Tag-Based User Fuzzy Fingerprints for Recommender Systems

André Carvalho, Pável Calado, and Joao Paulo Carvalho(✉)

INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal
{andre.silva.carvalho,pavel.calado}@tecnico.ulisboa.pt,
joao.carvalho@inesc-id.pt

Abstract. Most Recommender Systems rely exclusively on ratings and are known as Memory-based Collaborative Filtering systems. This is currently dominant approach outside of academia due to the low implementation effort and service maintenance, when compared with more complex Model-based approaches, Traditional Memory-based systems have as their main goal to predict ratings, using similarity metrics to determine similarities between the users' (or items) rating patterns. In this work, we propose a user-based Collaborative Filtering approach based on tags that does not rely on rating prediction, instead leveraging on Fuzzy Fingerprints to create a novel similarity metric. Fuzzy Fingerprints provide a concise and compact representation of users allowing the reduction of the dimensionality usually associated with user-based collaborative filtering. The proposed recommendation strategy combined with the Fuzzy Fingerprint similarity metric is able to outperform our baselines, in the Movielens-1M dataset.

Keywords: Recommender system · Collaborative Filtering
Fuzzy Fingerprint · Tags

1 Introduction

Users of the digital world are overloaded with information [16]. Recommender Systems (RSs) allow us to cope with this, by cataloging a vast list of items, that later can be recommended. Due to their success, RSs can be found in a number of services, providing recommendations for movies, music, news, products, events, services, among others [1]. However, turning state of the art solutions into real-world scenarios is still challenging, mainly due to a large amount of available data and the ensuing scalability issues. For this reason, more traditional approaches, such as Collaborative Filtering (CF) are still the most widely used [18]. Despite its simplicity, CF can provide quite accurate results, thus yielding an advantageous trade-off between engineering effort and user satisfaction.

Memory-based Collaborative Filtering can usually be implemented using one of two different strategies: *user-based* CF, which compares users ratings to determine a neighborhood of similar users; and *item-based* CF, which instead computes item similarities and forms item neighborhoods to produce the rating predictions.

J. Medina et al. (Eds.): IPMU 2018, CCIS 855, pp. 748–758, 2018.
https://doi.org/10.1007/978-3-319-91479-4_62

Over the years, item-based CF has replaced user-based CF, given its better scalability properties [11]. Since the number of users grows over time, and generally at a faster rate than the items, so does the number of similarities, thus posing a scalability problem. Similarities between users also vary more over time than similarities between items, since individual users tend to change their preferences, while the global opinion on a given item tends to remain stable.

In this work, we argue that an effective and efficient user-based CF system can be implemented. To this effect, we use Fuzzy Fingerprints (FFPs) to represent users based on item *tags* and ratings. *Tags*, i.e. short textual labels attached by the users to the items, provide an item description or categorization and are a common resource in current online RSs. They allow us to create a more detailed user representation than traditional CF, in a controlled manner, i.e. by controlling the number of tags used in the FFPs, we can easily fine-tune our system to improve recommendation quality or to speed up the similarity computation. In this work, we mainly focus on obtaining an improved recommendation quality.

Our main contributions are, therefore, (1) a new way to determine relevant items to recommend to users without requiring the computation of rating predictions for user-based CFs, and (2) a novel similarity metric for RSs, using the concept of FFPs [9] to represent users based on tags from rated items. More specifically, we propose to represent users by their low-dimensional Fingerprints, which can then be directly used to determine similarities between them. A similar idea has been previously applied to text authorship identification [9] with success. Our goal is to apply the same principle to RSs using tags from the items rated by each user, to obtain better recommendations. This solution has three major advantages: (1) provides overall better recommendations to users; (2) requires a minimal implementation effort; and (3) its representation of the users is scalable and easily maintainable.

To demonstrate our claims, experiments were performed on a movie dataset providing movies metadata information, allowing the creation of users FFPs.

The remainder of this paper is organized as follows: Sect. 2 contains literature review on similarity metrics for CF; Sect. 3 presents how FFPs can be applied to RSs; Sect. 4 presents an experimental evaluation; finally, in Sect. 5 some conclusions are drawn from the results and directions for future work are proposed.

2 Related Work

Fuzzy systems approaches have been previously used to improve the RS similarity metric [6] focusing exclusively on item-based CF. Our proposal applies concepts of Fuzzy Systems to the problem of user-based Collaborative Filtering. More specifically, we use Fuzzy Fingerprints, in a CF system, to represent users in a more compact way.

