Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı, S. 270-275, Kasım 2020

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue, pp. 270-275, November 2020

Research Article

A Study of Static and Dynamic Significance Weighting Multipliers on the Pearson Correlation for Collaborative Filtering

Savaş Okyay[1], Sercan Aygün[2*]

[1] Department of Computer Engineering, Eskişehir Osmangazi University, Eskişehir, Turkey (ORCID: 0000-0003-3955-6324)
[2] Department of Computer Engineering, Yıldız Technical University, İstanbul, Turkey (ORCID: 0000-0002-4615-7914)

(International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) 2020 – 22-24 October 2020)

(**DOI:** 10.31590/ejosat.822968)

ATIF/REFERENCE: Okyay, S. & Aygün, S. (2020). A Study of Static and Dynamic Significance Weighting Multipliers on the Pearson Correlation for Collaborative Filtering. *European Journal of Science and Technology*, (Special Issue), 270-275.

Abstract

Recommender systems as a field of data mining and knowledge discovery have a tremendous impact on movie recommendation platforms. Proper recommendation for the audience, considering profiles, is a measurable argument. By inferencing the linear combinations between some numerical data such as user rating actions, statistical analyses can be done. Thus, any item such as a movie can be recommended or not. The numerical calculation of correlations, namely the similarity weight, should be recomputed before prediction to increase the effect of user similarities for further constant multiplications. This method is named as the significance weighting that processes one more step to stress the impact of similarities. The affinity between users can simply be the total number of co-rated items, or any further inference using more complex computations. In this work, the significance weighting method related to Pearson Correlation is inspected using comparative approaches. The MovieLens dataset, both including `ML100K` and `ML1M` releases, are used in the experiments. *k*-fold cross-validation method is applied in a shifting fashion to increase the number of tests. After having Pearson Correlation Coefficients for *user-user* similarities, weights are signified using three different approaches. Then, neighbors are sorted to choose the top-*N* closest users for the user in the test. Concerning experimental results, over two other techniques, an explicit method that utilizes only the co-rated item count is preferred taking its simplicity and performance into account. In the plots of experimental results section, accuracy and error metrics are presented for three different significance weighting approaches. Especially for the `ML100K` dataset, the simple weighting method outperforms in terms of the error metrics.

Keywords: Collaborative filtering, MovieLens, Pearson similarity, recommender systems, significance weighting.

İşbirlikçi Filtreleme için Pearson Korelasyonu Üzerine Statik ve Dinamik Önem Ağırlıklandırma Çarpanları Çalışması

Öz

Veri madenciliği ve bilgi keşfinin bir alanı olarak öneri sistemleri, film tavsiye platformları üzerinde muazzam bir etkiye sahiptir. Profilleri göz önünde bulundurarak izleyiciler için uygun tavsiye ölçülebilir bir argümandır. Kullanıcı oylama eylemleri gibi bazı sayısal veri içerisindeki doğrusal kombinasyonları çıkararak istatistiksel analizler yapılabilir. Böylece, film gibi herhangi bir öğe kullanıcıya önerilebilir veya önerilmeyebilir. Korelasyonların sayısal hesaplaması, yani benzerlik ağırlığı, kullanıcı benzerliklerinin etkisini daha fazla sabit çarpımla arttırmak için tahminden önce yeniden hesaplanmalıdır. Bu yöntem, benzerliklerin etkisini vurgulamak için bir adım daha işleyen önem ağırlıklandırması olarak adlandırılır. Kullanıcılar arasındaki yakınlık, ortak oylanan öğelerin toplam sayısı veya daha karmaşık hesaplamalar yapılan başka bir çıkarım olabilir. Bu çalışmada, Pearson Korelasyonu ile

