Improving the Diversity of Top-N Recommendation via Determinantal Point Process

Laming Chen
Hulu LLC.
Beijing, China, 100120
laming.chen@hulu.com

Guoxin Zhang
Hulu LLC.
Beijing, China, 100120
guoxin.zhang@hulu.com

Hanning Zhou
Hulu LLC.
Beijing, China, 100120
eric.zhou@hulu.com

ABSTRACT

Recommender systems take the key responsibility to help users discover items that they might be interested in. Many recommendation algorithms are built upon similarity measures, which usually result in low intra-list diversity. The deficiency in capturing the whole range of user interest often leads to poor satisfaction. To solve this problem, increasing attention has been paid on improving the diversity of recommendation results in recent years.

In this paper, we propose a novel method to improve the diversity of top-N recommendation results based on the determinantal point process (DPP), which is an elegant model for characterizing the repulsion phenomenon. We propose an acceleration algorithm to greatly speed up the process of the result inference, making our algorithm practical for large-scale scenarios. We also incorporate a tunable parameter into the DPP model which allows the users to smoothly control the level of diversity. More diversity metrics are introduced to better evaluate diversification algorithms. We have evaluated our algorithm on several public datasets, and compared it thoroughly with other reference algorithms. Results show that our proposed algorithm provides a much better accuracy-diversity trade-off with comparable efficiency.

CCS CONCEPTS

• **Information systems** → **Information retrieval diversity**; **Recommender systems**; *Evaluation of retrieval results*;

KEYWORDS

Recommender system, diversity, determinantal point process, top-N recommendation, metrics

ACM Reference format:
Laming Chen, Guoxin Zhang, and Hanning Zhou. 2017. Improving the Diversity of Top-N Recommendation via Determinantal Point Process. In *Proceedings of 5th Large Scale Recommendation System Workshop, Como, Italy, August 2017 (LSRS 2017),* 8 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The increasing importance of discovering relevant information from explosively growing data has served as a driving force for the development of recommender systems technology [25]. With user's historical interactions with items, recommender systems either predict the rating value for an item, or recommend the top-N items. By recommending carefully selected items to users, relevant items are brought to the attention of users, thus helping to reduce overloaded information and users can have much better experiences.

Most research works [12, 16, 26] on recommender systems focus on accuracy metrics such as MAE and RMSE, precision and recall, which measure how well a system can predict an exact rating value for a specific item, or how likely the user will interact with the recommended results. However, as [23] suggests, the most accurate recommendations are sometimes not those that are most useful to users. Only focusing on accuracy metrics often leads to similar items, because they are designed to judge the accuracy of each individual item regardless of the entire recommendation list. This increases the risk that the user might not like any of these items, or get bored by the repeated recommendation of similar items. To improve users' satisfaction with the recommended results, some other factors are proposed [13], such as coverage, novelty, serendipity and diversity. Among these metrics, improving the diversity of recommendation results has gained much attention in recent years [2, 14, 32]. Recommending more diverse items gives the users more exploration opportunities to discover something novel and serendipitous, and also makes it easier for the service to discover potential interests of its users. However, although increasing diversity might be preferred as users often prefer diverse results, it usually comes at the expense of accuracy. It is still a challenge to find a better trade-off between accuracy and diversity.

The determinantal point process (DPP) was introduced in [21] with name "fermion process" to give the distributions of fermion systems in thermal equilibrium, and it precisely describes the repulsion and diversity phenomenon. Besides its successful applications in quantum physics and random matrix theory, it has also been applied recently to machine learning tasks such as document summarization [18] and image search [17].

In this paper, we propose a novel algorithm to improve the diversity of top-N recommendation results based on the DPP model [19]. Given a set of candidate items with relevance scores and pairwise similarities, our algorithm produces a ranking balancing accuracy and diversity. Experiments on various datasets show that our method can effectively generate recommendation results with a much better trade-off between accuracy and diversity compared to previous methods.

The main contributions of this paper can be summarized as follows:

- We propose a DPP based method for improving the diversity of top-N recommendation results. Unlike previous methods

LSRS 2017, August 2017, Como, Italy

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

which directly optimize the weighted accuracy and pairwise diversity metrics, our algorithm combines the accuracy and overall diversity into a unified model, based on which the diversified recommendations are inferred. To the best of our knowledge, this is the first time that the DPP model is utilized to solve the top-N recommendation diversification problem. We also introduce a tunable parameter allowing the users to control the level of diversity, smoothly varying from completely relevant results to completely diverse results. Our method provides a much better trade-off between accuracy and diversity compared to previous methods, as shown in our experiments on various datasets.

- We propose an efficient algorithm to approximately find the mode of DPP (the subset with highest probability to appear). Finding the mode of DPP is essential for applying DPP to practical tasks, but obtaining the exact solution is infeasible due to its NP-hardness [5]. The computational complexity of existing polynomial approximate solutions such as the greedy MAP approximation [19] is still too high for practical usage. We propose an acceleration technique to greatly speed up the approximation, and its efficiency is comparable to state-of-the-art online diversification methods.
- We introduce more diversity metrics from more perspectives to thoroughly compare and evaluate different diversification methods.

The rest of this paper is organized as follows. In Section 2, related works about top-N recommendation diversification algorithms and applications of the DPP model in recommendation tasks are discussed. In Section 3, we introduce some background of the DPP model. We present and discuss our DPP based top-N recommendation diversification algorithm in Section 4. We give the experimental results of the proposed algorithm in Section 5, which reveals superior performance of our algorithm in terms of efficiency and the accuracy-diversity trade-off.

1.1 Notation

Sets are represented by uppercase letters such as Z, and $\#Z$ denotes the number of elements in Z. Vectors and matrices are represented by bold lowercase letters and bold uppercase letters, respectively. $\langle \mathbf{x}, \mathbf{y} \rangle$ is the inner product of two vectors $\mathbf{x}$ and $\mathbf{y}$. $(\cdot)^\top$ denotes the transpose of the argument vector or matrix. $\det(\mathbf{L})$ and $\mathrm{rank}(\mathbf{L})$ are the determinant and the rank of matrix $\mathbf{L}$, respectively. $\mathbb{R}_{\geq 0}$ represents the set of non-negative real numbers.

