Title	Intent-aware Item-based Collaborative Filtering for Personalised Diversification
Authors(s)	Wasilewski, Jacek; Hurley, Neil J.
Publication date	2017-08-31
Publication information	Proceedings of the Poster Track of the 11th ACM Conference on Recommender Systems (RecSys 2017)
Publisher	ACM
Link to online version	http://ceur-ws.org/Vol-1905/; https://recsys.acm.org/recsys17/
Item record/more information	http://hdl.handle.net/10197/9980
Publisher's statement	© ACM, 2017. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in UMAP '18 Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization (pages 81-89). http://doi.acm.org/10.1145/3209219.3209234
Publisher's version (DOI)	10.1145/3209219.3209234

Intent-aware Item-based Collaborative Filtering for Personalised Diversification

Jacek Wasilewski
Insight Centre for Data Analytics
University College Dublin
Dublin, Ireland
jacek.wasilewski@insight-centre.org

Neil Hurley
Insight Centre for Data Analytics
University College Dublin
Dublin, Ireland
neil.hurley@insight-centre.org

ABSTRACT

Diversity has been identified as one of the key dimensions of recommendation utility that should be considered besides the overall accuracy of the system. A common diversification approach is to rerank results produced by a baseline recommendation engine according to a diversification criterion. The intent-aware framework is one of the frameworks that has been proposed for recommendations diversification. It assumes existence of a set of aspects associated with items, which also represent user intentions, and the framework promotes diversity across the aspects to address user expectations more accurately.

In this paper we consider item-based collaborative filtering and suggest that the traditional view of item similarity is lacking a user perspective. We argue that user preferences towards different aspects should be reflected in recommendations produced by the system. We incorporate the intent-aware framework into the item-based recommendation algorithm by injecting personalised intent-aware covariance into the item similarity measure, and explore the impact of such change on the performance of the algorithm. Our experiments show that the proposed method improves both accuracy and diversity of recommendations, offering better accuracy/diversity tradeoff than existing solutions.

1 INTRODUCTION

The goal of recommender systems is to help to address the choice overload problem by filtering a large set of possible selections into a much smaller set of recommended items. User interactions, as explicit or implicit evidence of user needs, involve a great deal of uncertainty. Engaging and holding a user's interest is a complex matter. Simply identifying relevant items, without taking into account the user's context, can lead to accurate recommendations when measured directly against user's past preferences, but may not meet current need of the user. For example, considering a movies streaming application, without any context it is hard to predict if at the time of a visit the user would like to see a comedy or rather a thriller, given the user has expressed an interest in these in the past. Recommendation diversification addresses this issue, by widening the range of possible item types recommended to the user, hoping that something will attract the user's attention. From this viewpoint, diversity can be treated as a strategy to favour practical accuracy in matching true user needs in an uncertain environment.

Different diversification frameworks have been proposed in the literature to define and enhance diversity. One of them, the intent-aware diversification framework has been proposed to deal with the ambiguity of users' intents. It attempts to ensure that all relevant interests are represented in the recommendations. It is a probabilistic framework based on the aspect model, which assumes the existence of aspects through which users' intents can be expressed, and by which items can be described. Given an aspect model, methods such as xQuAD [16] rerank recommendations produced by a baseline algorithm such that a user's propensity towards different aspects is reflected in the final recommendations, and multiple aspects are covered to maximise the likelihood of matching the user's current need. While post-filtering diversification approaches are confirmed to be successful in diversity enhancement, their performance heavily depends on the quality of the baseline recommendations. Rerankers will not succeed in their task if the baseline recommendations are homogeneous in terms of aspects.

In this paper we suggest that instead of generating recommendations first and reranking them such that they cover more aspects and reflect better the user's taste, we propose to combine information about relevance, user intentions and item-aspect associations to produce richer recommendations in one step. In order to achieve that, we introduce personalised item similarity based on the intent-aware personalised covariance proposed in [26], which we use to drive the item-based nearest-neighbour collaborative filtering algorithm. Item neighbourhoods based on such personalised similarity happen to be much more richer and diverse when compared to the state-of-the-art methods, making the final recommendations of better utility as well. Recommendations produced by the proposed solution are not only more diverse over aspects, but also of a higher accuracy than the item-based algorithm, and they offer better accuracy/diversity tradeoffs than existing diversification methods.

The rest of the paper is structured as follows. In section 2 we discuss popular variations of the neighbourhood-based recommenders, how additional sources of information can be incorporated to improve accuracy, and how utilities other than accuracy can be addressed. Section 3 contains a brief description of the item-based collaborative filtering algorithm for ranking and points out limitations of commonly used similarity functions. In section 4 we first introduce the intent-aware framework, later describe how intent-aware personalised covariance can be used as a measure of item similarity, and finally, we propose the intent-aware version of item-based collaborative filtering. Description and analysis of our experiments can be found in section 5. We conclude with a summary in section 6.

2 BACKGROUND

The task of a recommender system is simple: helping users in browsing huge catalogues of items by serving items of relevance, based on past user-item interactions, such as ratings given by a user to an item. Historically, given such input, recommender systems were trying to predict ratings of unobserved interactions in order to find the most relevant items, while now the ranking task—returning a list of items ordered by relevance—seems to be more related to real recommendation scenarios [7].

Neighbourhood-based collaborative filtering methods (kNN) are simple, yet still powerful (and one of the most popular) solutions proposed to address the recommendation task. They rely on the similarity assumption that similar profiles (in terms of past interactions) make good candidates of items to be recommended to users. There are two types of kNN algorithms, user-based and item-based, which differ in the type of input data used by similarity functions, and type of neighbourhood used to form final recommendations. Here we are interested in the item-based kNN algorithm which utilises neighbourhoods oriented around items. The performance of neighbourhood-based methods depends heavily on the characteristics of the similarity function used in the process of neighbourhood creation [9]. Over the years, a number of variations have been proposed to improve accuracy and reduce prediction error, or promote other utilities of recommendations, such as diversity. Some of the popular choices for similarity functions [8, 9, 17] are Cosine Vector similarity between item vectors of ratings given to them by users, Pearson Correlation similarity which removes differences in means and variances of items, and Adjusted Cosine similarity which is a modification of the Pearson Correlation similarity where user-mean-centred ratings are used. While these different similarity functions offer different performance, depending on the application, they all focus on improving the accuracy of recommendations, solely using ratings information.

In [13] it is suggested that the emphasis that has been put on the user similarity (in case of the user-based kNN) may be overstated and additional factors should be taken into consideration. They proposed that trustworthiness of users should be an important consideration. This indeed led to improved performance on the rating prediction task.

