Is diversity optimization always suitable? Toward a better understanding of diversity within recommendation approaches

Yu Du, Sylvie Ranwez, Nicolas Sutton-Charani, Vincent Ranwez

Information Processing and Management 58 (2021) 102721

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Is diversity optimization always suitable? Toward a better understanding of diversity within recommendation approaches

Yu Du [a], Sylvie Ranwez [a,*], Nicolas Sutton-Charani [a], Vincent Ranwez [b]

[a] EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Ales, Ales, France
[b] UMR AGAP Institut, Univ Montpellier, CIRAD, INRAE, Institut Agro, F-34398 Montpellier, France

ARTICLE INFO

Keywords:
Recommender system
Diversity
Greedy optimization
DBpedia
Knowledge graph embedding
Deep learning

ABSTRACT

The diversity of the item list suggested by recommender systems has been proven to impact user satisfaction significantly. Most of the existing diversity optimization approaches re-rank the list of candidate items during a post-processing step. However, the diversity level of the candidate list strongly depends on the recommender system used. Hence, applying the same post-processing diversification strategy may not be as effective for different recommendation approaches. Moreover, individual users' diversity needs are usually ignored in the diversification post-processing.

This article aims at providing an in-depth analysis of the diversity performances of different recommender systems. To the best of our knowledge, it is the first study to systematically compare diversity performances of the main types of recommendation models using benchmark datasets in different domains (movie, anime and book). Semantics related to items may be considered a key factor in measuring diversity within recommender systems. In this study, we leverage support from the knowledge engineering domain and take advantage of resources such as Linked Data and knowledge graphs, to assert the diversity of recommendations. We also propose a variant of the classic diversification post-processing objective that allows to take into account specific users' diversity needs. We measure the adequacy between the diversity levels a recommender system suggests to its users and those of users' profiles with the R^2 coefficient of determination.

Our study indicates that: (1) none of the tested recommender systems, even the most recent ones, provides items with levels of diversity that suit user profiles ($R^2 < 0.2$); (2) the classic post-processing diversification approach may lead to over-diversification compared to users' diversity needs and (3) the diversity adjustment that accounts for user profiles has more benefits (greater R^2 and smaller accuracy loss). All the source code and datasets used in our study are available to ensure the reproductibility of the study.

1. Introduction

Recommender systems (RSs) are effective solutions to help users find what they need in the current information overload. Based on item descriptions and/or available ratings supplied by users, RSs aim at selecting among unseen items those that may be of interest for a specific user. RSs have been proven successful in various fields including e-commerce, movie, music, travel, etc. For example, 35% of Amazon.com's revenue and 75% of what users watch on Netflix are generated by their recommendation

* Corresponding author.
E-mail addresses: yu.du@mines-ales.fr (Y. Du), sylvie.ranwez@mines-ales.fr (S. Ranwez), nicolas.sutton-charani@mines-ales.fr (N. Sutton-Charani), vincent.ranwez@supagro.fr (V. Ranwez).

https://doi.org/10.1016/j.ipm.2021.102721
Received 21 December 2020; Received in revised form 31 July 2021; Accepted 9 August 2021
Available online 5 September 2021

engines (MacKenzie et al., 2013). Different types of recommenders have been proposed in the literature. Although sharing the common task of user preference predictions for their unrated items, those recommenders can be categorized into four families based on the type of information used during the prediction procedure: Non-Personalized recommenders such as the approach based on item popularity (Ferrari Dacrema et al., 2019); Collaborative Filtering (CF) (Khan et al., 2017), Content-Based Filtering (CBF) (Lops et al., 2011) and hybrid approaches (Aggarwal, 2016). Specifically, non-personalized approaches often ignore users' preferences and provide them with the most popular items; CF approaches only use the rating matrix as input data for predictions; CBF recommenders use item content data, e.g. textual descriptions or item features along with the target user's own ratings, while ignoring preferences of other users; and hybrid approaches consider both user ratings and item features.

The performance evaluation of these recommendation approaches are typically driven by their ability to predict accurately user–item interactions, i.e. by the *accuracy* criterion. However, the most accurate RSs are not necessarily the most useful to users (McNee et al., 2006), according to the standard accuracy metrics. For example, imagine a user, who enjoys watching movies in which Brad Pitt starred. If a recommender system provides her/him with a list of Brad Pitt starred movies, all recommended movies may be accurate but the entire list might not be useful as the user might have watched most of them already. This is a typical lack-of-diversity problem, because items in the list are highly similar. The diversity of recommendations has therefore been recognized recently as an important criterion for evaluating the utility of recommendation lists (Kunaver & Požrl, 2017; McNee et al., 2006; Meymandpour & Davis, 2020; Vargas & Castells, 2011). The accuracy assessment and comparison have been rigorously conducted between the main families of RSs. In contrast, only few studies have systematically compared and analyzed the diversity performances between those recommender systems.

Most of the methods that aim at improving the diversity of recommendations consider diversity optimization as a post-processing step, regardless of the recommendation model (Adomavicius & Kwon, 2012; Aytekin & Karakaya, 2014; Kunaver & Požrl, 2017; Sha et al., 2016; Ziegler et al., 2005). In such approaches, the RS is first used to obtain a large set of candidate items (e.g. the top-100 most accurate items according to the RS); that list is then tailored down by the diversification post-processing to build the short, diversified recommendation list proposed to the user. As RSs use different approaches and algorithms, they may return candidate lists that have various levels of diversity. Therefore, applying the same diversification strategy might not be equally effective for different recommendation approaches. Intuitively, and as confirmed by this study: (1) CBF recommenders have the lowest ability of recommending diversified items as they favor items whose contents are similar with the ones highly rated by the target user; (2) CF recommenders lead to more diversified item lists, as they explore profiles of other users for predictions and (3) as hybrid approaches consider both collaborative and content information, they might lead to recommendations with intermediate diversity levels. The interest of blindly increasing the recommendation diversity for every RS remains questionable. Consequently, it might be reasonable to increase recommendation diversity for CBF recommenders, but this could be risky for the CF ones as their recommendations might already be diversified enough.

Previous works (Adomavicius & Kwon, 2012; Aytekin & Karakaya, 2014; Bradley & Smyth, 2001; Wu et al., 2018; Yigit-Sert et al., 2020; Ziegler et al., 2005) mainly attempted to increase the diversity of the recommended list while maintaining the recommendation accuracy. However, this could be unnecessary or even counterproductive for CF-based RSs. Moreover, the optimal level of diversity might vary from one user to another (Di Noia et al., 2014; Meymandpour & Davis, 2020; Wu et al., 2018). For example, a highly diversified recommendation list might please users who have heterogeneous tastes, but it might decrease the satisfaction of users with simple tastes. Hence, it seems reasonable to compare and analyze the behaviors of these RSs from both the *absolute* and the *relative* diversity perspectives. For a specific user, the *absolute* diversity refers to the amount of diversity in her/his recommendation list while the *relative* diversity represents the extent to which the *absolute* diversity of the recommendations would fit the user's diversity needs.

1.1. Research questions

The main research questions tackled in this paper are:

- RQ1: How do the main types of recommender systems perform in terms of *absolute* and *relative* diversities?
- RQ2: Is the diversity optimization driven by the post-processing method equally effective for different types of recommender systems? How would these recommender systems perform, in terms of diversity and accuracy, if they were combined with the same diversification post-processing?
- RQ3: What is the impact of considering individual users' diversity needs during the diversification post-processing for different types of RSs?

1.2. Proposed solutions

To investigate these research questions, we propose to analyze and compare the ability of the main families of RSs to provide diversified recommendations. To this aim, we compare seven different recommendation models representing the four different types of RSs on three real-world datasets (related to movies, books and anime). For non-personalized recommenders, we consider the item popularity-based approach (Ferrari Dacrema et al., 2019); For CBF recommenders, we consider a Linked Open Data based approach (Di Noia et al., 2012); For CF recommenders, we consider a memory-based approach (Sarwar et al., 2001), a model-based approach (Koren et al., 2009) and a state-of-the-art approach based on deep neural networks (He et al., 2017) and for hybrid recommenders we consider an approach based on knowledge graph embeddings (Palumbo et al., 2018a) and an approach that combines the content-based and item popularity-based models.

We propose to analyze the behavior of each RS during the diversification post-processing in terms of accuracy and absolute/relative diversity. The diversification post-processing can easily be adapted to optimize the *relative* diversity rather than the *absolute* one. We describe this extension and evaluate the impact of adjusting the recommendation diversity to individual users' needs.

The paper is organized as follows: In Section 2, the descriptions of the compared recommender systems are presented. Section 3 presents the diversification post-processing objectives and details the diversification methods that are compared in this study. In Section 4, we detail our evaluation protocol. In Section 5, we present and comment the experimental results and discuss the limitations and implications of the study. Finally, Section 6 presents our conclusions.

2. Compared recommender systems

Recommender systems are designed to predict, for each user u and each item i that he/she has not yet interacted with, the relevance score $rel(u,i)$,[1] quantifying the degree of user u's interest for item i. The top-n items, i.e. those with the highest predicted scores, are then recommended to the user. Different recommendation strategies provide different lists of recommendations, leading to different performances (in terms of accuracy and diversity for instance). As mentioned in the Introduction, this study considers seven recommenders, including: a non-personalized, item popularity-based approach (TopPopular), a content-based filtering approach based on linked open data (CBF), a memory-based CF approach (IBCF), a model-based CF approach (SVD), a CF approach based on deep neural networks (DNN), a hybrid approach based on knowledge graph embeddings (KGE) and another hybrid approach combining CBF and TopPopular (CBF-TopPopular). Summarized below are the principles of the systems compared in this analysis.

2.1. Top popular items approach (TopPopular)

Item popularity is known as an important factor of recommendation (Abdollahpouri, 2019). As a non-personalized approach, the model provides each user with the most popular items yet to be rated. Formally, the popularity of a given item i is defined as the number of users u in the user set U who have rated it, with I_u being the set of items rated by u:

$$Pop(i) = |\{u \in U | i \in I_u\}| \tag{1}$$

In spite of its simple approach, a recent study (Ferrari Dacrema et al., 2019) has shown that the *TopPopular* model could lead to more accurate recommendations on some datasets than some recently proposed deep neural network-based models, such as CMN (Ebesu et al., 2018).

2.2. Content-based filtering approach (CBF)

Content-based recommendations aim at providing users with items whose content are similar to what they have liked before. The content-based approach thus focuses on the ratings (preferences) of the target user and content information of items. In a typical content-based recommendation context, characteristic vectors $\mathbf{x}_i$, in which each dimension represents the weight of a specific item characteristic, are used to represent items. For example in the news recommendation scenario, we could model a piece of news as a weighted TF–IDF word term vector. For a given target user u and u's rated items I_u, the recommender generates a profile vector for u, i.e. $\mathbf{x}_u$ (Eq. (2)). The prediction of an unseen item j for user u consists in measuring the similarity between u's profile vector $\mathbf{x}_u$ and j's characteristic vector $\mathbf{x}_j$.

$$\mathbf{x}_u = \sum_{i \in I_u} r_{u,i} \mathbf{x}_i \tag{2}$$

However, it is a challenging task to find suitable resources containing rich item knowledge. The conventional characteristics based on item textual data need heavy pre-processing such as word sense disambiguation. Moreover, collecting such textual information also requires lots of human efforts. More robust content-based recommender systems have leveraged structured item characteristics stored by means of ontologies, i.e. a standardized structure of knowledge representations, providing a flexible way of managing item characteristics. For example, Di Noia et al. (2012) proposed an approach that uses item properties within the DBpedia[2] ontology to construct item feature vectors.

To predict relevance scores for users' unrated items, the CBF approach considered in our study is based on the formula proposed by Di Noia et al. (2012), and recalled in Eq. (3). In this formula, $profile(u)^+$ represents user u's liked items and the similarity function $sim(i,j)$ refers to the semantic similarity between items i and j. Following Di Noia et al. (2012), we make use of the item knowledge data within the DBpedia knowledge base to compute item semantic similarities. The model implementation details are provided in Section 4 (the experimental section).

$$rel(u,i) = \frac{\sum_{j \in profile(u)^+} sim(i,j)}{|profile(u)^+|} \tag{3}$$

[1] Note that for rating prediction-based RSs such as item-based CF and SVD, the $\hat{r}_{u,i}$ term is typically used to represent the relevance score, i.e. $rel(u,i)$.