2 RELATED WORK

Some research works have been devoted to improving the diversity of recommendation results. [4] proposed the maximal marginal relevance (MMR) method. In each iteration of MMR, for each item that is not selected, a score is calculated and the item which maximizes this score is added to the recommendation list, where the score is a linear combination of the relevance of the item and the *minimal* dissimilarity between this item and each already selected item. [28] and [3] formed the results by a strategy similar to MMR, with the difference that the score is a combination of the relevance of the item and the *average* dissimilarity between this item and each already selected item. In [28], the score is a product of these two terms, while in [3], the score is a linear combination of them. [32] linearly combined the reciprocals of these two terms, and selected the item which minimizes the score. In [30] and [14], the authors recast the optimization as a binary quadratic programming problem. Besides the aforementioned greedy algorithm, they also proposed a general strategy for this problem through relaxation and quantization. All these research works come down to simple combinations of the accuracy and diversity metrics. However, they use only pairwise dissimilarities to characterize the overall diversity property of the recommendation list, which may not capture some complex similarity relationships among items (e.g., the characteristics of one item can be described as a simple linear combination of another two).

The problem of increasing diversity has also been addressed by other approaches. [31] encouraged diversity in the framework of absorbing Markov chain random walks. However, this method needs to invert a large matrix in the first iteration, resulting in high computational complexity even for a short recommendation list. [27] proposed a latent factor portfolio model to capture each user's interest range and preference uncertainty by utilizing the variance of the learned user latent factors, resulting in an adaptive recommendation diversification approach. But this method needs to be combined with the latent factor models [16], and this additional dependency makes it unsuitable for direct integration into other existing recommender systems. Same disadvantage also applies to [6], where a DPP eigenmixture model was proposed to recommend diverse items. [29] integrated the concept of diversity into the traditional matrix factorization model and constructed a set-oriented collaborative filtering algorithm. However, this method incorporates the trade-off between accuracy and diversity into the latent factor model, which does not support the dynamic trade-off adjustment without retraining the model. In [2] and [20], the items were clustered in the training phase. Then the recommendation list was built by assigning weight to each cluster and selecting items from each cluster. However, this method is not appropriate for scenarios where items do not possess the clustering property. Moreover, the weight reassigning procedure may lead to very irrelevant recommendations.

Recently, the DPP model has been applied to a recommendation task — recommending complementary products to the ones in the shopping basket. [9] proposed to use the expectation-maximization algorithm to learn the full kernel matrix of DPP. In [22], a fixed-point Picard iteration was proposed to learn the full kernel matrix at a remarkable faster speed compared to previous approaches. [7, 8] learnt a low-rank factorization of the kernel matrix, which leads to better recommendation performance.

3 BACKGROUND OF DPP

DPP is an elegant probabilistic model of global, negative correlations [19]. Characterized by a kernel matrix that defines a global measure of similarity among items, DPP assigns higher probabilities to sets of items that are diverse. As a result, DPP is ideal for describing the diversity of the results.

Formally, a DPP $\mathcal{P}$ on a discrete set of items $Z = \{1, 2, \ldots, M\}$ is a probability measure on the set of all subsets of Z, i.e., for every subset $Y \subseteq Z$, $\mathcal{P}(Y)$ characterizes the likelyhood of observing Y.

There exists a matrix $\mathbf{K}$ such that for a random subset Y drawn according to $\mathcal{P}$ and for every $A \subseteq Z$,

$$\text{Prob}(A \subseteq Y) = \det(\mathbf{K}_A) \tag{1}$$

where $\mathbf{K}$ is a real, symmetric $M \times M$ matrix indexed by the elements of Z, $\mathbf{K}_A \doteq [\mathbf{K}_{ij}]_{i,j \in A}$ denotes the sub-matrix of $\mathbf{K}$ indexed by A, and $\det(\mathbf{K}_\emptyset) = 1$ by convention. When $\mathcal{P}$ gives nonzero probability to the empty set, DPP can also be defined through a real, symmetric matrix $\mathbf{L}$ indexed by Z:

$$\mathcal{P}(Y) \propto \det(\mathbf{L}_Y).$$

After calculating the normalization constant of $\mathcal{P}$, we have

$$\mathcal{P}(Y) = \frac{\det(\mathbf{L}_Y)}{\det(\mathbf{L} + \mathbf{I})} \tag{2}$$

where $\mathbf{I}$ is an identity matrix. Both $\mathbf{K}$ and $\mathbf{L}$ contain all the information needed to characterize DPP. Since (2) directly models the probabilities of observing each subset of Z while (1) gives marginal probabilities, we use formulation (2) in our paper, and refer to $\mathbf{L}$ as the kernel matrix.

For scenarios where the size of the desired set is known up front, [17] proposed k-DPP which is a conditional DPP that models only sets of cardinality k. For a k-DPP $\mathcal{P}^k$ with kernel matrix $\mathbf{L}$ and a subset $Y \subseteq Z$ with $\#Y = k$, we still have $\mathcal{P}^k(Y) \propto \det(\mathbf{L}_Y)$, but the normalization constant is different:

$$\mathcal{P}^k(Y) = \frac{\det(\mathbf{L}_Y)}{e_k(\lambda_1, \lambda_2, \ldots, \lambda_M)} \tag{3}$$

where $e_k(\lambda_1, \lambda_2, \ldots, \lambda_M)$ is the k-th elementary symmetric polynomial on $\mathbf{L}$'s eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_M$.

4 DIVERSIFICATION METHOD

In this section, we describe our DPP based method for improving the diversity of top-N recommendation results. Let Z denote the full item set with M items. For a user u with profile item set $P_u \subset Z$, a top-N recommendation system returns $R_u \subset Z$ satisfying $R_u \cap P_u = \emptyset$ and $\#R_u = N$. Besides common accuracy metrics which measure how relevant R_u is to the user with profile P_u, we should also consider diversity metrics within R_u.