[*] Corresponding Author: Department of Computer Engineering, Yıldız Technical University, İstanbul, Turkey. ORCID: 0000-0002-4615-7914, ayguns@yildiz.edu.tr

ilgili önem ağırlıklandırma yöntemi karşılaştırmalı yaklaşımlar kullanılarak incelenmiştir. Deneylerde hem `ML100K` hem de `ML1M` sürümlerini içeren MovieLens veri kümesi kullanılır. *k*-katlamalı çapraz doğrulama yöntemi, test sayısını artırmak için kaydırmalı tarzda uygulanır. *Kullanıcı-kullanıcı* benzerlikleri için Pearson Korelasyon Katsayılarını elde ettikten sonra, ağırlıklar üç farklı yaklaşım kullanılarak ifade edilir. Ardından komşular, testteki kullanıcı için en yakın *N* kullanıcıyı seçmek üzere sıralanır. Deneysel sonuçlarla ilgili olarak, diğer iki tekniğe göre, basitliği ve performansı hesaba katılarak, sadece ortak oylanan öğe sayısını kullanan açık yöntem tercih edilir. Deneysel grafiklerde, doğruluk ve hata ölçümleri üç farklı önem ağırlıklandırma yaklaşımı için sunulmuştur. Özellikle `ML100K` veri kümesi için, basit ağırlıklandırma yöntemi hata ölçümleri açısından daha iyi performans gösterir.

Anahtar Kelimeler: İşbirlikçi filtreleme, MovieLens, Pearson benzerliği, öneri sistemleri, önem ağırlıklandırma.

1. Introduction

Recommender systems (RS) are in a wide range of usage from movie recommendations to commercial item suggestions (Ahmad and Afzal 2020; Aiolli 2013; LVN et al., 2014; Nguyen et al., 2020; Philip et al., 2014; Samad et al., 2019; Singh et al., 2020). In RS, previous preferences are processed using data mining methods, and prospective personal choices are offered. Scientific researches show how the 3-step system implementation is created by measuring the performance. Concisely, the statistical correlation measurement between vectors is the first step. The vector is either users of the intended system or the items depending on the *user-based* or *item-based* similarities (Aygun and Okyay, 2015). The next step is the utilization of the obtained correlation value to attain the numerical prediction. Depending on the calculated prediction value, any system performance can be measured. Let's assume that user similarities are obtained, and for any custom user, an item of interest is tested. The item value can be categorical binary information such as *liked* or *disliked*, or multi-level ratings, i.e., *stars*, such as half-stars and full-stars. By trivially choosing an item that has actual value for test purposes, the obtained prediction via the similarity is compared with the real value. This is the performance measurement phase of the overall framework as the final step that donates the scientifically valuable information for any proposed algorithms in the literature (Hong-Xia, 2019; Nguyen et al., 2020; Powers, 2011).

For the primary step of our study, *user-user* based similarities are captured. In this part, the renowned linear correlation is computed via the *Pearson Correlation Coefficient* (PCC) (Dhawan et al., 2015; Madadipouya, 2015; Sheugh and Alizadeh, 2015). During the second step, the prediction is calculated with the aid of an approach named *mean centering* (Saric et al., 2009; Zeybek and Kaleli, 2018). Finally, the obtained prediction is measured in terms of the actual value. Thereabouts, the utilization of the confusion matrix is needed. We do perform a direct evaluation with *one-to-one* comparison, *F1-measure*, *accuracy*, *mean absolute error* (MAE), and *root mean square error* (RMSE).

When it comes to the literature, there are indicative studies that work on movie-based recommendations. Besides, a subset of RS science pays attention to weight significance (Bellogín et al., 2014; Gao et al., 2012; Ghazanfar and Prugel-Bennett, 2010; Hwang and Chen, 2007; Levinas, 2014; McLaughlin and Herlocker, 2004; Raeesi and Shajari, 2012; Weng et al., 2006; Zhang et al., 2020; Zhang and Yuan, 2017). We perform *user-user* similarities in this work, and we call the weighting signifier, i.e., multiplier, after the PCC result as weight, w, is obtained. One of the simplest methods is the consideration of commonly *co-rated item count* (CIC) in-between *user-neighbor*. This as a run-time compatible method, takes the intersection of commonly rated items between the active user and the neighbors for any correlation constant. In some papers, this method is called *user overlap* (Bellogín et al., 2014; Raeesi and Shajari, 2012). In addition to this, Bellogín et al. also discuss different weighting strategies, like the ones in (Herlocker et al., 2002; Hwang and Chen, 2007; McLaughlin and Herlocker, 2004; Weng et al., 2006). Besides, the *case amplification* method is proposed by Breese et al. (Breese et al., 2013). Still, Ghazanfar and Prugel-Bennett criticize the method reason why it is not including the number of items in common (Ghazanfar and Prugel-Bennett, 2010). In (Herlocker et al., 2017), Herlocker et al. consider the common item counts; however, Ghazanfar and Prugel-Bennett indicate that the approach is not performing well for the weights lower than zero (Ghazanfar and Prugel-Bennett, 2010). They also review the methods by Ma et al. (Ma et al., 2007) and McLaughlin & Herlocker (McLaughlin and Herlocker, 2004), indicating that either approach utilizing the *minimum* or *maximum* operations as an enhanced version of (Herlocker et al., 2017) has the generalization problem.

