataset*** was described in Section 5.4.1. Ratings range from 0.5 (min) to 4 (max).

***Movielens Dataset*** (HARPER; KONSTAN, 2015) collected by the GroupLens Research Project contains movies ratings from users that rated at least 20 movies in a 1 to 5 range.

***Flixster Dataset*** was also described in Section 3.4.1. Ratings range from 0.5 to 5.

### 6.4.2 Evaluation Criteria

We perform the experiments using the **0-rating protocol**. Ranking quality is measured computing the Normalized Discounted Cumulative Gain (nDCG) metric. We adapt

Eq. (4) to Eq. (13). In that equation, $r_{u,1}$ is the rating (according to the user feedback) of the item first recommended for user $u$, $t$ is the recommendation time, $r_{u,t}$ is the user feedback for the item recommended in turn $t$ and $T$ is the size of the ranked list. $DCG(u)$, see Eq. (14), is the discounted cumulative gain of predicted ranking for a target user $u$, $DCG^*(u)$ is the discounted cumulative gain of the user feedback and $N$ is the number of users in the result set.

$$nDCG = \frac{1}{N} \sum_{u} \frac{DCG(u)}{DCG^*(u)} \tag{13}$$

$$DCG(u) = r_{u,1} + \sum_{t=2}^{T} \frac{r_{u,t}}{\log_2 t} \tag{14}$$

### 6.4.3 Comparison Methods

To assess the effectiveness of our PdMS model, we compare it to the following baselines:

**Global Average:** A standard "popular" baseline, which recommends using the global average rate for an item.

**Random-MS:** Random selection of prediction models, see Section 6.3.2.

*Parameter Settings.* We use LibRec (GUO et al., 2015), which provides an implementation of Global Average. The implementation of Random-MS and PdMS was built on top of BiasedMF algorithm in LibRec library. We cluster BiasedMF prediction models and in the recommendation process we include the implementation of Random-MS, PdMS UCB1 and PdMS $\epsilon$-Greedy algorithm.

Experiments were executed with 10 latent factors and 100 iterations. We executed Random-MS 5 times and get the average result. With PdMS, we also experimentally test several cluster size. Then we set the optimal number of clusters to 4 clusters for FilmTrust, 3 clusters for Facebook, Movielens and Flixster. Beside this, we apply K-means (using the Euclidean distance measure) as the clustering algorithm. For PdMS $\epsilon$-Greedy the optimal value of $\epsilon$ is 0.3 to Facebook dataset and 0.2 to Filmtrust, Movielens and Flixster.

## 6.5 Result and Discussion

We aim to answer the following questions:

**Q1:** How effective is PdMS to offer initial recommendations to cold users?

**Q2:** Are the PdMS results reliable, or random and noisy?

Table 35 presents the $nDCG$ at rank size of 5, 10, 15, and 20 per method. Note that PdMS has two variants, UCB1 and $\epsilon$-Greedy.

Table 35 – nDCG per method against each dataset.

(a) Facebook

| Method | Rank size @5 | @10 | @15 | @20 |
|---|---|---|---|---|
| Global Average | 0.728 | 0.734 | 0.740 | 0.749 |
| Random-MS | 0.714 | 0.719 | 0.725 | 0.732 |
| PdMS $\epsilon$-Greedy | 0.858 | 0.850 | **0.850** | **0.849** |
| PdMS UCB1 | **0.859** | **0.851** | **0.850** | 0.848 |

(b) Filmtrust

| Method | Rank size @5 | @10 | @15 | @20 |
|---|---|---|---|---|
| Global Average | 0.800 | 0.808 | 0.814 | 0.821 |
| Random-MS | 0.805 | 0.811 | 0.815 | 0.818 |
| PdMS $\epsilon$-Greedy | 0.845 | 0.848 | 0.850 | 0.852 |
| PdMS UCB1 | **0.859** | **0.857** | **0.860** | **0.861** |

(c) Movielens

| Method | Rank size @5 | @10 | @15 | @20 |
|---|---|---|---|---|
| Global Average | 0.735 | 0.749 | 0.765 | 0.780 |
| Random-MS | 0.729 | 0.739 | 0.755 | 0.772 |
| PdMS $\epsilon$-Greedy | 0.857 | 0.858 | 0.865 | 0.874 |
| PdMS UCB1 | **0.858** | **0.859** | **0.865** | **0.874** |

(d) Flixster

| Method | Rank size @5 | @10 | @15 | @20 |
|---|---|---|---|---|
| Global Average | 0.708 | 0.718 | 0.721 | 0.724 |
| Random-MS | 0.721 | 0.723 | 0.724 | 0.726 |
| PdMS $\epsilon$-Greedy | **0.824** | **0.821** | **0.820** | **0.819** |
| PdMS UCB1 | 0.822 | **0.821** | 0.819 | 0.817 |

## Q1: Recommendation Effectiveness

Comparing the results of PdMS to Global Average, we note the following improvements regarding $nDCG$@5: 13% on Facebook, 5.9% on Filmtrust, 12.2% on Movielens and 11.6% on Flixster. The comparison against Random-MS is similar with the following improvements: 14.4% on Facebook, 5.3% on Filmtrust and 12.8% on Movielens.

From Table 35, we first observe that PdMS consistently outperforms all baselines in all four datasets. Note that on Filmtrust, PdMS gain is smaller. That might be so because of the rating distribution. 68.14% of Filmtrust ratings are greater or equal to 3, see Figure 21.

In particular, the difference between PdMS variants are small, and we now assess their significance.