CF systems usually rely on the ratings given to items by users to determine similarities between users (or items), through the use of a similarity metric. This allows the creation of neighborhoods of similar users, to predict new ratings.

Traditionally, the similarity is measured using metrics such as Pearson Correlation (PC) or the Cosine similarity (COS) [2]. Nevertheless, many other ways of measuring similarity have been proposed, ranging from simple variations of PC and COS, through the design of more complex functions.

An example is the work of [7], where ratings are combined with a measure of *trust* between users, which is inferred from social information. By introducing the degree of trust between users the authors show that it does improve the overall rating prediction. On a different approach, in [3], the authors propose a combination of the mean squared difference between the user's ratings with the Jaccard coefficient. Through experiments, they demonstrate that results are improved, when compared to traditional CF.

To determine the neighborhood of each user, usually, similarities are computed between the user and all other users, which are then sorted by their degree of similarity and only the top k are kept. In [17] an alternative way to determine neighborhoods is proposed. The authors randomly choose a possible neighbor from the set of all users. This neighbor is kept only if its similarity is above a given threshold. The process is then repeated until a certain amount of neighbors is obtained. Their work has two threshold variables that depend on the data and must be fine-tuned: (1) the minimum similarity for a user to be considered a neighbor; (2) the minimum number of users in the neighborhood.

Combining Recommender systems and tags is not a novel idea [10,13,15]. Tags can help alleviate the so-called *cold-start* and *data sparsity* problems. The cold-start problem occurs when new items, not yet rated by any user, or new users, who have not rated any item yet, cannot receive recommendations since they cannot be compared to other items/users. The *data sparsity* problem is also associated with CF systems since it is common for users and items to have very few ratings, and thus not enough information to produce valuable recommendations [4]. Tags can help address these issues, they only depend on the availability of metadata, for each item. Our RS takes advantage of tags to more accurately represent each user and, therefore, improve the quality of the user similarity computation.

Liu et al. [12] also propose a new similarity metric, which assigns penalties to *bad* similarities, while rewarding *good* similarities. Defining a similarity as good or bad depends on several factors, such as the popularity of the rated items or the similarity of the rating to the other user's ratings.

In [5] a FFP was applied to item-based CF using also movies synopsis to represent items. The FFP results from ratings and synopsis words that are also added as features. A normalization is applied to both ratings and synopsis words, separately, resulting in FFP which combines both. Note that in this work, we are currently creating a user-based CF to represent users with item tags weighted by the ratings, and not represent item using FFP.

The above works show that the selection process of neighbors and the improvement of the similarity measures have a beneficial impact on the overall RS results. This work presents a similarity metric based on FFPs, adapted for user-based CF, using the tags associated to each item, with the main goal of improving the recommendation quality.

3 Tag-Based User Fuzzy Fingerprints for Collaborative Filtering

A Fuzzy Fingerprint (FFP) is a fuzzified ranked vector containing information based on frequencies of occurrence of the elements being encoded [9]. In this Section, we explain how to build and apply a tag-based FFP to represent users in a CF recommender system.

Let N be the total number of tags in the system and let M be the total number of items in the system. Let θ_i represent the set of tags of a given item i: $\theta_i = (t_{1i}, t_{2i}, t_{3i}, \cdots, t_{Ni})$. Any element $t_{ni} \in \theta_i$ can assume the value 1 if the respective tag occurs in the item, or 0 if it does not.

Let r_u be the set of ratings for a given set of items $i_1 \cdots, i_M$, provided by a user u: $r_u = (r_{1u}, r_{2u}, \cdots, r_{Mu})$. We assume, without loss of generality, that $r_{mu} \geq 0$ and that a value of zero means that the user has not yet rated item i_m.

A Fingerprint ϕ_u is built by counting, for user u, the number of occurrences of each tag in the items rated by u, multiplied by the respective item's rating, i.e. $\phi_u = (c_{1u}, c_{2u}, \cdots, c_{Nu})$, where:

$$c_{nu} = \sum_{\forall i=1}^{M} t_{ni} \times r_{iu} \tag{1}$$

The rationale behind Eq. (1) is that tags from items a user has rated higher should also get a higher importance in the Fingerprint. The next step consists in ordering ϕ_u according to c_{nu} and keeping only the k highest values. The Fingerprint size k is a parameter of the system and can be optimized offline.

