

The importance of being dissimilar in Recommendation

Vito Walter Anelli*, Joseph Trotta*, Tommaso Di Noia*,
Eugenio Di Sciascio*, Azzurra Ragone*

*Polytechnic University of Bari

Bari - Italy

firstname.lastname@poliba.it

*Independent Researcher

azzurra.ragone@gmail.com

ABSTRACT

Similarity measures play a fundamental role in memory-based nearest neighbors approaches. They recommend items to a user based on the similarity of either items or users in a neighborhood. In this paper we argue that, although it keeps a leading importance in computing recommendations, similarity between users or items should be paired with a value of dissimilarity (computed not just as the complement of the similarity one). We formally modeled and injected this notion in some of the most used similarity measures and evaluated our approach showing its effectiveness in terms of accuracy results.

ACM Reference Format:

Vito Walter Anelli*, Joseph Trotta*, Tommaso Di Noia*, Eugenio Di Sciascio*, Azzurra Ragone*. 2018. The importance of being dissimilar in Recommendation. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Neighborhood-based approaches have been the first family of algorithms developed for collaborative filtering recommender systems. They identify similar users or items and they provide users a list of items they could be interested in by exploiting the degree of similarity. Though they have been around for many years, it has been shown that neighborhood-based approaches may perform better than latent model based methods to solve the *top-N* recommendation problem [3, 12, 16, 20]. In the *top-N* recommendation task, the focus is on providing an accurate ranked list rather than minimizing the rating prediction error. Among neighborhood-based methods the best-known are user-kNN, item-kNN and Sparse Linear Methods (SLIM) [20]. User-based and item-based schemes have proven to be effective in different settings despite they use the same logic behind the scenes. In details item-kNN and SLIM (which uses an item-based scheme) have shown to outperform user-kNN to solve the *top-N* recommendation problem [10] and several algorithms have been proposed in the literature to enhance neighbors models like GLSLIM [10] and Weighted kNN-GRU4REC [15], taking advantage of personalized models and recurrent neural networks. Moreover, we also have approaches focusing on the injection of time in neighborhood models [6] and modeling similarities by directly optimizing the pair-wise preferences error [23]. All in all, under the hood what makes neighborhood-based methods work is a similarity measure.

Several similarity measures have been proposed and used extensively such as Jaccard [13][22] and Tanimoto [11] coefficients, Cosine Vector similarity [1, 4, 7], Pearson Correlation [14], Constrained Pearson correlation [25], Adjusted Cosine similarity [24], Mean Squared Difference similarity [25], Spearman Rank Correlation [18], Frequency-Weighted Pearson Correlation [8], Target item weighted Pearson Correlation [5]. In the vast majority of cases, all these similarity measures are based on two assumptions:

- (1) the correlation between i and j is the same correlation between j and i (symmetry of similarity);
- (2) the correlation between two entities captures only how much they are similar with each other without taking into account their degree of dissimilarity.

To the best of our knowledge, a few works have been proposed in the past years related to asymmetric similarity, and they are mainly designed for a user-based scheme. An Asymmetric User Similarity has been proposed in [19] where the authors underline that a similarity measure should distinguish between a user with a rich profile and a cold user. Thus, given two users u and v , they slightly modify the Jaccard index in order to consider exclusively the number of ratings of the current user (instead of the overall number of both users). In HYBRTyco [17] the similarity proposed by Millan [19] is combined with the Sørensen index [26]. HYBRTyco [17] is an hybrid recommender system which combines matrix factorization with an asymmetric similarity model to realize a typicality-based collaborative filtering recommender system. The same approach is exploited in another asymmetric user similarity model [21] to feed a user-user similarity matrix that is then completed using a matrix factorization algorithm. Additionally, both of them provide an extension for the latter similarity measure based on explicit numerical feedbacks (ratings). Despite previous works are focused on the user-based scheme, we already underlined that item-kNN shows very good performance in *top-N* recommendation task. Moreover, when the number of users exceeds the number of items, as in most of the cases, item-based recommendation approaches require much less memory and time to compute the similarity weights than user-based ones, making them more scalable. Due to these reasons, both approaches have been considered in this work.