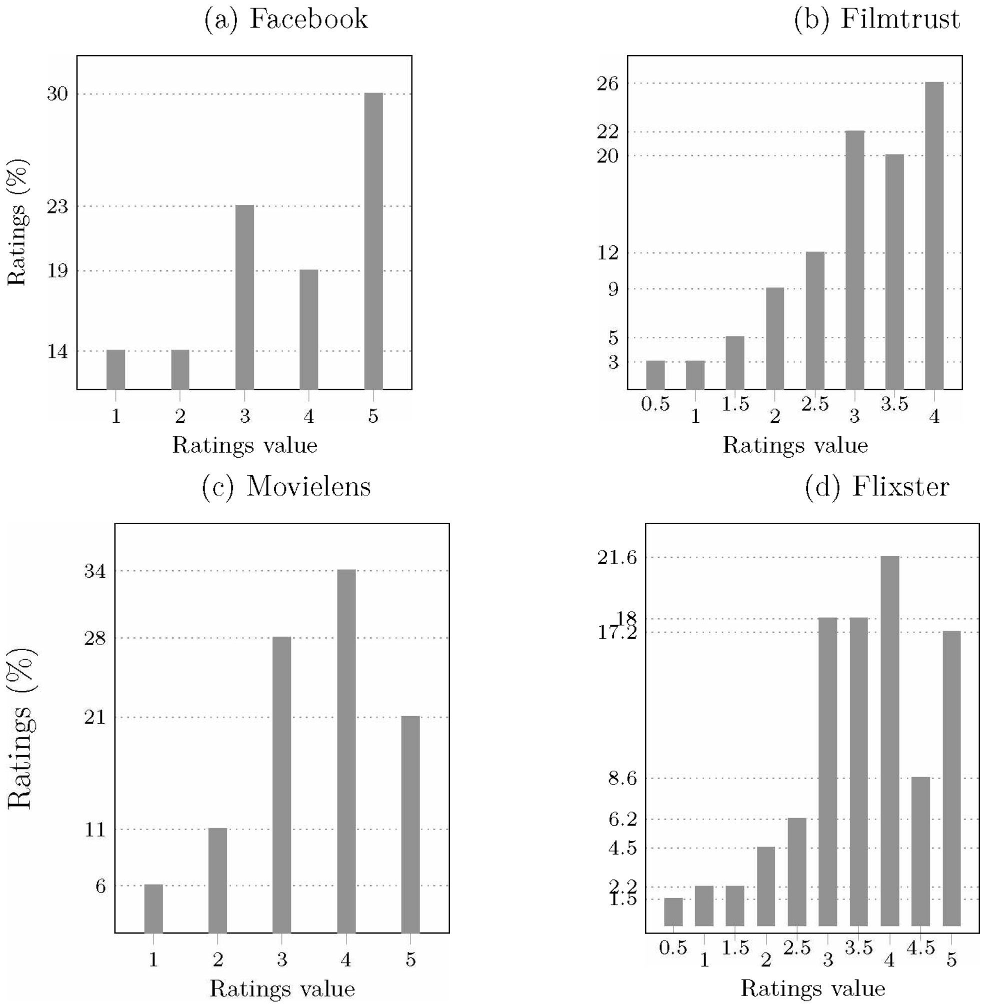


Figure 21 – Rating distribution per dataset.

## Q2: Randomness Analysis

We examine whether the recommendations generated by PdMS are significantly better than those made by baseline methods and its variants on the different datasets by performing a null hypothesis test. We express *H0* as: Recommendations offered from Global Average, Random-MS and PdMS ($\epsilon$-Greedy and UCB1) are distributed identically on all 4 datasets.

We checked the normality of the results with a Shapiro-Wilk test, and their homogeneity of the results ($nDCG$) using a Bartlett test. The tests reject both assumptions of normality, and homogeneity. Then, to check whether the null hypothesis holds, we run Kruskal-Wallis tests on the $nDCG$ results, using the 95% confidence level, (*i.e.*, p-value $< 0.05$). The Kruskal-Wallis test is a non-parametric test to assess whether samples originate from the same distribution.

In particular, we can see from the Table 35 that UCB1 outperforms $\epsilon$-Greedy only on Filmtrust dataset, showing that there are no significant differences between these two approaches. The same happened between our baselines, Global Average and Random-

MS. However, because of the difference between PdMS and the baselines, the p-value of Kruskal-Wallis test is less than 2.2e-16, therefore we reject the null hypothesis $H0$. In conclusion, PdMS performs significantly better than the 2 other baselines.

### Limitations and Future Work

In this study, we considered only one type of collaborative filtering algorithm, the Matrix Factorization. It is not clear whether conclusions generalize beyond this setting. Future work could compare it to state of the art systems. However, the approaches that are compared are all based on the same completed matrix, so that they should all suffer in the same way from the result of the matrix factorization. Further, we compare the two exploration model, $\epsilon$-Greedy and UCB1, however, we did not experiment different exploration parameter in UCB1 algorithm. Future work could also perform experiments with varying exploration parameter and using $\epsilon$-Greedy and UCB1 optimization algorithms.

Another threat arises from the evaluation criteria. We rely on $nDCG$ score to access our approach, mainly because we were investigating the recommendation quality. Overall it presents high accuracy levels, but we might check other metrics to ensure a fair evaluation (KLUVER; KONSTAN, 2014). For example, user coverage study would be required to reveal whether our approach can offer recommendation to a large audience; and likewise, catalog coverage (SAID; BELLOGÍN, 2014).

## 6.6 Summary

In this chapter, we showed how a careful model selection can provide better recommendations to full cold-start user. Furthermore, our approach, PdMS, performed reasonably well even with no side information. It achieves 85% accuracy levels of $nDCG$@5. To sum-up, our contributions are:

- ❏ A formalization of the model selection as a multi-armed bandit problem (Sections 6.2 and 6.3).
- ❏ PdMS, which is an effective approach to recommend for users without prior side information (Section 6.3).
- ❏ An empirical evaluation of PdMS against four real, public datasets (Section 6.4).

Looking forward, PdMS envisions recommender systems in which substantial amount of prediction models is available for analysis, making possible a new wave of intelligent recommender systems.

Chapter 7

# Conclusion and Future Work

## 7.1 As a Conclusion

Offering accurate recommendation to cold-start users is naturally a complex task. The impact of a poor user experience may be large and sometimes unknown. We need to ensure that initial recommendations are consistently performed.

