

Trust-Based Collaborative Filtering: Tackling the Cold Start Problem Using Regular Equivalence

Tomislav Duricic

Know-Center GmbH & Graz University of Technology
Graz, Austria
tduricic@know-center.at

Dominik Kowald

Know-Center GmbH
Graz, Austria
dkowald@know-center.at

Emanuel Lacic

Know-Center GmbH
Graz, Austria
elacic@know-center.at

Elisabeth Lex

Know-Center GmbH & Graz University of Technology
Graz, Austria
elisabeth.lex@tugraz.at

ABSTRACT

User-based Collaborative Filtering (CF) is one of the most popular approaches to create recommender systems. This approach is based on finding the most relevant k users from whose rating history we can extract items to recommend. CF, however, suffers from data sparsity and the cold-start problem since users often rate only a small fraction of available items. One solution is to incorporate additional information into the recommendation process such as explicit trust scores that are assigned by users to others or implicit trust relationships that result from social connections between users. Such relationships typically form a very sparse trust network, which can be utilized to generate recommendations for users based on people they trust. In our work, we explore the use of regular equivalence applied to a trust network to generate a similarity matrix that is used to select the k -nearest neighbors for recommending items. We evaluate our approach on Epinions and we find that we can outperform related methods for tackling cold-start users in terms of recommendation accuracy.

KEYWORDS

Trust; Recommender Systems; Collaborative Filtering; Cold-start; Network Science; Regular Equivalence; Katz similarity

1 INTRODUCTION

Ever since their introduction, user-based Collaborative Filtering (CF) approaches have been one of the most widely adopted and studied algorithms in the recommender systems literature [21]. CF is based on the intuition that those users, who have shown similar item rating behavior in the past, will likely give similar ratings to items in the future. Typically, CF comprises of three steps: first, we retrieve the k -nearest neighbors to the target user for whom the recommendations are generated. Second, we employ the ratings from these k neighbors to determine items, which were rated highly

by them but have not yet been rated by the target user. Third, these items are weighted or ranked by applying an appropriate algorithm.

In practice, each user's ratings are stored in a rating vector. These rating vectors are then used to calculate the correlation between the target user's vector and rating vectors of the rest of the users. The higher the correlation between the rating vectors of two users, the higher their similarity. This can be assessed, e.g., via the Pearson's correlation coefficient, Cosine similarity, Jaccard index or Mean Squared Difference (MSD) [16, 20]. However, such an approach to neighbor selection suffers from a cold-start user problem. This term refers to novel users which have rated a small number of items or have not yet rated any items at all [14, 22]. This means that we cannot use their rating vectors for finding similar users based on the pairwise vector correlation measure.

Apart from popularity-based or location-based approaches [11, 13, 19], trust-based CF methods have been suggested to mitigate cold-start user problems. Their basis are trust statements expressed on platforms such as, e.g., Epinions [17]. Trust statements can either be expressed explicitly by, for example, assigning trust scores or implicitly by engaging in social connections with trusted users. Based on such trust statements, trust networks can be created with the aim to generate recommendations for users based on people they trust [15]. Since trust networks are often also sparse, a particular property of trust, namely transitivity [2], can be exploited to propagate trust in the network. In this way, new connections are established between users, who are not directly connected, but are connected via intermediary users. Previous work in this respect proposed to perform a modified breadth first search in the trust network to compute a prediction. For example, TidalTrust [4] aggregates and weights the trust values between direct neighbors of two users. MoleTrust [17] works in a similar fashion, but does a backward exploration while considering all users up to a pre-defined maximum depth. In order to efficiently avoid the impact of noisy data while still considering enough ratings, the authors of [10] proposed TrustWalker. They combined trust-based and item-based recommendations, where a random walk model is utilized to compute the confidence in the predictions.

Present work. In this work, we focus on the first step of CF, i.e., finding the k -nearest neighbors. For this purpose, we explore the use of a similarity measure from network science referred to as "Katz similarity" (KS) by the author of [18]. Although Katz himself

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

RecSys '18, Vancouver, BC, Canada

© 2018 ACM. 978-1-4503-5901-6/18/10...\$15.00

DOI: 10.1145/3240323.3240404

never discussed it, KS captures regular equivalence of nodes in a network and can be applied in many different settings [6, 7]. As such, in this work, we explore how to use KS in a trust-based CF approach.