4.1 Kernel Matrix of DPP

To adopt the DPP model in the top-N recommendation task, we should construct the kernel matrix first. As revealed in [19], the kernel $\mathbf{L}$ can be written as a Gram matrix, $\mathbf{L} = \mathbf{B}^\top \mathbf{B}$, where the columns of $\mathbf{B}$ are vectors representing items in the set Z. Denoting the columns of $\mathbf{B}$ as $\mathbf{B}_i$ for $i \in Z$, we have

$$\mathcal{P}(Y) \propto \det(\mathbf{L}_Y) = (\text{Vol}(\{\mathbf{B}_i\}_{i \in Y}))^2 \tag{4}$$

where the right-hand side is the squared $\#Y$-dimensional volume of the parallelepiped spanned by the columns of $\mathbf{B}$ indexed by Y.

Equation (4) offers some insights into the DPP model. First, fixing all other factors, items with larger magnitude $\|\mathbf{B}_i\|_2$ are more likely to appear, because they increase the spanned volumes of sets containing them. Second, diverse item sets are more probable because their corresponding vectors $\{\mathbf{B}_i\}$ are more orthogonal, and hence span higher volumes. Therefore, we can write each column vector $\mathbf{B}_i$ as the product of a scalar $r_i \in \mathbb{R}_{\geq 0}$ and a normalized vector $\mathbf{f}_i \in \mathbb{R}^D$, $\|\mathbf{f}_i\|_2 = 1$. The entries of kernel $\mathbf{L}$ can be written as

$$\mathbf{L}_{ij} = \langle \mathbf{B}_i, \mathbf{B}_j \rangle = \langle r_i \mathbf{f}_i, r_j \mathbf{f}_j \rangle = r_i r_j \langle \mathbf{f}_i, \mathbf{f}_j \rangle.$$

We can think of r_i as measuring the relevance of item i to user u, and $\langle \mathbf{f}_i, \mathbf{f}_j \rangle$ as a measure of similarity between item i and item j. Therefore, as long as the relevance score vector $\mathbf{r} = [r_1, r_2, \ldots, r_M]^\top$ and the similarity matrix $\mathbf{S}$ are available, we can write the kernel matrix as

$$\mathbf{L} = \text{Diag}(\mathbf{r}) \cdot \mathbf{S} \cdot \text{Diag}(\mathbf{r}) \tag{5}$$

where $\text{Diag}(\mathbf{r})$ is a diagonal matrix whose diagonal entries are those of $\mathbf{r}$, and the (i, j)-th element of $\mathbf{S}$ is the similarity between item i and item j. Note that we do not need explicit item representations $\mathbf{B}$ to construct the kernel matrix. As a result,

$$\mathcal{P}(\mathbf{Y}) \propto \det(\mathbf{L}_Y) = \prod_{i \in Y} \mathbf{r}_i^2 \cdot \det(\mathbf{S}_Y). \tag{6}$$

Equation (6) clearly shows that the DPP model incorporates both relevance and diversity.

4.2 Diversification Algorithm

To get the diversified top-N recommendations, we should find the mode of DPP, i.e., solving

$$R_u = \arg\max_{Y \subseteq Z \setminus P_u} \mathcal{P}(Y) \propto \det(\mathbf{L}_Y) \quad \text{s.t.} \quad \#Y = N. \tag{7}$$

Finding the exact solution to (7) is NP-hard [5]. However, approximate solutions to (7) can be obtained by several algorithms, among which the greedy MAP approximation [19] was previously considered as the fastest one. Initialized as $R_u = \emptyset$, in each iteration an item j is added to R_u, where j is obtained by solving

$$j = \arg\max_{i \in (Z \setminus P_u) \setminus R_u} \det(\mathbf{L}_{R_u \cup \{i\}}). \tag{8}$$

In general, the computational complexity of calculating the determinant of an $k \times k$ matrix is $O(k^3)$. Therefore, the computational complexity of solving (8) is

$$O((\#R_u)^3 (M - \#P_u - \#R_u)).$$

Although greedy MAP approximation is considered as a fast algorithm, its computational complexity is still too high compared to methods [3, 4] with computational complexity

$$O(\#R_u (M - \#P_u - \#R_u))$$

per iteration. To speed up this algorithm, we utilize the Cholesky decomposition to reduce the computational complexity, which will be covered in detail in the rest of this subsection.

Recall (8) where we need to calculate determinant $\det(\mathbf{L}_{R_u \cup \{i\}})$ for each $i \in (Z \setminus P_u) \setminus R_u$. According to (5) and the fact that $\mathbf{S}$ is positive semidefinite, kernel $\mathbf{L}$ and all of its principal minors are also positive semidefinite. Assume $\mathbf{L}_{R_u}$ is positive definite which means the items in the already selected item set R_u are linearly independent, and suppose the Cholesky decomposition of $\mathbf{L}_{R_u}$ is available as

$$\mathbf{L}_{R_u} = \mathbf{V}\mathbf{V}^\top \tag{9}$$

where $\mathbf{V}$ is a lower triangular matrix. Then the keys of the accelerated algorithm are

- How to solve (8) efficiently if decomposition (9) is available;
- How to get the Cholesky decomposition of $\mathbf{L}_{R_u \cup \{j\}}$.