We proposed several approaches to support cold users. We organized them in four parts: (i) analysis of embedding social information into traditional recommender systems, (ii) improvements in recommendation with visual perception similarities, (iii) analysis of the benefits of a general framework to incorporate networked information into recommender systems, and (iv) analysis of the impact of prediction model selection for cold users. Existing approaches lack of a deep understanding of the benefits provided by a general framework to incorporate side-information, a better use of prediction models already built in the system, and an analysis of the impact of prediction model selection from one recommender to deal with cold users.

In this thesis, we argue for the need to analyze and improve recommender systems as to better support cold users experience. We provide (i) new experiments on the incorporation of social information into pairwise recommender system, (ii) a novel approach to deal with cold user using visual perception similarities, (iii) a large study to understand to which extent networked information might impact recommender systems, and (iv) a novel approach to switch prediction models at cold-start stage. We evaluated each approach against real datasets with the use of research questions that were answered when necessary with the use of statistical tests.

We summarize the four approaches in Table 36, highlighting the built systems used to validate each approach, the prediction model type, type of information used to cluster, the type of recommender system used to validate each approach and the type of information used in prediction model selection.

Next, we present a summary of the main results of each approach and we reiterate the most interesting conclusions we derived from our thesis.

| Approach | System | Prediction Model | Clusters | Recommender Type | Model Selection based on |
|---|---|---|---|---|---|
| 1 | Social PrefRec Preference | Pairwise Ratings | Users | Hybrid Network | Online Social |
| 2 | VP-Rec Preference | Pairwise Ratings | Users | Hybrid | VP-Network |
| 3 | ToSocialRec Factorization | Matrix Predictions | Ratings Filtering | Collaborative Network | Any User |
| 4 | PdMS Factorization | Matrix Predictions | Ratings Filtering | Collaborative | User Feedback |

Table 36 – The main features of proposed approaches.

## Pairwise Recommenders with Social Information

This work reported on a systematic study to investigate how to incorporate contextual social information into pairwise recommender systems. The study was performed against two real world dataset, one surveyed by us and another widely used by researchers. We devised Social PrefRec, an approach whose goal is to help pairwise preferences recommender systems to deal with 0-rating user's profile. We also carefully investigate the role of five well-known social metrics in pairwise preference recommendation and proposed a clustering based approach to incorporate social networks into recommender systems. Notably, the *mutual friends* social metric outperform the others to generate recommendations for a cold user (FELÍCIO et al., 2015; FELÍCIO et al., 2016a).

## Improving Recommendation with Visual Perception Similarities

Here, we proposed to two approaches to alleviate user cold-start problems in the domain of item/product recommendation based on images. In these solutions, we relied on extracted networks of visual perceptions to compute similarities among users and select a prediction model that might offer better recommendations. Our results are superior to state-of-art recommenders that are able to handle networked information. Further, visual perception network turned out to be a valuable source of information to handle cold users (FELÍCIO et al., 2016b; FELÍCIO et al., 2016c)

## Benefits of a General Framework to Incorporate Networked Information

Recommender system are already taking advantage of the massive information available from users. In this work, we proposed a novel approach to exploit existing prediction models instead of creating new ones, which allows improving the recommendations for cold users. We studied the network metrics in social and non-social contexts. We propose a general framework to incorporate networked information in recommender systems.

With this, now we are able to integrate networked data into legacy systems which were not possible before. Furthermore, the results of this paper suggest that it is fruitful to explore the predictions and users already using the recommendation system. The experiments provided statistical evidences that these existing models and users hold enough information to lead to more accurate item recommendations for cold users (FELÍCIO et al., 2016e).

### Learning to Select Prediction Model at Cold-Start Stage

Different prediction models are used to deal with distinct stages of a user experience. Herein, we showed how a careful prediction model selection provide better recommendations to full cold-start user. Furthermore, our approach, called PdMS, performed reasonably well even with no side information. It achieves 85% accuracy levels of $nDCG@5$. We contribute to this line of research with a formalization of the model selection as a Multi-Armed Bandit problem based on user feedback, where we enable recommendation without prior knowledge about the user. We advocate that a recommender systems, which substantial amount of prediction models is available for analysis, just need to switch between the models already existent to learning the better one for cold users recommendations. (FELÍCIO et al., 2016d; FELÍCIO et al., 2017).

## 7.2 Future Work

There are some limitations that were not addressed in this thesis, but should be explored in future work.

### Clustering Preferences/Prediction Models

In this thesis, we focus on maximize the existing information in recommender systems. To do so, we rely on the clustering of preferences or prediction models at the first stage of our approaches. The purpose of using clustering is for driving the proposed algorithms. However, we set the clustering parameters, such as the number of clusters, manually. We run our method with multiple parameter values to select the best one. Automatizing such parameters selection might help extend the applicability of our approaches to industry.

On the other hand, considering that in our work we use only K-means as cluster algorithm, explore other cluster techniques mainly for high dimensional data is a good direction to future work.

### User-Centric Directions

Because we were envisioning methods to provide accurate recommendation for cold users, we evaluate our approaches with accuracy metrics. However, those standard metrics might not reflect the usefulness of our approach to cold users. Sometimes the recommendation might not be interesting to a user, although top ranked. So, future work should evaluate our approaches taking into consideration other metrics, such as catalog coverage, but also other aspects of a recommendation. For example, evaluation of serendipity, which is the assessment of the user experience receiving an unexpected recommendation. In the same direction, should be important to evaluate user satisfaction, which often is not expressed by high accuracy (ZIEGLER et al., 2005; MCNEE; RIEDL; KONSTAN, 2006).

### User Preferences Dynamics

Users exploit a wide range of information insofar as modify their tastes over time (RAFAILIDIS; NANOPOULOS, 2014; PEREIRA; AMO; GAMA, 2016). In our experiments we did not evaluate the impact of changes in the users preferences or network features in the proposed approaches. This type of study would be interesting to analyze the systems behavior when we have the prediction models or network updating.