Firstly, we utilize the trust connections to create an adjacency matrix where each entry represents a directed trust link between two users. Secondly, we apply the KS measure on the created trust adjacency matrix. More specifically, we calculate the pairwise similarities between users by using the iterative approach on calculating KS. The iterative approach does not only allow us to calculate the similarity between two nodes in the network, but additionally provides the possibility to choose the maximum used path length in doing so. This approach effectively gives us the ability to decide how far do we want to propagate trust in the network. Lastly, we use the resulting similarity matrix and apply various normalization techniques in order to get a better distribution of similarity values and better evaluation results in return. We evaluate these approaches on the Epinions dataset.

Contributions and findings. The contributions of this work are three-fold: (i) we explore the application of KS measure in the neighbor selection step of the trust-based CF approach for cold-start users, (ii) we evaluate different normalization techniques on the resulting similarity matrix to achieve better recommendation accuracy, and (iii) we introduce an adapted KS measure that gives higher similarity values to node pairs with path lengths of 2. In the trust-based CF setting, this means that propagated trust connections are given a higher importance than by using the standard KS measure.

Taken together, this study may help researchers to get an insight on how to apply KS on trust networks in combination with different normalization techniques to address the cold-start user problem in CF-based recommender systems. Moreover, we show that our approach for boosting the propagated trust values can result in increasing the impact of newly created trust connections on recommendation accuracy.

2 APPROACH

Our approach utilizes Katz similarity, which is a measure of regular equivalence, i.e., a measure of the extent to which two nodes share the same neighbors but also the extent to which their neighbors are similar. As described in [3], two nodes may have few or no neighbors in common, but they may still be similar in an indirect, global way. The idea behind KS is that paths of any length are contributing to the value of similarity between two nodes in the network, with shorter paths having a stronger impact. KS can be mathematically expressed in a matrix form as follows:

$$\sigma = \sum_{l=0}^{\infty} (\alpha A)^l = (\mathbf{I} - \alpha A)^{-1} \quad (1)$$

where σ represents the similarity matrix and each value $\sigma_{i,j}$ is a similarity value between nodes i and j , A represents the adjacency matrix of the network, \mathbf{I} is the identity matrix which is necessary to make sure that each node is similar to itself, α is the attenuation factor which weights the contribution of a path of length l . In our trust-based setting, the adjacency matrix A is asymmetric and it

represents an unweighted directed trust network, in which each node corresponds to a single user and each link represents a trust statement issued by one user to another:

$$A_{i,j} = \begin{cases} 1, & \text{if user } j \text{ expressed a trust statement to user } i \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

This also makes the similarity matrix σ asymmetric, which means that $\sigma_{i,j}$ does not have to be equal to $\sigma_{j,i}$, which is of advantage because in this way, the asymmetric property of trust is preserved. Furthermore, one important thing to note is that for (1) to converge, the attenuation factor has to satisfy the following condition:

$$\alpha < \frac{1}{\lambda_A} \quad (3)$$

where λ_A is the largest eigenvalue of A . The largest eigenvalue for the Epinions trust network (see Section 3) is 120.54, hence α needs to be less than 0.0083 and we set it to 0.008 throughout all of our experiments.¹ Since calculating the matrix inverse is computationally expensive, we can evaluate the above summation expression starting from $l = 0$ for a fixed maximum l (i.e., l_{max}) and get the following:

$$\begin{aligned} \sigma^{(0)} &= 0 \\ \sigma^{(1)} &= \mathbf{I} \\ \sigma^{(2)} &= \alpha A + \mathbf{I} \\ \sigma^{(3)} &= \alpha^2 A^2 + \alpha A + \mathbf{I} \\ &\dots \\ \sigma^{(l_{max}+1)} &= \sum_{l=0}^{l_{max}} (\alpha A)^l \end{aligned} \quad (4)$$

Step 1: Setting l_{max} . By using this approach and setting l_{max} to a positive integer value, we can define how far down the network do we want to propagate similarity or in this case, trust. In the conducted experiments, we used values 1 and 2 as l_{max} , which means that we either have not propagated similarities through the network at all or that we propagated them through the network using a maximum path length of 2.