On the other side, single test item dynamicity at each test attempt converges to the computation load, which is against the performance. In this work, we trivially show how it is possible to use a relatively acceptable weighting signifier that is also compatible with the dynamical approach. In the following sections, first, the methodology will be presented. The details of how to signify the weights properly will be given in that section. Three approaches will be considered. Then, the results will be shown in the following section by addressing the `100K`-sized and `1M`-sized datasets of MovieLens (Harper and Konstan, 2015).

2. Methodology

The overall methodology will be presented in this section. First, the similarity and prediction equations will be given. After that, the weight significance will follow, presenting three approaches.

2.1. Equations

The primary step is about how to compute linear similarities between two arguments. As *user-user* similarities are in our focus, the similarity coefficient between two users is calculated using the following formula in Equation 1.

$$w_{a,u}^{PCC} = \frac{\sum_{i\in(\mathrm{I}_a\cap\mathrm{I}_u)}\left((r_{a,i}-\bar{r}_a)\times(r_{u,i}-\bar{r}_u)\right)}{\sqrt{\sum_{i\in(\mathrm{I}_a\cap\mathrm{I}_u)}(r_{a,i}-\bar{r}_a)^2}\times\sqrt{\sum_{i\in(\mathrm{I}_a\cap\mathrm{I}_u)}(r_{u,i}-\bar{r}_u)^2}} \quad (1)$$

The PCC weight value, $w_{a,u}^{PCC}$, is calculated using the ratings of active (a) user $r_{a,i}$ and the ratings of prospective neighbor (u), $r_{u,i}$, who has commonly rated the item of interest, i. The overall rated items of the active user, I_a and the other user items I_u are also utilized for the intersection subset in the denominator, where each rating deviation from the overall rating mean ($\bar{r}$) is employed.

The calculated weight is then utilized in Equation 2, as it stands for the numerical prediction calculation. This will give to obtain rating value, which is to be then checked in terms of the performance comparison concerning the actual rating value. Therefore, the weight parameter in the equation is quite crucial to decide the prediction; even more, there can be an enhancement over w. Thus, a *significance weighting* (SW) is a method to highlight the correlation between two users if there is any other inference between the two of them.

$$p_{a,i} = \bar{r}_a + \frac{\sum_u\left((r_{u,i}-\bar{r}_u)\times w_{a,u}^{PCC}\right)}{\sum_u\left(w_{a,u}^{PCC}\right)} \quad (2)$$

In the next subsection, three perspectives are given on the utilization of the commonly rated item counts. During the prediction, the number of neighbors to be included in the calculation is a well-known phenomenon. In this work, the *best neighbor count* (BNC) is decided by being set parametrically starting from 5 to 100 with a 5-neighbor increment at every attempt.

2.2. Significance Weighting

Significance weighting can be thought of as a constant multiplication for the calculated weight as the *user-user* correlation. This constant is denoted as μ and given in Equation 3. The multiplication constant, μ, can be based on a static or dynamic approach. In this work, we group these two perspectives, where the first is a static multiplication based on a predefined value, α. Then, the dynamic approach is proposed to see the effect of multiple commonly rated user relations.

$$w_{a,u}^{sw} = \mu \times w_{a,u} \quad (3)$$

All figures in the following subsections are based on the real data (rounded to a 3-digit fraction) obtained from the *15th* active user and the *18th* item test pair (*a*=15, *i*=18, for all *u* values in the randomly folded train-test sets).