While the item-based kNN method is understood to be more stable, efficient and justifiable, especially in cases where there are many more users than items, recommender systems using this approach will tend to recommend highly similar items to those previously appreciated by the users. This may lead to safer recommendations, however less surprising or broadening of user awareness of different types of items [9].

In [20], Smyth et al. propose to retrieve k diverse neighbours instead of retrieving the k most similar neighbours. Most similar item profiles in item neighbourhoods tend to be very similar to each other, therefore the recommendation space may be limited if left as is. Selecting a diverse set of most similar items leads to better utilisation of the item space and a more diverse final recommendation. With the same motivation, in [27] a neighbour diversification-based kNN algorithm has been proposed, extending the user-based kNN approach, where a diverse set of users is used instead of the most similar ones to the target user. In this case, a set of users is diverse if they have few items in common. Another user-based diversification algorithm has been proposed in [29] where users are associated with interests and a diverse set of users consist of users that have different interests.

A different approach has been proposed in [15] where using k furthest neighbours is suggested instead of forming neighbourhoods of k nearest neighbours. In this technique, the final recommendations are based on what is disliked by the least similar users to the target user. As a result, recommendations tend to be more diverse if compared to the original algorithm, with a tolerable loss in accuracy.

In the literature, two lines of research can be identified for diversity enhancement, these being post-filtering and novel solutions enhancing the utility of interest. Most of the approaches discussed above fall into the former group. State-of-the-art diversification methods, such as MMR [4], IA-Select [1], xQuAD [16] are examples of the post-filtering approach. Their advantage is that they can be applied on any candidate recommendation lists. These are examples of a greedy optimisation scheme that selects a subset from a candidate set by balancing accuracy of recommendations with other utilities—originally diversity but also extended to improve unexpectedness, serendipity and other types of novelty. In [29] it is argued that the performance of post-filtering optimisation techniques is limited by the quality of the input candidate items generated by the baseline recommendation system engine. If the candidate set is not diverse enough, the best subset selected by a post-filtering algorithm will not be diverse enough as well.

In [25], an extension of matrix factorisation techniques has been proposed to produce recommendations that are both relevant and diverse. This has been done through regularisation of the learning objective, similarly to regularisers that have been used in machine learning models to prevent overfitting. This method, however, had troubles in producing recommendations of better accuracy/diversity tradeoff when compared with the current state-of-the-art, like the MMR post-filtering algorithm.

3 ITEM-BASED COLLABORATIVE FILTERING

The recommendation task can be formulated as defining a scoring function $s : \mathcal{U} \times I \to \mathbb{R}$ for pairs of users $u \in \mathcal{U}$ and items $i \in I$ so that, for each user, a ranked list of items $\mathcal{R}_u \subset I$ (or simply $\mathcal{R}$ if the user context is clear) is defined by sorting items by decreasing score order, and keeping the top-N items. The scoring function of a recommender algorithm is based on previous interactions between users and items.

3.1 Item-based kNN

Given a similarity function $\text{sim}(i, j)$ which estimates how much two items are alike, we define the neighbourhood $N^k(j)$ of item j such that $|N^k(j)| \le k$ and items in the neighbourhood $N^k(j)$ are the most similar items to j according to the function $\text{sim}(i, j)$. Then, the item-based (IB) kNN recommender for the top-N [2, 7] can be defined according to the following formula:

$$s(u, i) = \sum_{j \in I_u} \mathbb{1}_{i \in N^k(j)} \cdot \text{sim}(i, j) \cdot r_{uj} \tag{1}$$

where indicator function $\mathbb{1}_{i \in N^k(j)}$ filters out items if they do not belong to the neighbourhood $N^k(j)$, r_{uj} denotes the rating that user u gave in the past to the item j which works as a relevance weighting, and $\mathcal{I}_u$ is the set of items that user u interacted with in the past. Top-N items with highest scores s form the final recommendation list.

Equation (1) shows that the utility of the recommendation generated by the item-based kNN algorithm depends strongly on the similarity, used both to weight items in the neighbourhoods, and also to select those neighbourhoods, which are made of top-k items according to the same similarity function $\text{sim}(i, j)$.

In our study we use the cosine similarity as a baseline function to represent the similarity between items. Using the rating feedback given by users, similarity between two items can be defined as:

$$\text{sim}(i, j) = \frac{\sum_{u \in \mathcal{U}} r_{ui} r_{uj}}{\sqrt{\sum_{u \in \mathcal{U}} r_{ui}^2} \sqrt{\sum_{u \in \mathcal{U}} r_{uj}^2}}$$

The above assigns a high similarity value if ratings for items i and j are aligned.

3.2 Analysis of item neighbourhoods

The quality of neighbourhoods impacts the quality of recommendations. We investigate the items forming item neighbourhoods based on the cosine similarity to judge their quality. We define neighbourhood's quality through the neighbourhood novelty, richness and diversity. We conduct this analysis on the MovieLens 1M dataset (the dataset's description can be found in section 5.1). We consider the neighbourhoods of $k = 10$ most similar movies to the target movie, according to the cosine similarity. Each movie in the dataset is enriched with genre information, which we lean our analysis on.

Interactions-based similarity functions build item neighbourhoods by aggregating items that are liked together. It is a common belief that neighbourhoods of alike items in terms of interactions also tend to represent the same genre, as people who liked e.g. drama movies highly likely also watched other drama movies. While this is logical, it introduces choice limitations as only certain item types— the most similar ones— are promoted to be recommended from. As this is not a desired situation, we would like to understand how many of the items in the neighbourhoods represent the same genre as the target item. We looked at the neighbourhoods of all items in the dataset and we found that on average 8 out of 10 movies in each of the neighbourhoods have a common genre with the target movie. This suggests that neighbourhoods are of low novelty.

Instead of simply counting how many items are representing the same item type, it is possible to quantify how consistent a neighbourhood is in terms of all genres, by measuring its *richness*. We define richness as a number of different genres found among movies in the neighbourhood (this is similar to S-recall metric defined in section 5.2 however we simply count genres). On average, neighbourhoods cover 5 genres out of 18 available in the dataset. As comparison, we looked at user profiles in the dataset and we measured the richness of movies that users have interacted with. The average user profile shows interactions with movies covering 15 genres all together. If we constrain the interactions to those that users have rated positively (at least with a rating of 4), and ignore single occurrences of genres, the average user has shown an interest

in 11 genres. We can see that interactions-based neighbourhoods do not trace these interests well. It tells us that to find the most suitable movies for a target movie we take a narrow perspective, limited in terms of genres.