[2] https://wiki.dbpedia.org/

2.3. Item-based collaborative filtering approach (IBCF)

The item-based CF approach (Sarwar et al., 2001) is a variant of the user-based CF approach whose main idea is that the rating $\hat{r}_{u,i}$ (relevance score) that a user u would assign to an item i is expected to be similar to user v's rating on i, i.e. $r_{v,i}$, if u and v have similar ratings on other items. Likewise, the IBCF model assumes that $\hat{r}_{u,i}$ would be similar to u's own ratings on item j if i and j have similar rating distribution among all users. More specifically, the prediction of the rating $\hat{r}_{u,i}$ is based on u's ratings of item i's neighbor (similar) items. Formally, to compute $\hat{r}_{u,i}$, the ratings assigned by the target user u, to each of the target item i's neighbors j (denoted as N_i), are combined using a weighted average so that closer neighbors have more impact on the predicted rating $\hat{r}_{u,i}$ (see Eq (4)). The neighborhood is computed based on the similarity measurement between items, with pairwise distance metrics (e.g. cosine) between their rating vectors. Normalization techniques are often adopted to handle the disparity among rating distributions (Herlocker et al., 2002):

$$\hat{r}_{u,i} = \bar{r}_i + \sigma_i \frac{\sum_{j \in N_i} [(\frac{r_{u,j} - \bar{r}_j}{\sigma_j}) * sim(i,j)]}{\sum_{j \in N_i} sim(i,j)} \tag{4}$$

with $\bar{r}_i$ and σ_i (resp. $\bar{r}_j$ and σ_j) representing the average and the standard deviation of item i's (resp. j's) rating vector.

It is worth noticing that in the current study we omitted the user-based CF approach (UBCF) for three reasons. First, researchers (Ekstrand et al., 2014; Sarwar et al., 2001) have demonstrated that IBCF generally outperforms UBCF, in terms of recommendation accuracy and user satisfaction. Second, IBCF is more flexible than UBCF as it requires less computer memory to construct the similarity matrix for items than for users, as the number of items in a real-world recommender system, e.g. amazon, is generally much lower than the number of users. Third, the user study carried out by Ekstrand et al. (2014) has shown that there is no significant difference between IBCF and UBCF, in terms of the recommendation diversity, on which the current study focuses.

2.4. Singular value decomposition approach (SVD)

The SVD approach (Koren et al., 2009) is one of the most popular model-based CF approaches. The model applies matrix factorization techniques to map users and items into a joint latent factor space of a reduced dimensionality d. The learnt representations are then used to predict ratings. Formally, each item i is associated with a factor vector $\mathbf{q}_i \in \mathbb{R}^d$ and each user u is associated with a factor vector $\mathbf{p}_u \in \mathbb{R}^d$. The inner product of these two vectors in the latent vector space represents the predicted rating, i.e. $\hat{r}_{u,i} = \mathbf{q}_i^T \mathbf{p}_u$. To learn the factor vectors, the SVD model minimizes the loss function representing the regularized squared error between predicted and real ratings (Eq. (5)). To achieve this optimization, stochastic gradient descent (SGD) is applied.

$$Loss = \sum_{r_{u,i} \in R_{train}} (r_{u,i} - \mathbf{q}_i^T \mathbf{p}_u)^2 + \lambda(\|\mathbf{q}_i\|^2 + \|\mathbf{p}_u\|^2) \tag{5}$$

2.5. Deep neural networks-based collaborative filtering approach (DNN)

In general, recommender systems based on neural networks take the user rating matrix as the input layer and generate scores for each user–item pair on the output layer. The hidden layers, i.e. a multi-layer neural architecture, brings to light the latent structures of user–item interactions. The considered DNN approach (He et al., 2017), also known as NeuMF (Neural Matrix Factorization), is one of the most cited deep neural networks-based recommendation approaches (Ferrari Dacrema et al., 2019). Specifically, the DNN model concatenates two neural networks: GMF (Generalized Matrix Factorization) and MLP (Multi-Layer Perceptron). The main idea of the GMF neural architecture is to generalize the matrix factorization model i.e. Eq. (5), in which latent factors are treated equally and a linear function (i.e. the inner product) is used to model the user–item interaction. The GMF network can automatically learn different weights for each latent factor and consider a non-linear setting for user–item interactions. Unlike GMF that only uses a fixed element-wise product of $\mathbf{p}_u$ and $\mathbf{q}_i$, the MLP neural network allows to learn subtler user–item interactions by adding hidden layers to the concatenated vector of $\mathbf{p}_u$ and $\mathbf{q}_i$. Finally, the last hidden layers of GMF and MLP are concatenated to generate the final score of a given user–item pair.

2.6. Knowledge graph embeddings-based approach (KGE)

Knowledge graphs (KG) are data structures that allow to represent both human-readable and machine-understandable ontological domain knowledge. KGs are typically composed of $\langle s, p, o \rangle$ triples in which s and o are *subject* and *object* entities while p is the *predicate* indicating the corresponding semantic relation of s toward o (e.g., $\langle$Brad_Pitt, starring_of, 12_Monkeys$\rangle$). Famous large-scale KGs such as DBpedia and Google Knowledge Graph[3] have been successfully applied to real-world applications. Although they are effective in representing heterogeneous information, the handling of such KGs remains difficult due to their underlying graph structure, notably for large-scale KGs. Knowledge graph embedding is a technique that makes the handling of a KG more flexible. The principle is to embed the KG's elements (its entities and relations) into a latent vector space while preserving its inherent structure. Two types of embedding models have been proposed in the literature, including translational distance models e.g. TransE (Bordes

[3] https://en.wikipedia.org/wiki/Knowledge_Graph

et al., 2013) and semantic matching models e.g. RESCAL (Nickel et al., 2011), SME (Bordes et al., 2014). For example, TransE considers both entities and relations as d dimensional vectors in the same latent space $\mathbb{R}^d$. For a given triple $\langle s, p, o\rangle$, the *predicate* p is interpreted as a translation vector $\mathbf{p}$ so that the embedding vectors of s and o, i.e. $\mathbf{s}$ and $\mathbf{o}$, can be connected by $\mathbf{p}$ with low error, i.e. $\mathbf{s}+\mathbf{p} \approx \mathbf{o}$ when $\langle s, p, o\rangle$ holds. Readers may refer to Wang et al. (2017) for more detailed descriptions of knowledge graph embedding models and their applications.

In the past few years, knowledge graph embedding models have been proven highly effective for recommender systems (Palumbo et al., 2020, 2018a, 2018b). In the RS context, two knowledge graphs can be constructed: ICKG (Item Content Knowledge Graph) and UPKG (User Preference Knowledge Graph). The former contains triples describing knowledge related to item contents (e.g. the director of a movie) while the latter represents knowledge related to user preferences (e.g. the movies liked by a user).

As proposed in Palumbo et al. (2018b), the KGE recommender that we consider in this study is based on the whole knowledge graph (denoted as HybridKG) combining ICKG and UPKG, in which user preference data (user ratings) and item knowledge are hybridized. Specifically, entities within HybridKG are made of users, items and item property objects while relations within HybridKG are made of user preferences (i.e. *likes*) and item attributes, e.g. *genre*, *starring*. For example in the following KG composed by the two triples $\{\langle \text{Citizen, genre, Drama\_film}\rangle; \langle \text{John, likes, Citizen}\rangle\}$, {John, Citizen, Drama_film} are entities and {likes, genre} are relations.

The prediction of the relevance score for each item i in the user u's unrated items, i.e. $rel(u, i)$, is based on the latent vector representations (i.e. embeddings) learnt by a knowledge graph embedding model, for entities and relations within the HybridKG. In this framework, the recommendation can be considered as a knowledge graph completion task or a link prediction task for each triple $\langle u, \text{likes}, i\rangle \notin$ HybridKG. For example, for translational distance-based embedding models, we could use the score function $rel(u, i) = -f(\mathbf{u} + \mathbf{likes} - \mathbf{i})$ with $\mathbf{u}$, $\mathbf{likes}$ and $\mathbf{i}$ being the embedding vectors of the entity (user) u, the relation *likes* and the entity (item) i, respectively. The $f(\cdot)$ term typically refers to the norm of the result vector.

2.7. Hybridizing content-based and popularity-based approaches (CBF-TopPopular)

As presented in Sections 2.1 and 2.2 , popularity-based recommendations are non-personalized and privilege popular items over niche ones while content-based recommenders are generally over-personalized and would favor obscure and long-tail items. So, we would like to consider a simple hybrid approach combining the above CBF and TopPopular models. Formally, given user u and one of u's unrated items i, the relevance score $rel(u, i)$ of the CBF-TopPopular model is defined by Eq. (6) as follows:

$$rel(u, i) = \omega * TopPopular(i) + (1 - \omega) * CBF(u, i) \tag{6}$$

where $TopPopular(i)$ and $CBF(u, i)$ refer respectively to the popularity of item i (cf. Eq. (1)) and the relevance score $rel(u, i)$ computed by the CBF model (cf. Eq. (3)). ω represents a hyper-parameter, weighting the importance of the two combined models.

3. Diversity of recommendations

3.1. Definition & evaluation metrics

According to the Cambridge dictionary, the concept of diversity is defined as: *the fact of many different types of things or people being included in something; a range of different things or people.* According to that definition, *diversity* could be roughly considered as the opposite of *similarity* (Bradley & Smyth, 2001). It is worth noticing that the similarity notion is typically employed for a pair of items (i.e. pairwise) whereas the diversity is generally considered for a list of items (i.e. listwise). In the recommender system context, researchers (Bradley & Smyth, 2001; Castagnos et al., 2013; Di Noia et al., 2014; Ekstrand et al., 2014; Jannach et al., 2015; Jugovac et al., 2017; Vargas & Castells, 2011; Wang et al., 2019) often adopt the *intra-list-diversity* (ILD) metric to measure the diversity of a recommendation list, which is defined as the average pairwise dissimilarity among the recommended items (Eq. (7)). Given the recommended item list L, the ILD of L is defined as follows:

$$\text{ILD}(L) = \frac{2}{|L| \times (|L| - 1)} \sum_{\forall 1 \le j < k \le |L|} (1 - sim(L[j], L[k])) \tag{7}$$

where $sim(.,.)$ is a similarity function which measures the similarity value (conventionally normalized between [0, 1]) for a given pair of items. For example, this similarity metric could refer to item proximities based on item knowledge data or on the user ratings.

3.1.1. Individual (absolute) diversity & aggregate diversity

The diversity of a RS can be measured at the individual level (i.e. the diversity of the list recommended to a specific user) or at the RS level (i.e an aggregate of the diversity of the recommendations proposed by the RS to its users). The ILD metric presented in the previous subsection generally refers to the individual diversity, as the list of items based on which the ILD is measured is provided for individual users. Unlike the individual diversity, one can also deal with the so-called aggregate diversity (Adomavicius & Kwon, 2012), which mainly aims at measuring the ability of a RS to provide different items at the system level. The aggregate diversity can be computed as the proportion of items in the entire item catalog that can be recommended to users. A recommender with high aggregate diversity performance should have the ability to provide users with, for instance, long-tail/niche items.

The current study aims at analyzing and comparing the performances of different families of recommender systems, in terms of individual diversity (ILD). As we intended to study the impact of considering individual users' diversity needs during the

diversification post-processing (i.e. RQ3), the aggregate diversity would not fit our goal. Nevertheless, the aggregate diversity comparison between various recommendation approaches would also have its own benefits, especially for the e-commerce, as item popularity generally biases users to choose popular items against long-tail ones. For example, the findings of Lee and Hosanagar (2014) demonstrated that CF recommenders produce generally low aggregate diversity among sales.

We use the term *absolute* diversity to refer to the ILD of the item list recommended to a particular user, as opposed to the *relative* diversity discussed in the following subsection.

3.1.2. Personalized (relative) diversity

Personalized diversity has recently attracted some attention. The main assumption is that different users may have different needs in terms of recommendation diversification. Unlike the non-personalized *absolute* diversity that measures the amount of diversity (ILD) within a list of recommended items, the personalized *relative* diversity refers to the extent to which the ILD of the recommended items would fit users' needs for diversity. As the *relative* diversity depends on the profile of each particular user, one needs to first quantify the user's diversity needs.

The most intuitive way of quantifying a particular user's diversity needs is to compute the ILD value among the items that the user has rated, i.e. the user's profile items (Jugovac et al., 2017; Meymandpour & Davis, 2020). Likewise, one may also quantify the user's diversity needs by measuring the Shannon's entropy of the user profile, with respect to some item categories. For example, authors of Di Noia et al. (2014) propose to consider individual users' propensity toward diversity by computing the entropy based on different attributes, e.g. genre, actor, year, etc. for users' profile items. Those approaches rely mainly on users' historical data, i.e. their ratings. Authors of Wu et al. (2018) propose an alternative way of quantifying users' diversity needs, which is based on a personality model. To be specific, they assumed that individual users' personalities might influence their needs for diversity, e.g. a user with a high level of *Openness* would expect high diversity. They carried out a user study to collect personality data and built a linear regression model to analyze the impact of a user's personality on her/his diversity needs.