In this work we investigate the effect on recommendation accuracy when we go beyond the above two assumptions and define (and include) the concepts of dissimilarity and asymmetry in similarity measures. In our proposal, we start from a probabilistic interpretation of similarity to define symmetric and asymmetric dissimilarities. The dissimilarity measures are then combined with traditional similarity values using additive and multiplicative strategies. The

experimental evaluation shows that our approach outperforms the non-dissimilarity-aware counterparts improving accuracy of results or diversity or both.

The rest of the paper is organized as follows: Section 2 presents the motivation behind our work and the proposed approach. Section 3 presents the evaluation protocol, metrics, datasets and performance of the method. Finally, in Section 4 concluding considerations are provided.

2 PROPOSED APPROACH

2.1 Motivation

A symmetric similarity may not be sufficient to capture subtle interactions between items. Our assumption is that representing the similarity through traditional measures can lead to imperfect results as important information might not be considered. Let us consider some examples in an item-kNN scenario. Suppose we are dealing with a dataset containing ratings data from the book domain on the following books:

Title	Short name	Author	# Votes
A Game of Thrones	GoT	G. R. R. Martin	100
A Dance with Dragons	DwD	G. R. R. Martin	10
Shroud of Eternity	SoE	T. Goodkind	120

In details both "A Game of Thrones" and "A Dance with Dragons" belong to the same saga "A song of ice and fire" and they are, respectively, the first and the fifth volume. "Shroud of Eternity" is the second volume of Nicci Chronicles's saga. We may assume that all the users who rated DwD also rated GoT, i.e. $U_{DwD} \subseteq U_{GoT}$. Analogously, we may assume that a number of readers of SoE also voted GoT, $U_{SoE} \cap U_{GoT} \neq \emptyset$. Suppose now that we have $U_{SoE} \cap U_{GoT} = 20$ and $U_{DwD} \cap U_{GoT} = 10$. If we compute the Jaccard similarity between the pairs SoE, GoT and DwD, GoT we have

$$JS(DwD, GoT) = \frac{|U_{DwD} \cap U_{GoT}|}{|U_{DwD} \cup U_{GoT}|} = 0.1$$

$$JS(SoE, GoT) = \frac{|U_{SoE} \cap U_{GoT}|}{|U_{SoE} \cup U_{GoT}|} = 0.1$$

In our opinion, a lot of relevant information has been lost in this simple example. The scenario is shown graphically in Figure 1.

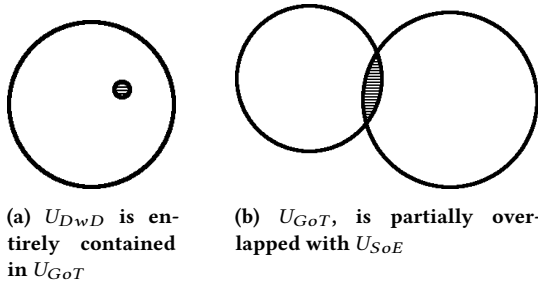


Figure 1: Representation of the motivation example

It is clear that U_{DwD} is a proper subset of U_{GoT} , on the contrary there are a lot of users in U_{SoE} that have not experienced GoT. This information is mostly lost after the similarity measurement even

though a piece of this information is retained in the denominator of the Jaccard coefficient through the overall value of $U_{GoT} \cup U_{DwD}$ and $U_{GoT} \cup U_{SoE}$. Let us focus on a more formal description of the scenario to explicit the reason why we do not consider this remaining information sufficient.

Many similarity measures (like Jaccard in the previous example) mainly rely on a value that denotes the similarity between two items normalized by their overall weight. This can be represented as a probability. Let $\mathcal{I}(u) = \{\langle i, r_{ui} \rangle \mid u \text{ rated } i \text{ with } r_{ui}\}$ be the user profile containing the pairs item-rating and $\mathcal{U}(i) = \{\langle u, r_{ui} \rangle \mid \langle i, r_{ui} \rangle \in \mathcal{I}(u)\}$ be the set of users that experienced i . If we consider Jaccard similarity we see it represents the following probability:

$$p^{JS}(i, j) = p(\langle u, r_{ui} \rangle \in \mathcal{U}(i) \wedge \langle u, r_{uj} \rangle \in \mathcal{U}(j)) \quad (1)$$

which can be read as the probability that u experienced both i and j . Analogously, we may define a dissimilarity measure, given by two different probabilities:

- the probability that a generic user u experienced the item i but never experienced the item j
- the probability that a generic user u experienced the item j but never experienced the item i

This consideration is certainly important but these two different probabilities are not equally important.