We show how to solve (8) efficiently. For any $i \in (Z \setminus P_u) \setminus R_u$, $\mathbf{L}_{R_u \cup \{i\}}$ can be written as

$$\mathbf{L}_{R_u \cup \{i\}} = \begin{bmatrix} \mathbf{L}_{R_u} & \mathbf{L}_{R_u,i} \\ \mathbf{L}_{i,R_u} & \mathbf{L}_{ii} \end{bmatrix}$$

where $\mathbf{L}_{R_u,i}$ is a column vector indexed by R_u in rows and i in column. According to (9), the Cholesky decomposition of $\mathbf{L}_{R_u \cup \{i\}}$ is

$$\begin{bmatrix} \mathbf{L}_{R_u} & \mathbf{L}_{R_u,i} \\ \mathbf{L}_{i,R_u} & \mathbf{L}_{ii} \end{bmatrix} = \begin{bmatrix} \mathbf{V} & \mathbf{0} \\ \mathbf{c}_i & d_i \end{bmatrix} \begin{bmatrix} \mathbf{V}^\top & \mathbf{c}_i^\top \\ \mathbf{0} & d_i \end{bmatrix}$$

where row vector $\mathbf{c}_i$ and $d_i \geq 0$ satisfies

$$\mathbf{V}\mathbf{c}_i^\top = \mathbf{L}_{R_u,i}, \tag{10}$$

$$d_i^2 = \mathbf{L}_{ii} - \|\mathbf{c}_i\|_2^2. \tag{11}$$

In addition, since $\mathbf{V}$ is lower triangular, it can be derived that

$$\det(\mathbf{L}_{R_u \cup \{i\}}) = \det(\mathbf{V}\mathbf{V}^\top) \cdot d_i^2 = \det(\mathbf{L}_{R_u}) \cdot d_i^2. \tag{12}$$

Therefore, optimization (8) is equivalent to solving

$$j = \arg\max_{i \in (Z \setminus P_u) \setminus R_u} d_i. \tag{13}$$

Getting $\mathbf{c}_i$ and d_i by (10) and (11) involves solving a linear equation with a lower triangular matrix, whose computational complexity is $O((\#R_u)^2)$. This is greatly reduced compared with the original $O((\#R_u)^3)$ for calculating the determinant.

Once j is found by (13), the Cholesky decomposition of $\mathbf{L}_{R_u \cup \{j\}}$ is

$$\mathbf{L}_{R_u \cup \{j\}} = \begin{bmatrix} \mathbf{V} & \mathbf{0} \\ \mathbf{c}_j & d_j \end{bmatrix} \begin{bmatrix} \mathbf{V}^\top & \mathbf{c}_j^\top \\ \mathbf{0} & d_j \end{bmatrix}. \tag{14}$$

This completes the keys of our algorithm.

In fact, we can further reduce the computational complexity by updating $\mathbf{c}_i$ and d_i recursively. Define $\tilde{\mathbf{c}}_i$ and $\tilde{d}_i$ as the associated vector and scalar to item i after item j is added to R_u. According to (10) and (14), for any $i \in (Z \setminus P_u) \setminus (R_u \cup \{j\})$, we have

$$\begin{bmatrix} \mathbf{V} & \mathbf{0} \\ \mathbf{c}_j & d_j \end{bmatrix} \tilde{\mathbf{c}}_i^\top = \mathbf{L}_{R_u \cup \{j\},i} = \begin{bmatrix} \mathbf{L}_{R_u,i} \\ \mathbf{L}_{ji} \end{bmatrix}. \tag{15}$$

Combining (15) with (10), we have

$$\tilde{\mathbf{c}}_i = \begin{bmatrix} \mathbf{c}_i & \left(\mathbf{L}_{ji} - \langle \mathbf{c}_j, \mathbf{c}_i \rangle\right) d_j^{-1} \end{bmatrix}.$$

Define

$$e_i = \left(\mathbf{L}_{ji} - \langle \mathbf{c}_j, \mathbf{c}_i \rangle\right) d_j^{-1}. \tag{16}$$

Then

$$\tilde{\mathbf{c}}_i = [\mathbf{c}_i \; e_i], \tag{17}$$

and (11) implies

$$\tilde{d}_i^2 = \mathbf{L}_{ii} - \|\tilde{\mathbf{c}}_i\|_2^2 = \mathbf{L}_{ii} - \|\mathbf{c}_i\|_2^2 - e_i^2 = d_i^2 - e_i^2. \tag{18}$$

Recursively updating $\mathbf{c}_i$ and d_i by (16), (17), and (18) involves calculating the inner product of two vectors, whose computational complexity is further reduced to $O(\#R_u)$. Note that we need additional space complexity $O(MN)$ for $\mathbf{c}_i$ and d_i to achieve the time complexity.

In summary, the computational complexity of solving (13) with (16), (17), and (18) is

$$O(\#R_u(M - \#P_u - \#R_u))$$

which is the same as methods [3, 4].

4.3 Discussion on Numerical Stability

As introduced in the previous subsection, updating $\mathbf{c}_i$ and d_i involves calculating e_i by (16), where d_j^{-1} is involved. If d_j is approximately zero, our algorithm encounters the numerical instability issue. According to (12), d_j satisfies

$$d_j^2 = \frac{\det(\mathbf{L}_{R_u \cup \{j\}})}{\det(\mathbf{L}_{R_u})}. \tag{19}$$

Let $d^{\#R_u} = d_j$ where j satisfies (13). We give some results about the sequence $\{d^k\}$, as in the following theorem.

Theorem 4.1. *Suppose $N \leq \operatorname{rank}(\mathbf{L}_{Z \setminus P_u})$. Then*
(1) $d^k > 0$ holds for $k = 0, 1, \ldots, N-1$;
(2) The sequence $\{d^k\}$ is non-increasing.

The proof of Theorem 4.1 is given in Appendix A.1.

The assumption in Theorem 4.1 is equivalent to assuming that there are at least N linearly independent items in $Z \setminus P_u$, which is reasonable in practice because generally $Z \setminus P_u$ is a large candidate set and N is a small number. Under this assumption, Theorem 4.1 shows that

$$d^0 \geq d^1 \geq \cdots \geq d^{N-1} > 0.$$

Therefore, to avoid the numerical instability issue, we pre-define a small tolerance $\varepsilon > 0$ so that the iteration of our proposed algorithm stops when

$$d_j \leq \varepsilon. \tag{20}$$

According to (19), this means that the best selected item j can be almost linearly expressed in terms of the items from R_u. In this case, the proposed algorithm should be terminated.