With users' profile (*desired*) diversities $Div(p_u)$ and that of their recommended items $Div(r_u)$, the *relative* diversity can be represented by using error-based metrics. Generally, the *relative* diversity quantifies how well a recommendation list would match a user's historical tendency with respect to the diversity aspect. Jugovac et al. (2017), Wu et al. (2018) propose to compute, for each user u, the absolute error between u's profile diversity and that in u's recommendation list, i.e. $|Div(p_u) - Div(r_u)|$. Similarly, Meymandpour and Davis (2020) use the root mean square (diversity) error (RMSDE) to represent the *relative* diversity (Eq. (8)).

$$\text{RMSDE} = \sqrt{\frac{\sum_{u \in U} (Div(p_u) - Div(r_u))^2}{|U|}} \tag{8}$$

In statistics, the coefficient of determination, i.e. R^2, is a typical metric to measure the correlation of two given lists of values. In the context of *relative* diversity, one can use this metric to measure how well the values of $Div(r_u)$ (diversities recommended to users) would fit the values of *profile* $Div(p_u)$ (users' diversity needs), over the entire set of users (cf. Eq. (9)). The $\overline{Div(p_u)}$ term refers to the average diversity over profiles of the user set U.

$$R^2 = 1 - \frac{\sum_{u \in U} (Div(p_u) - Div(r_u))^2}{\sum_{u \in U} (Div(p_u) - \overline{Div(p_u)})^2} \tag{9}$$

The R^2 approach is a normalized version of the RMSDE metrics, in which the values vary from 0 to 1. Such a normalization might be useful in terms of interpretation. A value of $R^2 = 1$ means that the diversities proposed by the recommender system fit its users' diversity needs perfectly.

Alternative ways of measuring the *relative* diversity consider metrics based on distributions, such as the Jensen–Shannon divergence (Eskandanian et al., 2017) or the Kullback–Leibler (KL) divergence. For example, Steck (2018) propose *calibration* metrics $C_{\text{KL}}(p, q)$, that are based on the KL divergence of two probability distributions p and q, with p being the distribution of genres in users' profile items (movies in their study) and q being the distribution of genres in users' recommendations (Eq. (10)).

$$C_{\text{KL}}(p, q) = \sum_{g \in \text{genres}} p(g|u) \log \frac{p(g|u)}{q(g|u)} \tag{10}$$

Lower values of $C_{\text{KL}}(p, q)$ stand for better calibrated recommendations. Given that $C_{\text{KL}}(p, q)$ may diverge when $q(g|u) = 0$ and $p(g|u) > 0$, (i.e. the genre g is presented in u's profile but not in u's recommendations), the authors propose to replace the $q(g|u)$ term with $(1 - a) \cdot q(g|u) + a \cdot p(g|u)$ in Eq. (10), with a small value of $a > 0$ (e.g. 0.01).

The work of Steck (2018) aims at providing users with a list of movies in which the proportion of each genre matches its corresponding proportion in the user profile. Though related, the authors discussed that calibrated recommendations are quite different from the *relative* diversity perspective. The main difference is that calibrated recommendations are restricted to a particular feature such as a movie's genre, whereas for the diversity measurement, it might be better not to use such a restriction since items (films) have more than one feature (e.g. directors, actors etc.). One may also consider computing the KL divergence between multivariate attribute vectors, for instance. Nevertheless, it is often difficult to gather exactly the same attribute data for all items (e.g. missing values in the knowledge graph), and the KL computation is not straightforward.

3.2. Objective functions for diversification

This section discusses the main objective functions on which diversification may be based.

3.2.1. Classic objective function for diversity consideration

In general, increasing diversity is associated with a decrease of recommendation accuracy (Adomavicius & Kwon, 2012; Wang et al., 2019; Zhou et al., 2010). It is a challenging task to balance the accuracy-diversity trade-off (Kunaver & Požrl, 2017). On the one hand, always providing users with accuracy-based recommendations, regardless of diversity, can weary them and thus reduce their satisfaction. For instance, a list only recommending movies of a single genre may seems repetitive even to a user who appreciates that type of movies. This point refers to the over-specialization problem, notably for content-based recommenders, which provide users with items similar to their highly rated items (Lops et al., 2011). On the other hand, the overmuch consideration of the recommendation diversity would sacrifice the relevance, which is no more acceptable. For example, recommending horror and action movies to a user who only consumed comedy movies might decrease the quality of her/his experiences on the system. Suggesting irrelevant items not only fails to achieve the main goal of RSs (i.e. helping users to find interesting items), but it also undermines the confidence that users have in the RSs. Both accuracy and diversity have to be considered. Meanwhile, accuracy should remain the preliminary goal of recommenders because significant accuracy loss would not be acceptable in most real-life applications (Adomavicius & Kwon, 2012). In other words, it could be preferable to recommend items achieving a considerable gain in diversity while maintaining roughly the same level of accuracy. The diversified recommendation could therefore be treated as an ordered, or weighted, bi-criterion optimization problem, in which one seeks to maximize the overall relevance of a recommendation list while minimizing the redundancy between the recommended items. A widespread (Bradley & Smyth, 2001) objective function used to reach this goal is provided by Eq. (11):

$$f_{obj}(L,\alpha) = (1-\alpha) \times \frac{\sum_{\forall i \in L} rel(i)}{|L|} + \alpha \times \mathrm{ILD}(L) \tag{11}$$

where $rel(i)$ represents the relevance score of item i and the parameter $\alpha \in [0,1]$ is a diversification factor balancing the accuracy-diversity trade-off.

3.2.2. Personalized objective function for diversity

This objective function is user independent and does not account for individual users' diversity needs. As discussed in Section 3.1.2, metrics taking into account individual users' diversification needs have been proposed. Thus, one may consider variants of Eq. (11) with respect to the metrics used for personalizing users' diversity needs. Generally considered as a bi-criterion optimization problem regarding both the accuracy and diversity, the personalization of diversity mainly leans on the right part of Eq. (11), as the accuracy (left) part, which depends only on the recommenders, is fixed. Di Noia et al. (2014) propose to weight pairwise item similarity values according to a given user's diversity needs (measured by Shannon's entropy). Thus, for a user with a high entropy value (diversity needs), their objective function penalizes scores of candidate items which are similar to the items already in the recommended list. Wu et al. (2018) model users' personalities for measuring $Div(p_u)$, i.e. users' needs for diversification. The right part of the objective function in their approach is defined by $-|Div(p_u) - Div(r_u)|$, i.e. the difference between users' profile diversity and the recommended diversity. Note that the negation of this difference is considered since the goal is to maximize the whole objective function. Likewise, the calibrated recommendation approach proposed by Steck (2018) adopts the Kullback–Leibler divergence to measure the distance between two distributions (see Section 3.1.2). Hence, the KL term becomes the right part of their objective function, aiming at recommending items that best fit the user's profile, regarding the calibration metric they proposed.

Though different, all the previously discussed personalized objective functions share one common goal: minimizing the difference between users' profile diversity and that of their recommended items. In the current study, as described by RQ3 (cf. Section 1.1), we also aim at studying the impact of the diversity personalization on the performance behaviors for different families of recommender systems. A straightforward and intuitive way to achieve this goal is to optimize the following variant of the classic objective function (Eq. (11)) that accounts for individual users' diversity profiles, as defined by Eq. (12):

$$f_{obj}(u,L,\alpha) = (1-\alpha) \times \frac{\sum_{\forall i \in L} rel(i)}{|L|} + \alpha \times (-|\mathrm{ILD}(L) - \mathrm{ILD}_p(u)|) \tag{12}$$

where $\mathrm{ILD}_p(u)$ represents the ILD of the user u's rated items (u's profile). The idea here is to recommend a list L in which the items would be relevant and the ILD of L would be as close as possible to the ILD of the user profile, rather than being as high as possible.

3.3. Greedy optimization heuristics

The greedy optimization (Bradley & Smyth, 2001) is a commonly used post-processing approach, in which candidate items preliminarily ranked by their relevance scores are later re-ranked. Specifically, the greedy approach proceeds in n sequential steps with n being the length of the recommendation list to build. During each iteration it seeks among the candidate items the one that maximizes a bi-criterion objective function combining accuracy and diversity. Formally, let $L(u)$ be the list of unrated items for user u, in which items are ordered by their relevance scores. Suppose that we want to recommend a list of n items to u, denoted as $L_{rec}(u)$. Without considering the diversity optimization, the first n items of $L(u)$ are returned to u, which corresponds to a typical recommender system scenario. On the contrary, the greedy optimization illustrated by Algorithm 1 selects at each step the item maximizing the objective function f_{obj} (lines 4 and 6), which takes both diversity and accuracy into account. Note that different objective functions are used based on the value of the Boolean parameter *personalized*.

As pointed out in Bradley and Smyth (2001), this algorithm is expensive in terms of time complexity because $L(u)$ can be very large as users often rate a small portion of the items catalog, i.e. $|L(u)| \gg n$. Thus, the authors proposed a bounded greedy approach,

Algorithm 1: The greedy optimization

Input : $L(u)$, α, *personalized* (Boolean), n
Output: $L_{rec}(u)$
$L_{rec}(u) \leftarrow \{\}$;
while $|L_{rec}(u)| < n$ **do**
  if *personalized* **then**
    $i^* \leftarrow \underset{i \in L(u) \setminus L_{rec}(u)}{\text{argmax}} f_{obj}(u, L_{rec}(u) \cup \{i\}, \alpha)$;
  else
    $i^* \leftarrow \underset{i \in L(u) \setminus L_{rec}(u)}{\text{argmax}} f_{obj}(L_{rec}(u) \cup \{i\}, \alpha)$;
  end
  $L_{rec}(u) \leftarrow L_{rec}(u) \cup \{i^*\}$;
end
return $L_{rec}(u)$

which simply applies the greedy algorithm to the first m elements of $L(u)$ (with $|L(u)| \gg m > n$). Thus, the calculation time is greatly reduced without much impact on the objective score of the recommended list. The bounded version of greedy algorithm is commonly adopted in the literature (Di Noia et al., 2014; Sha et al., 2016; Wu et al., 2018; Ziegler et al., 2005). Here we denote $L_{candidates}(u)$ as the list of the top-m items of $L(u)$ that constitute the input of the bounded greedy algorithm.

3.4. Other diversification approaches

As mentioned previously, most of the existing diversification optimization approaches are post-processing ones. However, some recent works (Cheng et al., 2017; Li et al., 2017) view the problem differently and consider diversity during the recommendation phase using LTR (learning to rank) approaches. The main idea behind these approaches is to train supervised learning models on a set of empirically determined training instances. To optimize the model parameters, the gradient descent method is applied with a loss function which considers both diversity and accuracy. It is worth noting that the LTR method is linked to a specific recommendation model, e.g. the matrix factorization model as done in Cheng et al. (2017), Li et al. (2017). Unlike the post-processing approaches, LTR is thus unfit for direct integration into existing recommender systems. In addition, the greedy optimization approach is a classic re-ranking method largely and continuously used in the literature on account of its simplicity (Adomavicius & Kwon, 2012; Bradley & Smyth, 2001; Di Noia et al., 2014; Jugovac et al., 2017; Sha et al., 2016; Steck, 2018; Vargas & Castells, 2011; Wang et al., 2019; Wu et al., 2018; Ziegler et al., 2005). Therefore, we choose to omit the LTR approaches in the experiments of our study and focus on the greedy post-processing heuristics for the diversity optimization of recommendations. This allows us to compare the diversity performances for different recommender systems and to analyze the interests of adding to them a diversification post-processing.

4. Experimental protocol

4.1. Datasets and preprocessing

Our experiments are based on three real-world datasets in different recommendation domains: *MovieLens-1M*[4] (the movie domain), *Anime*[5] (the Japanese manga domain) and *LibraryThing*[6] (the book domain). *MovieLens-1M* is a well known benchmark dataset containing 1 million ratings from 6,040 users on 3,900 movies. The *Anime* dataset is a public Kaggle dataset containing 1,597,830 ratings from 37,100 users on 12,294 anime. *LibraryThing* is a book recommendation dataset that contains 626,000 ratings of 7,112 users on 37,231 books.