We have found that items in the neighbourhoods are similar to the target item and cover only few genres. The quality of the neighbourhood is not only defined by how items are similar to the target item but also how items in the neighbourhood relate to each other. To assess that, we measure the intra-neighbourhood dissimilarity in terms of genres—genre-based distance between each two items in the neighbourhood is calculated and averaged over all pairs (this follows the EILD metric which is described in section 5.2). Similarly as with other properties, neighbourhoods present low genre diversity—approximately 0.37 (on 0-1 scale). This stresses even stronger that neighbourhoods are generally homogeneous, and promoting similar movies.

This analysis has looked at the different properties that item neighbourhoods can be assessed on. General conclusions are that items in neighbourhood are similar to target items, represent only a fraction of genres and are of low diversity. As neighbourhoods are driving the item-based collaborative filtering, these characteristics may propagate to final recommendations, leading to less appealing or engaging recommendations. The neighbourhoods (based on the cosine similarity in this case) not only aggregate items representing the same genre as the target, but also they assume that perception of the similarity is independent from the user. We believe that a personal perspective on items should be taken into account in similarity calculations. The individual perception of similarity and diversity has been studied by cognitivists. Studies, like [19], looked into this problem by considering how thematically related and unrelated concepts differ in perception. They demonstrated the necessity of an individualised model for perceived similarity and suggested that models of similarity should be tuned to individual variability. If we were to apply the above into the neighbourhood creation process, items should not only be selected based on the interactions-based similarity but also on user's taste profile.

4 INTENT-AWARE RECOMMENDATIONS

In this section we first introduce the intent-aware framework that typically has been used for diversification. We explain the intent-aware personalised covariance of two items upon which we build the intent-aware item-based kNN algorithm.

4.1 Intent-aware Framework

The intent-aware diversification framework, proposed in [16] and extended in [23], has been used in information retrieval to mitigate the uncertainty of user queries. If a search engine is queried with "jaguar", without any additional context, it is uncertain whether the results are expected to be related to the animal or the car manufacturer. In information retrieval, if multiple aspects are related to a search query (like the animal and the car manufacturer), without explicit information about the query's intent, the intent-aware diversification framework ensures a good spread of explicit aspects among the relevant items in the result list, hoping that at least one will satisfy user's needs.

In the recommender systems setting, Vargas et al. [22] introduced the notion of user intent as an analogue of query intent. User intents are described in terms of a probability distribution over a set of aspects based on the interests previously expressed through interactions with items.

The intent-aware framework assumes the existence of a predefined set of aspects (latent or explicit), $\mathcal{A}$, over which the aspect model [11] is formed. It is composed of two components: user's intents in terms of a probability $p(a|u)$, such that $\sum_{a\in\mathcal{A}} p(a|u) = 1$, which can be seen as the user's taste profile, and the probability $p(\text{rel}_i|a, u)$ which holds the information about relevance of item i if we know the user u is interested in aspect a. Combining the probabilities of the aspect model entails a recommendation relevance model:

$$p(\text{rel}_i|u) = \sum_{a\in\mathcal{A}} p(a|u)p(\text{rel}_i|a, u)$$

In the relevance model, $p(\text{rel}_i|u)$ models the binary relevance of an item i to a given user u. An item is classed as either relevant or not relevant and it is possible that many items may be simultaneously relevant to the user. We also assume that relevance is independent, given the aspect, so that:

$$p(\text{rel}_i \wedge \text{rel}_j|u) = \sum_{a\in\mathcal{A}} p(a|u)p(\text{rel}_i|a, u)p(\text{rel}_j|a, u)$$

4.2 Intent-aware Personalised Covariance

The portfolio diversification framework proposed by Markowitz [12] has been used in the financial context to maximise the expected return on investment, while minimising the risk as measured by the variance of the return. Diversification across negatively correlated assets allows the risk to be hedged, and more stable returns. This idea has been adapted to the context of information retrieval and recommender systems [24] where, similarly as in the financial domain, recommender systems seek to maximise a recommendation's return—user satisfaction with the recommended items—while the variance of the relevance (i.e. the risk) is minimised.

In order for the framework to work, accurate estimates of item relevance and covariance of relevance are required. These can be estimated using historical ratings or item-aspect associations, as suggested in [24], however these are global estimates, independent of the user's preferences. An intent-aware item relevance covariance has been proposed in [26] which incorporates items' relevances, aspect relationships between items, and user preferences towards aspects. Using the conditional independence of relevance, the user-dependent covariance can be derived from the aspect model as:

$$\text{cov}_{\text{IA}}(\text{rel}_i, \text{rel}_j, u) = \sum_a p(a|u)p(\text{rel}_i|a, u)p(\text{rel}_j|a, u)$$
$$- \sum_a p(a|u)p(\text{rel}_i|a, u)\sum_a p(a|u)p(\text{rel}_j|a, u)$$

for $i \neq j$, and

$$\text{cov}_{\text{IA}}(\text{rel}_i, \text{rel}_i, u) = \sum_a p(a|u)p(\text{rel}_i|a, u)(1 - \sum_a p(a|u)p(\text{rel}_i|a, u)).$$

As the personalised covariance is based on more than one source of information, the interpretation differs from the standard one. Positive value of $\text{cov}_{\text{IA}}(.)$ indicates that both items i and j are relevant on aspects important to the user, and items share these aspects.

Negative values of covariance occur when two items are still highly relevant to the aspects liked by the user, but items do not share these aspects. The value of 0 is observed when at least one item is not relevant to the aspect the user is interested in.

The above shows that if $\text{cov}_{\text{IA}}(.)$ was used to promote items, relevant items with aspects exactly like the target item's would be preferred, resulting in low novelty and diversity. In our application it makes more sense to promote items with a negative covariance, as these items still would be relevant, but representing totally different aspects and this is the analogue of risk minimisation in portfolio optimisation.

4.3 Intent-aware Item-based kNN

Similarly as in [13, 20], the intent-aware covariance can be combined with different components of the IB algorithm, namely the neighbourhood selection and the weighting. We explore these possibilities by using the intent-aware weighting to form neighbourhoods and/or weight the items returned from the neighbourhoods.

4.3.1 Intent-aware Weighting.

The simplest way to incorporate intent-aware relationships between items into the recommendation process is to replace the pure similarity function with a weighting based on the intent-aware covariance, $w_{\text{IA}}(i, j, u)$. We define the weighting as a product of intent-aware covariance and the similarity value:

$$w_{\text{IA}}(i, j, u) = (- \text{cov}_{\text{IA}}(\text{rel}_i, \text{rel}_j, u)) \cdot \text{sim}(i, j) \tag{2}$$

Negative covariance is used as we would like to promote items of different aspects. We chose a multiplicative combination of intent-aware covariance and similarity, over addition, arithmetic, harmonic or geometric mean, or pure intent-aware covariance as it performed best in our preliminary experiments.