2.2.1. Static Multiplier

In the static multiplication, each co-rated item count of the neighbors is processed with an only constant, α. As shown in Equation 4, α is applied to CIC with a condition (Herlocker et al., 2017). In this work, we set α = 10, 25, 50, 75, 100 for our parametric tests. All the weights in progress free from their values are processed with stable α.

$$\mu = \begin{cases} \dfrac{|\mathrm{I}_a\cap\mathrm{I}_u|}{\alpha} & if\ |\mathrm{I}_a\cap\mathrm{I}_u| < \alpha \\ 1 & otherwise \end{cases} \quad (4)$$

In Figure 1, the real example of the static multiplier is shown. Each CIC between a and u is considered together with the static α, which is then processed for signified weight, $w_{a,u}^{sw}$.

$w_{a,u}$ ↓	u	$CIC_{a,u}$
0.535	585	5
0.273	59	41
0.154	1	32
0.136	207	37
0.060	181	88
-0.315	385	16

(a)

→

$w_{a,u}^{sw}$	u
0.054	585
0.224	59
0.099	1
0.100	207
0.060	181
-0.101	385

(b)

→

$w_{a,u}^{sw}$ ↓	u
0.224	59
0.100	207
0.099	1
0.060	181
0.054	585
-0.101	385

(c)

Figure 1. Example of applying multiplier α (=50), (a) Sorted original PCC weights, (b) Significance weighted (α applied) PCC weights, (c) Sorted significance weighted (α applied) PCC weights.

2.2.2. Dynamic Multiplier

In the dynamic approach, instead of a constant predefined α, an inference-based technique is performed during the prediction calculation. By considering all neighbors in the co-rating list, the mean value is obtained as $\frac{\sum CICs}{|CICs|}$. Instead of taking the exact maximum depending on a single value, the adaptive solution is preferred. Thus, the mean is doubled by treating all elements in a vector. Then, it is normalized with a fraction that is an intentional parameter, namely β, to show the effect of different mappings by setting β = 1/4, 1/3, 1/2, 2/3, 3/4 during our experiment. With this approach, a generalized α is obtained fitting into the current values of *a*,*i* pair. Then, the same procedure in Equation 4 is applied to all weights from neighbors. In Equation 5, the calculation of the aforementioned general solution is shown as it is rounded to the nearest integer as either $\left\lfloor \beta \times 2 \times \frac{\sum CICs}{|CICs|} \right\rfloor$ or $\left\lceil \beta \times 2 \times \frac{\sum CICs}{|CICs|} \right\rceil$.

$$\alpha \approx \left(\beta \times 2 \times \frac{\sum CICs}{|CICs|} \right) \quad (5)$$

In Figure 2, the example from the real dataset is shown.

(a)

$w_{a,u}$ ↓	u	$CIC_{a,u}$
0.535	585	5
0.273	59	41
0.154	1	32
0.136	207	37
0.060	181	88
-0.315	385	16

→

(b)

$w_{a,u}^{sw}$	u
0.072	585
0.273	59
0.133	1
0.136	207
0.060	181
-0.136	385

→

(c)

$w_{a,u}^{sw}$ ↓	u
0.273	59
0.136	207
0.133	1
0.072	585
0.060	181
-0.136	385

Figure 2. Example of applying the multiplier β = 1/2 → Equation (5) → α = 37, (a) Sorted original PCC weights, (b) Significance weighted (β applied) PCC weights, (c) Sorted significance weighted (β applied) PCC weights.

2.2.3. Direct CIC Multiplier

Last but not least, a pure CIC-based approach without an additional operation is applied apart from the above. The multiplier constant is directly taken as $\mu = |I_a \cap I_u|$ (Bellogín et al., 2014; Raeesi and Shajari, 2012). The bright side of the CIC usage is more than the calculation simplicity of it. In the first and the second approaches, the CIC as a threshold is considered with a further normalization, wherein the second, one more adaptive solution is designed with the overall CICs. However, it is experimentally proved in the next section that CIC between users, neither with a further normalization nor with the mean of all intersections, gives the top solution. Especially with the performance in real-time systems, this approach as a single expander of weights works well. In Figure 3, an example of the direct CIC multiplication is shown.