The DBpedia knowledge base is an indispensable support for our study as we used it to build the CBF recommender, to construct the ICKG and UPKG knowledge graphs (see their definitions in Section 2.6) for the KGE recommender and to compute item semantic similarities for the estimation of recommendation diversity (cf. Eq. (7)). To leverage this gigantic source of knowledge, items need to be mapped with corresponding DBpedia entities that are identified through their URIs. Items in the *MovieLens-1M* and the *LibraryThing* datasets have been mapped by previous work (Di Noia et al., 2012) and the mappings are publicly available. The DBpedia knowledge graph has been evolving since these mappings were done. Some of them are now invalid, pointing to entity URIs that no longer exist. So, we corrected these mappings manually. For the Anime dataset, we first extracted all DBpedia entities of the *dbo:Manga* type and their English labels by using SPARQL queries through the DBpedia endpoint.[7] We then compared manga labels of DBpedia entities and anime names provided in the dataset's meta-data to define a similarity between Anime items and

[4] https://grouplens.org/datasets/movielens/1m/
[5] https://www.kaggle.com/CooperUnion/anime-recommendations-database
[6] https://www.librarything.com/
[7] https://dbpedia.org/sparql

Table 1
Characteristics of the (filtered) datasets.

Dataset	domain	# users (%kept)	# items (%kept)	# ratings (%kept)	sparsity
MovieLens-1M	movie	6,040 (100%)	3,301 (85%)	948,840 (95%)	95.09%
Anime	anime	27,454 (74%)	656 (5%)	1,007,384 (63%)	94.41%
LibraryThing	book	6,789 (95%)	11,695 (31%)	403,860 (65%)	99.49%

The original anime ratings contain implicit preference values i.e. −1, which means that the user watched the anime without rating it. These ratings were excluded from our experiments.

Table 2
DBpedia ontology properties taken into account for each dataset.

Dataset	Properties
MovieLens-1M	dbo:director, dbo:starring, dbo:distributor, dbo:writer, dbo:musicComposer, dbo:producer, dbo:cinematography, dbo:editing, dct:subject
Anime	dbo:author, dbo:publisher, dbo:magazine, dbp:director, dbp:genre, dbo:illustrator, dbp:writer, dct:subject
LibraryThing	dbo:author, dbo:publisher, dbo:literaryGenre, dbo:mediaType, dbo:subsequentWork, dbo:previousWork, dbo:series, dbo:country, dbo:language, dbo:coverArtist, dct:subject

Table 3
Number of triples making up the two knowledge graphs constructed for each dataset.

Dataset	ICKG (#triples)	UPKG (#triples)
MovieLens-1M	93,269	442,838
Anime	12,266	490,427
LibraryThing	134,333	207,884

DBpedia entities. An Anime dataset item was then mapped to a DBpedia entity when the normalized Levenshtein similarity between their labels was larger than 0.95. The mapping procedure allowed us to obtain 3,301 items of the movie database (85%); 656 items of the anime database (5%) and 11,695 items of the book database (31%) successfully mapped to DBpedia entities. We then removed unmapped items, and the associated ratings, from the corresponding datasets. It is worth noting that the dataset preprocessing leads to removing many items (especially from the anime dataset); however, most of the removed items have few ratings anyway. The most famous (manga, book, movie) items are often those present in DBpedia and having received the most ratings; whereas the less known items, often missing from DBpedia, tend to have few ratings. Hence, this pre-filtering of the datasets is not as drastic as it may seems. Even for the anime dataset, in which only 5% of the items are kept, 63% of the ratings are preserved.

Table 1 summarizes the main characteristics of the three preprocessed datasets. The sparsity of a dataset represents the proportion of the missing values among the whole user–item rating matrix. Finally, we split each dataset into the corresponding training set (80% of ratings) and test set (20% of ratings).

4.2. Knowledge graph constructions

For each dataset, we built two knowledge graphs, ICKG and UPKG. To build the three ICKG instances, we selected a subset of properties within the DBpedia ontology, as proposed in Palumbo et al. (2018b). The properties selected for each dataset are listed in Table 2. To build the three UPKG instances, the relation "*likes*" should be established between user and item entities. Following Di Noia et al. (2016), Palumbo et al. (2018b), we considered that a user u likes an item i if the rating $r_{u,i} \geq 4$ for the *MovieLens-1M* dataset (where ratings vary between 1 and 5) and $r_{u,i} \geq 8$ for the *Anime* and the *LibraryThing* datasets (where ratings vary between 1 and 10). Table 3 illustrates the number of triples encompassed in the ICKG and UPKG instances built for each dataset.

4.3. Recommender constructions

We followed the proposition of Steck (2013) that all the items in the catalog are eligible for recommendation. Hence, in our experiments, the $L(u)$ set contains all the items that u has not rated before, regardless of whether they are in the training set or in the test set. In our experiments, we compare seven recommenders of four different types that are listed in Table 4. For each dataset, we proceed as follows to construct the recommenders: (1) hyper-parameters tuning for each of the compared RSs toward accuracy and (2) models training with optimal hyper-parameters.

Table 4
Recapitulation of the tested recommenders.

Recommender (Reference)	Type of recommender	Brief description
CBF (Di Noia et al., 2012)	Content-based filtering	Method based on item semantic similarity
TopPopular (Ferrari Dacrema et al., 2019)	Non-personalized	Method based on item popularity
CBF-TopPopular (Eq. (6))	Hybrid	Method combining CBF and TopPopular
IBCF (Sarwar et al., 2001)	Collaborative filtering	Method based on item–item collaborative filtering
SVD (Koren et al., 2009)	Collaborative filtering	Method based on latent factors (matrix factorization)
KGE (Palumbo et al., 2018b)	Hybrid	Method based on knowledge graph embeddings
DNN (NeuMF) (He et al., 2017)	Collaborative filtering	Method based on deep neural networks

Hyper-parameters tuning. Tuning hyper-parameters of machine learning models is an optimization step aiming at finding the optimal combination of hyper-parameters that leads to the best performances. This is a crucial step for modern recommendation models as the behavior of recommenders could be strongly impacted by the used dataset and hyper-parameters (Jannach et al., 2015).

We adopted the Bayesian Optimization (BO) approach (Snoek et al., 2012) to tune hyper-parameters for the compared RSs. Compared with the random search and the grid search methods which are generally time-consuming, BO is an iterative procedure that takes into account the past choices made in order to consider the next set of hyper-parameters to evaluate, thus reducing largely the computation time. In our experiments, Nogueira (2014)'s *BayesianOptimization* package was used to tune hyper-parameters of RS models.

Among the seven recommenders compared in our study, the TopPopular and CBF models do not have hyper-parameters to be tuned. Specifically, for the TopPopular model, the relevance score of each item i in user u's unrated items $L(u)$ is computed using Eq. (1). For the CBF recommender, the relevance score of a given user–item pair i.e. $rel(u, i)$ is computed using Eq. (3) in which the semantic similarity between items is estimated by measuring the cosine similarity between item embedding vectors, learnt from the corresponding item content knowledge graph (ICKG). Therefore, high similarity scores are assigned to items which are semantically similar to users' highly rated ones.

For the remaining 5 recommenders, i.e. CBF-TopPopular, IBCF, SVD, DNN and KGE, we ran 20 BO iterations on each of the three datasets, to tune each model's hyper-parameters on each dataset. To this end, we opted firstly to split each training set (representing 80% of the dataset) into two sets: a sub-training set (80% of the training set) and a validation set (20% of the training set). The BO process takes the sub-training sets to train models and the hyper-parameters are tuned on the corresponding validation sets, by optimizing (maximizing) the mean average precision (MAP) metric (cf. Section 4.4). After the BO step, the resulting optimal hyper-parameter configurations were used to train each model on each entire training set (80% of the corresponding dataset) and the fine-tuned models were finally tested on each test set.

The sets of hyper-parameters that we tuned for each model are the followings:

- CBF-TopPopular: the weighting factor ω. In Eq. (6), we opted to normalize the popularity score of items, i.e. $TopPopular(i)$ into [0, 1] so that it has the same scale as $CBF(u, i)$.
- IBCF: the number k of the target item i's neighbors i.e. $|N_i|$ in Eq. (4).
- SVD: the number of factors $n_{factors}$, the number of training epochs of stochastic gradient descent, the learning rate η and the regularization term λ.
- KGE: we trained the TransE model on HybridKG (hybridization of ICKG and UPKG, cf. Section 2.6) for the corresponding dataset. TransE is a state-of-the-art knowledge graph embedding model that has been shown to perform better than alternative translation models on recommendation tasks (Palumbo et al., 2018a). Thus, the hyper-parameters include the embedding dimension d, the margin loss γ, the learning rate η, the batch size s_{batch} and the number of training epochs.
- DNN: the batch size s_{batch}, the number of embedding factors $n_{factors}$, the number of hidden layers in the MLP network n_{layers}, the learning rate η, the number of epochs and the number of negative samples for training n_{neg}.

The implementation of these models was achieved with open-source packages. Specifically, Hug (2020)'s *Surprise* Python library[8] was used to tune and train the IBCF and SVD models. For the DNN model, we used an available *PyTorch*[9] implementation.[10] The learning of the KGE model was done with the *Pykeen* package (Ali et al., 2019), i.e. a reference package for learning knowledge graph embeddings.

The configuration details of the hyper-parameters tuning along with the found optimal hyper-parameters for each dataset is provided in Tables A.7 and A.8 in Appendix. All resources of our study (datasets, scripts, mappings, embeddings etc.) are available in the GitHub repository.[11] This ensures the reproductibility of this study and should ease further analyses.

4.4. Evaluation metrics

We adopt the following metrics to evaluate the quality of a recommendation list of n items, in terms of accuracy and diversity.

[8] https://surprise.readthedocs.io/en/stable/prediction_algorithms_package.html
[9] https://pytorch.org/
[10] https://github.com/guoyang9/NCF
[11] https://github.com/lgi2p/Rec_Sys_Diversity_Study

4.4.1. Accuracy metrics

We evaluate the accuracy of the tested recommenders using the following standard information retrieval metrics:

- **Precision** represents the number of relevant items among the n recommended items for a given user u.
- **Recall** refers to the proportion of items liked by the user u that are included in her/his recommended list.
- **F1-measure** is the harmonic mean of precision and recall i.e. $\frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}$.
- **Mean Average Precision (MAP)** represents the mean of the Average Precision (AP) measures over all users. Unlike precision and recall, which do not consider positions of the relevant items, the AP measure is a ranked precision metric that places emphasis on highly ranked relevant items (Steck, 2013).

4.4.2. Diversity metrics

The diversity performance of recommenders is evaluated with the following metrics based on ILD and ILD_p measures (defined in Section 3.2.2):

- $\overline{\textbf{ILD}}$ represents the average of the intra-list diversity (ILD) values over all users. We also call this measure the *absolute* diversity as opposed to the next measure which evaluates the diversity of the recommendation lists in relation to the diversity of the items present in user profiles.
- $\mathbf{R}^2(\text{ILD}, \text{ILD}_p)$, denoted as the *relative* diversity, is the coefficient of determination of the ILD values (i.e. diversities of items recommended to users by a RS) vs. ILD_p values (i.e. diversities of corresponding users' profile items), throughout the entire user set (cf. Eq. (9)). This metric measures how well the diversity of recommendations provided by a RS meets its users' expectations in terms of diversity.

4.4.3. Semantic similarity

The similarity function $sim(i, j)$ in Eq. (7) is crucial for measuring the ILD of an item list (i.e. for both *absolute* and *relative* diversities). The similarity measurement between two items could be based on the user–item rating matrix. This has the advantage of always being directly feasible. On the other hand, semantic-based similarity measures require to have some semantic features associated to items, which could be difficult to obtain (see Section 4.1). However, rating-based similarities have some drawbacks of their own. The first one is related to the sparsity of the rating matrices: Having many missing values, as is often the case since most items are generally rated by only few users, could lead to unreliable similarity measurements (Aytekin & Karakaya, 2014). Considering a new item, with (almost) no available ratings, its similarity to other items will be inaccurately measured by rating-based similarity measures (low precision). The second drawback of rating-based similarities is the popularity bias that they introduce. Compared with obscure items, popular items tend to have more users in common (users having rated both items) and thus tend to be considered more similar with each other than unpopular items (Hou et al., 2018) (low accuracy). Therefore, the number of ratings per item influences the precision of the similarity estimation but also its accuracy.

Recent studies (Trattner & Jannach, 2020; Yao & Harper, 2018) have investigated item similarity functions with regard to human judgments. The main aim of those works was to study users' perceptions on the similarity between items. Yao and Harper (2018) asked users to rate how similar (to a target movie) the movies resulting from different similarity functions (e.g. rating-based and content-based) were. According to their user study, content-based movies are mostly considered to be "moderately" or "extremely" similar to the target movie. Trattner and Jannach (2020) investigated user perceptions on item similarities more deeply. Their study relied mainly on the content-based item similarities, trying to understand how different item features (e.g. title, plot, image etc. for movies) could determine users' perceived similarity between items and how users evaluate the usefulness of recommendations obtained by different similarity functions.