In a classical memory-based Item-kNN, $sim(i, j)$ is used to compare i to different j to find out which js are the most similar to i . In practical terms, we are interested in how much j is similar to i . If we focus on Figure 1a, we realize that DwD is more similar to GoT than the opposite situation. The reason behind this behavior is not directly related to the size of the involved sets but it depends on the probability that a user who experienced GoT did not experience DwD.

Though some interesting asymmetric similarities have been proposed in the last years, to our knowledge, no one focused on this probability that represents a negative asymmetric dissimilarity.

2.2 Metrics

In this work we propose a general asymmetric similarity model in which items i, j similarities are computed by taking into account the probability that users that experienced j never experienced i . The idea behind our work is preserving the core meaning of a specific similarity, applying a corrective factor encoding the dissimilarity we mentioned before.

We introduced this correction into two binary symmetric similarities: Jaccard and Sørensen index, and in the asymmetric variant to Jaccard coefficient proposed in the literature [19]. We tested this correction both as an additive and as a multiplicative factor.

For the sake of completeness, we reintroduce *Jaccard coefficient similarity* (JS) that, for a memory-based item-kNN model can be expressed as:

$$JS(i, j) = \frac{|\mathcal{U}(i) \cap \mathcal{U}(j)|}{|\mathcal{U}(i) \cup \mathcal{U}(j)|} \quad (2)$$

The probability that users who experienced j never experienced i can be modeled as the complementary probability of Equation (1) w.r.t. $\mathcal{U}(i)$ over $|\mathcal{U}(i) \cup \mathcal{U}(j)|$. This probability, we name *Jaccard*

Asymmetric Dissimilarity(JAD) can be formulated as follows:

$$JAD(i, j) = \frac{|\mathcal{U}(j)| - |\mathcal{U}(i) \cap \mathcal{U}(j)|}{|\mathcal{U}(i) \cup \mathcal{U}(j)|} \quad (3)$$

Again, Equation (3) can be seen in terms of probability as

$$p^{JAD}(i, j) = p(\langle u, r_{uj} \rangle \in \mathcal{U}(j)) - p^{JS}(i, j)$$

We now propose to modify the original similarity injecting the former negative correction weighted with a parameter λ that can be easily customized. The overall similarity, *Additive Adjusted Jaccard* (AAJ), is then formulated as:

$$AAJ(i, j) = JS(i, j) - \lambda \cdot JAD(i, j) \quad (4)$$

where λ is a parameter that could depend on many factors such as the number of users and items in the dataset and the intensity of interactions between them.

Recalling that the overall formula should represent a degree of similarity between the two different items we defined the *Multiplicative Adjusted Jaccard* as the product of Jaccard similarity with the inverse of Jaccard Asymmetric Dissimilarity (IJAD). In other words, we use $\frac{1}{p^{JAD}(i, j)}$ as a corrective factor for $p^{JS}(i, j)$. We then define the *Multiplicative Adjusted Jaccard* (MAJ) as:

$$MAJ(i, j) = \frac{JS(i, j)}{JAD(i, j)} = JS(i, j) \cdot IJAD(i, j) \quad (5)$$

In the multiplicative variant, in order to avoid division by zero the minimum value for JAD is set to $\frac{1}{|\mathcal{U}(i) \cup \mathcal{U}(j)|}$. As mentioned in Section 2.1 a symmetric variant of Jaccard coefficient (named *Jaccard Symmetric Dissimilarity* (JSD)) can be used composing Equation (3) with the probability that a user u experienced the item i but never experienced the item j :