(a)

$w_{a,u}$ ↓	u	$CIC_{a,u}$
0.535	585	5
0.273	59	41
0.154	1	32
0.136	207	37
0.060	181	88
-0.315	385	16

→

(b)

$w_{a,u}^{sw}$	u
2.676	585
11.212	59
4.932	1
5.025	207
5.276	181
-5.047	385

→

(c)

$w_{a,u}^{sw}$ ↓	u
11.212	59
5.276	181
5.025	207
4.932	1
2.676	585
-5.047	385

Figure 3. Example of applying the pure CIC multiplier, (a) Sorted original PCC weights, (b) Significance weighted (CIC applied) PCC weights, (c) Sorted significance weighted (CIC applied) PCC weights.

3. Tests and Results

In this section, test results of *significance weighting* methods are given comparatively. For each user, five rated items are randomly chosen to be tested. At each test, the dataset is divided into ten folds stochastically, and tests are repeated 100 times. On each test, prediction values are computed distinctively for the same *train-test* set couples used in the compared methods for a fair analogy. Predicted values are labeled as *liked* or *disliked* depending on whether being greater or less than 3.5 of 5-scale ratings. Then, actual ratings and calculated results are processed for binary analyses on behalf of four renowned performance metrics given in Figure 4.

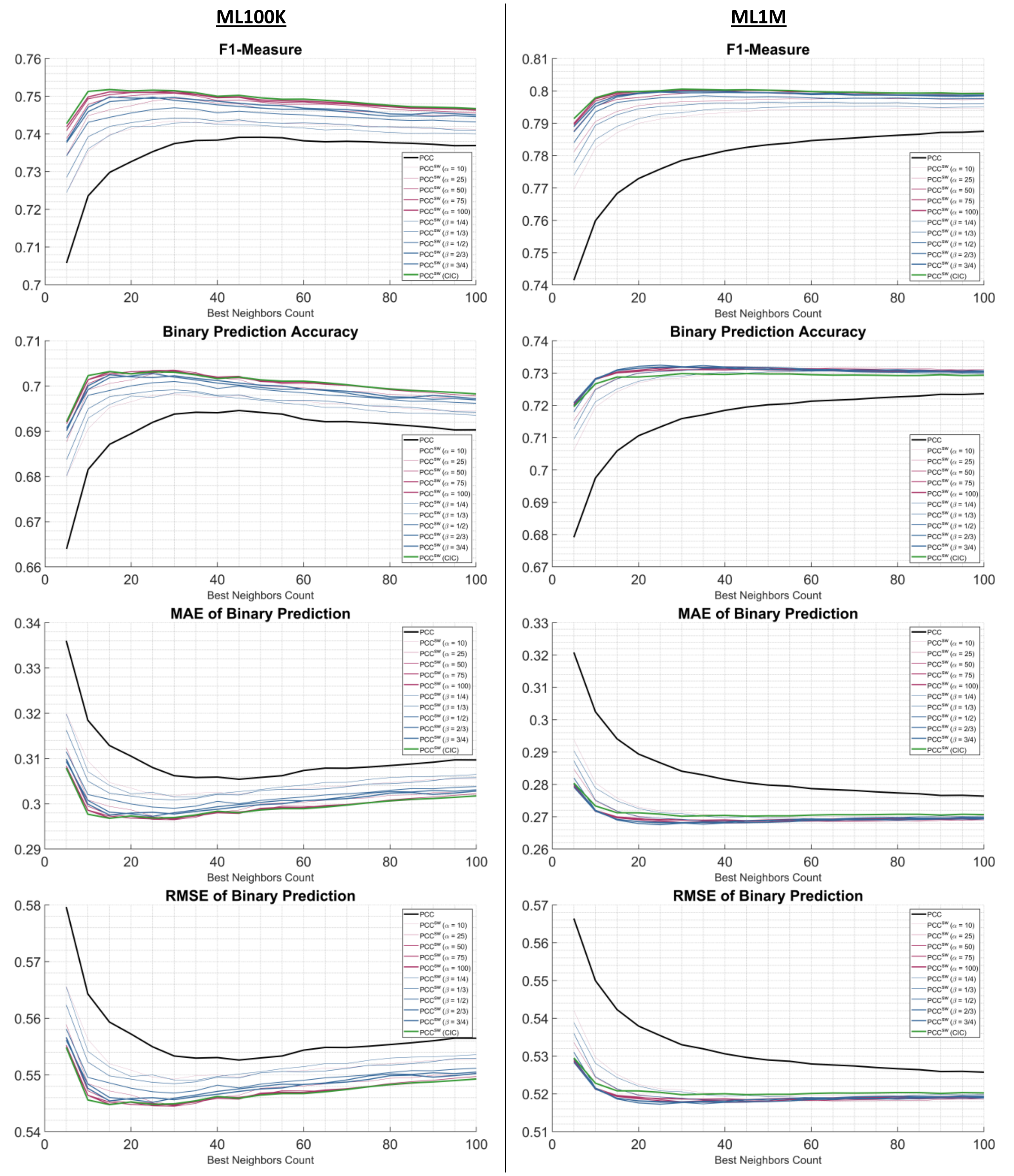

Figure 4. Comparative test results of SW methods over PCC taking all μ-based approaches. The results are given as the average of all individual tests.

The plots in Figure 4 show that the standard approach without SW (line in black color) falls behind the ones with SW. Focusing on α (lines in red color) and β (lines in blue color) parameters, the enhanced performance results are recorded for their increased values within the approach that brings less erroneous results. Besides, the pure CIC-based method (line in green color) outperforms dominantly in error metrics for the `ML100K`. The pure CIC-based method is recommended to be applied with a decreased number of neighbors when there is a large-sized dataset in use. Even though the results related to methods concerning α and β vary in different metrics, the pure CIC-based approach outperforms in the *F1-measure* that supplies compound information holding both *precision* and *recall* metrics together inside.