## 7.3 Collaboration

During the Ph.D, I had the opportunity to work with the SequeL - INRIA Lille research group (Lille, France). I was visiting the group from September, 2015 to July, 2016. We have mainly worked in the application of machine learning to recommender systems. So far, we have collaborated in four manuscript (FELÍCIO et al., 2016e; FELÍCIO et al., 2016d; FELÍCIO et al., 2016a; FELÍCIO et al., 2017).

# Bibliography

ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. **IEEE Transactions on Knowledge and Data Engineering**, v. 17, n. 6, p. 734–749, June 2005. ISSN 1041-4347.

ADOMAVICIUS, G.; TUZHILIN, A. Context-aware recommender systems. In: **Recommender Systems handbook**. Cham: Springer, 2015.

ALAHMADI, D. H.; ZENG, X. J. Twitter-based recommender system to address cold-start: A genetic algorithm based trust modelling and probabilistic sentiment analysis. In: **27th International Conference on Tools with Artificial Intelligence**. [S.l.]: IEEE, 2015. (ICTAI), p. 1045–1052. ISSN 1082-3409.

ALMEIDA, C. M. M. de. **Estudo da Percepção Humana e Mineração de Preferências Contextuais na Recomendação de Imagens para o Usuário**. Dissertação (Mestrado) — Universidade Federal de Uberlândia, Uberlândia, 2016.

AMO, S. de et al. Mining user contextual preferences. **JIDM**, v. 4, n. 1, p. 37–46, 2013.

AMO, S. de et al. Contextual preference mining for user profile construction. **Information Systems**, v. 49, p. 182 – 199, 2015. ISSN 0306-4379.

AMO, S. de; OLIVEIRA, C. G. Towards a tunable framework for recommendation systems based on pairwise preference mining algorithms. In: **Advances in Artificial Intelligence: 27th Canadian Conference on Artificial Intelligence, Canadian AI, Montréal, QC, Canada, May 6-9. Proceedings**. Cham: Springer International Publishing, 2014. p. 282–288. ISBN 978-3-319-06483-3.

AMO, S. de; RAMOS, J. Improving pairwise preference mining algorithms using preference degrees. In: **29th Brazilian Symposium on Databases**. Curitiba, Brazil: [s.n.], 2014. (SBBD '14), p. 107–116.

AUER, P.; CESA-BIANCHI, N.; FISCHER, P. Finite-time analysis of the multiarmed bandit problem. **Machine Learning**, v. 47, n. 2-3, p. 235–256, maio 2002. ISSN 0885-6125.

AUER, P.; CESA-BIANCHI, N.; FISCHER, P. Finite-time analysis of the multiarmed bandit problem. **Machine Learning**, v. 47, n. 2, p. 235–256, 2002. ISSN 1573-0565.

AUER, P. et al. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In: **Proceedings of IEEE 36th Annual Foundations of Computer Science**. [S.l.: s.n.], 1995. p. 322–331. ISSN 0272-5428.

BALABANOVIĆ, M.; SHOHAM, Y. Fab: content-based, collaborative recommendation. **Communications of the ACM**, ACM, v. 40, n. 3, p. 66–72, 1997.

BALAKRISHNAN, S.; CHOPRA, S. Two of a kind or the ratings game? adaptive pairwise preferences and latent factor models. **Frontiers of Computer Science**, SP Higher Education Press, v. 6, n. 2, p. 197–208, 2012. ISSN 2095-2228.

BARJASTEH, I. et al. Cold-start recommendation with provable guarantees: A decoupled approach. **IEEE Transactions on Knowledge and Data Engineering**, v. 28, n. 6, p. 1462–1474, June 2016. ISSN 1041-4347.

BASU, C. et al. Recommendation as classification: Using social and content-based information in recommendation. In: **AAAI/IAAI**. [S.l.: s.n.], 1998. p. 714–720.

BELLOGÍN, A.; SAID, A.; VRIES, A. P. de. The magic barrier of recommender systems - no magic, just ratings. In: **User Modeling, Adaptation, and Personalization - 22nd International Conference, UMAP 2014, Aalborg, Denmark, July 7-11, 2014. Proceedings**. Cham.: Springer, 2014. (Lecture Notes in Computer Science, v. 8538), p. 25–36. ISBN 978-3-319-08785-6.

BILLSUS, D.; PAZZANI, M. J. User modeling for adaptive news access. **User Modeling and User-Adapted Interaction**, v. 10, n. 2, p. 147–180, 2000. ISSN 1573-1391.

BOBADILLA, J. et al. Recommender systems survey. **Knowledge-Based Systems**, v. 46, p. 109 – 132, 2013. ISSN 0950-7051.

BOCCALETTI, S. et al. Complex networks: Structure and dynamics. **Physics Reports**, v. 424, n. 4–5, p. 175 – 308, 2006. ISSN 0370-1573.

BONHARD, P.; SASSE, M. A. 'knowing me, knowing you' – using profiles and social networking to improve recommender systems. **BT Technology Journal**, Kluwer Academic Publishers, Hingham, MA, USA, v. 24, n. 3, p. 84–98, jul. 2006. ISSN 1358-3948.

BOUNEFFOUF, D.; BOUZEGHOUB, A.; GANCARSKI, A. L. A contextual-bandit algorithm for mobile context-aware recommender system. In: **Proceedings of the 19th International Conference on Neural Information Processing - Volume Part III**. Berlin, Heidelberg: Springer-Verlag, 2012. (ICONIP'12), p. 324–331. ISBN 978-3-642-34486-2.

BRAUNHOFER, M.; CODINA, V.; RICCI, F. Switching hybrid for cold-starting context-aware recommender systems. In: **Proceedings of the 8th ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2014. (RecSys), p. 349–352. ISBN 978-1-4503-2668-1.

BURKE, R. Hybrid recommender systems: Survey and experiments. **User Modeling and User-Adapted Interaction**, Kluwer Academic Publishers, Hingham, MA, USA, v. 12, n. 4, p. 331–370, nov. 2002. ISSN 0924-1868.