It seems preferable therefore to adopt semantic similarities for measuring $sim(i, j)$ when possible. Semantic similarity is typically used in the knowledge engineering domain to represent the relatedness between the meanings of two concepts and has been widely adopted in recommender systems (Carrer-Neto et al., 2012; Di Noia et al., 2014; Di Noia et al., 2016; García-Sánchez et al., 2020; Meymandpour & Davis, 2020; Passant, 2010). Our study also relies on semantic similarity between items. To estimate the semantic similarity between two items, we consider the cosine of their embedding vectors in the item content knowledge graph (ICKG). This semantic pairwise item similarity is used for the rating prediction of the tested CBF recommender (Eq. (3)) as well as for the ILD estimation of an item list (Eq. (7)).

4.5. Chosen parameters for studying diversification within recommenders

First, each of the tested RSs is supposed to recommend a list of $n = 10$ items with the highest predicted relevance scores. The recommenders are then evaluated in terms of accuracy and diversity (both *absolute* and *relative*).

Second, to analyze the impact of the diversification post-processing on each RS, we ran the bounded greedy algorithm (Algorithm 1) with two different objective functions, i.e. the classic (Eq. (11)) and the personalized (Eq. (12)) ones. For the bounded greedy algorithm, the bound m was set to 100 (n^2), i.e. $L_{candidates}(u)$ contains the top-100 rated items. For different recommenders and datasets, the ranges of their predicted scores, i.e. the $rel(\cdot)$ term in the objective functions, are quite different. To avoid possible bias, we opted to normalize these $rel(\cdot)$ values. Specifically, to normalize a given predicted relevance score $rel(\cdot)$, we applied the formula $\frac{rel(\cdot)-rel_{min}}{rel_{max}-rel_{min}}$ with rel_{min} and rel_{max} being the minimum and the maximum values of the $rel(\cdot)$ function for a specific method and dataset. Finally for the diversification factor α, we tested 10 different values ranging in [0, 0.9] with the step=0.1.

Last, as our study is fully based on offline experiments, we studied the impact of the bound size m of the bounded greedy post-processing on the RSs' performances. Indeed, larger bound sizes (e.g. $m > 100$) may promote RSs to find more diversified items, but it would also sacrifice more accuracy.

Table 5
Accuracy performance comparison of tested recommenders.

Measure	Dataset	CBF	TopPopular	CBF-TopPopular	IBCF	SVD	KGE	DNN
Precision	MovieLens	0.0322	0.0808	0.0869	0.0175	0.0406	**0.0887**	*0.0857*
	Anime	0.0586	0.1308	0.1352	0.0356	0.0734	*0.1313*	**0.1786**
	LibraryThing	*0.0302*	*0.0302*	0.0512	0.0019	0.0217	**0.0695**	0.0299
Recall	MovieLens	0.0259	0.0498	*0.0608*	0.0110	0.0278	0.0625	**0.0821**
	Anime	0.1256	0.3065	0.3159	0.0847	0.1772	*0.3114*	**0.4139**
	LibraryThing	*0.0539*	0.0434	0.0847	0.0024	0.0307	**0.1132**	0.0449
F1-Measure	MovieLens	0.0287	0.0616	*0.0715*	0.0135	0.0330	0.0733	**0.0839**
	Anime	0.0799	0.1834	0.1894	0.0501	0.1038	*0.1847*	**0.2495**
	LibraryThing	*0.0387*	0.0356	0.0638	0.0021	0.0254	**0.0861**	0.0359
MAP	MovieLens	0.0849	0.1661	0.1889	0.0442	0.0951	*0.1860*	**0.1956**
	Anime	0.1591	0.3476	0.3635	0.1026	0.1593	*0.3498*	**0.4270**
	LibraryThing	*0.0980*	0.0819	0.1617	0.0045	0.0707	**0.1966**	0.0889

For each dataset and metric, the best score is bolded, the second-to-best is underlined and the third one is in italics.

5. Results and discussion

In this section, we first present and discuss the performances of the tested RSs, in terms of accuracy, *absolute* diversity and *relative* diversity (RQ1). Second, we study the sensibility of the tested RSs to diversification post-processing (RQ2 and RQ3). Third, we report our findings w.r.t the experiments discussed in the last paragraph of the previous section. Finally, we discuss the limitations and implications of our study.

5.1. Accuracy and diversity performances of the tested RSs

Table 5 summarizes the accuracy metrics obtained by the seven tested RSs (of 4 different families) on the three considered datasets. Let us first consider the MovieLens and Anime datasets. RSs based on more recent techniques (i.e. KGE for knowledge graph embeddings and DNN for deep neural networks) are always more accurate than other standard models. Indeed, for all the 8 configurations (4 measures × 2 datasets), DNN is ranked the best model 7 times and KGE once. Item popularity-based models (TopPopular and CBF-TopPopular) are slightly less accurate than the best models with CBF-TopPopular being slightly better than TopPopular. Note that a simple, while fine-tuned, mix of CBF and TopPopular leads to 6 times (out of 8) the second most accurate model. The SVD and the CBF models have comparable accuracy performances on these two datasets and the IBCF approach has the lowest accuracy. For instance, on the MovieLens dataset, the MAP is ~ 0.19 for DNN, KGE and CBF-TopPopular; around 0.16 for TopPopular; ~ 0.09 for SVD and CBF and as low as 0.04 for IBCF. The performances rank is unchanged when considering the MAP for the Anime dataset, while the difference between DNN (~ 0.43) and KGE/CBF-TopPopular (~ 0.35/~ 0.36) is larger. The same observation can be made for other metrics on the Anime dataset, e.g. ~ 0.25 for DNN v.s ~ 0.18 for KGE and CBF-TopPopular, considering F1-Measure.

Surprisingly enough, the accuracy results are quite different on the LibraryThing dataset. There, recommendation approaches leveraging item knowledge, i.e. either content-based (CBF) or hybrid (KGE and CBF-TopPopular) are more accurate than pure CF-based approaches. Indeed, on this dataset, hybrid models are always the best models with KGE always coming first and CBF-TopPopular second, for all 4 metrics. They are followed by CBF, always the third ranked model. For instance, KGE obtains the highest MAP score (~ 0.20) followed by CBF-TopPopular (~ 0.16) and CBF (~ 0.10). The deep neural network approach is only fourth (~ 0.09) followed by TopPopular (~ 0.08), SVD (~ 0.07) and IBCF (~ 0.005). The poor performances of CF methods on the LibraryThing dataset may be due to the high sparsity of its rating matrix (99.49%, see Table 1). This may prevent CF methods to learn an adequate representation of the user and item factor vectors. In such a high data sparsity context, even a simple content-based approach could lead to more accurate recommendations than the state-of-the-art deep learning approach. This confirms, if needed, that considering item knowledge allows to alleviate parts of the cold-start and data sparsity problems faced by CF approaches (Natarajan et al., 2020).

Table 6 summarizes the diversity metrics scored by the seven tested RSs on the three considered datasets. Let us first discuss the absolute diversity ($\overline{\text{ILD}}$) of the recommendations. The CBF model, which relies exclusively on item knowledge, leads to much less diversified recommendations than other models. For instance, on the MovieLens dataset, the average ILD of CBF is only ~ 0.28 while it ranges from ~ 0.40 (CBF-PopPopular) to ~ 0.70 (IBCF) for other RSs. The same trend is observed for the other two datasets. This confirms that content-based recommendations are generally over-specialized. At the other end of the spectrum, IBCF and SVD, the two traditional CF models, provide the most diversified recommendations. The remaining 4 models stand between those extremes: DNN is slightly more diversified than the popularity-based model (TopPopular) and the two hybrid models (KGE and CBF-TopPopular) on the movie and book datasets; TopPopular is the second most diversified model on the anime dataset. If the aim is simply to maximize the diversity of the recommendations, then the traditional CF in general, and the IBCF model in particular, appear to be an optimal choice. However, one can wonder how these absolute levels of diversity compare to users' expectations or needs.

Table 6
Diversity performance comparison of the tested recommenders.

Measure	Dataset	CBF	TopPopular	CBF-TopPopular	IBCF	SVD	KGE	DNN
$\overline{\text{ILD}}$	MovieLens ($\overline{\text{ILD}_p}$: 0.59)	0.2761	0.4953	0.3988	**0.6998**	0.6623	0.5144	*0.5417*
	Anime ($\overline{\text{ILD}_p}$: 0.75)	0.5952	*0.7406*	0.7039	**0.8057**	0.7786	0.7389	0.7260
	LibraryThing ($\overline{\text{ILD}_p}$: 0.75)	0.3283	0.6019	0.4006	**0.8405**	0.7110	0.5901	*0.6576*
$R^2(\text{ILD}, \text{ILD}_p)$	MovieLens	0.0605	0.0065	0.0362	0.0001	0.0073	*0.0485*	**0.1030**
	Anime	**0.0542**	0.0428	0.0057	0.0005	0.0093	*0.0205*	0.0103
	LibraryThing	0.1870	0.0001	**0.3119**	0.0001	0.0001	*0.1342*	0.0126

For each dataset and metric, the best score is bolded, the second-to-best is underlined and the third one is in italics.

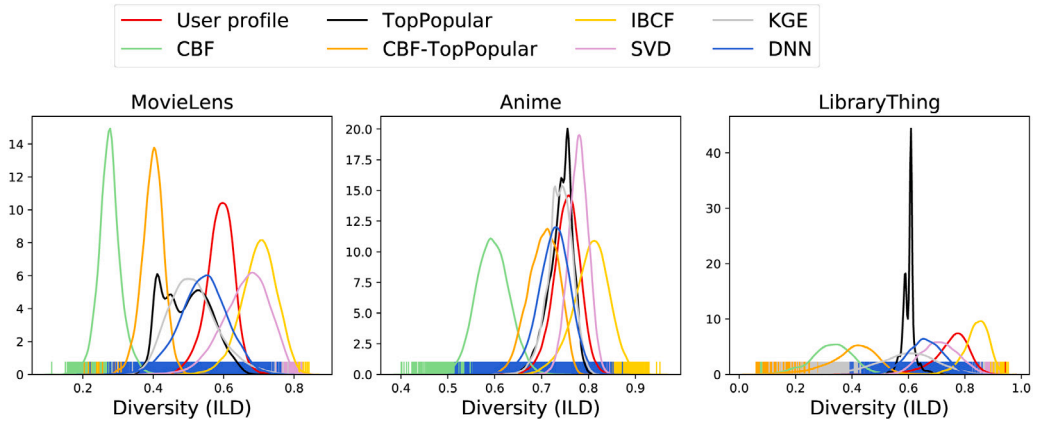

Fig. 1. Diversity (ILD) distributions of the lists of items provided by different recommenders as well as users' profiles diversity (ILD_p) distributions (red curves), for all the datasets. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Considering the relative diversity metric, i.e. confronting the diversity of the recommendations ($\overline{\text{ILD}}$) to that observed within user profiles ($\overline{\text{ILD}_p}$), allows to go a step further. Indeed, this comparison estimates the adequacy between the level of diversity proposed by these RSs and the level expected by users. According to Table 6, recent KGE and DNN models seem to perform the best in terms of relative diversity. The intermediate diversity level of their recommendation lists is more coherent with the diversity observed in the user profiles. For instance, on the Anime dataset where $\overline{\text{ILD}_p}$ is about 0.75, the $\overline{\text{ILD}}$ is $\sim 0.74/\sim 0.73$ for KGE/DNN, whereas it exceeds at ~ 0.80 for IBCF and is below 0.60 for CBF. According to these average metrics, KGE and DNN seem to perform reasonably well and be close to user expectations in terms of diversity. Fig. 1 displays, for all considered datasets, the distribution of ILD values of each tested RS along with the distribution of the ILD_p values (red curves). This visual representation confirms that IBCF (gold curves) leads to over-diversified recommendations whereas content-based models such as CBF and CBF-TopPopular (cf. green and orange curves) tend to produce under-diversified recommendations. The more recent RSs (KGE and DNN) lead to ILD distributions closer to the ILD_p ones.

However, even if the ILD distribution and the average ILD of some RS models are close enough to those of ILD_p (notably for the Anime dataset), their most diversified recommendations do not seem to be suggested to the right users. Indeed, the most striking result here is probably the strong discrepancy between the diversity of individual user profiles (ILD_p) and the diversity of the recommendations made to them by a RS (ILD). The R^2 values between these two sets of values (indicating the extent to which they are linearly correlated) are quite low for all 21 configurations (3 datasets × 7 RSs). All R^2 values are below 0.2 as shown in Table 6 except for CBF-TopPopular on LibraryThing, which is ~ 0.31 but remains still low). Such low correlation values indicate that even DNN and KGE fail to propose the correct diversity level to the right user. This is clearly illustrated by Fig. 2 that displays the scatter plot representation of the ILD values v.s ILD_p values.