Step 2: Degree normalization. As described in [18], σ as defined in (1), tends to give high similarity to nodes that have a high degree. In some cases this might be desirable but if we want to get rid of this bias, we could apply a degree normalization on σ , which would give higher similarity values to pairs of nodes that, independently of their degrees, are similar, while lower values would correspond to pairs of nodes that are dissimilar. Mathematically, for a given l_{max} , this step can be written as follows:

$$\sigma_{Dnorm}^{(l_{max}+1)} = \mathbf{D}^{-1} \left(\sum_{l=0}^{l_{max}} (\alpha A)^l \right) \mathbf{D}^{-1} \quad (5)$$

¹Although we used the iterative approach to calculate KS where l_{max} was set to a small integer value and α could have been set to any value between 0 and 1, we investigate the impact of α when the condition in Eq. 3 is also satisfied.

where \mathbf{D} represents a degree matrix of a network. In the conducted experiments, we evaluated approaches with an in-degree normalization, a combined-degree normalization and without a degree normalization.²

Step 3: Row normalization. After applying degree normalization, we found that all of the values in the degree normalized similarity matrix are very close to 0, including the maximum value. Therefore, we introduced an additional step where we individually scale rows of the final resulting matrix using one of the three vector norms: $l1$, $l2$ or max .³

Step 4: Boosting propagated similarities. As already mentioned, the attenuation factor α is used to decrease similarity the further it gets propagated in the network. Since we set the α to 0.008, similarity decays fast with each propagation step. Therefore, propagated similarity values become much smaller already in the first propagation step, i.e., for $l = 2$. This would mean that trust connections created through propagation in comparison with direct trust connections have an almost insignificant impact on the resulting recommendations, and therefore, this additional boosting step would increase the impact of propagated similarities with respect to the recommendation accuracy.

Largest value for l_{max} in the conducted experiments was set to 2. This could be interpreted as using user's neighbors and their neighbors for generating item recommendations. One of the contributions of this paper was to increase the impact of propagated trust values generated with KS for $l_{max} = 2$. Our proposed approach for doing so consists of the following four steps: (i) calculate $\sigma^{(3)}$ as described above using trust network as \mathbf{A} , (ii) create a new similarity matrix $\hat{\sigma}$ such that:

$$\hat{\sigma}_{i,j} = \begin{cases} \sigma_{i,j}^{(3)}, & \text{if } A_{i,j} = 0 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

(iii) create $\hat{\sigma}_{norm}$ matrix by individually scaling rows of $\hat{\sigma}$ using $l1$, $l2$ or max vector norm and lastly, (iv) create a similarity matrix σ_{boost} such that:

$$\sigma_{boost} = \mathbf{A} + \hat{\sigma}_{norm} \quad (7)$$

With this approach, we achieve that each entry in σ_{boost} has a similarity value of 1 between pairs of nodes for which there exists an explicit trust connection in \mathbf{A} and for pairs of nodes for which the similarity has been calculated through propagation, the similarity values are not exclusively small values close to zero increasing their impact on the resulting recommendations.

Recommendation strategy. As already outlined in Section 1, in this work, we focus on user-based CF. We first create a similarity matrix using the above mentioned four steps: (i) calculate σ using Eq. (4) with $l_{max} \in \{1, 2\}$, (ii) normalize the similarity matrix using in-degree or combined-degree normalization, (iii) normalize similarity matrix rows using $l1$, $l2$ or max vector norm, and (iv) apply boosting of propagated similarities. Steps (ii), (iii) and (iv)

²Combined-degree matrix is a diagonal matrix where each value on the diagonal corresponds to the sum of in-degree and out-degree of a particular node.