$$\begin{aligned} JSD(i, j) &= \frac{(|\mathcal{U}(j)| - |\mathcal{U}(i) \cap \mathcal{U}(j)|) + (|\mathcal{U}(i)| - |\mathcal{U}(i) \cap \mathcal{U}(j)|)}{|\mathcal{U}(i) \cup \mathcal{U}(j)|} \\ &= \frac{|\mathcal{U}(i)| + |\mathcal{U}(j)| - 2 \cdot |\mathcal{U}(i) \cap \mathcal{U}(j)|}{|\mathcal{U}(i) \cup \mathcal{U}(j)|} \end{aligned}$$

leading to the corresponding probability

$$p^{JSD}(i, j) = p^{JAD}(i, j) + p^{JAD}(j, i)$$

Thus the *Symmetric Additive Adjusted Jaccard* (S-AAJ) and *Symmetric Multiplicative Adjusted Jaccard* (S-MAJ) can be defined as follows:

$$S-AAJ(i, j) = JS(i, j) - \lambda \cdot JSD(i, j)$$

$$S-MAJ(i, j) = \frac{JS(i, j)}{JSD(i, j)}$$

In order to test our idea, we applied all the variants previously introduced for Jaccard similarity to two more popular similarity measures: *Asymmetric Jaccard Similarity* (AJS) and *Sørensen coefficient* (SOR). All the derived variants are represented, respectively, in Table 1 and Table 2.

All the above metrics have been introduced having in mind a item-kNN approach but, without loss of generality, they can be applied to user-kNN model as well.

Table 1: Asymmetric Jaccard considered variants.

Short name	Extended	Formula
AJS(i,j)	Asymm. Jacc. Similarity	$\frac{ \mathcal{U}(i) \cap \mathcal{U}(j) }{ \mathcal{U}(i) }$
AJD(i,j)	Asymm. Jacc. Dissimilarity	$\frac{ \mathcal{U}(j) - \mathcal{U}(i) \cap \mathcal{U}(j) }{ \mathcal{U}(i) }$
AAAJ(i,j)	Additive Adjusted As. Jacc.	$AJS(i, j) - \lambda \cdot AJD(i, j)$
MAAJ(i,j)	Multiplicative Adjus. As. Jacc.	$AJS(i, j) \cdot IJAD(i, j)$
S-AAAJ(i,j)	Symmetric AAAJ	$AJS(i, j) - \lambda \cdot AJD(i, j)$
S-MAAJ(i,j)	Symmetric MAAJ	$AJS(i, j) \cdot IJSD(i, j)$

Table 2: Sørensen similarity considered variants.

Short name	Extended	Formula
SOR(i,j)	Sørensen Similarity	$\frac{ \mathcal{U}(i) \cap \mathcal{U}(j) }{ \mathcal{U}(i) + \mathcal{U}(j) }$
ASD(i,j)	Asymm. Sør. Dissimilarity	$\frac{ \mathcal{U}(j) - \mathcal{U}(i) \cap \mathcal{U}(j) }{ \mathcal{U}(i) + \mathcal{U}(j) }$
AAS(i,j)	Additive Adjusted As. Sør.	$SOR(i, j) - \lambda ASD(i, j)$
MAS(i,j)	Multiplicative Ad. As. Sør.	$SOR(i, j) \cdot ASD(i, j)$
S-AAS(i,j)	Symmetric AAS	$SOR(i, j) - \lambda \frac{ \mathcal{U}(i) + \mathcal{U}(j) - 2 \cdot \mathcal{U}(i) \cap \mathcal{U}(j) }{ \mathcal{U}(i) + \mathcal{U}(j) }$
S-MAS(i,j)	Symmetric MAS	$SOR(i, j) \cdot IJSD(i, j)$

3 EXPERIMENTAL EVALUATION

Datasets. We evaluated the effectiveness of our approach on three datasets belonging to different domains (Music, Books, and Movies), shown in Table 3. The Last.fm dataset [9] corresponds to transactions with Last.fm online music system released in HetRec 2011¹. It contains social networking, tagging, and music artist listening information from a set of 2K users. LibraryThing represents books ratings collected in the LibraryThing community website. It contains social networking, tagging, and rating information on a [1..10] scale. Yahoo!Movies (Yahoo! Webscope dataset ydata-ymovies-user-movie-ratings-content-v1_0)² contains movies ratings generated by Yahoo! Movies up to November 2003. It provides content, demographic, ratings information, and mappings to MovieLens and EachMovie datasets.