Using the $w_{\text{IA}}(i, j, u)$, we reformulate Equation (1) into:

$$s(u, i) = \sum_{j\in\mathcal{I}_u} \mathbb{1}_{i\in N_{\text{IA}}^k(j)} w_{\text{IA}}(i, j, u) \cdot r_{uj} \tag{3}$$

where $N_{\text{IA}}^k(j)$ is the neighbourhood of j—a set of k items most similar to j according to $w_{\text{IA}}(i, j, u)$.

4.3.2 Intent-aware Bounding.

Intent-aware weighting requires calculation of $w_{\text{IA}}(i, j, u)$ for all item pairs in order to build the neighbourhoods. To reduce complexity, similarly to [20], a bounded version of $N_{\text{IA}}^k(j)$ neighbourhoods can be used. We build the bounded neighbourhood $B_{\text{IA}}^{bk}(j)$ of item j by selecting the bk (multiple b of k) most similar items to j from the original neighbourhood $N^{bk}(j)$, which are later reordered by $w_{\text{IA}}(i, j, u)$. The top k items form the bounded neighbourhood $B_{\text{IA}}^{bk}(j)$ which are selected instead of $N^k(j)$. The modified recommendation formula is then:

$$s(u, i) = \sum_{j\in\mathcal{I}_u} \mathbb{1}_{i\in B_{\text{IA}}^{bk}(j)} \text{sim}(i, j) \cdot r_{uj} \tag{4}$$

Note that the original similarity weighting is used to aggregate items coming from the neighbourhood, not the intent-aware one. The difference then, between the original IB and the above approach is that here we force the neighbourhood to return items that are similar in terms of ratings, relevant to the user, but representing different aspects.

Because we are no longer examining all of the pairs of items to form a neighbourhood, we may miss an item with a marginally higher intent-aware weight value then the best bk items retrieved from $N^{bk}(j)$. However the likelihood of this happening decreases with the item similarity $\text{sim}(i, j)$ so that for suitable values of b it becomes unlikely.

4.3.3 Combining Intent-aware Weighting and Bounding.
Both of the proposed schemes can be combined into one, such that neighbourhoods of relevant and different items are first found, which later are re-weighted to form final recommendations. Equation (5) shows both approaches used in combination:

$$s(u, i) = \sum_{j \in I_u} \mathbb{1}_{i \in B_{\text{IA}}^{bk}(j)} w_{\text{IA}}(i, j, u) \cdot r_{uj} \tag{5}$$

4.4 Aspect Model Estimation

Covariance $\text{cov}_{\text{IA}}(\text{rel}_i, \text{rel}_j, u)$ depends on the existence of an aspect model. Following [26], we build the aspect model using a source of good quality covariance estimation, in this case based on past recommendations which form scores $s(u, i)$.

To transform scores into probabilities, the Platt scaling [14] is applied. Given a threshold τ, we create class labels $\text{rel}(i, u) = 1$ if $r_{ui} \geq \tau$, and $\text{rel}(i, u) = 0$ otherwise. We generate a sample of scores $s(i|u)$ for a set of randomly chosen user-item pairs, and choose a class label $\text{rated}(i, u) = 1$ if (u, i) pair is in our training set, and $\text{rated}(i, u) = 0$ otherwise. The relevance function $g(.)$ is inferred using logistic regression as a combination of relevance of a (u, i) pair given that it has been rated by the user ($l(s)$), and the probability that a user-item pair is rated ($r(s)$), given score s:

$$g(s) = p(\text{rel}\,|\text{rated}, s)p(\text{rated}|s) + p(\text{rel}\,|\neg\text{rated}, s)p(\neg\text{rated}|s)$$
$$= l(s)r(s) + \text{relb}(1 - r(s))$$

where relb is a background (prior) relevance score for unrated user-item pairs, which we take $\text{relb} = 0$.

To estimate components of the aspect model we consider the combination of aspect coverage and learned relevance function, following [5, 21, 22]. Given the function $g(s)$ and items rated by users, I_u, we can estimate components of the aspect model as:

$$p(a|u) = \frac{|\{i \in I_u : a \in \mathcal{A}_i\}|}{\sum_{a' \in \mathcal{A}} |\{i \in I_u : a' \in \mathcal{A}_i\}|}$$

$$p(\text{rel}_i\,|a, u) = \frac{2^{\mathbb{1}_{a \in \mathcal{A}_i} \frac{g(s(u,i))}{g(s^*(u,a))}} - 1}{2}$$

where $s^*(u, a)$ is the max score that any item with aspect a obtained for the user u. Probabilities $p(a|u)$ are based purely on the items in the user's profile, and $p(\text{rel}_i\,|a, u)$ are based on item-aspect associations, the relevance function $g(s)$ and scores s.

5 EVALUATION

5.1 Datasets

To show the effectiveness of our approach, we perform our evaluation on three datasets: MovieLens 1M, MovieLens 20M [10] and Netflix [3].

The biggest MovieLens dataset, MovieLens 20M (ML-20M), consists of about 20M ratings on scale from 0.5 to 5, with a step-size of 0.5, from 138K users on 28K movies collected thought the

https://movielens.org application. Movies are enriched by 20 genres, however interactions for items without any genre information have been removed from the dataset. All users had rated at least 20 items. We also use the smaller MovieLens dataset, MovieLens 1M (ML-1M), which contains 1M ratings given by 6K users who joined the application in 2000. It holds 1M ratings on 3.7K movies, enriched by 18 genres. Ratings are made on a 5-star scale.

The full Netflix dataset consists of 100M ratings from 1 to 5, from 480K users on 18K movies. Using IMDb, 28 movie genres have been identified and associated with the movies in the dataset, such that 9K movies have at least one associated genre. Similarly as before, ratings for movies without genres have been removed.

The datasets have been split into 5 folds, where in each turn 1 fold becomes the test set, and the rest form the training set. Results of this 5-fold cross-validation are averaged over all runs. For the MovieLens 20M and Netflix datasets, evaluation is performed on a randomly sampled 10K users.

5.2 Metrics

In our experiments, to measure the effectiveness of the ranked recommendation list we use two common metrics: Precision and nDCG. In order to calculate these, we set the relevance threshold as rating 4, meaning we consider items with rating at least 4 to be relevant. Same threshold has been used in the relevance-aware measures of diversity.