³For example, by utilizing the scikit-learn library in Python: <http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.normalize.html>.

are optional and can be skipped. Utilizing the created trust-based similarity matrix, we first find the k -nearest similar users and afterwards recommend the items of those users as a ranked list of top- n items to the target user. According to the literature, the maximum number of nearest neighbors should be a value between 20 and 60 [8], we used 60 in all of our experiments. The final ranking of the items to recommend is calculated by summing up the similarities of neighboring users as done in [11, 23].

3 EXPERIMENTAL SETUP

Dataset. To evaluate the performance of our trust-based CF approaches for cold-start users, the well-known *Epinions* dataset has been used [17]. This dataset was crawled from the consumer reviewing platform Epinions.com. Here, registered users can rate items available on the Epinions platform on a scale of 1 – 5. Additionally, users can issue trust statements to other users on the platform, i.e., they can express how much they trust other users. In this dataset, there are only positive values for trust statements, meaning there are no negative trust statements (i.e., distrust).

Taken together, there is a total number of 49,290 users in our dataset, which rated 139,738 different items with 664,824 ratings. Moreover, users have issued a total number of 487,181 trust connections. We utilized the trust connections issued by the users to create an unweighted trust network, in which each node represents a user and each directed link represents a trust statement expressed by one user to another. The resulting trust network provides a graph density value of 0.0002, making the trust network adjacency matrix very sparse.

Baseline algorithms. We compare our proposed approach to three baselines algorithms from the literature, which were shown to be useful methods in cold-start settings:

MP. MostPopular is a classic approach in recommender systems, which recommends the most frequently used items in the dataset to every user. Thus, it can be also applied in a cold-start setting.

Trust_{exp}. This naive trust-based approach uses explicit trust values in order to create the neighborhood of a user. Basically, adjacency matrix \mathbf{A} created from a trust network is used as a similarity matrix which does not allow for ranking of similar users because similarity values are binary, i.e., either 0 or 1.

Trust_{jac}. This is a trust-based approach using Jaccard coefficient on explicit trust values and was also used by the authors of [1]. The idea behind this approach is that two users are more similar the more trusted users they have in common. Jaccard coefficient is a statistic used to measure the similarity and diversity of sample sets and it can be written as:

$$J(\mathbf{A}_{*,a}, \mathbf{A}_{*,b}) = \frac{|\mathbf{A}_{*,a} \cap \mathbf{A}_{*,b}|}{|\mathbf{A}_{*,a} \cup \mathbf{A}_{*,b}|} \quad (8)$$

where $J(\mathbf{A}_{*,a}, \mathbf{A}_{*,b})$ is used to calculate similarity between users a and b , $\mathbf{A}_{*,a}$ corresponds to explicit values given to other users in the trust network by user a and the same applies to $\mathbf{A}_{*,b}$ for user b .

Evaluation method and metrics. In order to compare our proposed approach to these baseline algorithms in a cold-start setting, we extracted all users with no more than 10 rated items from the dataset. This resulted in 25,393 users, for which we put all of their

Approach	l_{max}	Degree norm.	Row norm.	Boost	nDCG	R	P
<i>Trust_{exp}</i>					.0224	.0296	.0110
<i>Trust_{jac}</i>					.0176	.0219	.0087
<i>MP</i>					.0134	.0202	.0070
<i>KSP_{CMB}</i>	2	Combined	Max	Yes	.0303	.0425	.0117
<i>KSP_{CMN}</i>	2	Combined	Max	No	.0295	.0422	.0113
<i>KSP_{CL₁B}</i>	2	Combined	L1	Yes	.0273	.0358	.0106
<i>KSP_{NL₂B}</i>	2	No degree	L2	Yes	.0257	.0340	.0106
<i>KSN_{CMN}</i>	1	Combined	Max	No	.0213	.0289	.0106
<i>KSN_{INN}</i>	1	In degree	N/A	No	.0161	.0243	.0087
<i>KSP_{NNN}</i>	2	No degree	N/A	No	.0036	.0057	.0020

Table 1: Evaluation results for $n = 10$. The reported subset of the 33 evaluated KS-based approaches are additionally labeled for an easier result comparison between different step combinations (i.e., columns 2 to 5).

rated items into the test set. To finally quantify the performance of our evaluated algorithms, we used the well-established accuracy metrics *nDCG*, *Precision* and *Recall* for $n = 1 - 10$ recommended items [9, 24].