To illustrate the previous procedure, let $r_u = (5, 2, 4)$ for items a, b, and c. Assume there are only 5 tags and let $\theta_a = (1, 0, 0, 1, 1)$, $\theta_b = (0, 1, 0, 0, 1)$, and $\theta_c = (0, 0, 1, 1, 0\}$. Assuming that $k = 4$, the resulting Fingerprint ϕ_u will be $(c_{4u} = 9, c_{5u} = 7, c_{1u} = 5, c_{3u} = 4)$.

The Fingerprint ϕ_u is, therefore, an *ordered set* of tags. The rank of each tag reflects its importance in representing the user. This Fingerprint still needs to be transformed into a Fuzzy Fingerprint. The fuzzification of the Fingerprint leverages the importance of the order (and not of the frequency) to distinguish between users. The FFP of user u, Φ_u, is obtained by fuzzifying the rank (the position in the Fingerprint) of each tag.

The choice of the fuzzifying function can affect the obtained results [8,9]. Here, we have tested the linear approach, shown in Eq. 2, where p_{u_j} is the rank of tag t_n within ϕ_u (starting with $t = 0$).

$$\mu_{linear}(\mathrm{p}_{t_n}) = \frac{k - \mathrm{p}_{t_n}}{k} \tag{2}$$

Preliminary experiments indicate that using other fuzzifying functions does not significantly improve or degrade the quality of the results in this approach.

After the fuzzification step, we can now define the FFP Φ_u as:

$$\Phi_u = \{(t_n, \mu(\mathrm{p}_{t_n})), \forall t_n \in \phi_u\} \tag{3}$$

The FFP is, therefore, a ranked set of tags, each of which is associated with a membership value, built based on the description of the items rated by the user.

Once the FFP for each user is determined, it is possible to compute similarities between users.

Consider Φ_u and Φ_j the FFPs of users u and j. The FFP similarity between users u and j is defined as:

$$sim_{FFP} = (\Phi_u, \Phi_j) = \sum_{t_n \in U_i \cap U_j} \frac{\min(\Phi_u(t_n), \Phi_j(t_n))}{k} \tag{4}$$

where $\Phi_x(t_u)$ denotes the membership value associated to tag t_n in Φ_x. Note that the use of k in this equation as a normalization factor is only needed to facilitate development and parameter optimization. It can be omitted during system operation when computing similarities, largely improving computational efficiency.

The recommendation process of the proposed RS does not rely upon rating predictions as in traditional Collaborative Filtering (see Sect. 4). Instead, it identifies the user's nearest neighbors (according to Eq. 4) and uses the items seen and liked by them to extrapolate possible items to recommend to the user.

The RS starts by computing which users are the nearest neighbors of user u, based on the FFP similarity metric. Users are considered neighbors if the similarity is greater than a defined threshold $sim_{\text{threshold}}$.

We consider that any item rated highly by a neighbor (e.g., 4 or 5 on a 0–5 scale) and rated higher than that neighbor's item rating average, is recommendable to the user.

The final step in the recommendation process consists in getting the difference between the rating of the recommendable item, the average rating given to that item by the neighbor, and multiplying it by the similarity between the user and the neighbor. This allows to create a ranking of recommendable items.

4 Evaluation

To assert the effectiveness of the proposed RS experiments were performed using a movie dataset. Precision, Recall, and F1-score are used as evaluation metrics.

The similarity metrics used as baselines for comparison are the traditional Pearson Correlation (PC) and Cosine similarity (COS). In addition, we also include the Jaccard Mean Squared Difference (JMSD) [3], an improvement on previous metrics that offers a high rating prediction accuracy, while using a lower number of neighbors. Finally, a similarity metric, that uses FFPs [6] yet is only applicable to traditional item-based CF and which only relies upon ratings to compute similarities. We refer to this baseline [6] throughout the rest of this document as FFP_{rating}. While the FFP proposed in this document will be referenced as FFP_{tags}. All similarity metrics baselines use both user-based and item-based, except FFP_{rating} that is only applicable to item-based CF.