A large number of metrics have been proposed to measure diversity. A good review can be found in [21]. For the investigation of the tradeoff of the relevance and diversity, we utilise a number of metrics: S-recall, DNG, α-nDCG, ERR-IA and EILD. Except for the S-recall, we use relevance-aware versions of metrics, following [22]. For EILD we report also the relevance-unaware version. Also, we apply a logarithmic discount to items in the ranked list, making items higher in the list more impactful. Below we briefly summarise each of the considered diversity metrics:

- Subtopic Recall (S-recall) [28] is a simple metric which measures how well the recommendations cover the aspect space:

$$\text{S-recall}(\mathcal{R}) = \frac{|\cup_{i \in \mathcal{R}} \mathcal{A}_i|}{|\mathcal{A}|}$$

- DNG, proposed in [18], measures how early new aspects are introduced to a ranked list:

$$\text{DNG}(\mathcal{R}) = \sum_{k=1}^{|\mathcal{R}|} \frac{1}{\log_2(k + 1)} \text{rel}(i_k|u)G(k)$$

where $G(k)$ is the number of new aspects at rank k at which a relevant item appears; i_k denotes the item at rank k.

- Intra-List Diversity (ILD) [22, 30, 31] measures the average pairwise distance of the items in a recommendation set:

$$\text{ILD}(\mathcal{R}) = \frac{1}{|\mathcal{R}|(|\mathcal{R}| - 1)} \sum_{i, j \in \mathcal{R}} \text{dist}(i, j)$$

where $\text{dist}(i, j)$ is a distance function based on item features. We use the Expected Intra-List Diversity (EILD) which is a rank and relevance-aware version of the ILD metric.

- Aspect-aware version of the nDCG metric has been proposed in [6], the α-nDCG metric. It can be defined as:

$$\alpha-\text{nDCG}(\mathcal{R}) = \frac{\alpha-\text{DCG}(\mathcal{R})}{\alpha-\text{IDCG}(\mathcal{R})}$$

in which

$$\alpha-\text{DCG}(\mathcal{R}) = \sum_{i \in \mathcal{R}} \frac{1}{\log_2(k_i + 1)} \sum_{a \in \mathcal{A}} \text{rel}(i|u, a)$$
$$\prod_{j \in \mathcal{R}: k_j < k_i} (1 - \alpha \, \text{rel}(j|u, a))$$

where k_i denotes the position of item i in the ranked list, and $\text{rel}(i|u, a) = \mathbb{1}_{a \in \mathcal{A}_i} \text{rel}(i|u)$. $\alpha-\text{IDCG}(\mathcal{R})$ denotes the highest possible value of $\alpha-\text{DCG}(\mathcal{R})$ representing the case when the recommendations is made of ideally diversified relevant items. α is a constant set to control the penalty for the redundancy of the recommended items - in our experiment we use $\alpha = 0.5$ for moderate penalty.

- In [1], Agrawal et al. proposed an intent-aware generalisation of some standard metrics to account for aspects, for instance ERR-IA [6], the intent-aware expected reciprocal rank, which is similar to α-nDCG but it takes personal preferences towards aspects into account, different rank discount and it is not normalised:

$$\text{ERR-IA}(\mathcal{R}) = \sum_{a \in \mathcal{A}} p(a|u) \sum_{i \in \mathcal{R}} \frac{1}{k_i} p(\text{rel}\,|i, u, a)$$
$$\prod_{j \in \mathcal{R}: k_j < k_i} (1 - p(\text{rel}\,|j, u, a))$$

5.3 Setup

In this paper we are interested to assess whether using intent-aware item similarity to drive the item-based collaborative filtering has advantages over the original algorithm and other diversification approaches. For the considered solutions, we generate ranked lists of top $N = 10$ recommendations. As we do not aim to find the best parameters for the dataset used in evaluation, we set the $k = 10$ for both methods for fair comparison. To distinguish different versions we refer to the original item-based algorithm as IB, IB version using intent-aware weighting as IA-IB$_w$, IB version using intent-aware bounding as IA-IB$_b$, and version following both as IA-IB$_{wb}$. Both IA-IB$_b$ and IA-IB$_{wb}$ depend on an additional parameter b. We explore its impact by setting it to $b = \{1, 2, 5, 10, 20\}$.

We compare the diversification performance of the proposed methods with the state-of-the-art methods: xQuAD [16] and MMR [4] reranking applied on the IB baseline, and neighbourhood diversification as in [20]—DivIB. For the reranking methods, we generate a set C of 50 candidate items for each user, then we iteratively construct the reranked list, $\mathcal{S}$, by greedily selecting at each iteration the item i that satisfies:

$$i^* = \underset{i \in C \setminus \mathcal{S}}{\arg\max}(1 - \lambda)s(u, i) + \lambda \, \text{div}(i, \mathcal{S}),$$

and updating $\mathcal{S} \leftarrow \mathcal{S} \cup \{i^*\}$. The two terms in this expression represent the item quality component and the item diversity component, and are mixed together using a tradeoff controlling parameter $0 \leq \lambda \leq 1$. Both components are first standardised before mixing.

Depending on the reranking method used, we consider $\text{div}(i, \mathcal{S})$ of one of the following forms:

$$\text{div}_{\text{xQuAD}}(i, \mathcal{S}) = \sum_{a \in \mathcal{A}} p(a|u)p(\text{rel}_i\,|u, a) \prod_{j \in \mathcal{S}} (1 - p(\text{rel}_j\,|u, a))$$
$$\text{div}_{\text{MMR}}(i, \mathcal{S}) = \min_{j \in \mathcal{S}} \text{dist}(i, j)$$

The $\text{dist}(i, j)$ function depicts a distance between two items, i and j, in the item features space. For the DivIB method, we perform greedy neighbourhood diversification in a similar fashion to the MMR algorithm. We construct the neighbourhood $N_{\text{div}}^{bk}(j)$ by iteratively selecting from the top bk candidates produced by the baseline neighbourhood, items with the highest weight:

$$q(i, j) = (1 - \lambda) \text{sim}(i, j) + \lambda \frac{\sum_{i \in N_{\text{div}}^{bk}(j)} \text{dist}(i, j)}{|N_{\text{div}}^{bk}(j)|}$$

until $|N_{\text{div}}^{bk}(j)| = k$. Additionally, weight $q(i, j)$ is used instead of the similarity function in Equation (1). For DivIB, the parameter b has been set to $b = 2$, as greater values resulted in huge performance drop.