Table 3: Datasets statistics.

Dataset	#Users	#Items	#Transactions	Sparsity
Yahoo! Movies	7642	11,916	221,367	99.76%
LibraryThing	7279	37,232	2,056,487	99.24%
Last FM	1850	11,247	59,071	99.72%

Columns corresponding to #Users, #Items and #Transactions show the number of users, number of items and number of transactions, respectively, in each dataset. The last column shows the sparsity of the dataset.

Evaluation Protocol and Experimental Setting. We employed the *all unrated items* [27] evaluation protocol to evaluate the proposed method. In *all unrated items* the recommendation list is computed from a candidate list given by the cartesian product between users and items minus the items each user experimented in the

¹<http://ir.ii.uam.es/hetrec2011/>

²http://research.yahoo.com/Academic_Relations

Table 4: Comparison in terms of Precision and Aggregate Diversity for User-kNN scheme

Precision - P@10																												
	JS		AAJ		MAJ		S-AAJ		S-MAJ		ASOR		SOR		S-AAS		S-MAS		AJS		AAAJ		MAAJ		S-AAAJ		S-MAAJ	
Datasets	P@10	λ	P@10	P@10	λ	P@10	P@10	P@10	P@10	λ	P@10	P@10	P@10	λ	P@10	P@10	P@10	P@10	λ	P@10	P@10	λ	P@10	P@10	λ	P@10	P@10	P@10
LibraryThing	0.0363	0.2	0.0582	0.0627	0.4	0.0364	0.0010	0.0394	0.0363	0.2	0.0586	0.0629	0.2	0.0405	0.0364	0.0364	0.2	0.0558	0.0603	0.2	0.0057	0.0375						
Yahoo	0.0437	0.4	0.0561	0.0676	0.2	0.0403	0.0433	0.0442	0.0438	0.4	0.0573	0.0687	0.2	0.0568	0.0435	0.0607	0.6	0.0529	0.0664	0.2	0.0378	0.0529						
Last FM	0.0164	0.4	0.0242	0.0257	0.2	0.0037	0.0160	0.0215	0.0166	0.4	0.0241	0.0253	0.2	0.0275	0.0164	0.0248	0.2	0.0228	0.0323	0.2	0.0037	0.0242						

Aggregate Diversity - D@10																												
	JS		AAJ		MAJ		S-AAJ		S-MAJ		ASOR		SOR		S-AAS		S-MAS		AJS		AAAJ		MAAJ		S-AAAJ		S-MAAJ	
Datasets	D@10	λ	D@10	D@10	λ	D@10	D@10	D@10	D@10	λ	D@10	D@10	λ	D@10	D@10	D@10	D@10	D@10	λ	D@10	D@10	λ	D@10	D@10	λ	D@10	D@10	
LibraryThing	2136	0.2	7367	2406	0.4	2246	873	2819	2083	0.2	7330	2292	0.2	5528	2171	1299	0.2	7501	2060	0.2	3331	1486						
Yahoo	734	0.4	2070	2187	0.2	835	825	936	695	0.4	2048	786	0.2	1114	784	451	0.6	1773	717	0.2	2175	587						
Last FM	1449	0.4	1345	1654	0.2	1203	1460	1433	1324	0.4	1324	1625	0.2	1410	1420	774	0.2	1665	1077	0.2	1207	941						

Table 5: Comparison in terms of Precision and Aggregate Diversity for Item-kNN scheme