4 RESULTS

In our study, we evaluated 33 approaches for all possible step combinations when creating the similarity matrix (i.e., as defined in Section 2). However, for the sake of space, in Table 1, we only report the results for a subset of these approaches that provide the most insightful findings. All of the evaluation results are reported for $n = 10$, i.e., for 10 recommended items. As it can be seen in Table 1, the best performing approach in terms of all accuracy measures was *KSP_{CMB}*, where we used trust propagation ($l_{max} = 2$) with combined degree normalization, row normalization with *max* norm as well as boosting of the propagated similarity values.

One interesting finding was that if similarity propagation was not used, i.e., l_{max} was set to 1, better results were achieved if degree and row normalization were not applied (i.e., basically the *Trust_{exp}* baseline). However, if l_{max} was set to 2, we noticed result improvements in almost all of the cases except when no row normalization was applied, e.g., in the case of *KSP_{NNN}*.

Additionally, similarity propagation with $l_{max} = 2$ increased the similarity matrix density from 0.0002 to 0.008. It turned out that row normalization was a very important step in using KS with similarity propagation for neighbor selection. Another important finding was that the combined-degree normalization provided better results than in-degree normalization in most of the cases. Also, with respect to row normalization, *max* norm provided better results than *l1* and *l2* norms in most of the cases. Lastly, with degree normalization and row normalization unchanged, boosting of propagated similarities often provided better results.

Finally, in Figure 1, we show the performance of all approaches listed in Table 1 in form of Recall-Precision plots for different number of recommended items (i.e., $n = 1 - 10$). The results clearly show that the best performing algorithm (i.e., *KSP_{CMB}*) again outperforms all of the other approaches also for a smaller number of recommended items (i.e., for $n < 10$).

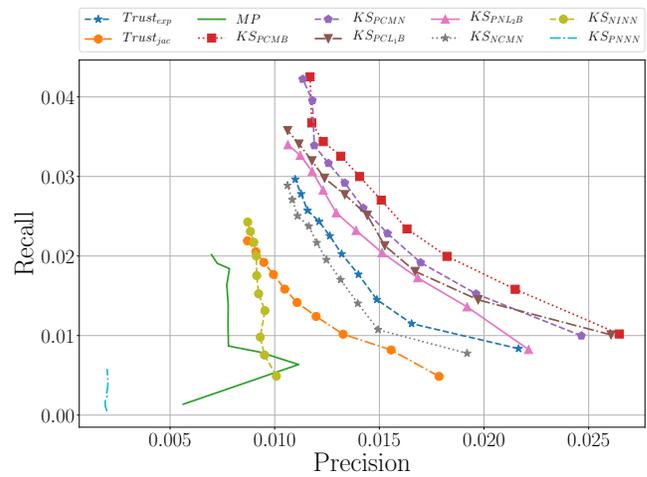


Figure 1: Recall-Precision plots of the described approaches for $n = 1 - 10$ recommended items. Again, we can observe that the approach labeled as *KSP_{CMB}* outperforms all three baselines as well as the other KS-based approaches.

5 CONCLUSION & FUTURE WORK

In this paper, we explored the use of Katz similarity (KS), a similarity measure of regular equivalence in networks, for selecting k -nearest neighbors in a Collaborative Filtering (CF) algorithm for cold-start users. We used an iterative approach for calculating KS since it provides the ability to restrict the length of paths in the network used for similarity calculation. We found that KS can be a very useful measure for neighbor selection if it is used with degree-normalization and row normalization, especially when using similarity propagation. When these techniques are properly combined with KS, we managed to outperform related approaches for tackling the cold-start problem. Our results also indicate that trust propagation is a very important feature when using trust networks in a CF setting as well as that KS is a useful technique for efficiently propagating trust in a network. Summed up, our study may help researchers to get an insight on how to apply KS on trust networks in combination with different normalization techniques to address the cold-start user problem in recommender systems.