BUSA-FEKETE, R.; HÜLLERMEIER, E. A survey of preference-based online learning with bandit algorithms. In: **Algorithmic Learning Theory - 25th International Conference, ALT 2014, Bled, Slovenia, October 8-10, 2014. Proceedings**. Cham.: Springer, 2014. (Lecture Notes in Computer Science, v. 8776), p. 18–39. ISBN 978-3-319-11661-7.

CARON, S.; BHAGAT, S. Mixing bandits: A recipe for improved cold-start recommendations in a social network. In: **Proceedings of the 7th Workshop on Social Network Mining and Analysis**. New York, NY, USA: ACM, 2013. (SNAKDD '13), p. 11:1–11:9. ISBN 978-1-4503-2330-7.

CARRER-NETO, W. et al. Social knowledge-based recommender system. application to the movies domain. **Expert Systems with Applications**, v. 39, n. 12, p. 10990 – 11000, 2012. ISSN 0957-4174.

CESA-BIANCHI, N. Multi-armed bandit problem. In: KAO, M.-Y. (Ed.). **Encyclopedia of Algorithms**. Boston, MA: Springer US, 2008. p. 1–5. ISBN 978-3-642-27848-8.

CHANG, S.; HARPER, F. M.; TERVEEN, L. Using groups of items for preference elicitation in recommender systems. In: **18th ACM Conference on Computer Supported Cooperative Work & Social Computing**. New York, NY, USA: ACM, 2015. (CSCW), p. 1258–1269. ISBN 978-1-4503-2922-4.

CHENG, C. et al. Fused matrix factorization with geographical and social influence in location-based social networks. In: **AAAI Conference on Artificial Intelligence**. [S.l.]: AAAI, 2012.

CHENG, C. et al. Fused matrix factorization with geographical and social influence in location-based social networks. In: **Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence**. [S.l.]: AAAI Press, 2012. (AAAI'12), p. 17–23.

CHENG, Y.; LIU, J.; YU, X. Online social trust reinforced personalized recommendation. **Personal and Ubiquitous Computing**, v. 20, n. 3, p. 457–467, 2016. ISSN 1617-4917.

CLAYPOOL, M. et al. Combining content-based and collaborative filters in an online newspaper. In: **Proceedings of ACM SIGIR workshop on recommender systems**. [S.l.: s.n.], 1999. v. 60.

COHEN, M. P. Stratified sampling. In: LOVRIC, M. (Ed.). **International Encyclopedia of Statistical Science**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 1547–1550. ISBN 978-3-642-04898-2.

COHEN, W. W.; SCHAPIRE, R. E.; SINGER, Y. Learning to order things. **J. Artif. Int. Res.**, AI Access Foundation, USA, v. 10, n. 1, p. 243–270, maio 1999. ISSN 1076-9757.

DELPORTE, J. et al. Socially enabled preference learning from implicit feedback data. In: **Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD, Prague, Czech Republic, September 23-27, Proceedings, Part II**. Berlin, Heidelberg: Springer, 2013. p. 145–160. ISBN 978-3-642-40991-2.

DOOMS, S.; PESSEMIER, T.; MARTENS, L. Online optimization for user-specific hybrid recommender systems. **Multimedia Tools Appl.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 74, n. 24, p. 11297–11329, dez. 2015. ISSN 1380-7501.

DRENNER, S.; SEN, S.; TERVEEN, L. Crafting the initial user experience to achieve community goals. In: **Proceedings of the 2008 ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2008. (RecSys), p. 187–194. ISBN 978-1-60558-093-7.

DROR, G. et al. The yahoo! music dataset and kddcup. In: **in JMLR Workshop and Conference Proceedings: Proceedings of KDD Cup Competition**. [S.l.: s.n.], 2012. p. 3–18.

EKSTRAND, M.; RIEDL, J. When recommenders fail: Predicting recommender failure for algorithm selection and combination. In: **Proceedings of the Sixth ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2012. (RecSys), p. 233–236. ISBN 978-1-4503-1270-7.

EKSTRAND, M. D. et al. Letting users choose recommender algorithms: An experimental study. In: **Proceedings of the 9th ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2015. (RecSys), p. 11–18. ISBN 978-1-4503-3692-5.

FELÍCIO, C. Z. et al. Social prefrec framework: leveraging recommender systems based on social information. In: **3rd Symposium on Knowledge Discovery, Mining and Learning**. Petrópolis (RJ), Brazil: [s.n.], 2015. (KDMiLe), p. 66–73.

FELÍCIO, C. Z. et al. Exploiting social information in pairwise preference recommender system. **Journal of Information and Data Management (JIDM)**, v. 7, n. 2, p. 99–115, 2016.

FELÍCIO, C. Z. et al. Visual perception similarities to improve the quality of user cold start recommendations. In: **Advances in Artificial Intelligence: 29th Canadian Conference on Artificial Intelligence**. Victoria, BC, Canada: Springer International Publishing, 2016. (Lecture Notes in Computer Science, v. 9673), p. 96–101. ISBN 978-3-319-34110-1.

FELÍCIO, C. Z. et al. Vp-rec: A hybrid image recommender using visual perception network. In: **28th International Conference on Tools with Artificial Intelligence**. San Jose (CA), USA: IEEE, 2016. (ICTAI), p. 70–77. ISSN 2375-0197.

FELÍCIO, C. Z. et al. Multi-armed bandits to recommend for cold-start user. In: **4rd Symposium on Knowledge Discovery, Mining and Learning**. Recife (PE), Brazil: [s.n.], 2016. (KDMiLe), p. 66–73.

FELÍCIO, C. Z. et al. Preference-like score to cope with cold-start user in recommender systems. In: **28th International Conference on Tools with Artificial Intelligence**. San Jose (CA), USA: IEEE, 2016. (ICTAI), p. 62–69. ISSN 2375-0197.

FELÍCIO, C. Z. et al. A multi-armed bandit model selection for cold-start user recommendation. In: **Proceedings of the 2017 Conference on User Modeling Adaptation and Personalization**. [S.l.: s.n.], 2017. (UMAP 17).