5.2. Impact of the diversification post-processing

The diversification post-processing can be seen as a re-ranking process during which items, initially ordered according to their relevance score, are re-ordered so that the top items are both relevant and diverse. Here, we compare the results of the greedy re-ranking method (Algorithm 1) using two different objective functions: the classic one and the personalized one, denoted as *cls-greedy* and *pers-greedy*, respectively. The two methods rely on a parameter, denoted as α, to adjust the compromise between relevance (accuracy) and diversity.

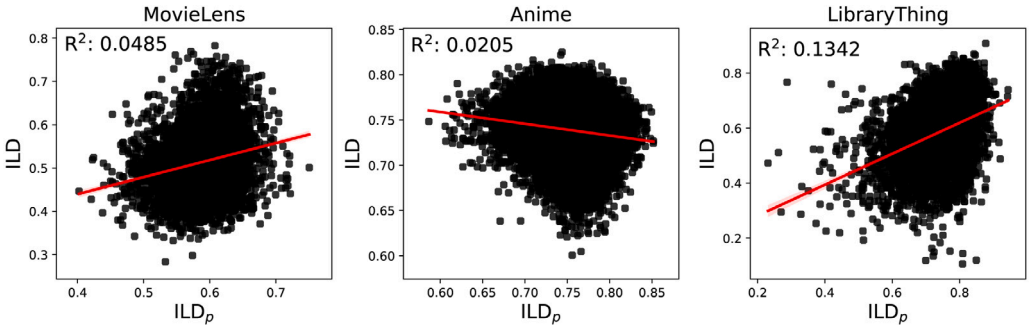

Fig. 2. Scatter plot of recommended diversity v.s profile diversity values (ILD values v.s ILD_p ones) for different datasets. The recommendations are based on the KGE model.

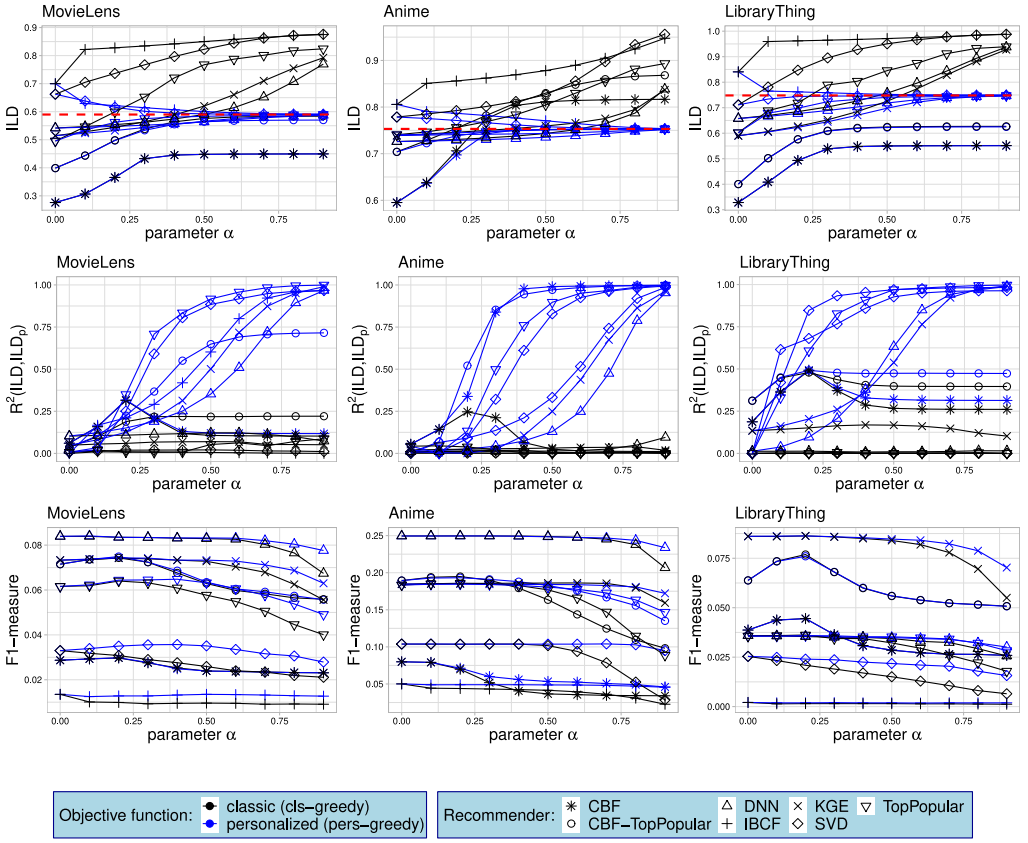

Fig. 3. The *absolute* diversity (top row), the *relative* diversity (mid row) and the accuracy (bottom row) evaluations for the classic and the personalized greedy objectives.

These two post-processing approaches were tested in combination to the seven recommendation approaches on the three benchmark datasets. Fig. 3 provides, for each of these 42 combinations, a graphic representation of the evolution of the *absolute* diversity (ILD), the *relative* diversity (R^2) and the accuracy of the recommendations when increasing α. Note that, for simplicity sake, the unique accuracy metric discussed in this subsection is the F1-measure since the same trends are observed for the other accuracy metrics, i.e. Precision, Recall and MAP.

As expected, the *cls-greedy* post-processing increases the ILD values for all RSs and datasets and the higher the α values the larger the ILD (Fig. 3 top row). The only exceptions to this rule are observed for content-based RSs, i.e. CBF (∗) and CBF-TopPopular (∘), on the MovieLens or LibraryThing benchmarks where the $\overline{ILD}$ reaches a plateau. It seems that content-based RSs provide so little diversity in their candidate lists (i.e. the top-100 ranked items in our context), that it imposes a limitation on the diversification post-processing. In other words the top-100 items returned by content-based approaches are sometimes so similar that there is no

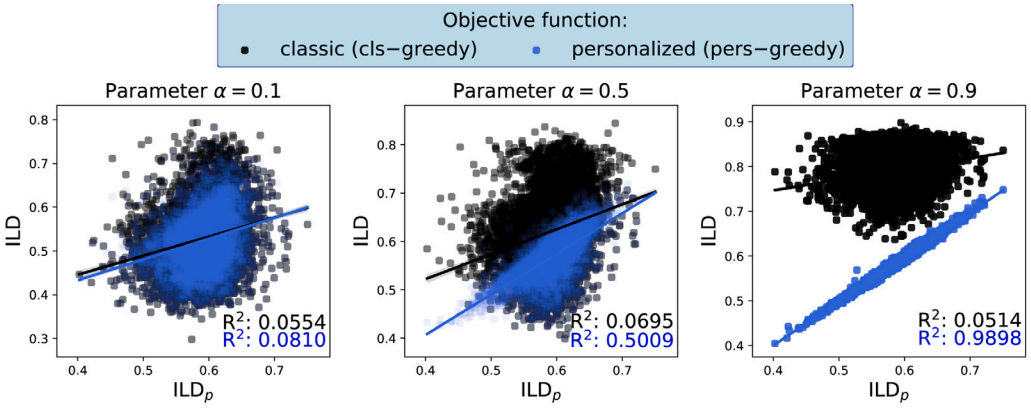

Fig. 4. Scatter plot of ILD (recommended diversity) vs. ILD_p (profile diversity) values for three diversification factor values (α) with two different objective functions. The recommendations are provided by the KGE model on the MovieLens dataset.

way to build a highly diversified subset of them. To overcome this limitation, larger sizes of candidate items may be useful, as we shall discuss in the next subsection. In all other cases, when the α parameter of *cls-greedy* is increased, the $\overline{ILD}$ is also largely increased, regardless of users' diversity needs (i.e., the $\overline{ILD}_p$, represented by the red dashed lines in Fig. 3 top row). In contrast, when the α parameter of *pers-greedy* is increased, the $\overline{ILD}$ tends toward $\overline{ILD}_p$ (Fig. 3 top row). The plot of the relative diversity (Fig. 3 mid row) indicates that, when using *pers-greedy*, increasing α improves the consistency of the recommendation and user profile diversities (R^2 values tend toward 1). In contrast, when using *cls-greedy*, increasing α often leads to decreasing the correlation between the diversity of a user profile and that in her/his recommendations (R^2 values tend toward 0) even more. The difference between those two post-processings, regarding the correlation between ILD and ILD_p values, is clarified by the dot plots of Fig. 4.

Regarding the impact of the α parameter, content-based RSs have, once again, an atypical behavior. Indeed, when combined with CBF and CBF-TopPopular recommenders, the *cls-greedy* post-processing also allows to increase the R^2 metrics when mid-range α values are used. This is probably due to the low diversity within their recommendations so that introducing some diversity, thanks to a post-processing (personalized or not), has only advantages. Indeed, CBF ($*$) and CBF-TopPopular ($\circ$) are the only two tested RSs for which it is possible to improve simultaneously the *absolute* diversity (ILD), the *relative* diversity (R^2) and the accuracy (F1-measure). An α value close to 0.25 seems optimal for content-based approaches: ILD is much better than with $\alpha = 0$ while the F1-measure is almost unchanged (α from 0 to 0.25) for MovieLens and Anime and even improved for LibraryThing. For all other RSs, improving the diversity metrics always comes at the cost of a lower accuracy (Fig. 3 bottom row). However, the *cls-greedy* optimization leads to a more important accuracy sacrifice in comparison with the *pers-greedy* optimization (notably when α becomes larger than 0.5). The latter tends to preserve higher levels of accuracy, considering all three datasets.

5.3. Impact of the bound size on recommendation diversity and accuracy

In this subsection, we report our findings w.r.t the last experimentation described at the end of Section 4.

As previously discussed, the bound size m used in the bounded greedy (BG) optimization might also be a kind of trade-off between diversity and accuracy for RSs. When the number of candidate items increases, BG would, on the one hand, offer more opportunity for RSs to find more diversified items but, on the other hand, it could also lead to a larger decrease of the accuracy. Fig. 5 illustrates such a case where increasing the bound size leads CBF and CBF-TopPopular to recommend item lists that are more diverse but less accurate. More generally, for most of the tested RSs (notably for content-based ones, i.e. CBF and CBF-TopPopular), their ILD values tend to increase with the candidate list sizes (m from 50 to 300), for both classic and personalized optimization objectives. Interestingly, the impact of the m parameter on the recommendation diversity seems to be negligible for KGE and DNN, i.e. the two recent RSs with best accuracy performances (see Table 5). Fig. 5 suggests that for these two RSs the set of 50 top-ranked items would be enough (optimal) for applying a greedy search. In terms of F1-measure, the impact of m seems negligible for all RSs except for the content-based ones (CBF and CBF-TopPopular), in which the accuracy decreases sufficiently to be visually obvious on this plot.

Overall, the impact of the bound size m is not as remarkable as we expected. The choice of $m = 100$ (square of the number of recommendations) that we made for our study is quite reasonable as Fig. 5 shows that the diversity (ILD) and the accuracy (F1-measure) results do not change much from 100 candidate items.

5.4. Limitation, discussion and implication

In our study, individual users' diversity needs are estimated from user profile items (i.e. ILD_p), which may have some limitations. First, different users may have different rating behaviors as some users rate only few items while others rate much more. Even

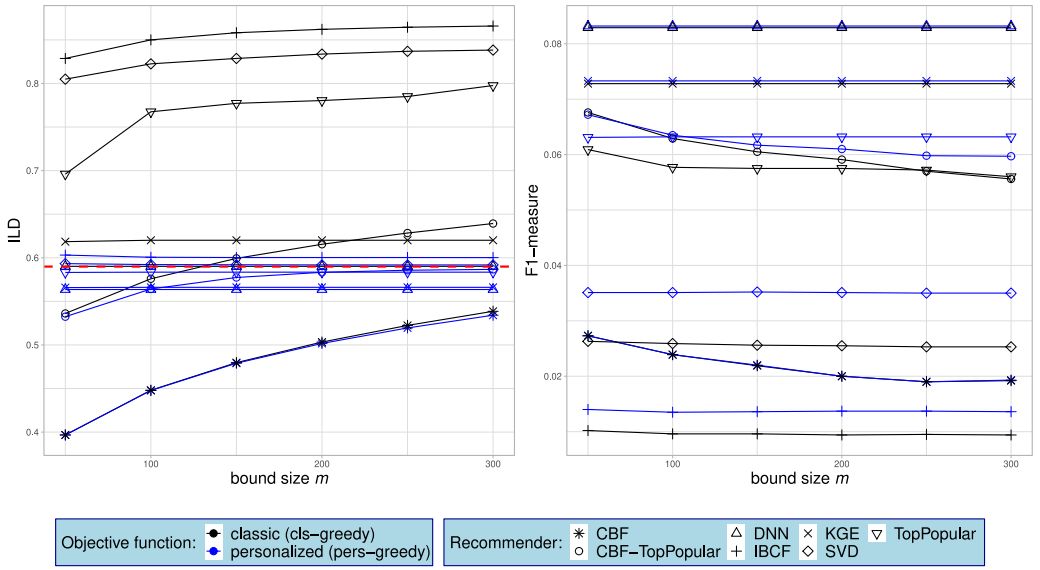

Fig. 5. Accuracy (F1-measure) and absolute diversity (ILD) of RSs on MovieLens for different bound sizes m during the greedy post-processing. The diversification factor α is set to 0.5 for both objective functions.

though the ILD metric estimates an average value, the resulting profile diversities might lead to different levels of reliability. Since we currently ignore the actual ratings of the items to estimate user diversity expectation, a possible alternative would be to consider, if available, implicit feedbacks (Aggarwal, 2016) and to measure the ILD of the items with which users have interacted (e.g. items bought on a merchant site, movie watched on a streaming platform, etc.). This generally constitutes a much larger set of items than the set of rated ones, and could hence lead to a more robust estimation of the diversity adapted to each user, notably for those having rated few items.