4.4 Trade-off Parameter

A nice feature of methods [2, 4, 30] is that they involve a tunable parameter which allows users to adjust the trade-off between accuracy and diversity. However, according to (6), the original DPP model does not offer such a mechanism. To solve this problem, we define a mapping $m : \mathbb{R}_{\geq 0} \to [1, +\infty]$ by

$$m(\mathbf{r}_i) = \alpha^{\mathbf{r}_i}, \quad \alpha \geq 1 \tag{21}$$

and use this mapping to construct the kernel instead of (5):

$$\mathbf{L} = \operatorname{Diag}(m(\mathbf{r})) \cdot \mathbf{S} \cdot \operatorname{Diag}(m(\mathbf{r})). \tag{22}$$

When $\alpha = 1$, kernel $\mathbf{L}$ equals $\mathbf{S}$ and the DPP model only captures similarities among items. The following theorem shows that with sufficiently large α, the set of the most relevant items has the highest probability to be observed. Therefore, we can smoothly adjust between accuracy and diversity by varying parameter α.

Theorem 4.2. *Define top-N relevance set as a set of N items with the largest relevance scores. Let r_{MIN} denote the smallest relevance of items in a top-N relevance set. Let $r_{\max}$ denote the largest value in $\mathbf{r}$ that is strictly smaller than r_{MIN}. Suppose Y is a top-N relevance set and satisfies $\det(\mathbf{S}_Y) > 0$. When*

$$\alpha > (\det(\mathbf{S}_Y))^{-2(r_{\text{MIN}} - r_{\max})}, \tag{23}$$

any set X of N items that is not a top-N relevance set satisfies

$$\mathcal{P}(X) < \mathcal{P}(Y). \tag{24}$$

Algorithm 1 Div-DPP

Input: Profile P_u, $\alpha_u \geq 1$, similarity $\mathbf{S}$, relevance $\mathbf{r}_u$, $\varepsilon > 0$
Initialization: $R_u = \emptyset$, $\mathbf{r}^u = \alpha_u^{\mathbf{r}_u}$, $\mathbf{c}_i = []$ and $d_i = \mathbf{r}_i^u$ for $i \in Z$
while $\#R_u < N$ and $\#R_u + \#P_u < M$ **do**
  $j = \arg\max_{i \in (Z \setminus P_u) \setminus R_u} d_i$
  if $d_j \leq \varepsilon$ **then**
    break
  end if
  $R_u = R_u \cup \{j\}$
  for $i \in (Z \setminus P_u) \setminus R_u$ **do**
    $e_i = \left(\mathbf{r}_j^u \mathbf{r}_i^u \mathbf{S}_{ji} - \langle \mathbf{c}_j, \mathbf{c}_i \rangle\right) d_j^{-1}$
    $\mathbf{c}_i = [\mathbf{c}_i \; e_i]$
    $d_i^2 = d_i^2 - e_i^2$
  end for
end while
Return: R_u

The proof of Theorem 4.2 is given in Appendix A.2.

Combining (13), (16), (17), (18), (20), and (21), our proposed algorithm Div-DPP is summarized in Algorithm 1.

5 EXPERIMENTAL RESULTS

In this section, we describe the experimental results of the proposed algorithm in terms of efficiency and the accuracy-diversity trade-off. The reference algorithms include:

- Random: This is the simplest strategy for increasing the diversity of a set of N items. This algorithm randomly selects N items from a larger set comprised of the $N+b$ most relevant items to the user, with $0 \leq b \leq M - N$. When $b = 0$, this is equivalent to selecting the most relevant items, which is named as Top.
- MMR [4]: Initialize $R_u = \emptyset$. In each iteration, the item

$$j = \arg\max_{i \in (Z \setminus P_u) \setminus R_u} \theta \mathbf{r}_i + (1 - \theta) \min_{k \in R_u} (1 - \mathbf{S}_{ki}) \tag{25}$$

  is added to R_u, until $\#R_u = N$. $\theta \in [0, 1]$ is a trade-off parameter to balance the relevance and the minimal dissimilarity. As θ increases, MMR returns more relevent results.
- Greedy [3]: Initialize $R_u = \emptyset$. In each iteration, the item

$$j = \arg\max_{i \in (Z \setminus P_u) \setminus R_u} \theta \mathbf{r}_i + (1 - \theta) \operatorname{mean}_{k \in R_u} (1 - \mathbf{S}_{ki}) \tag{26}$$

  is added to R_u, until $\#R_u = N$. (26) uses the average dissimilarity while (25) uses the minimal dissimilarity.

5.1 Efficiency

In this experiment, the efficiency of different algorithms is compared. Algorithms are implemented in Python and running on a laptop with 2.2GHz Intel Core i7 and 16GB of RAM. Synthetic data are used in this experiment without loss of any generality. The total number of items is fixed to $M = 1000$. This is a reasonable number because in large scale recommender systems, a few hundreds of related items are usually pre-selected for each user [14]. The entries of the relevance score vector $\mathbf{r} \in \mathbb{R}^M$ are drawn i.i.d. from the standard uniform distribution. For the similarity matrix $\mathbf{S}$, the dimension of the normalized vectors for items is set to $D = 100$. We first generate a random matrix $\mathbf{F} \in \mathbb{R}^{D \times M}$ whose entries are drawn i.i.d. from the standard uniform distribution, then normalize each column of $\mathbf{F}$ to have unit ℓ_2 norm, and finally let $\mathbf{S} = \mathbf{F}^\top \mathbf{F}$. For DPP, the kernel $\mathbf{L}$ is formed by (5). The number of recommended items N increases from 0 to 50 with step size 5, and the running time of different algorithms is compared after averaging over 10 independent trials.

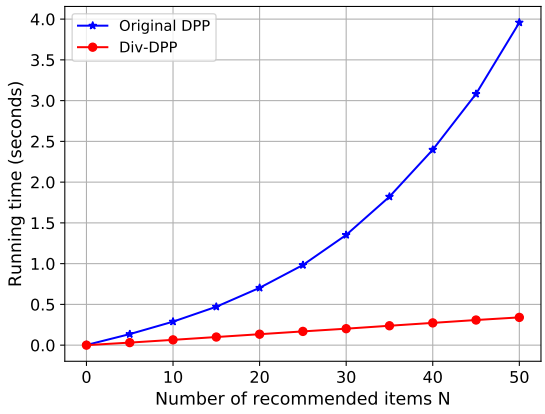

Figure 1: Running time comparison of original DPP and proposed Div-DPP.