GIRGIN, S. et al. Managing advertising campaigns — an approximate planning approach. **Frontiers of Computer Science**, v. 6, n. 2, p. 209–229, 2012. ISSN 1673-7466.

GOMEZ-URIBE, C. A.; HUNT, N. The netflix recommender system: Algorithms, business value, and innovation. **ACM Transactions on Management Information Systems**, ACM, New York, NY, USA, v. 6, n. 4, p. 13:1–13:19, dez. 2015. ISSN 2158-656X.

GOOD, N. et al. Combining collaborative filtering with personal agents for better recommendations. In: **AAAI/IAAI**. [S.l.: s.n.], 1999. p. 439–446.

GROH, G.; EHMIG, C. Recommendations in taste related domains: Collaborative filtering vs. social filtering. In: **Proceedings of the 2007 International ACM Conference on Supporting Group Work**. New York, NY, USA: ACM, 2007. (GROUP '07), p. 127–136. ISBN 978-1-59593-845-9.

GUNAWARDANA, A.; MEEK, C. A unified approach to building hybrid recommender systems. In: **Proceedings of the Third ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2009. (RecSys '09), p. 117–124. ISBN 978-1-60558-435-5.

GUO, G. et al. Librec: A java library for recommender systems. In: **Posters, Demos, Late-breaking Results and Workshop Proc. 23rd Conf. on User Modeling, Adaptation, and Personalization (UMAP), Dublin, Ireland, June.** [S.l.: s.n.], 2015.

GUO, G.; ZHANG, J.; YORKE-SMITH, N. A novel bayesian similarity measure for recommender systems. In: **Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence**. [S.l.]: AAAI Press, 2013. (IJCAI), p. 2619–2625. ISBN 978-1-57735-633-2.

GUO, G.; ZHANG, J.; YORKE-SMITH, N. A novel evidence-based bayesian similarity measure for recommender systems. **ACM Trans. Web**, ACM, New York, NY, USA, v. 10, n. 2, p. 8:1–8:30, maio 2016. ISSN 1559-1131.

GUY, I. Social recommender systems. In: **Recommender Systems Handbook**. Boston, MA: Springer US, 2015. p. 511–543. ISBN 978-1-4899-7637-6.

HARPER, F. M.; KONSTAN, J. A. The movielens datasets: History and context. **ACM Transactions on Interactive Intelligent Systems**, ACM, New York, NY, USA, v. 5, n. 4, p. 19:1–19:19, dez. 2015. ISSN 2160-6455.

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.

JAMALI, M.; ESTER, M. A matrix factorization technique with trust propagation for recommendation in social networks. In: **Proceedings of the Fourth ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2010. (RecSys), p. 135–142. ISBN 978-1-60558-906-0.

KAUTZ, H.; SELMAN, B.; SHAH, M. Referral web: Combining social networks and collaborative filtering. **Commun. ACM**, ACM, New York, NY, USA, v. 40, n. 3, p. 63–65, mar. 1997. ISSN 0001-0782.

KLUVER, D.; KONSTAN, J. A. Evaluating recommender behavior for new users. In: **Proceedings of the 8th ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2014. (RecSys '14), p. 121–128. ISBN 978-1-4503-2668-1.

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.

LACERDA, A. et al. Improving daily deals recommendation using explore-then-exploit strategies. **Inf. Retr.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 18, n. 2, p. 95–122, abr. 2015. ISSN 1386-4564.

LACERDA, A.; VELOSO, A.; ZIVIANI, N. Exploratory and interactive daily deals recommendation. In: **Proceedings of the 7th ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2013. (RecSys '13), p. 439–442. ISBN 978-1-4503-2409-0.

LI, H.; WU, D.; MAMOULIS, N. A revisit to social network-based recommender systems. In: **Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval**. New York, NY, USA: ACM, 2014. (SIGIR '14), p. 1239–1242. ISBN 978-1-4503-2257-7.

LI, L. et al. A contextual-bandit approach to personalized news article recommendation. In: **Proceedings of the 19th International Conference on World Wide Web**. New York, NY, USA: ACM, 2010. (WWW '10), p. 661–670. ISBN 978-1-60558-799-8.

LI, S.; KARATZOGLOU, A.; GENTILE, C. Collaborative filtering bandits. In: **Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: ACM, 2016. (SIGIR '16), p. 539–548. ISBN 978-1-4503-4069-4.

LIU, F.; LEE, H. J. Use of social network information to enhance collaborative filtering performance. **Expert Systems with Applications**, v. 37, n. 7, p. 4772 – 4778, 2010. ISSN 0957-4174.

LIU, J.; DOLAN, P.; PEDERSEN, E. R. Personalized news recommendation based on click behavior. In: **15th International Conference on Intelligent User Interfaces**. Hong Kong, China: ACM, 2010. p. 31–40.

LOPS, P.; GEMMIS, M.; SEMERARO, G. Content-based recommender systems: State of the art and trends. In: RICCI, F. et al. (Ed.). **Recommender Systems Handbook**. [S.l.]: Springer US, 2011. p. 73–105.

MA, H. An experimental study on implicit social recommendation. In: **Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: ACM, 2013. (SIGIR '13), p. 73–82. ISBN 978-1-4503-2034-4.

MA, H. On measuring social friend interest similarities in recommender systems. In: **Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval**. New York, NY, USA: ACM, 2014. (SIGIR '14), p. 465–474. ISBN 978-1-4503-2257-7.

MA, H. et al. Sorec: Social recommendation using probabilistic matrix factorization. In: **Proceedings of the 17th ACM Conference on Information and Knowledge Management**. New York, NY, USA: ACM, 2008. (CIKM '08), p. 931–940. ISBN 978-1-59593-991-3.

MA, H. et al. Improving recommender systems by incorporating social contextual information. **ACM Trans. Inf. Syst.**, ACM, New York, NY, USA, v. 29, n. 2, p. 9:1–9:23, abr. 2011. ISSN 1046-8188.