Second, we followed Di Noia et al. (2014), Meymandpour and Davis (2020) to consider all the items rated by the user for estimating her/his ILD_p value. As an alternative, one may consider focusing only on users' liked items (referring also to users' positive profile) while measuring the ILD_p values. Indeed, it is more reasonable for RSs to only consider users' liked items in cases when the goal is to provide users with items they might like (i.e. accuracy criterion). In the diversity optimization context, only considering users' liked items might lead to slightly lower ILD_p values as users' highly rated items may share some common semantic properties. The low $\overline{ILD}$ values of the tested CBF approach demonstrate this point as the algorithm returns items which are semantically similar to users' highly rated ones. On the other hand, considering all of users' rated items allows to take a larger range of items into account. An assumption is made here that as long as a user has rated an item (even with a low rating), it has attracted her/his attention. However, this proposition is far from obvious and probably depends on each user's rating behavior.

The third point is that users' diversity needs may also depend on other factors such as their mood, personality, etc. and can change over time. For example, 62% of the participants involved in Yao and Harper (2018)'s study considered mood as an important feature while choosing the next movie to watch. It might be worth considering these factors while measuring an individual user's diversity needs. Another main limitation of our study is that it is firmly based on the quality of item knowledge encoded in the DBpedia knowledge graph, which may contain only partial facts about items (Färber et al., 2018).

The present study is exclusively based on offline experiments and assumes that the semantic similarity-based ILD metric could capture active users' perception on the diversity (also on the variation of diversity) of the recommended items. These common assumptions could be valid as demonstrated by the user study presented in Castagnos et al. (2013). In their study, users were presented with TV programs generated from some classic RSs including content-based and CF-based ones. The diversity of recommendations was measured with the semantic similarity-based ILD metric by using attributes of items. Ekstrand et al. (2014) also reported active users' perceptions on their recommended movie lists w.r.t different evaluation dimensions e.g. accuracy, diversity, satisfaction etc. Their diversity measurement was again based on the ILD metric with item similarity being estimated by tag genome vectors (Vig et al., 2012), modeling items in a community knowledge-based tag space. Users were asked to communicate their perceptions on recommendation lists from three RSs (IBCF, UBCF and SVD). Their findings demonstrated that IBCF provided more diversified items than SVD, which is coherent with our findings (see Table 6). That is why we believe that the findings of our offline-based experiments should reflect effectively the diversity performances of the tested RSs.

In terms of research implications, the results of our study indicate that the *absolute* diversity level of recommendations provided by the state-of-the-art memory-based collaborative filtering algorithm (IBCF) is always much higher than that observed in user profiles (e.g. the gold curve is almost right-shifted from the red one in Fig. 1 for the movie dataset). This type of RSs is still widely used in real-world recommendation engines (Amazon for instance) and our results provide a clear guideline concerning them: it is counterproductive to further increase their *absolute* diversity by chaining them with a classic bounded greedy diversification

procedure. On the other hand, for content-based approaches in general, our study suggests that diversity optimization methods are much more relevant as those RSs provide low levels of *absolute* diversity. Recent recommendation approaches such as the studied KGE and DNN models provide highly accurate recommendations which are, on average, diversified enough. However, they do not perform so well in terms of the *relative* diversity as is also the case with all other tested RSs. None of the tested RSs was able to recommend to users item lists with optimal diversity levels relevant to their rating profiles.

As the current trend of the recommender system community is based on deep neural networks and graph embeddings (Ferrari Dacrema et al., 2019; Xu et al., 2020; Zhang et al., 2019), the results of our study might inspire researchers or RS developers to design new models that account for users' diversity needs, as they highlight the suboptimality of individual diversification. We also hope that this work could encourage the provision of benchmark datasets containing implicit feedbacks (such as the Deskdrop[12] plaform) in order to get information about items users have interacted with and not just the ones they have rated. This is a crucial information as most users watch much more movies or read much more books than they rate. Cross checking these two types of information could probably help better capture users' needs, especially regarding the diversification criterion.

6. Conclusion

Recommender systems have long been evaluated based on their accuracy. The importance of providing diversified recommendations is also acknowledged now. However, the impact of diversification post-processing on the main types of recommender systems has not been extensively studied. This paper provides a systematic and comprehensive study of the four main families of recommender systems (non-personalized, content-based, collaborative filtering-based and hybrid) with respect to both accuracy and diversity. This study was done using three large datasets classically used to benchmark RSs. We argue that considering only the absolute level of diversity of a recommendation list is not enough and that the diversity of each recommendation list should also be evaluated with respect to the target user of the recommendations. We proposed to use the diversity of the items rated by a user as a proxy of the diversification she/he is expecting from a RS.

The main findings of this study are the followings. First, different RS approaches lead to recommendations that have very different levels of diversification. Therefore, while it may seem reasonable to increase the diversity for some of them, others would benefit from a post-processing taming down their tendency to over-diversification. Considering the average of the user profile diversities as a reference, content-based approaches appear to provide under-diversified items while the classical collaborative filtering approaches rather lead to over-diversified ones. More recent approaches, based on deep neural networks or knowledge graph embeddings, provide recommendations whose average diversity is much closer to that of users' profiles. These results were somehow expected: to recommend items using only their similarities does not leave much room for diversification while recommending items based only on peers' suggestions may rapidly lead to a list *à la Prévert*.

The second main finding is that none of the tested RSs, even the most recent ones, seems to provide recommendations that have a diversity level in line with the user profile for whom the recommendations are intended. Coupling these approaches to a classical post-processing diversification makes things even worse. Such a post-processing only aims at maximizing the diversity of the recommended item list while favoring as much as possible the top rated items. The result is a decrease in the RS accuracy coupled with an increase in the recommendation list diversity, even when it was already over-diversified. This kind of post-processing seems only beneficial for content-based approaches as they are the only ones that really lead to under-diversified recommendations. For all other tested RS models, this post-processing decreases recommendation accuracy, moves the average recommendation diversity further away from the average profile diversity and decreases the correlation between user profile diversities and recommendation diversities.

The third main finding is that an alternative diversification post-processing is worth studying. We proposed a simple variant of the classical diversification post-processing. It aims at building a recommendation list with a diversity that is as close as possible to the user profile diversity. In other words, instead of maximizing an absolute level of diversity, it aims at maximizing the adequation between the diversity of the user profile and that of the recommendations made to the user. This simple shift in perspective allows to drastically improve things. For all tested RSs and benchmarks, using this personalized post-processing allows to move the average recommendation diversity closer to the average user profile diversity and to increase the correlation between user profile diversities and recommendation diversities while decreasing the accuracy much less than when the classical diversification objective is used.

Increasing recommendation diversity via a post-processing is usually perceived as beneficial. This study demonstrates that such post-treatment may indeed be detrimental for most RSs for which the level of diversity is already close to, or even above, the diversity observed in user profiles. This paper provides a simple solution to take users' specificity into account when adjusting the level of diversity of their recommendations. This simple approach proves to be relevant and we hope that it would incite researchers to work on this topic so that recommender systems could better fit user expectations in terms of both accuracy and diversity.

CRediT authorship contribution statement

Yu Du: Conceptualization, Methodology, Software, Validation, Writing – original draft, Visualization. **Sylvie Ranwez:** Supervision, Writing – review & editing, Validation. **Nicolas Sutton-Charani:** Supervision, Writing – review & editing, Formal analysis. **Vincent Ranwez:** Supervision, Writing – review & editing, Visualization, Formal analysis, Validation.

[12] https://www.kaggle.com/gspmoreira/articles-sharing-reading-from-cit-deskdrop

Table A.7
Search spaces of compared RS models' hyper-parameters during the Bayesian optimization process.

Recommender	Hyper-parameter search space
CBF-TopPopular	$\omega \in [0, 1]$
IBCF	$k \in [1, 60]$
SVD	$n_{factors} \in [50, 200]$, $epoch \in [1, 100]$, $\eta \in [0, 0.1]$, $\lambda \in [0, 0.1]$
KGE	$d \in [10, 200]$, $\gamma \in [1, 10]$, $\eta \in [0.001, 0.1]$, $s_{batch} \in [16, 256]$, $epochs \in [100, 1000]$
DNN	$s_{batch} \in [16, 256]$, $n_{factors} \in [8, 128]$, $n_{layers} \in [1, 4]$, $\eta \in [0.001, 0.1]$, $epochs \in [1, 30]$, $n_{neg} \in [0, 10]$

Table A.8
Found optimal configurations of RS models' hyper-parameters for each dataset.

Dataset	Recommender	Found optimal hyper-parameters
Anime	CBF-TopPopular	$\omega = 0.4176$
	IBCF	$k = 3$
	SVD	$n_{factors} = 200$, $epoch = 3$, $\eta = 0.0003$, $\lambda = 0.0626$
	KGE	$d = 147$, $\gamma = 4$, $\eta = 0.001$, $s_{batch} = 116$, $epochs = 232$
	DNN	$s_{batch} = 174$, $n_{factors} = 128$, $n_{layers} = 1$, $\eta = 0.001$, $epochs = 10$, $n_{neg} = 10$
MovieLens-1M	CBF-TopPopular	$\omega = 0.2275$
	IBCF	$k = 2$
	SVD	$n_{factors} = 175$, $epoch = 11$, $\eta = 0.0354$, $\lambda = 0.0813$
	KGE	$d = 199$, $\gamma = 5$, $\eta = 0.0421$, $s_{batch} = 255$, $epochs = 715$
	DNN	$s_{batch} = 256$, $n_{factors} = 66$, $n_{layers} = 1$, $\eta = 0.001$, $epochs = 30$, $n_{neg} = 10$
LibraryThing	CBF-TopPopular	$\omega = 0.1211$
	IBCF	$k = 3$
	SVD	$n_{factors} = 199$, $epoch = 1$, $\eta = 0.0011$, $\lambda = 0.0815$
	KGE	$d = 194$, $\gamma = 6$, $\eta = 0.0435$, $s_{batch} = 21$, $epochs = 439$
	DNN	$s_{batch} = 144$, $n_{factors} = 44$, $n_{layers} = 1$, $\eta = 0.001$, $epochs = 30$, $n_{neg} = 10$

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Configuration details of Bayesian optimization for hyper-parameters tuning of the compared RS models

See Tables A.7 and A.8.

References

Abdollahpouri, H. (2019). Popularity bias in ranking and recommendation. In *AIES '19*, *Proceedings of the 2019 AAAI/ACM conference on AI, ethics, and society* (pp. 529–530). New York, USA: http://dx.doi.org/10.1145/3306618.3314309.

Adomavicius, G., & Kwon, Y. (2012). Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering*, *24*(5), 896–911. http://dx.doi.org/10.1109/TKDE.2011.15.

Aggarwal, C. C. (2016). An introduction to recommender systems. In *Recommender systems: the textbook* (pp. 1–28). Cham: Springer International Publishing, http://dx.doi.org/10.1007/978-3-319-29659-3_1.

Ali, M., Jabeen, H., Hoyt, C. T., & Lehmann, J. (2019). The KEEN universe. In C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois, & F. Gandon (Eds.), *Vol. 11779*, *The semantic web - ISWC 2019* (pp. 3–18). Cham: Springer International Publishing, http://dx.doi.org/10.1007/978-3-030-30796-7_1.