Precision - P@10																												
	JS		AAJ		MAJ		S-AAJ		S-MAJ		ASOR		SOR		S-AAS		S-MAS		AJS		AAAJ		MAAJ		S-AAAJ		S-MAAJ	
Datasets	P@10	λ	P@10	P@10	λ	P@10	P@10	P@10	P@10	λ	P@10	P@10	λ	P@10	P@10	P@10	P@10	P@10	λ	P@10	P@10	λ	P@10	P@10	λ	P@10	P@10	
LibraryThing	0.0869	0.2	0.0949	0.0451	0.4	0.0822	0.0944	0.1019	0.0815	0.2	0.0914	0.0374	0.4	0.0826	0.0901	0.0598	0.2	0.1011	0.0303	0.4	0.0830	0.0792						
Yahoo	0.0331	0.2	0.0510	0.0016	0.2	0.0535	0.0373	0.0447	0.0297	0.2	0.0504	0.0014	0.2	0.0531	0.0352	0.0046	0.4	0.0509	0.0012	0.2	0.0527	0.0083						
Last FM	0.0141	0.2	0.0248	0.0036	0.2	0.0230	0.0158	0.0127	0.0124	0.4	0.0195	0.0031	0.2	0.0230	0.0155	0.0036	0.2	0.0237	0.0032	0.2	0.0229	0.0068						

Aggregate Diversity - D@10																												
	JS		AAJ		MAJ		S-AAJ		S-MAJ		ASOR		SOR		S-AAS		S-MAS		AJS		AAAJ		MAAJ		S-AAAJ		S-MAAJ	
Datasets	D@10	λ	D@10	D@10	λ	D@10	D@10	D@10	D@10	λ	D@10	D@10	λ	D@10	D@10	D@10	D@10	D@10	λ	D@10	D@10	λ	D@10	D@10	λ	D@10	D@10	
LibraryThing	9745	0.2	12004	17510	0.4	12304	10249	10399	9556	0.2	11727	16557	0.4	13306	9998	17361	0.2	11221	11132	0.4	12315	14819						
Yahoo	3103	0.2	3718	3463	0.2	4315	2541	3346	3184	0.2	3935	3397	0.2	3541	2475	2948	0.4	4092	2334	0.2	3756	2295						
Last FM	3508	0.2	3538	3486	0.2	3462	2911	3779	3265	0.4	305	3175	0.2	3509	2887	2992	0.2	3935	2644	0.2	3692	3457						

training set. We performed a temporal 80-20 hold-out split retaining the last 20% of ratings as test set when temporal information is available. We measured the performance by computing Precision@N ($Prec@N$) for $top-N$ recommendation lists as accuracy metric. Precision has been computed on a per-user basis and then results have been averaged. As Precision needs relevant items to be computed, we set the relevance threshold to 8 over 10 for LibraryThing and Yahoo!Movies, and to 0 for Last.fm since in the latter no ratings are provided but the number of user-item transactions. We measured Diversity through *catalog coverage* (aggregate diversity in $top-N$ list). The *catalog coverage*, also called diversity-in-top-N ($D@N$) [2], is measured by computing the overall number of different items recommended within all recommendation lists. It represents the propensity of a system to recommend always the same items.

Baselines. We compared our approaches with in both User-kNN and Item-kNN settings. The former finds the k -nearest user neighbors based on a similarity function and then exploits them to predict a score for each user-item pair. The latter is the item-based version of the k -nearest neighbors algorithm that uses the k -nearest items to make the predictions. For both the algorithms $k = 80$ has been set. However, we are not interested in the algorithm itself but on the similarity measures that are used to compute neighbors and predictions. As baseline to compare with, we used both symmetric and asymmetric measures, namely, *Jaccard (JS)* and *Sorensen (SOR)* (for symmetric measures) and *asymmetric Jaccard (AJS)* and *asymmetric Jaccard weighted with the Sorensen Index (ASOR)* [17] (for asymmetric measures). For all the similarities that make use of λ we evaluated them varying λ in 0.2,0.4,0.6,0.8. Only best results have been shown in Tables 4 and 5³. The best values in terms of Precision and Aggregate Diversity are highlighted in bold. We computed significance tests for precision highlighted results and

we found them statistically significant at the 0.05 level w.r.t. their respective baselines.