4. Conclusion

In RS science, there are loads of efforts to increase recommendation efficiency using different methods. In this work, we have shared the observations related to the three approaches for correlation weight significance. Especially for the real-time systems, the less complicated but higher performable approaches are required during the correlation calculation and prediction measurement. Therefore, we perform three different approaches of SW. Detailed experiments in the previous section have shown that the pure CIC method gives indicative results, especially for the ML100K dataset. In addition to the simple computation facility of pure CIC SW multiplier, satisfactory results are obtained. In a small set of the neighborhood, the acceptable results are gathered. For future work, the extensive performance metrics of CIC-based SW methods can be performed.

References

Ahmad, S. & Afzal, M. T. (2020). Combining metadata and co-citations for recommending related papers. *Turkish Journal of Electrical Engineering and Computer Sciences* 28 (3), 1519–34.

Aiolli, F. (2013). Efficient top-N recommendation for very large scale binary rated datasets.” *RecSys 2013 - Proceedings of the 7th ACM Conference on Recommender Systems, Hong Kong, China*, 273–280.

Aygun, S. & Okyay, S. (2015). Improving the Pearson similarity equation for recommender systems by age parameter. *Adv. Information, Electron. Electr. Eng. AIEEE*, Riga, Latvia.

Bellogín, A., Castells, P., & Cantador, I. (2014). Neighbor selection and weighting in user-based collaborative filtering: a performance prediction approach. *ACM Trans. on the Web*, (8), 12.

Breese, J. S., Heckerman, D., & Kadie, C. (2013). Empirical analysis of predictive algorithms for collaborative filtering. *UAI'98, The 14th Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, USA, 43–52.

Dhawan, S., Singh, K., & Jyoti. (2015). High rating recent preferences based recommendation system. *Procedia Computer Science* (70), 259–64.

Gao, M., Fu, Y, Chen, Y, & Jiang, F. (2012). User-weight model for item-based recommendation systems. *Journal of Software* 7 (9), 2133–2140.