Aytekin, T., & Karakaya, M. O. (2014). Clustering-based diversity improvement in top-n recommendation. *Journal of Intelligent Information Systems*, *42*(1), 1–18. http://dx.doi.org/10.1007/s10844-013-0252-9.

Bordes, A., Glorot, X., Weston, J., & Bengio, Y. (2014). A semantic matching energy function for learning with multi-relational data. *Machine Learning*, *94*(2), 233–259. http://dx.doi.org/10.1007/s10994-013-5363-6.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th international conference on neural information processing systems, NIPS'13* (pp. 2787–2795).

Bradley, K., & Smyth, B. (2001). Improving recommendation diversity. In *Proceedings of the 12th irish conference on artificial intelligence and cognitive science, Maynooth, Ireland, AICS'01* (pp. 85–94).

Carrer-Neto, W., Hernández-Alcaraz, M. L., Valencia-García, R., & García-Sánchez, F. (2012). Social knowledge-based recommender system. Application to the movies domain. *Expert Systems with Applications*, *39*(12), 10990–11000. http://dx.doi.org/10.1016/j.eswa.2012.03.025.

Castagnos, S., Brun, A., & Boyer, A. (2013). When diversity is needed... but not expected!. In *Proceedings of the 3rd international conference on advances in information mining and management, IMMM'13, Lisbon, Portugal* (pp. 44–50).

Cheng, P., Wang, S., Ma, J., Sun, J., & Xiong, H. (2017). Learning to recommend accurate and diverse items. In *WWW '17*, *Proceedings of the 26th international conference on world wide web* (pp. 183–192). http://dx.doi.org/10.1145/3038912.3052585.

Di Noia, T., Mirizzi, R., Ostuni, V. C., Romito, D., & Zanker, M. (2012). Linked open data to support content-based recommender systems. In *I-SEMANTICS '12*, *Proceedings of the 8th international conference on semantic systems* (pp. 1–8). New York, USA: http://dx.doi.org/10.1145/2362499.2362501.

Di Noia, T., Ostuni, V. C., Rosati, J., Tomeo, P., & Di Sciascio, E. (2014). An analysis of users' propensity toward diversity in recommendations. In *RecSys '14*, *Proceedings of the 8th ACM conference on recommender systems* (pp. 285–288). New York, USA: http://dx.doi.org/10.1145/2645710.2645774.

Di Noia, T., Ostuni, V. C., Tomeo, P., & Sciascio, E. D. (2016). Sprank: Semantic path-based ranking for top-n recommendations using linked open data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *8*(1), http://dx.doi.org/10.1145/2899005.

Ebesu, T., Shen, B., & Fang, Y. (2018). Collaborative memory network for recommendation systems. In *SIGIR '18*, *Proceedings of the 41st international ACM SIGIR conference on research & development in information retrieval* (pp. 515–524). New York, USA: http://dx.doi.org/10.1145/3209978.3209991.

Ekstrand, M. D., Harper, F. M., Willemsen, M. C., & Konstan, J. A. (2014). User perception of differences in recommender algorithms. In *RecSys '14*, *Proceedings of the 8th ACM conference on recommender systems* (pp. 161–168). New York, USA: http://dx.doi.org/10.1145/2645710.2645737.

Eskandanian, F., Mobasher, B., & Burke, R. (2017). A clustering approach for personalizing diversity in collaborative recommender systems. In *UMAP '17*, *Proceedings of the 25th conference on user modeling, adaptation and personalization* (pp. 280–284). http://dx.doi.org/10.1145/3079628.3079699.

Färber, M., Bartscherer, F., Menne, C., & Rettinger, A. (2018). Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago. *Semantic Web*, *9*(1), 77–129. http://dx.doi.org/10.3233/SW-170275.

Ferrari Dacrema, M., Cremonesi, P., & Jannach, D. (2019). Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In *RecSys '19*, *Proceedings of the 13th ACM conference on recommender systems* (pp. 101–109). New York, USA: http://dx.doi.org/10.1145/3298689.3347058.

García-Sánchez, F., Colomo-Palacios, R., & Valencia-García, R. (2020). A social-semantic recommender system for advertisements. *Information Processing & Management*, *57*(2), Article 102153. http://dx.doi.org/10.1016/j.ipm.2019.102153.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. In *WWW '17*, *Proceedings of the 26th international conference on world wide web* (pp. 173–182). http://dx.doi.org/10.1145/3038912.3052569.

Herlocker, J., Konstan, J. A., & Riedl, J. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, 287–310. http://dx.doi.org/10.1023/A:1020443909834.

Hou, L., Pan, X., & Liu, K. (2018). Balancing the popularity bias of object similarities for personalised recommendation. *The European Physical Journal B*, *91*, http://dx.doi.org/10.1140/epjb/e2018-80374-8.

Hug, N. (2020). Surprise: A python library for recommender systems. *Journal of Open Source Software*, *5*(52), 2174. http://dx.doi.org/10.21105/joss.02174.

Jannach, D., Lerche, L., Kamehkhosh, I., & Jugovac, M. (2015). What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Modeling and User-Adapted Interaction*, *25*(5), 427–491. http://dx.doi.org/10.1007/s11257-015-9165-3.

Jugovac, M., Jannach, D., & Lerche, L. (2017). Efficient optimization of multiple recommendation quality factors according to individual user tendencies. *Expert Systems with Applications*, *81*, 321–331. http://dx.doi.org/10.1016/j.eswa.2017.03.055.

Khan, B. M., Mansha, A., Khan, F. H., & Bashir, S. (2017). Collaborative filtering based online recommendation systems: A survey. In *ICICT '17*, *Proceedings of the 7th international conference on information and communication technologies* (pp. 125–130). http://dx.doi.org/10.1109/ICICT.2017.8320176.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 30–37. http://dx.doi.org/10.1109/MC.2009.263.

Kunaver, M., & Požrl, T. (2017). Diversity in recommender systems–a survey. *Knowledge-Based Systems*, *123*, 154–162. http://dx.doi.org/10.1016/j.knosys.2017.02.009.

Lee, D., & Hosanagar, K. (2014). Impact of recommender systems on sales volume and diversity. In *Proceedings of the 35th international conference on information systems, ICIS '14*.

Li, S., Zhou, Y., Zhang, D., Zhang, Y., & Lan, X. (2017). Learning to diversify recommendations based on matrix factorization. In *Proceedings of the 15th international conference on dependable, autonomic and secure computing, 15th international conference on pervasive intelligence and computing, 3rd international conference on big data intelligence and computing and cyber science and technology congress, IEEE '17* (pp. 68–74). http://dx.doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec.2017.26.

Lops, P., de Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook* (pp. 73–105). Boston, MA: Springer US, http://dx.doi.org/10.1007/978-0-387-85820-3_3.

MacKenzie, I., Meyer, C., & Noble, S. (2013). How retailers can keep up with consumers. *McKinsey & Company*, *18*.

McNee, S. M., Riedl, J., & Konstan, J. A. (2006). Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI '06*, *Proceedings of the conference on human factors in computing systems* (pp. 1097–1101). http://dx.doi.org/10.1145/1125451.1125659.

Meymandpour, R., & Davis, J. G. (2020). Measuring the diversity of recommendations: a preference-aware approach for evaluating and adjusting diversity. *Knowledge and Information Systems*, *62*(2), 787–811. http://dx.doi.org/10.1007/s10115-019-01371-0.

Natarajan, S., Vairavasundaram, S., Natarajan, S., & Gandomi, A. H. (2020). Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. *Expert Systems with Applications*, Article 113248. http://dx.doi.org/10.1016/j.eswa.2020.113248.

Nickel, M., Tresp, V., & Kriegel, H. P. (2011). A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th international conference on machine learning, ICML '11* (pp. 809–816).

Nogueira, F. (2014). Bayesian optimization: Open source constrained global optimization tool for python. https://github.com/fmfn/BayesianOptimization.

Palumbo, E., Monti, D., Rizzo, G., Troncy, R., & Baralis, E. (2020). Entity2rec: Property-specific knowledge graph embeddings for item recommendation. *Expert Systems with Applications*, Article 113235. http://dx.doi.org/10.1016/j.eswa.2020.113235.

Palumbo, E., Rizzo, G., Troncy, R., Baralis, E., Osella, M., & Ferro, E. (2018a). An empirical comparison of knowledge graph embeddings for item recommendation. In *Proceedings of the 1st workshop on deep learning for knowledge graphs and semantic technologies, DL4KGS@ESWC '18* (pp. 14–20).

Palumbo, E., Rizzo, G., Troncy, R., Baralis, E., Osella, M., & Ferro, E. (2018b). Translational models for item recommendation. In A. Gangemi, A. L. Gentile, A. G. Nuzzolese, S. Rudolph, M. Maleshkova, H. Paulheim, J. Z. Pan, & M. Alam (Eds.), *The semantic web: ESWC 2018 satellite events* (pp. 478–490). http://dx.doi.org/10.1007/978-3-319-98192-5_61.

Passant, A. (2010). Measuring semantic distance on linking data and using it for resources recommendations. In *AAAI spring symposium: linked data meets artificial intelligence: Vol. 77* (p. 123).

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *WWW '01*, *Proceedings of the 10th international conference on world wide web* (pp. 285–295). http://dx.doi.org/10.1145/371920.372071.

Sha, C., Wu, X., & Niu, J. (2016). A framework for recommending relevant and diverse items. In *Proceedings of the 25th international joint conference on artificial intelligence. IJCAI'16* (pp. 3868–3874).

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Proceedings of the 25th international conference on neural information processing systems. NIPS'12* (pp. 2951–2959).

Steck, H. (2013). Evaluation of recommendations: Rating-prediction and ranking. In *RecSys '13*, *Proceedings of the 7th ACM conference on recommender systems* (pp. 213–220). http://dx.doi.org/10.1145/2507157.2507160.

Steck, H. (2018). Calibrated recommendations. In *RecSys '18*, *Proceedings of the 12th ACM conference on recommender systems* (pp. 154–162). http://dx.doi.org/10.1145/3240323.3240372.

Trattner, C., & Jannach, D. (2020). Learning to recommend similar items from human judgments. *User Modeling and User-Adapted Interaction*, *30*(1), 1–49. http://dx.doi.org/10.1007/s11257-019-09245-4.

Vargas, S., & Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. In *RecSys '11*, *Proceedings of the 5th ACM conference on recommender systems* (pp. 109–116). http://dx.doi.org/10.1145/2043932.2043955.

Vig, J., Sen, S., & Riedl, J. (2012). The tag genome: Encoding community knowledge to support novel interaction. *ACM Transactions on Interactive Intelligent Systems*, *2*(3), http://dx.doi.org/10.1145/2362394.2362395.

Wang, Q., Mao, Z., Wang, B., & Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, *29*(12), 2724–2743. http://dx.doi.org/10.1109/TKDE.2017.2754499.

Wang, Q., Yu, J., & Deng, W. (2019). An adjustable re-ranking approach for improving the individual and aggregate diversities of product recommendations. *Electronic Commerce Research*, *19*(1), 59–79. http://dx.doi.org/10.1007/s10660-018-09325-4.

Wu, W., Chen, L., & Zhao, Y. (2018). Personalizing recommendation diversity based on user personality. *User Modeling and User-Adapted Interaction*, *28*(3), 237–276. http://dx.doi.org/10.1007/s11257-018-9205-x.

Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., & Achan, K. (2020). Product knowledge graph embedding for E-commerce. In *WSDM '20*, *Proceedings of the 13th international conference on web search and data mining* (pp. 672–680). http://dx.doi.org/10.1145/3336191.3371778.

Yao, Y., & Harper, F. M. (2018). Judging similarity: A user-centric study of related item recommendations. In *RecSys '18*, *Proceedings of the 12th ACM conference on recommender systems* (pp. 288–296). http://dx.doi.org/10.1145/3240323.3240351.

Yigit-Sert, S., Altingovde, I. S., Macdonald, C., Ounis, I., & Ulusoy, Ö. (2020). Supervised approaches for explicit search result diversification. *Information Processing & Management*, *57*(6), Article 102356. http://dx.doi.org/10.1016/j.ipm.2020.102356.

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system. *ACM Computing Surveys*, *52*(1), 1–38. http://dx.doi.org/10.1145/3285029.

Zhou, T., Kuscsik, Z., Liu, J. G., Medo, M., Wakeling, J. R., & Zhang, Y. C. (2010). Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences of the United States of America*, *107*(10), 4511–4515. http://dx.doi.org/10.1073/pnas.1000488107.

Ziegler, C. N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005). Improving recommendation lists through topic diversification. In *WWW '05*, *Proceedings of the 14th international conference on world wide web* (p. 22). http://dx.doi.org/10.1145/1060745.1060754.