Performance of the proposed methods. The results in Table 4 show that our approach always outperforms baseline variants in User-kNN scheme. In details **additive** asymmetric similarity and **multiplicative** asymmetric similarity significantly perform better than JS, SOR and ASOR for all three dataset. Among these two variants of similarity, the multiplicative variant is absolutely the best-performing one. Quite interestingly, modifying AJS, which is asymmetric in its inner nature with our asymmetric dissimilarity factor leads to an improvement irrespective of the considered dataset. It is worth noticing that, other than the accuracy improvements, aggregate diversity also increases due to the dissimilarity injection. In details, the asymmetric additive variant achieves the best results and triples catalog coverage values for LibraryThing and Yahoo!Movies.

Table 5 shows Precision and Catalog Coverage results for an item-based scheme. These results are very interesting for many reasons. First of all, it is clear that, same similarities can lead to very different results depending on the scheme adopted. In particular asymmetric Jaccard (AJS) performs very badly for item-kNN algorithm. Under the dissimilarities perspective, it happens exactly the same and multiplicative approach performs bad. Quite surprisingly the additive version is able to always outperform the base variants. This suggests that adopting an additive strategy for item-kNN may lead to better results. This is probably due to the wide number of items pairs without any common user. Focusing on additive symmetric and asymmetric similarities we can note that aggregate diversity results reflect the same improvements observed in accuracy results. The only case that appear to behave differently is AAAJ that registered a catalog coverage lower than AJS. Actually, this happens as we considered the best performing λ in terms of precision. In case of $\lambda \in \{0.4, 0.6\}$ we obtain aggregate diversity values of 16,205 and

³The complete results are publicly available at <https://github.com/sisinflab/The-importance-of-being-dissimilar-in-Recommendation>.

18,071, respectively, with precision values constantly higher than AJS (0.09374 and 0.08820).

4 CONCLUSION AND FUTURE WORK

In this work we propose a method to improve performance of neighborhood based models, by capturing subtle interactions between users and items, which cannot be appreciated using a traditional similarity measure. We defined a dissimilarity measure, that can be used combined with traditional user-based and item-based schemes. The proposed approach takes into account the single asymmetric components, leading to more fine improvements in terms of precision and aggregate diversity. We performed a comparative experimental evaluation using three well-known datasets, varying the tuning parameter λ . Experiments show that our approach outperforms competing algorithms, denoting the usefulness of incorporating symmetric and asymmetric dissimilarity in neighborhood-based models. We are currently working on an extension of our idea that takes into account also user ratings and not just set-based measures. As further extension we are also interested in making the approach even more personalized by weighting dissimilarity with user-centered lambdas.