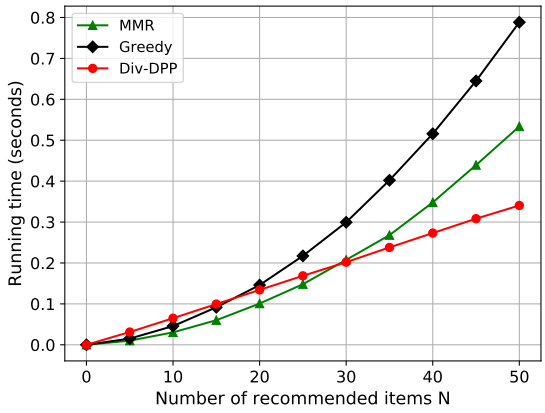

Figure 2: Running time comparison of MMR, Greedy, and Div-DPP.

Figure 1 compares the efficiency of our proposed algorithm with the original one which solves (8) directly in each iteration, where the determinant is calculated using `numpy.linalg.det`. As can be seen, our proposed algorithm generates the same results with significantly reduced running time. It needs to be noted that our proposed acceleration technique is applicable not only to the top-N recommendation diversification problem, but also to a much wider range of problems where the DPP model and the greedy MAP approximation are applied. Figure 2 compares the efficiency of our proposed algorithm with MMR and Greedy. It shows that Div-DPP

has competitive performance in terms of efficiency compared with fast diversification algorithms.

5.2 Accuracy vs. Diversity

In this experiment, we compare our proposed algorithm with the reference algorithms on the trade-off between accuracy and diversity.

5.2.1 Experimental Setup. We evaluate the algorithms on the following public datasets, and subsample the data as in [2]:

- MovieLens [11]: It contains 1,000,209 ratings of 3,706 movies made by 6,040 users. We subsample this dataset so that each user rates at least 20 movies and each movie is rated by at least 10 users. The reduced dataset contains 6,034 users and 3,260 movies with 998,428 ratings.
- Last.FM[1]: It contains 92,834 listening counts of 17,632 artists listened by 1,892 users. We use the same subsample strategy and the reduced dataset contains 1,545 users and 1,280 artists with 56,423 listening counts.
- Jester [10]: It contains 1,761,439 continuous ratings of 140 jokes from 59,132 users. Again, we subsample this dataset so that each user rates 20 to 80 jokes and each joke is rated by at least 10 users. The reduced dataset contains 19,547 users and 140 jokes with 745,360 ratings.

For each dataset, we split it into a training set and a test set by randomly selecting an interacted item for each user to be part of the test set, and using the rest of the data for training. We adopt an item-based recommendation algorithm SUGGEST described in [15] on the training set to learn an item-item similarity matrix. For each user, the interacted items in the training set form the profile set, and the most K similar items of each item in the profile set form the candidate set. For MovieLens, we choose $K = 30$, while $K = 20$ for the other two. The relevance score of any item in the candidate set is calculated as its aggregated similarity to all items in the profile set as in [14]. With the similarity matrix (same for all users) and relevance scores for each user, our algorithm is evaluated and compared with the reference algorithms in top-N recommendation, where $N = 20$ for MovieLens and $N = 10$ for the other two.

5.2.2 Evaluation Metric. The accuracy is measured by the recall, i.e., the percentage of items in the test set that are also in the top-N recommended results returned for all users. Suppose the item in the test set for user u is t_u and U is the set of all users. Then

$$\text{recall} = \frac{\sum_{u \in U} \mathbf{1}_{t_u \in R_u}}{\#U}$$

where $\mathbf{1}_P$ is the indicator function that has value 1 when P is true and 0 elsewhere.

The diversity is usually measured by the average dissimilarity within R_u [14, 28],

$$\text{average dissimilarity} = \operatorname*{mean}_{i,j \in R_u, i \neq j} (1 - \mathbf{S}_{ij}).$$

[1]Last.fm website, http://www.lastfm.com

In this experiment, we also propose to use the minimum and median of dissimilarity to measure the diversity,

$$\text{minimum dissimilarity} = \min_{i,j \in R_u, i \neq j} (1 - \mathbf{S}_{ij}),$$
$$\text{median dissimilarity} = \operatorname*{median}_{i,j \in R_u, i \neq j} (1 - \mathbf{S}_{ij}).$$

The minimum dissimilarity characterizes the least dissimilar pair of items in the recommended results. The median dissimilarity is not skewed by extremely large or small values, and it may give a better idea of a "typical" value than the average dissimilarity.

5.2.3 Comparison Results. By varying parameters b of Random, θ of MMR and Greedy, and α of Div-DPP, we are able to compare their trade-off between the accuracy and diversity metrics. For MovieLens and Last.FM, b varies from 0 to 2, while for Jester, b varies from 0 to 1. When $b = 0$, Random is identical to Top. For ease of comparison, the range of θ and α are chosen such that different algorithms have the same range of Recall. For a fixed parameter choice, the accuracy and diversity metrics are averaged over all users. The experiment is repeated for 10 independent trials, each trial using a different random partitioning of data for training and test, to ensure that our results are statistically accurate.

The comparison results are shown in Figure 3. The rows correspond to the results of MovieLens 1M, Last.FM, and Jester, while the columns use average, minimum, and median dissimilarity as the diversity metric. According to the results, compared with the reference algorithms, our proposed algorithm Div-DPP is able to return more diverse items with the same accuracy, especially in terms of the minimum dissimilarity.