MACEDO, A. Q.; MARINHO, L. B.; SANTOS, R. L. Context-aware event recommendation in event-based social networks. In: **Proceedings of the 9th ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2015. (RecSys '15), p. 123–130. ISBN 978-1-4503-3692-5.

MARY, J.; GAUDEL, R.; PREUX, P. Bandits and recommender systems. In: SPRINGER. **International Workshop on Machine Learning, Optimization and Big Data**. [S.l.], 2015. p. 325–336.

MASSA, P.; AVESANI, P. Trust-aware recommender systems. In: **Proceedings of the 2007 ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2007. (RecSys '07), p. 17–24. ISBN 978-1-59593-730–8.

MCAULEY, J.; PANDEY, R.; LESKOVEC, J. Inferring networks of substitutable and complementary products. In: **21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2015. (KDD), p. 785–794. ISBN 978-1-4503-3664-2.

MCAULEY, J. et al. Image-based recommendations on styles and substitutes. In: **38th International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: ACM, 2015. (SIGIR), p. 43–52. ISBN 978-1-4503-3621-5.

MCNEE, S. M.; RIEDL, J.; KONSTAN, J. A. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In: **CHI '06 Extended Abstracts on Human Factors in Computing Systems**. New York, NY, USA: ACM, 2006. (CHI EA '06), p. 1097–1101. ISBN 1-59593-298-4.

MCPHERSON, M.; SMITH-LOVIN, L.; COOK, J. M. Birds of a feather: Homophily in social networks. **Annual review of sociology**, JSTOR, p. 415–444, 2001.

MELO, E. V. de; NOGUEIRA, E. A.; GULIATO, D. Content-based filtering enhanced by human visual attention applied to clothing recommendation. In: **27th International Conference on Tools with Artificial Intelligence**. [S.l.]: IEEE, 2015. (ICTAI), p. 644–651. ISSN 1082-3409.

MELVILLE, P.; MOONEY, R. J.; NAGARAJAN, R. Content-boosted collaborative filtering for improved recommendations. In: **Eighteenth National Conference on**

**Artificial Intelligence**. Menlo Park, CA, USA: American Association for Artificial Intelligence, 2002. p. 187–192. ISBN 0-262-51129-0.

MISLOVE, A. et al. Measurement and analysis of online social networks. In: **Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement**. New York, NY, USA: ACM, 2007. (IMC), p. 29–42. ISBN 978-1-59593-908-1.

MISLOVE, A. et al. You are who you know: Inferring user profiles in online social networks. In: **Proceedings of the Third ACM International Conference on Web Search and Data Mining**. New York, NY, USA: ACM, 2010. (WSDM '10), p. 251–260. ISBN 978-1-60558-889-6.

MUTLU, B.; VEAS, E.; TRATTNER, C. Vizrec: Recommending personalized visualizations. **ACM Trans. Interact. Intell. Syst.**, ACM, New York, NY, USA, v. 6, n. 4, p. 31:1–31:39, nov. 2016. ISSN 2160-6455.

NIELSEN, T. D.; JENSEN, F. V. **Bayesian networks and decision graphs**. [S.l.]: Springer Science & Business Media, 2009.

NING, X.; KARYPIS, G. Sparse linear methods with side information for top-n recommendations. In: **ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2012. (RecSys), p. 155–162. ISBN 978-1-4503-1270-7.

PAZZANI, M. J. A framework for collaborative, content-based and demographic filtering. **Artificial Intelligence Review**, Springer, v. 13, n. 5-6, p. 393–408, 1999.

PAZZANI, M. J.; BILLSUS, D. Content-based recommendation systems. In: **The adaptive web**. [S.l.]: Springer, 2007. p. 325–341.

PENNOCK, D. M. et al. Collaborative filtering by personality diagnosis: A hybrid memory and model-based approach. In: **Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000. (UAI '00), p. 473–480. ISBN 1-55860-709-9.

PEREIRA, A. L. V.; HRUSCHKA, E. R. Simultaneous co-clustering and learning to address the cold start problem in recommender systems. **Know.-Based Syst.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 82, n. C, p. 11–19, jul. 2015. ISSN 0950-7051.

PEREIRA, F. S. F.; AMO, S. de; GAMA, J. On using temporal networks to analyze user preferences dynamics. In: **Discovery Science - 19th International Conference, Bari, Italy, October 19-21, 2016, Proceedings**. Cham.: Springer, 2016. (Lecture Notes in Computer Science, v. 9956), p. 408–423. ISBN 978-3-319-46306-3.

POPESCUL, A.; PENNOCK, D. M.; LAWRENCE, S. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In: MORGAN KAUFMANN PUBLISHERS INC. **Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence**. [S.l.], 2001. p. 437–444.

QUIJANO-SáNCHEZ, L.; RECIO-GARCíA, J. A.; DíAZ-AGUDO, B. A reusable methodology for the instantiation of social recommender systems. In: **2013 IEEE 25th International Conference on Tools with Artificial Intelligence**. [S.l.: s.n.], 2013. p. 775–782. ISSN 1082-3409.

RAFAILIDIS, D.; NANOPOULOS, A. Modeling the dynamics of user preferences in coupled tensor factorization. In: **Proceedings of the 8th ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2014. (RecSys '14), p. 321–324. ISBN 978-1-4503-2668-1.

RASHID, A. M.; KARYPIS, G.; RIEDL, J. Learning preferences of new users in recommender systems: An information theoretic approach. **SIGKDD Explor. Newsl.**, ACM, New York, NY, USA, v. 10, n. 2, p. 90–100, dez. 2008. ISSN 1931-0145.

REAFEE, W.; SALIM, N.; KHAN, A. The power of implicit social relation in rating prediction of social recommender systems. **PLoS ONE**, Public Library of Science, v. 11, n. 5, p. 1–20, 05 2016.

RESNICK, P.; VARIAN, H. R. Recommender systems. **Communications of the ACM**, ACM, New York, NY, USA, v. 40, n. 3, p. 56–58, mar. 1997. ISSN 0001-0782.