Pearson Correlation coefficient has been widely used since it is simple to implement, intuitive, and provides good quality results [3]. PC is defined in Eq. 5, where I is the set items both user u and j rated.

$$sim_{PC}(u,j) = \frac{\sum_{i \in I}(r_{u,i} - \bar{r}_u) \times (r_{j,i} - \bar{r}_j)}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_u)^2} \times \sqrt{\sum_{i \in I}(r_{j,i} - \bar{r}_j)^2}} \tag{5}$$

The resulting similarity will be in within the interval $[-1, 1]$, where -1 corresponds to an inverse correlation, $+1$ to a positive correlation, and values near zero show that no linear correlation exists between the two users.

Another often used similarity measure is the Cosine similarity, as defined in Eq. 6. COS will yield a value between 0 and 1, where 0 corresponds to no similarity between u and j and 1 to exactly proportional ratings between both users.

$$sim_{COS}(u,j) = \frac{\sum_{i \in I} r_{u,i} \times r_{j,i}}{\sqrt{\sum_{i \in I} r_{u,i}^2} \times \sqrt{\sum_{i \in I} r_{j,i}^2}} \tag{6}$$

The idea behind Jaccard Mean Squared Difference (JMSD) is to combine the Jaccard coefficient, which captures the number of ratings in common between users, with the Mean Square Difference (MSD) of those ratings, resulting in Eq. 7:

$$sim_{JMSD}(u,j) = Jaccard(u,j) \times \big(1 - MSD(u,j)\big); \tag{7}$$

where $Jaccard$ and MSD are defined as:

$$Jaccard(i,j) = \frac{|I_u \cap I_j|}{|I_u \cup I_j|} \qquad MSD(i,j) = \frac{\sum_{i \in I}(r_{u,i} - r_{j,i})^2}{|I|} \tag{8}$$

where I_s is the set of items rated by user s.

The FFP$_{rating}$ metric uses an approach that is totally different to the one proposed in this work: each item has its own FFP and the recommendation is based exclusively on ratings. The user's ratings constitute the item Fingerprint and ratings are sorted taking into consideration the total amount of ratings from each user.

We now explain how a traditional CF computes rating predictions. Let $\hat{r}_{ui}$ be the *predicted* rating that a given user u would assign to item i. We start by computing the *neighborhood* N_u, of user u, i.e. the set of n users in the database that are more similar to u, using a similarity function. The value of $\hat{r}_{ui}$ is defined as:

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N_u} \mathrm{sim}(u,v) \times (r_{vi} - \bar{r}_v)}{\sum_{v \in N_v} \mathrm{sim}(u,v)} \tag{9}$$

where r_{vi} is the rating assigned by user v to item i, $\bar{r}_x$ is the average of all ratings assigned to user x. A traditional CF system usually performs these predictions for a large set of items and returns those with the highest rating predictions, as recommendations.

An evaluation was conducted using MovieLens-1M (ML-1M) dataset, from the movie domain. By using Dbpedia[1], Tags and other meta-data, regarding each movie, were collected. In this work, we focus exclusively on Tag information.

The ML-1M dataset has 1 million ratings, 6040 users, 3706 items, a sparsity of 95.53% and has an average of 125 ratings per user.

The evaluation process was performed through 5-fold cross-validation, using RiVal [14], a framework to make RSs evaluation fair process, completely separating the recommendation task of a RS from the Evaluation of the recommendations.

We define any item with rating greater than or equal to 4 as a relevant (i.e. should be recommended) to the user.

Precision can be computed using Eq. 10 and Recall using Eq. 11. In this work, we do not set a threshold for a maximum number of recommendations i.e. the RS can recommend as many relevant items to a user as possible. Even though we calculate the F1-score (Eq. 12.), we support the idea that Precision is a far better indication for a good RS, as long as Recall is within a range that allows the retrieval of a sufficient number of relevant items (in the tested cases, all approaches fulfill the Recall criteria).