5.4 Results and Analysis

5.4.1 Analysis of item neighbourhoods.

Similarly as in section 3.2, we performed an analysis of the intent-aware personalised neighbourhoods. We compare the neighbourhoods based on pure cosine-based similarity with neighbourhoods used in DivIB and IA-IB methods. As the item neighbourhoods in the IA-IB methods are user-dependent, unlike cosine-based neighbourhoods which are user-independent, we need to project the neighbourhoods for a fair comparison. For each item in our dataset, we select a set of $m = 100$ random users. For each item-user pair we assess the quality of the intent-aware neighbourhood, and we average the results over users to obtain an item quality score.

Analysis of cosine-based neighbourhoods showed a low neighbourhood novelty—8 items out of 10 sharing a genre with the target item. Neighbourhood diversification, DivIB, improves the novelty, however not significantly—after diversification, 7 items out of 10 share a genre. On the other hand, the intent-aware neighbourhood reduced the number of similar items (based on genres) to only 2, showing that neighbourhoods now are rather made of items that are different to the target items but as the later analysis shows, still relevant to the user.

Not only more items in the intent-aware neighbourhoods represent different aspect than the target items but also more aspects are covered. The richness of the neighbourhoods has risen significantly from 5 to almost 10 aspects appearing in the neighbourhoods. Although the neighbourhood diversification method has also improved aspects coverage, it does not perform as well as the intent-aware method—between 6 and 7 aspects can be seen in the results, but leading to a deterioration in accuracy.

Finally, the intra-neighbourhood diversity expresses how on average items in the neighbourhoods differ from each other. We showed that cosine-based neighbourhoods are characterised as of low diversity meaning items are really similar to each other in terms of genres. Neighbourhoods in DivIB have been shown an increase in diversity, from 0.37 to 0.45. However more notable improvements could be achieved with greater values of λ, the advantages

		Precision	nDCG	α-nDCG	ERR-IA	S-recall	DNG	EILD	EILD (rel)
ML-1M	IB	0.2425	0.2450	0.2885	0.1826	0.5292	1.6129	0.6399	0.1499
	+ xQuAD ($\lambda = 0.7$)	0.2278	0.2295	△0.3058	▲0.1964	▲0.6651	△1.7631	△0.7189	△0.1530
	+ MMR ($\lambda = 0.5$)	0.2296	0.2348	0.2827	0.1786	△0.6246	1.6306	▲0.7688	△0.1625
	DivIB ($\lambda = 0.1$)	0.2342	0.2361	△0.2918	0.1731	△0.5925	△1.6420	△0.7021	△0.1535
	IA-IB$_w$	△0.2561	△0.2570	▲0.3521	△0.1876	△0.6295	△1.9428	△0.6960	△0.1691
	IA-IB$_b$ ($b = 20$)	△0.2525	△0.2491	△0.3372	*0.1775*	△0.6468	△1.8632	△0.7065	△0.1675
	IA-IB$_{wb}$ ($b = 20$)	▲0.2584	▲0.2595	△0.3510	△0.1901	△0.6381	▲1.9557	△0.6977	▲0.1714
ML-20M	IB	0.1941	0.2230	0.2643	0.1343	0.5703	1.4626	0.6536	0.1219
	+ xQuAD ($\lambda = 1.0$)	0.1821	0.2073	△0.2857	▲0.1593	▲0.7164	△1.7227	△0.7005	0.1182
	+ MMR ($\lambda = 0.5$)	0.1899	0.2200	0.2614	0.1322	△0.6586	△1.4912	▲0.7786	△0.1355
	DivIB ($\lambda = 0.1$)	0.1875	0.2149	0.2537	0.1199	△0.6175	1.3704	△0.7209	△0.1257
	IA-IB$_b$ ($b = 20$)	△0.2186	△0.2435	△0.3102	△0.1360	△0.6797	△1.7468	△0.7194	△0.1460
	IA-IB$_{wb}$ ($b = 20$)	▲0.2233	▲0.2572	▲0.3289	△0.1520	△0.6743	▲1.9187	△0.7060	▲0.1489
Netflix	IB	0.2254	0.2195	0.2310	0.1390	0.3718	1.4485	0.6688	0.1458
	+ xQuAD ($\lambda = 0.7$)	0.2183	0.2092	△0.2520	▲0.1592	△0.4530	△1.6391	△0.7143	△0.1476
	+ MMR ($\lambda = 0.5$)	0.2179	0.2122	0.2246	0.1361	△0.4281	△1.4615	▲0.7921	△0.1616
	DivIB ($\lambda = 0.1$)	0.2111	0.2021	0.2162	0.1192	△0.4179	1.3250	△0.7557	△*0.1479*
	IA-IB$_b$ ($b = 20$)	△0.2466	△0.2369	△0.2906	△0.1441	▲0.4688	△1.8357	△0.7175	△0.1661
	IA-IB$_{wb}$ ($b = 20$)	▲0.2542	▲0.2482	▲0.3073	△0.1567	△0.4649	▲1.9799	△0.7070	▲0.1709

Table 1: Accuracy and diversity performance results obtained on three dataset. For each dataset and metric, improvements over the baseline are denoted with $^\triangle$, best results are underlined and denoted with $^\blacktriangle$. All differences with respect to the IB are statistically significant (Wilcoxon $p < 0.05$), except when in italics. Presented λ values represent settings with the best performance on *ERR-IA* (for xQuAD) and *EILD (rel)* (for MMR and DivIB), b parameter for IA-IB has been selected such that best α-nDCG is obtained. Parameters are reported in parentheses.

would be diminish by the loss in accuracy of the DivIB. The personalised neighbourhoods show a much higher performance on the intra-neighbourhood diversity measure—it doubles the DivIB's neighbourhoods average diversity by reaching an a diversity value of 0.9.

All of the above show superiority of the intent-aware neighbourhoods over simple cosine-based neighbourhoods or diversified ones. We expect that seeing these improved properties leads to improvements of final recommendations produced by the item-based algorithm.

5.4.2 Accuracy analysis.
Table 1 presents the results obtained of the proposed methods, together with the results of the state-of-the-art for item-based collaborative filtering, neighbourhood diversification and reranking approaches for diversity and personalised diversity. Results of our experiments on all analysed datasets show that IA-IB methods outperform the original IB algorithm in terms of precision and nDCG. We could expect that from the IA-IB methods, the weighted IA-IB (IA-IB$_w$) would be the best performing as it analyses utility of all item pairs to create the neighbourhoods, however the weighted-bounded IA-IB performs slight better. Due to the computational complexity of the weighted IA-IB we only run it on the ML-1M dataset.