SAID, A.; BELLOGÍN, A. Comparative recommender system evaluation: Benchmarking recommendation frameworks. In: **Proc. RecSys**. [S.l.: s.n.], 2014.

SALAKHUTDINOV, R.; MNIH, A. Probabilistic matrix factorization. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2008. v. 20.

SAMMUT, C.; WEBB, G. I. Leave-one-out cross-validation. In: **Encyclopedia of Machine Learning**. Boston, MA: Springer US, 2010. p. 600–601. ISBN 978-0-387-30164-8.

SHANI, G.; GUNAWARDANA, A. Evaluating recommendation systems. In: RICCI, F. et al. (Ed.). **Recommender Systems Handbook**. [S.l.]: Springer US, 2011. p. 257–297. ISBN 978-0-387-85819-7.

SOBOROFF, I.; NICHOLAS, C. Combining content and collaboration in text filtering. In: **Proceedings of the IJCAI**. [S.l.: s.n.], 1999. v. 99, p. 86–91.

STROGATZ, S. H. Exploring complex networks. **Nature**, Nature Publishing Group, v. 410, n. 6825, p. 268–276, 2001.

SU, X.; KHOSHGOFTAAR, T. M. A survey of collaborative filtering techniques. **Adv. in Artif. Intell.**, Hindawi Publishing Corp., New York, NY, United States, v. 2009, p. 4:2–4:2, jan. 2009. ISSN 1687-7470.

SUGANO, Y. et al. Image preference estimation with a data-driven approach: A comparative study between gaze and image features. **Journal of Eye Movement Research**, v. 7, n. 3, p. 5, 2014.

SUGANO, Y.; ZHANG, X.; BULLING, A. Aggregaze: Collective estimation of audience attention on public displays. In: **Proceedings of the 29th Annual Symposium on User Interface Software and Technology**. New York, NY, USA: ACM, 2016. (UIST '16), p. 821–831. ISBN 978-1-4503-4189-9.

SUN, Z. et al. Recommender systems based on social networks. **Journal of Systems and Software**, v. 99, p. 109 – 119, 2015. ISSN 0164-1212.

SUTTON, R. S.; BARTO, A. G. **Reinforcement learning: An introduction**. [S.l.]: MIT press Cambridge, 1998. v. 1.

TANG, J.; HU, X.; LIU, H. Social recommendation: a review. **Social Network Analysis and Mining**, Springer, v. 3, n. 4, p. 1113–1133, 2013.

TANG, L. et al. Ensemble contextual bandits for personalized recommendation. In: **Proceedings of the 8th ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2014. (RecSys), p. 73–80. ISBN 978-1-4503-2668-1.

TEO, C. H. et al. Adaptive, personalized diversity for visual discovery. In: **Proceedings of the 10th ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2016. (RecSys), p. 35–38. ISBN 978-1-4503-4035-9.

TÖSCHER, A.; JAHRER, M.; BELL, R. M. **The BigChaos Solution to the Netflix Grand Prize**. 2009.

UMEMOTO, K. et al. Search intent estimation from user's eye movements for supporting information seeking. In: **Proceedings of the International Working Conference on Advanced Visual Interfaces**. [S.l.: s.n.], 2012. p. 349–356.

WANG, D. et al. Recommendation based on weighted social trusts and item relationships. In: **Proceedings of the 29th Annual ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2014. (SAC '14), p. 254–259. ISBN 978-1-4503-2469-4.

WRÓBLEWSKA, A.; RACZKOWSKI, L. Visual recommendation use case for an online marketplace platform: Allegro.pl. In: **Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: ACM, 2016. (SIGIR '16), p. 591–594. ISBN 978-1-4503-4069-4.

XU, S.; JIANG, H.; LAU, F. C. Personalized online document, image and video recommendation via commodity eye-tracking. In: **Proceedings of the 2008 ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2008. (RecSys '08), p. 83–90. ISBN 978-1-60558-093-7.

XU, S.; JIANG, H.; LAU, F. C. M. Observing facial expressions and gaze positions for personalized webpage recommendation. In: **Proceedings of the 12th International Conference on Electronic Commerce: Roadmap for the Future of Electronic Business**. [S.l.: s.n.], 2010. p. 78–87.

YANG, B. et al. Social collaborative filtering by trust. In: **Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence**. [S.l.]: AAAI Press, 2013. (IJCAI '13), p. 2747–2753. ISBN 978-1-57735-633-2.

YANG, X.; STECK, H.; LIU, Y. Circle-based recommendation in online social networks. In: **Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2012. (KDD '12), p. 1267–1275. ISBN 978-1-4503-1462-6.

YE, M.; LIU, X.; LEE, W.-C. Exploring social influence for recommendation: A generative model approach. In: **Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: ACM, 2012. (SIGIR '12), p. 671–680. ISBN 978-1-4503-1472-5.

YUAN, T. et al. How friends affect user behaviors? an exploration of social relation analysis for recommendation. **Knowledge-Based Systems**, v. 88, p. 70 – 84, 2015. ISSN 0950-7051.

ZAFARANI, R.; ABBASI, M. A.; LIU, H. **Social Media Mining: An Introduction**. New York, NY, USA: Cambridge University Press, 2014. ISBN 1107018854, 9781107018853.

ZANKER, M.; JESSENITSCHNIG, M. Case-studies on exploiting explicit customer requirements in recommender systems. **User Modeling and User-Adapted Interaction**, v. 19, n. 1, p. 133–166, 2009. ISSN 1573-1391.

ZHOU, T. et al. Kernelized probabilistic matrix factorization: Exploiting graphs and side information. In: **SIAM International Conference on Data Mining**. California, USA: SIAM, 2012. (SDM), p. 403–414.

ZIEGLER, C.-N. et al. Improving recommendation lists through topic diversification. In: **Proceedings of the 14th International Conference on World Wide Web**. New York, NY, USA: ACM, 2005. (WWW '05), p. 22–32. ISBN 1-59593-046-9.