6 CONCLUSIONS AND FUTURE WORKS

In this paper, we present a novel method to improve the diversity of top-N recommendation list based on DPP. We propose an accelerated algorithm to approximately find the mode of DPP, which gives the diversified recommendation results. This algorithm generates the same results with significantly reduced running time compared to the original one, and this acceleration technique is also applicable to other DPP based methods. A tunable parameter is incorporated into the DPP model so that we can smoothly adjust between accuracy and diversity for each user. Besides the average dissimilarity, we propose two new metrics of diversity: the minimum dissimilarity and the median dissimilarity. Experimental results on public datasets show that compared with the reference algorithms, our proposed algorithm is able to recommend more diverse items with the same accuracy and comparable efficiency.

A drawback of our algorithm is that in theory it may recommend less than N items due to the stopping criteria (20), although this is very unlikely to happen if the required item size is evidently smaller than the total number of candidate items, and we have not observed such a phenomenon in all our experiments. This can also be easily avoided by appending other algorithms until it returns enough items, but it worth further studying a nontrivial unified method. Another research direction is to incorporate the temporal diversity into our diversification method. Finally, we can also consider to extend our method to improve the coverage (also known as the aggregate diversity [1, 24]).

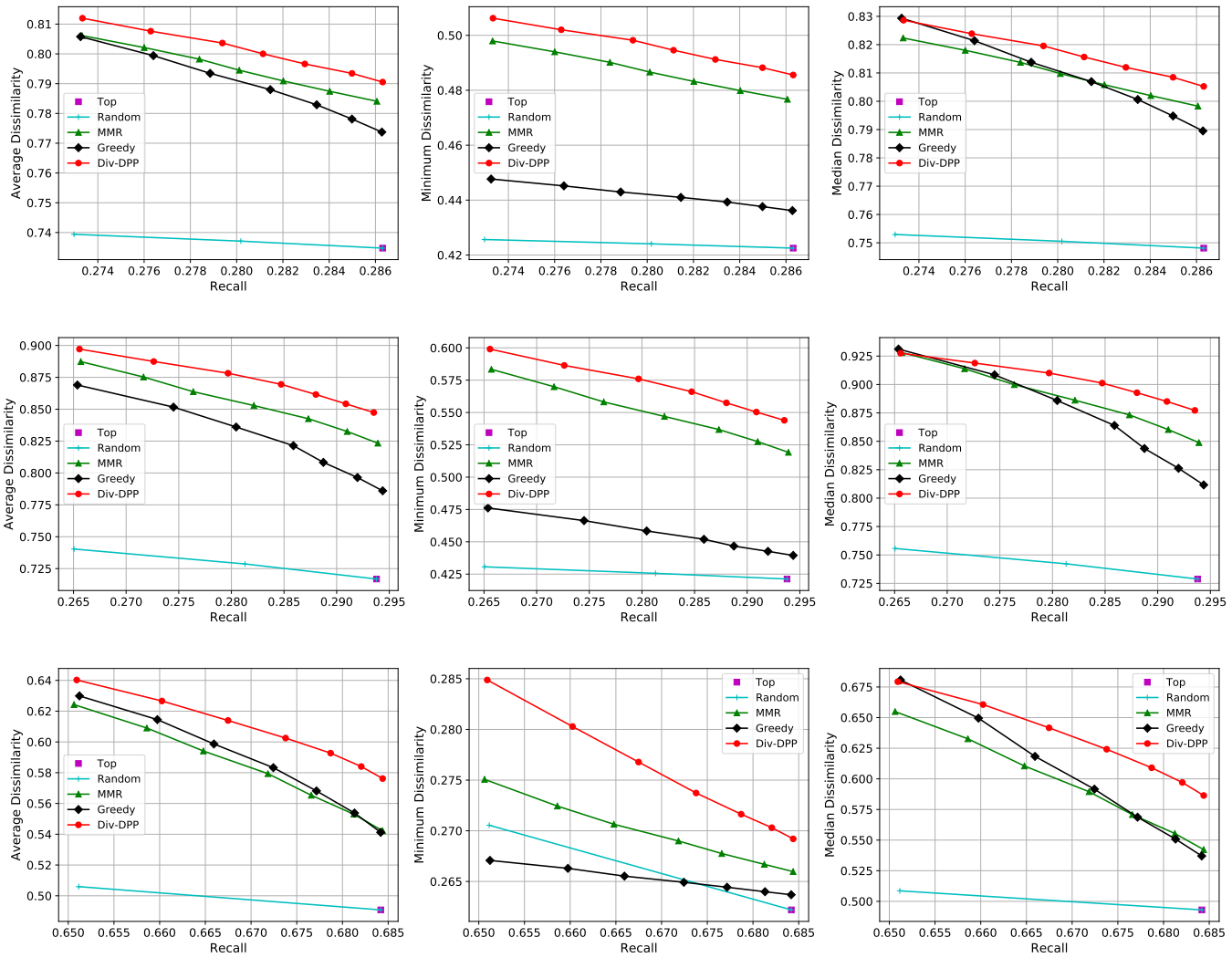

Figure 3: Comparison of the trade-off between accuracy and diversity metrics. (Top row) on MovieLens 1M. (Middle row) on Last.FM. (Bottom row) on Jester. (Left column) average dissimilarity versus recall. (Middle column) minimum dissimilarity versus recall. (Right column) median dissimilarity versus recall.

A PROOFS

A.1 Proof of Theorem 4.1

Firstly, we prove part (1). Let $V_k \subset Z \setminus P_u$ be the set of items that have been chosen by greedy MAP approximation at the end of the k-th step. Let $W_k \subset Z \setminus P_u$ be a set of k items such that $\det(\mathbf{L}_{W_k})$ is maximum. According to Theorem 3.3 in [5], we have

$$\det(\mathbf{L}_{V_k}) \geq \left(\frac{1}{k!}\right)^2 \cdot \det(\mathbf{L}_{W_k}). \tag{27}$$

Since $N \leq \text{rank}(\mathbf{L}_{Z \setminus P_u})$, W_N satisfies $\det(\mathbf{L}_{W_N}) > 0$, and therefore

$$\det(\mathbf{L}_{V_N}) > 0. \tag{28}$$

According to (12), we have

$$\det(\mathbf{L}_{V_N}) = \prod_{k=0}^{N-1} \left(d^k\right)^2. \tag{29}$$

As a result, $d^k > 0$ holds for $k = 0, 1, \ldots, N-1$.