REFERENCES

- [1] S Abiteboul, P Buneman, D Suciu, J Allan, R Papka, V Lavrenko, A Blum, T Mitchell, A Bonifati, S Ceri, et al. 1995. News Weeder: Learning to Filter News. In *Proceedings of the 12th International Conference of Machine Learning (ICML-95)*.
- [2] Gediminas Adomavicius and YoungOk Kwon. 2012. Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques. *IEEE Trans. Knowl. Data Eng.* 24, 5 (2012), 896–911.
- [3] Fabio Aioli. 2013. A Preliminary Study on a Recommender System for the Million Songs Dataset Challenge. In *Proceedings of the 4th Italian Information Retrieval Workshop, Pisa, Italy, January 16-17, 2013*. 73–83. <http://ceur-ws.org/Vol-964/paper12.pdf>
- [4] Marko Balabanovic and Yoav Shoham. 1997. Content-Based, Collaborative Recommendation. *Commun. ACM* 40, 3 (1997), 66–72. <https://doi.org/10.1145/245108.245124>
- [5] Linas Baltrunas and Francesco Ricci. 2009. Item Weighting Techniques for Collaborative Filtering. In *Knowledge Discovery Enhanced with Semantic and Social Information*. 109–126. https://doi.org/10.1007/978-3-642-01891-6_7
- [6] Alejandro Bellogin and Pablo Sánchez. 2017. Revisiting Neighbourhood-Based Recommenders For Temporal Scenarios. In *Proceedings of the 1st Workshop on Temporal Reasoning in Recommender Systems co-located with 11th International Conference on Recommender Systems (RecSys 2017), Como, Italy, August 27-31, 2017*. 40–44. <http://ceur-ws.org/Vol-1922/paper8.pdf>
- [7] Daniel Billsus and Michael J. Pazzani. 2000. User Modeling for Adaptive News Access. *User Model. User-Adapt. Interact.* 10, 2-3 (2000), 147–180. <https://doi.org/10.1023/A:1026501525781>
- [8] John S. Breese, David Heckerman, and Carl Myers Kadie. 1998. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *UAI '98: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, University of Wisconsin Business School, Madison, Wisconsin, USA, July 24-26, 1998*. 43–52. https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=231&proceeding_id=14
- [9] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2011. 2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011). In *Proceedings of the 5th ACM conference on Recommender systems (RecSys 2011)*. ACM, New York, NY, USA.
- [10] Evangelia Christakopoulou and George Karypis. 2016. Local Item-Item Models For Top-N Recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*. 67–74. <https://doi.org/10.1145/2959100.2959185>
- [11] Edjalma Q. da Silva, Celso G. Camilo-Junior, Luiz Mario L. Pascoal, and Thier-son Couto Rosa. 2016. An evolutionary approach for combining results of recommender systems techniques based on collaborative filtering. *Expert Syst. Appl.* 53 (2016), 204–218. <https://doi.org/10.1016/j.eswa.2015.12.050>
- [12] Mukund Deshpande and George Karypis. 2004. Item-based top-N recommendation algorithms. *ACM Trans. Inf. Syst.* 22, 1 (2004), 143–177. <https://doi.org/10.1145/963770.963776>
- [13] Wei Dong, Charikar Moses, and Kai Li. 2011. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th international conference on World wide web*. ACM, 577–586.
- [14] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. 2002. An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms. *Inf. Retr.* 5, 4 (2002), 287–310. <https://doi.org/10.1023/A:1020443909834>
- [15] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27-31, 2017*. 306–310. <https://doi.org/10.1145/3109859.3109872>
- [16] Santosh Kabbur, Xia Ning, and George Karypis. 2013. FISM: factored item similarity models for top-N recommender systems. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*. 659–667. <https://doi.org/10.1145/2487575.2487589>
- [17] Rahul Katarya and Om Prakash Verma. 2016. Effective collaborative movie recommender system using asymmetric user similarity and matrix factorization. In *Computing, Communication and Automation (ICCCA), 2016 International Conference on*. IEEE, 71–75.
- [18] MG Kendall and JD Gibbons. 1990. Rank correlation methods, trans. *JD Gibbons (5th edn ed.)*. Edward Arnold: London (1990).
- [19] Marta Millan, Maria Trujillo, and Edward Ortiz. 2007. A collaborative recommender system based on asymmetric user similarity. In *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 663–672.
- [20] Xia Ning and George Karypis. 2011. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, December 11-14, 2011*. 497–506. <https://doi.org/10.1109/ICDM.2011.134>
- [21] Parivash Pirasteh, Dosam Hwang, and Jason J Jung. 2015. Exploiting matrix factorization to asymmetric user similarities in recommendation systems. *Knowledge-Based Systems* 83 (2015), 51–57.
- [22] Ali Mustafa Qamar, Eric Gaussier, Jean-Pierre Chevallet, and Joo Hwee Lim. 2008. Similarity learning for nearest neighbor classification. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 983–988.
- [23] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*. 452–461. https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1630&proceeding_id=25
- [24] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*. 285–295. <https://doi.org/10.1145/371920.372071>
- [25] Upenra Shardanand and Pattie Maes. 1995. Social Information Filtering: Algorithms for Automating "Word of Mouth". In *Human Factors in Computing Systems, CHI '95 Conference Proceedings, Denver, Colorado, USA, May 7-11, 1995*. 210–217. <https://doi.org/10.1145/223904.223931>
- [26] T. Sørensen. 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biol. Skr.* 5 (1948), 1–34.
- [27] Harald Steck. 2013. Evaluation of recommendations: rating-prediction and ranking. In *RecSys*. 213–220.