One limitation of this study was that we only evaluated our approaches using recommender accuracy, although optimizing on non-accuracy measures has been closely tied to user satisfaction [12, 25]. As such, in the future we plan to investigate the impact of trust-based networks on beyond accuracy metrics such as novelty, diversity and coverage. Additionally, we would like to evaluate our approach not only on cold-start users, but rather to run the experiments on the complete dataset. We would also like to run additional experiments using different values for α and l_{max} . Moreover, we also plan to explore the use of recently popularized node embeddings (e.g., Node2Vec [5]) for trust networks to further improve our results. And finally, we plan to conduct a more extensive evaluation to see how our method compares with other popular approaches which support trust propagation [4, 10, 17].

Acknowledgments. This work was supported by the Know-Center (Austrian COMET program) and the AFEL project (GA: 687916).

REFERENCES

- [1] P. H. Chia and G. Pitsilis. Exploring the use of explicit trust links for filtering recommenders: a study on epinions. com. *Journal of Information Processing*, 19:332–344, 2011.
- [2] S. Fazeli, B. Loni, A. Bellogin, H. Drachslers, and P. Sloep. Implicit vs. explicit trust in social matrix factorization. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 317–320, New York, NY, USA, 2014. ACM.
- [3] M. Franceschet. Global similarity. Available at <https://www.sci.unich.it/~francesc/teaching/network/globalsimilarity.html>. Accessed: 2018-05-05.
- [4] J. A. Golbeck. *Computing and applying trust in web-based social networks*. PhD thesis, 2005.
- [5] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.
- [6] I. Hasani-Mavriqi, D. Kowald, D. Helic, and E. Lex. Consensus dynamics in online collaboration systems. *Computational social networks*, 5(1):2, 2018.
- [7] D. Helic. Regular equivalence in informed network search. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*, pages 1088–1093. IEEE, 2014.
- [8] J. Herlocker, J. A. Konstan, and J. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information retrieval*, 5(4):287–310, 2002.
- [9] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22, 2004.
- [10] M. Jamali and M. Ester. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 397–406. ACM, 2009.
- [11] E. Lacic, D. Kowald, L. Eberhard, C. Trattner, D. Parra, and L. B. Marinho. Utilizing online social network and location-based data to recommend products and categories in online marketplaces. In *Mining, Modeling, and Recommending 'Things' in Social Media*, pages 96–115. Springer, 2015.
- [12] E. Lacic, D. Kowald, M. Reiter-Haas, V. Slawicek, and E. Lex. Beyond accuracy optimization: On the value of item embeddings for student job recommendations. *arXiv preprint arXiv:1711.07762*, 2017.
- [13] E. Lacic, D. Kowald, M. Traub, G. Luzhnica, J. Simon, and E. Lex. Tackling cold-start users in recommender systems with indoor positioning systems. In *RecSys Posters*, 2015.
- [14] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pages 208–211. ACM, 2008.
- [15] N. Lathia, S. Hailes, and L. Capra. Trust-based collaborative filtering. In *IFIP international conference on trust management*, pages 119–134. Springer, 2008.
- [16] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu. A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems*, 56:156–166, 2014.
- [17] P. Massa and P. Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 17–24. ACM, 2007.
- [18] M. Newman. *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA, 2010.
- [19] S.-T. Park and W. Chu. Pairwise preference regression for cold-start recommendation. In *Proceedings of the third ACM conference on Recommender systems*, pages 21–28. ACM, 2009.
- [20] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.
- [21] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.
- [22] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002.
- [23] P. Seitlinger, D. Kowald, S. Kopeinik, I. Hasani-Mavriqi, E. Lex, and T. Ley. Attention please! a hybrid resource recommender mimicking attention-interpretation dynamics. In *Proceedings of the 24th International Conference on World Wide Web*, pages 339–345. ACM, 2015.
- [24] B. Smyth and P. McClave. Similarity vs. diversity. In *Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development*, ICCBR '01. Springer-Verlag, 2001.
- [25] Y. C. Zhang, D. Ó. Séaghdha, D. Quercia, and T. Jambor. Auralist: introducing serendipity into music recommendation. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 13–22. ACM, 2012.