Now we turn to part (2). Suppose in two adjacent iterations, the first iteration selects item j and the second one selects item $\tilde{j}$. According to (18) and (13),

$$\tilde{d}_{\tilde{j}}^2 = d_{\tilde{j}}^2 - e_{\tilde{j}}^2 \leq d_{\tilde{j}}^2 \leq d_{\tilde{j}}^2. \tag{30}$$

As a result, the sequence $\{d^k\}$ is non-increasing.

A.2 Proof of Theorem 4.2

Since Y is a top-N relevance set while X is not, according to the definitions of r_{MIN} and $r_{\max}$, we have

$$\sum_{i \in Y} \mathbf{r}_i - \sum_{i \in X} \mathbf{r}_i \geq r_{\text{MIN}} - r_{\max} > 0. \tag{31}$$

Together with (22) and (6), we have

$$\frac{\mathcal{P}(\mathbf{Y})}{\mathcal{P}(\mathbf{X})} = \frac{\prod_{i \in Y} \alpha^{2\mathbf{r}_i} \det(\mathbf{S}_Y)}{\prod_{i \in X} \alpha^{2\mathbf{r}_i} \det(\mathbf{S}_X)} \geq \alpha^{2(r_{\text{MIN}} - r_{\max})} \frac{\det(\mathbf{S}_Y)}{\det(\mathbf{S}_X)}. \tag{32}$$

Since $\mathbf{S}$ is a similarity matrix, $\det(\mathbf{S}_X) \leq 1$ holds. Due to (23), it can be derived that

$$\frac{\mathcal{P}(\mathbf{Y})}{\mathcal{P}(\mathbf{X})} \geq \alpha^{2(r_{\text{MIN}} - r_{\max})} \det(\mathbf{S}_Y) > 1, \tag{33}$$

which completes the proof.

REFERENCES

[1] Gediminas Adomavicius and YoungOk Kwon. 2012. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering* 24, 5 (2012), 896–911.

[2] Tevfik Aytekin and Mahmut Özge Karakaya. 2014. Clustering-based diversity improvement in top-N recommendation. *Journal of Intelligent Information Systems* 42, 1 (2014), 1–18.

[3] Keith Bradley and Barry Smyth. 2001. Improving recommendation diversity. In *Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science, Maynooth, Ireland*. Citeseer, 85–94.

[4] Jaime Carbonell and Jade 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*. ACM, 335–336.

[5] Ali Çivril and Malik Magdon-Ismail. 2009. On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science* 410, 47-49 (2009), 4801–4811.

[6] James Foulds and Dilan Görür. 2013. Diverse Personalization with Determinantal Point Process Eigenmixtures. (2013).

[7] Mike Gartrell, Ulrich Paquet, and Noam Koenigstein. 2016. Bayesian low-rank determinantal point processes. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 349–356.

[8] Mike Gartrell, Ulrich Paquet, and Noam Koenigstein. 2016. Low-rank factorization of determinantal point processes for recommendation. *arXiv preprint arXiv:1602.05436* (2016).

[9] Jennifer A Gillenwater, Alex Kulesza, Emily Fox, and Ben Taskar. 2014. Expectation-maximization for learning determinantal point processes. In *Advances in Neural Information Processing Systems*. 3149–3157.

[10] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. 2001. Eigentaste: A constant time collaborative filtering algorithm. *information retrieval* 4, 2 (2001), 133–151.

[11] F Maxwell Harper and Joseph A Konstan. 2016. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5, 4 (2016), 19.

[12] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 230–237.

[13] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 5–53.

[14] Neil Hurley and Mi Zhang. 2011. Novelty and diversity in top-n recommendation–analysis and evaluation. *ACM Transactions on Internet Technology (TOIT)* 10, 4 (2011), 14.

[15] George Karypis. 2001. Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the tenth international conference on Information and knowledge management*. ACM, 247–254.

[16] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009).

[17] Alex Kulesza and Ben Taskar. 2011. k-DPPs: Fixed-size determinantal point processes. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 1193–1200.

[18] Alex Kulesza and Ben Taskar. 2011. Learning determinantal point processes. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 419–427.

[19] Alex Kulesza, Ben Taskar, et al. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning* 5, 2–3 (2012), 123–286.

[20] Sang-Chul Lee, Sang-Wook Kim, Sunju Park, and Dong-Kyu Chae. 2017. A Single-Step Approach to Recommendation Diversification. In *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 809–810.

[21] Odile Macchi. 1975. The coincidence approach to stochastic point processes. *Advances in Applied Probability* 7, 01 (1975), 83–122.

[22] Zelda Mariet and Suvrit Sra. 2015. Fixed-point algorithms for learning determinantal point processes. In *ICML*. 2389–2397.

[23] Sean M McNee, John Riedl, and Joseph A Konstan. 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems*. ACM, 1097–1101.

[24] Katja Niemann and Martin Wolpers. 2013. A new collaborative filtering approach for increasing the aggregate diversity of recommender systems. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 955–963.

[25] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. *Introduction to recommender systems handbook*. Springer.

[26] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. ACM, 285–295.

[27] Yue Shi, Xiaoxue Zhao, Jun Wang, Martha Larson, and Alan 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*. ACM, 175–184.

[28] Barry Smyth and Paul McClave. 2001. Similarity vs. diversity. In *International Conference on Case-Based Reasoning*. Springer, 347–361.

[29] Ruilong Su, Li'Ang Yin, Kailong Chen, and Yong Yu. 2013. Set-oriented personalized ranking for diversified top-n recommendation. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 415–418.

[30] Mi Zhang and Neil Hurley. 2008. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems*. ACM, 123–130.

[31] Xiaojin Zhu, Andrew B Goldberg, Jurgen Van Gael, and David Andrzejewski. 2007. Improving Diversity in Ranking using Absorbing Random Walks.. In *HLT-NAACL*. 97–104.

[32] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*. ACM, 22–32.