$$PR = \frac{\#TruePositives}{\#TruePositives + \#FalsePositives} \tag{10}$$

$$RC = \frac{\#TruePositives}{\#TruePositives + \#FalseNegatives} \tag{11}$$

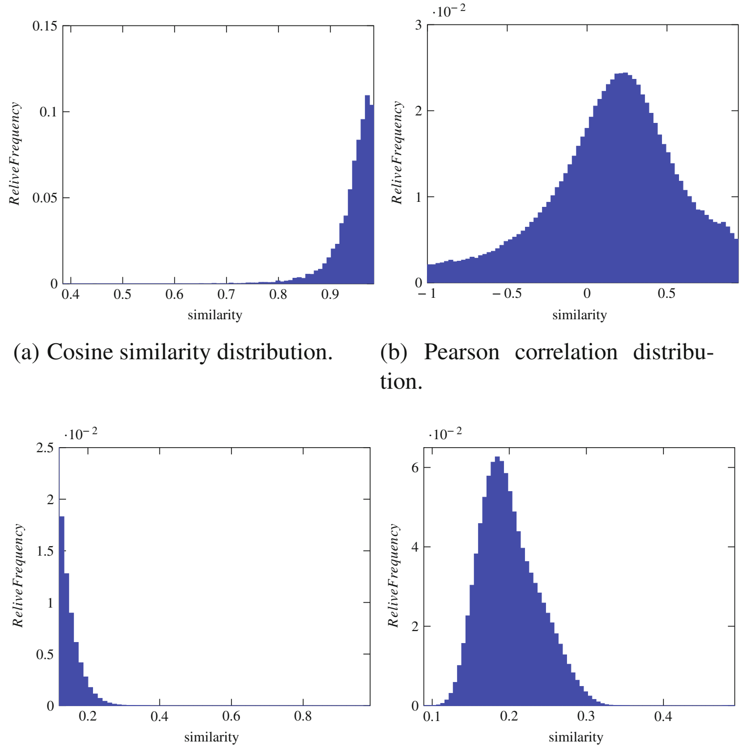
$$F1 = 2 \times \frac{PR \times RC}{PR + RC} \tag{12}$$

We start by comparing the similarity distribution using our similarity metric and the baselines, this allows us to determine the best $sim_{\text{threshold}}$ when selecting the neighborhood. We then vary the number of neighbors used by the FFP$_{tags}$ over different sizes of k. This allows us to determine not only the best k for the FFP$_{tags}$ but also the most adequate number of neighbors to use. Finally, we present a summary table with baselines and how do they perform in comparison to the proposed RS.

Figure 1 shows the similarity distributions. By analyzing Fig. 1d, we notice that the average similarity is around 0.2. This provides a good indicator to experiment different $sim_{\text{threshold}}$ around 0.2. Experimentally, we determined that using 0.25 provides good results, for this dataset.

Figure 2 compares different sizes for the FFP and for each size we vary the number of neighbors used by the RS. According to the $F1 - measure$ the best results are obtained using k equal to 200. Knowing that, on average, each user has 637 tags associated to rated movies, the proposed FFP similarity metric uses only 31% of existing tags, being able to correctly select relevant tags to represent each user.

[1] Dbpedia: http://www.dbpedia.org.



(a) Cosine similarity distribution.

(b) Pearson correlation distribution.

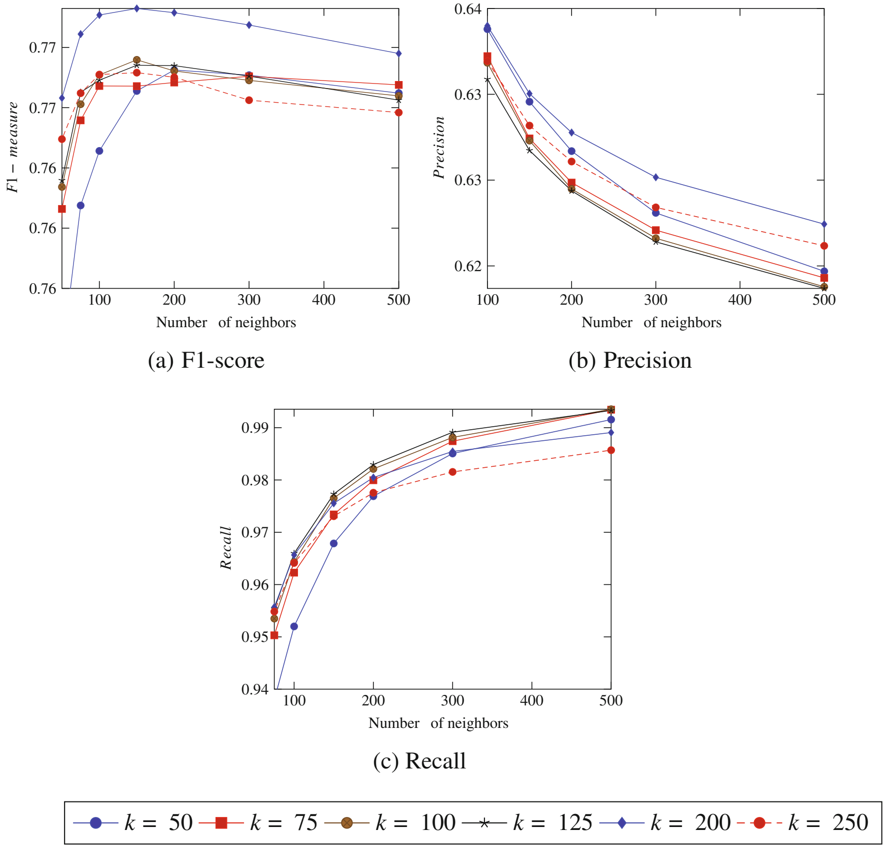
(c) JMSD similarity distribution.