Ghazanfar, M. A. & Prugel-Bennett, A. (2010). Novel significance weighting schemes for collaborative filtering: generating improved recommendations in sparse environments. *DMIN’10, International Conference on Data Mining*, USA.

Harper, F. M. & Konstan, J. A. (2015). The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems* 5 (4), 1–19.

Herlocker, J. L. , Konstan, J. A., Borchers, A., & Riedl, J. (2017). An algorithmic framework for performing collaborative filtering. *ACM SIGIR Forum*.

Herlocker, J., Konstan, J. A., & Riedl, J. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval* 5 (4), 287–310.

Hong-Xia, W. (2019). An improved collaborative filtering recommendation algorithm.” *4th IEEE International Conference on Big Data Analytics, ICBDA*, Suzhou, China, 431–35.

Hwang, C. S. & Chen Y. P. (2007). Using trust in collaborative filtering recommendation. *Lect. Notes Comput. Sci., New Trends in Applied Artificial Intelligence*, (4570).

Levinas, C. A. (2014). An analysis of memory based collaborative filtering recommender systems with improvement proposals. *M.Sc. Thesis, UPC*, 1–89.

LVN, R. Wang, R., & Raj, J. D. (2014). Recommending news articles using Cosine Similarity function. *Proc. SAS Glob. Forum*, Washington, DC, USA.

Ma, H., King, I., & Lyu, M. R. (2007). Effective missing data prediction for collaborative filtering. *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval, SIGIR’07,* 39–46.

Madadipouya, K. (2015). A location-based movie recommender system using collaborative filtering. *International Journal in Foundations of Computer Science & Technology* 5 (4), 13–19.

McLaughlin, M. R. & Herlocker, J. L. (2004). A collaborative filtering algorithm and evaluation metric that accurately model the user experience. *27th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, 329–336.

Nguyen, L. V., Hong, M. S., Jung, J. J., & Sohn, B. S. (2020). Cognitive similarity-based collaborative filtering recommendation system. *Applied Sciences*, 10(12), 1–14.

Philip, S., Shola, P. B., & John, A. O. (2014). Application of content-based approach in research paper recommendation system for a digital library. *International Journal of Advanced Computer Science and Applications* 5 (10), 37–40.

Powers, D. M. W. (2007). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies* 2 (1), 37–63.

Raeesi, M. & Shajari, M. (2012). An enhanced significance weighting approach for collaborative filtering. *6th International Symposium on Telecommunications, IST 2012*, Tehran, Iran, 1165–1169.

Samad, A., Islam, M. A., Iqbal, M. A., & Aleem, M. (2019). Centrality-based paper citation recommender system. *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems* 6 (19).

Saric, A., Hadzikadic, M., & Wilson, D. (2009). Alternative formulas for rating prediction using collaborative filtering,” *Int. Symp. on Method. for Intell. Sys.*, Lecture Notes in Computer Science, Springer, 5722, 301–310.

Sheugh, L. & Alizadeh, S. H. (2015). A note on Pearson correlation coefficient as a metric of similarity in recommender system. *2015 AI and Robotics, IRANOPEN 2015*, Qazvin, Iran.

Singh, R. H., et al. (2020). Movie recommendation system using cosine similarity and KNN. *International Journal of Engineering and Advanced Technology* 9 (5), 556–559.

Weng, J., Miao, C., & and Goh., A. (2006). Improving collaborative filtering with trust-based metrics.” *Proceedings of the ACM Symposium on Applied Computing*, 1860–1864.

Zeybek, H., & Kaleli, C. (2018). Dynamic k neighbor selection for collaborative filtering. *Anadolu Universıty Journal of Science And Technology A - Applied Sciences and Engineering,* 19 (2), 303-315.

Zhang, B. & Yuan, B. (2017). Improved collaborative filtering recommendation algorithm of similarity measure. *Materials Science, Energy Technology, and Power Engineering I, AIP Conf. Proc.*, 1839.

Zhang, L. et al. (2020). Diversity balancing for two-stage collaborative filtering in recommender systems. *Applied Sciences* 10 (4), 1-16.