It is commonly said that diversification by its nature results in deterioration in accuracy as a cost of introducing diversity. The results of the IA-IB methods show that by introducing diversity the
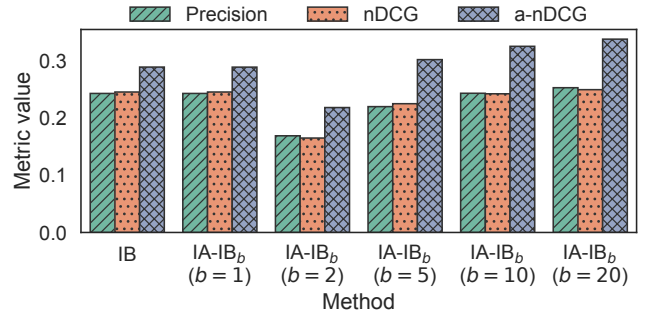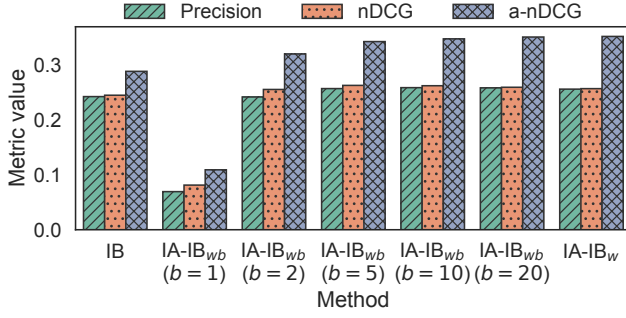

Figure 1: Comparison of the IA-IB$_b$ method and the original IB algorithm, on ML-1M dataset. Similar characteristics can be seen on the two other datasets.

accuracy got improved as well, suggesting that user's needs have been addressed more accurately.

5.4.3 Impact of bounding.
The weighted IA-IB method requires the computation of covariance of all item pairs in order to build the intent-aware neighbourhoods. This may be an impractical task, but also unnecessary as our definition of weighting depends both on the similarity and intent-aware covariance. In such case, if items are not similar to each other in terms of ratings, the likelihood of them being relevant according to the intent-aware covariance is low. This means that instead of

Figure 2: Comparison of the IA-IB$_w$**,** IA-IB$_{wb}$ **and the original IB algorithm, on ML-1M dataset. Similar characteristics can be seen on the two other datasets.**

analysing the covariance of all item pairs, it is enough to calculate the covariance of top bk candidate items. However it is important to understand the impact of such bounding.

Figure 1 compares the original IB algorithm with the bounded version of the IA-IB where the number of candidate items from which the final neighbourhoods are create, varies from 10 to 200 (bk, for $k = 10$ and $b = 1, 2, 5, 10, 20$). As the $b = 1$ is equal to the original IB, the performance on three analysed metrics is exactly the same. We can see that if more items are considered, the performance drops at first, at $b = 2$. This can be explained by the fact that top candidate items tend to be similar in terms of genres thus it is hard for the intent-aware covariance to find good matches. However, for $b = 10$ performance matches the IB, and for $b = 20$ it exceeds the performance of IB. It is worth to remind that here the method uses intent-aware covariance only in the neighbourhood creation process, and not in item weighting. This is done by the weighted-bounded IA-IB.

Figure 2 shows the performance of the IA-IB$_{wb}$ method for different b values, as before it is compared with the IB method, but also with the IA-IB$_w$. IA-IB$_w$ can be seen as the IA-IB$_{wb}$ method with $b = \infty$. Differently than before, for $b = 1$ the performance of the methods is not the same as the IB but it drops significantly. While neighbourhoods are made of the same items as the original IB, the intent-aware weighting diminishes their usefulness. When more candidate items are available, the performance on all metrics visibly is improved. As the b parameter increase, all considered metrics improve—α-nDCG most significantly. Also, we can notice that performance of IA-IB$_{wb}$ tends to IA-IB$_w$ when b increases, thus good results can be obtained without the computation of similarity for all item pairs.

5.4.4 Diversity analysis.
Finally, we compare the diversity performance of proposed methods with related state-of-the-art diversification techniques. The IA-IB methods show general improvements on all diversity metrics over the IB, and offer the best accuracy/diversity tradeoffs. It is worth noting that IA-IB methods do not directly optimise for any of the metrics, improvements come from the nature of the proposed method.

α-nDCG and ERR-IA are measures of diversity which favour recommendations in which many aspects are represented and not

over-represented. IA-IB methods perform particularly well on the α-nDCG, showing over 20% improvement over the baseline. While xQuAD method is known for offering good performance on that metric, it does not outperform IA-IB. However, it does perform the best on ERR-IA and S-recall (with an exception on the Netflix dataset). This is expected as the xQuAD explicitly optimises the ERR-IA objective, however the difference between xQuAD and IA-IB$_{wb}$ is relatively small. Similarly with the S-recall, xQuAD obtains the highest performance, however IA-IB produces better accuracy/S-recall tradeoff.

While S-recall measures the coverage of aspects in the recommendations, DNG measures how early new aspects are introduced in the list. All analysed methods show improvements in this matter when compared to the baseline however the best performance is obtained by the IA-IB$_{wb}$.

EILD and relevance-aware EILD are simpler measures of diversity in comparison to α-nDCG or ERR-IA. They measure dissimilarity between items in the recommendation, ignoring redundancy or personal preferences. While it is expected that the MMR offers best performance on the EILD, due to improvements in accuracy, the IA-IB methods obtain the best performance on the relevance-aware version of EILD. However IA-IB does not offer performance on EILD close to the one obtained by MMR, it is still a positive improvement compared to the baseline, comparable to the DivIB method.

The neighbourhood diversification method, DivIB, which we directly compare our solution against, does not show comparable results on metrics other than EILD. The diversification level of the DivIB, expressed through the λ parameter, is mild, however higher diversity level leads to significant loss in accuracy.

6 CONCLUSIONS

In this paper, we tackled the problem of intent-aware recommendations. We showed that item neighbourhoods used by the item-based collaborative filtering are homogeneous, of little novelty and diversity, which impacts the recommendations produced by the item-based algorithm. We proposed a novel approach, combining the item-based scheme with the intent-aware framework through the intent-aware personalised covariance which has been used to measure items similarity. Our experiments showed that neighbourhoods based on such similarity are richer, much more novel and diverse, impacting positively the recommendations. In comparison to the existing diversification approaches, the proposed methods offer superior accuracy/diversity tradeoff.

Many of the existing recommender systems utilise the item similarity component. As brining the user factor into the item similarity component has shown improvements in terms of diversity and accuracy of recommendations, the future work will explore new areas where the personalised item similarity could be applied, and how it is perceived by users in user studies.

REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. 2009. Diversifying Search Results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM '09)*. ACM, New York, NY, USA, 5–14. https://doi.org/10.1145/1498759.1498766

[2] F. Aiolli. 2013. Efficient Top-n Recommendation for Very Large Scale Binary Rated Datasets. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*. ACM, New York, NY, USA, 273–280. https://doi.org/10.1145/2507157.2507189

[3] J. Bennett and S. Lanning. 2007. The Netflix Prize. In *In KDD Cup and Workshop in conjunction with KDD*.

[4] J. Carbonell and J. Goldstein. 1998. The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)*. ACM, New York, NY, USA, 335–336. https://doi.org/10.1145/290941.291025

[5] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. 2009. Expected Reciprocal Rank for Graded Relevance. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM '09)*. ACM, New York, NY, USA, 621–630. https://doi.org/10.1145/1645953.1646033

[6] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, Z. Ashkan, S. Büttcher, and I. MacKinnon. 2008. Novelty and Diversity in Information Retrieval Evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*. ACM, New York, NY, USA, 659–666. https://doi.org/10.1145/1390334.1390446

[7] P. Cremonesi, Y. Koren, and R. Turrin. 2010. Performance of Recommender Algorithms on Top-n Recommendation Tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, USA, 39–46. https://doi.org/10.1145/1864708.1864721

[8] M. Deshpande and G. Karypis. 2004. Item-based top-N Recommendation Algorithms. *ACM Trans. Inf. Syst.* 22, 1 (Jan. 2004), 143–177. https://doi.org/10.1145/963770.963776

[9] C. Desrosiers and G. Karypis. 2011. *A Comprehensive Survey of Neighborhood-based Recommendation Methods*. Springer US, Boston, MA, 107–144. https://doi.org/10.1007/978-0-387-85820-3_4

[10] F. M. Harper and J. A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages. https://doi.org/10.1145/2827872

[11] T. Hofmann. 2004. Latent Semantic Models for Collaborative Filtering. *ACM Trans. Inf. Syst.* 22, 1 (Jan. 2004), 89–115. https://doi.org/10.1145/963770.963774

[12] H. Markowitz. 1952. Portfolio Selection. *The Journal of Finance* 7, 1 (1952), 77–91. http://www.jstor.org/stable/2975974

[13] J. O'Donovan and B. Smyth. 2005. Trust in Recommender Systems. In *Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI '05)*. ACM, New York, NY, USA, 167–174. https://doi.org/10.1145/1040830.1040870

[14] J. Platt. 2000. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers*.

[15] A. Said, B. J. Jain, B. Kille, and S. Albayrak. 2012. Increasing Diversity Through Furthest Neighbor-Based Recommendation. In *Proceedings of the WSDM'12 Workshop on Diversity in Document Retrieval (DDR'12)*.

[16] R. L. T. Santos, C. Macdonald, and I. Ounis. 2012. On the role of novelty for search result diversification. *Information Retrieval* 15, 5 (01 Oct 2012), 478–502. https://doi.org/10.1007/s10791-011-9180-x

[17] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*. ACM, New York, NY, USA, 285–295. https://doi.org/10.1145/371920.372071

[18] Y. Shi, X. Zhao, J. Wang, M. Larson, and A. Hanjalic. 2012. Adaptive Diversification of Recommendation Results via Latent Factor Portfolio. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '12)*. ACM, New York, NY, USA, 175–184. https://doi.org/10.1145/2348283.2348310

[19] S. Simmons and Zachary Estes. 2008. Individual differences in the perception of similarity and difference. *Cognition* 108, 3 (21 Aug 2008), 781–795. https://doi.org/10.1016/j.cognition.2008.07.003

[20] B. Smyth and P. McClave. 2001. Similarity vs. Diversity. In *Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development (ICCBR '01)*. Springer-Verlag, London, UK, UK, 347–361. http://dl.acm.org/citation.cfm?id=646268.758890

[21] S. Vargas. 2015. *Novelty and Diversity Evaluation and Enhancement in Recommender Systems*. Ph.D. Dissertation. Universidad Autónoma de Madrid.

[22] S. Vargas, P. Castells, and D. Vallet. 2011. Intent-oriented Diversity in Recommender Systems. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '11)*. ACM, New York, NY, USA, 1211–1212. https://doi.org/10.1145/2009916.2010124

[23] S. Vargas, P. Castells, and D. Vallet. 2012. Explicit Relevance Models in Intent-oriented Information Retrieval Diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '12)*. ACM, New York, NY, USA, 75–84. https://doi.org/10.1145/2348283.2348297

[24] J. Wang. 2009. Mean-Variance analysis: A new document ranking theory in information retrieval. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5478 LNCS (2009), 4–16. https://doi.org/10.1007/978-3-642-00958-7_4

[25] J. Wasilewski and N. Hurley. 2016. Incorporating Diversity in a Learning to Rank Recommender System. In *Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2016, Key Largo, Florida, May 16-18, 2016*. 572–578. http://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS16/paper/view/12944

[26] J. Wasilewski and N. Hurley. 2017. Personalised Diversification Using Intent-Aware Portfolio. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP '17)*. ACM, New York, NY, USA, 71–76. https://doi.org/10.1145/3099023.3099067

[27] C. Yang, C. C. Ai, and R. F. Li. 2013. Neighbor Diversification-Based Collaborative Filtering for Improving Recommendation Lists. In *2013 IEEE 10th International Conference on High Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing*. 1658–1664. https://doi.org/10.1109/HPCC.and.EUC.2013.234

[28] C. X. Zhai, W. W. Cohen, and J. Lafferty. 2003. Beyond Independent Relevance: Methods and Evaluation Metrics for Subtopic Retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval (SIGIR '03)*. ACM, New York, NY, USA, 10–17. https://doi.org/10.1145/860435.860440

[29] F. Zhang. 2009. Improving recommendation lists through neighbor diversification. In *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, Vol. 3. 222–225. https://doi.org/10.1109/ICICISYS.2009.5358201

[30] M. Zhang and N. Hurley. 2008. Avoiding Monotony: Improving the Diversity of Recommendation Lists. In *Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys '08)*. ACM, New York, NY, USA, 123–130. https://doi.org/10.1145/1454008.1454030

[31] C. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. 2005. Improving Recommendation Lists Through Topic Diversification. In *Proceedings of the 14th International Conference on World Wide Web (WWW '05)*. ACM, New York, NY, USA, 22–32. https://doi.org/10.1145/1060745.1060754