(d) FFP_{tags} similarity distribution.

Fig. 1. Histograms show the similarity distribution of different similarity metrics when applied to user similarity computation. FFP_{tags} uses $k = 200$ tags to represent a user FFP.

Table 1 shows how the different tested approaches perform. The proposed Tag-user based FFP performs better overall than any other approach, even when compared to the state-of-the-art JMSD, although the improvement is not significant.

An interesting result is how much better the proposed approach is when compared to other previously proposed user-based approaches, thus opening the door to further developments in user-based RS. It should be noted that item-based approaches have been thoroughly used in the past and have been highly optimized. Yet user-based approaches are also viable. For example, it is very easy to enrich the FFP using data other than simple tags, from movie descriptions to a user's favorite actors, directors or genres.



(a) F1-score (b) Precision

(c) Recall

Fig. 2. Comparison between different sizes of the FFP, while varying the number of neighbors used.

Table 1. Summary results in which FFP_{tags} (using $k = 200$) combined with the proposed recommendation algorithm is compared with several baselines using item-based and user-based CF.

Similarity Metric	Approach	Num. neighbors	F1-score	Precision	Recall
FFP_{tags}	User-based	150	**0.76929**	**0.63504**	0.97554
COS	Item-based	50	0.76622	0.62112	**0.99978**
PC	Item-based	75	0.76621	0.62115	0.99969
JMSD	Item-based	20	0.76623	0.62112	0.99980
$FPP_{ratings}$	Item-based	20	0.76623	0.62113	0.99979
COS	User-based	200	0.42356	0.26869	0.99989
PC	User-based	100	0.42338	0.26854	0.99990
JMSD	User-based	100	0.42356	0.26869	0.99989

5 Conclusion

In this work, we have applied the concept of Fuzzy Fingerprints to user-based Collaborative Filtering and represented users based on tags according to the items they rated. FFPs are used to create a new concise user representation that improves the F1-score and Precision of an RS. The best result for the proposed approach was obtained for $k = 200$. In this dataset, each user has on average 637 tags, which shows that the FFPs are able to reduce the problem complexity while still improving recommendation quality.

We have experimentally compared our proposal to two traditional similarity measures, Pearson Correlation and Cosine similarity, and a state-of-the-art similarity metrics such as Jaccard Mean Squared Difference.

Results show that FFPs are a promising approach since they can be applied with success in recommendation tasks. In fact, using FFPs we are able to represent a user using, on average, 68% less features. In addition, and even though we do not address such issue in this paper, FFP similarity is a much more computationally efficient process than any of the other similarity measures. This can be arguably enough to compensate for the fact that there are usually much more users than items in RS, as we will try to show in a future work.

Future work includes more extensive parameter optimization, enriching the FFP with other features, and improving the last step of the recommendation algorithm by using more sophisticated ways to aggregate the influence of each neighbor.

Acknowledgments. This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013, by project GoLocal (ref. CMUPERI/TIC/0046/2014) and co-financed by the University of Lisbon and INESC-ID.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005). https://doi.org/10.1109/TKDE.2005.99
2. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender systems survey. Knowl.-Based Syst. **46**, 109–132 (2013). https://doi.org/10.1016/j.knosys.2013.03.012. http://www.sciencedirect.com/science/article/pii/S0950705113001044
3. Bobadilla, J., Serradilla, F., Bernal, J.: A new collaborative filtering metric that improves the behavior of recommender systems. Knowl.-Based Syst. **23**(6), 520–528 (2010). https://doi.org/10.1016/j.knosys.2010.03.009. http://www.sciencedirect.com/science/article/pii/S0950705110000444
4. Bogdanov, D., Haro, M., Fuhrmann, F., Xambó, A., Gómez, E., Herrera, P.: Semantic audio content-based music recommendation and visualization based on user preference examples. Inf. Process. Manage. **49**(1), 13–33 (2013). https://doi.org/10.1016/j.ipm.2012.06.004. http://www.sciencedirect.com/science/article/pii/S0306457312000763

5. Carvalho, A., Calado, P., Carvalho, J.P.: Combining ratings and item descriptions in recommendation systems using fuzzy fingerprints. In: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–6 (2017). https://doi.org/10.1109/FUZZ-IEEE.2017.8015604
6. Carvalho, A., Calado, P., Carvalho, J.P.: Fuzzy fingerprints for item-based collaborative filtering. In: Kacprzyk, J., Szmidt, E., Zadrożny, S., Atanassov, K.T., Krawczak, M. (eds.) IWIFSGN/EUSFLAT -2017. AISC, vol. 641, pp. 419–430. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-66830-7_38
7. Chen, S., Luo, T., Liu, W., Xu, Y.: Incorporating similarity and trust for collaborative filtering. In: Sixth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2009, vol. 2, pp. 487–493 (2009). https://doi.org/10.1109/FSKD.2009.720
8. Dimiev, V.: Fuzzifying functions. Fuzzy Sets Syst. **33**(1), 47–58 (1989). https://doi.org/10.1016/0165-0114(89)90216-9. http://www.sciencedirect.com/science/article/pii/0165011489902169
9. Homem, N., Carvalho, J.P.: Authorship identification and author fuzzy 'fingerprints'. In: Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), pp. 1–6 (2011). https://doi.org/10.1109/NAFIPS.2011.5751998
10. Kim, H.N., Ji, A.T., Ha, I., Jo, G.S.: Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation. Electron. Commer. Res. Appl. **9**(1), 73–83 (2010). https://doi.org/10.1016/j.elerap.2009.08.004. http://www.sciencedirect.com/science/article/pii/S1567422309000544, Special Issue: Social Networks and Web 2.0
11. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. **7**(1), 76–80 (2003). https://doi.org/10.1109/MIC.2003.1167344
12. Liu, H., Hu, Z., Mian, A., Tian, H., Zhu, X.: A new user similarity model to improve the accuracy of collaborative filtering. Knowl.-Based Syst. **56**, 156–166 (2014). https://doi.org/10.1016/j.knosys.2013.11.006. http://www.sciencedirect.com/science/article/pii/S0950705113003560
13. Osmanli, O., Toroslu, I.: Using tag similarity in SVD-based recommendation systems. In: 2011 5th International Conference on Application of Information and Communication Technologies (AICT), pp. 1–4 (2011). https://doi.org/10.1109/ICAICT.2011.6111034
14. Said, A., Bellogín, A.: RiVal: a toolkit to foster reproducibility in recommender system evaluation. In: RecSys 2014 Proceedings of the 8th ACM Conference on Recommender Systems, pp. 371–372. ACM Press (2014). https://doi.org/10.1145/2645710.2645712, http://dl.acm.org/citation.cfm?doid=2645710.2645712
15. Smith, G.: Tagging: People-Powered Metadata for the Social Web, 1st edn. New Riders Publishing, Thousand Oaks (2007)
16. Tsai, C.F., Hung, C.: Cluster ensembles in collaborative filtering recommendation. Appl. Soft Comput. **12**(4), 1417–1425 (2012). https://doi.org/10.1016/j.asoc.2011.11.016. http://www.sciencedirect.com/science/article/pii/S1568494611004583
17. Wibowo, A.T., Rahmawati, A.: Naive random neighbor selection for memory based collaborative filtering. In: 2015 International Seminar on Intelligent Technology and Its Applications (ISITIA), pp. 351–356 (2015). https://doi.org/10.1109/ISITIA.2015.7220005
18. Ye, T., Bickson, D., Ampazis, N., Benczur, A.: LSRS'15: Workshop on large-scale recommender systems. In: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, pp. 349–350. ACM, New York (2015). https://doi.org/10.1145/2792838.2798715, http://doi.acm.org/10.1145/2792838.2798715