A Survey and Comparison of Content-Based, Collaborative Filtering, and Hybrid Recommender Systems

JOSEPH BROCK, Davidson College, USA
JAKE CARVER, Davidson College, USA
NEIL PATEL, Davidson College, USA
ANNABEL WINTERS-MCCABE, Davidson College, USA

Recommender Systems provide users with product/service recommendations in order to save time and effort maneuvering through the plethora of choices available. This work reviews content-based, collaborative filtering, and hybrid recommender systems looking at their accuracy and ability to generate recommendations across a large dataset. We find that hybrid recommenders represent a system which completes recommendations for all or nearly all users while also generating very accurate recommendations.

Additional Key Words and Phrases: Recommender Systems, Collaborative Filtering, Similarity and Prediction techniques

ACM Reference Format:
Joseph Brock, Jake Carver, Neil Patel, and Annabel Winters-McCabe. . A Survey and Comparison of Content-Based, Collaborative Filtering, and Hybrid Recommender Systems. In *Proceedings of (Recommender Systems Research).* ACM, New York, NY, USA, 9 pages.

1 INTRODUCTION

Recommender systems have been a rapidly evolving area of research since the publication of the first papers on collaborative filtering in the 1990s [1]. Today, recommender systems have many uses ranging from making recommendations on e-commerce websites based on a customer's purchase history to predicting which YouTube video a user may want to watch next. Recommender systems research has grown considerably since the age of email filtering with Tapestry [3], and remains a high-demand research area because of the many practical applications of these algorithms. Collaborative filtering [6], one of the most popular methods used in recommender systems, matches users with similar preferences together to create personalized recommendations based on the preference data from the group of similar users, or neighborhood. While neighborhood-based collaborative filtering is widely accepted as an effective method to analyze user data and create predictions [6], there is research that shows advantages and disadvantages of both user-based and item-based algorithms, as well as content-based algorithms [9]. An alternative approach to neighborhood-based collaborative filtering are latent factor models, such as matrix factorization which "characterizes both items and users by vectors of latent factors inferred from item rating patterns" [7]. Content-based methods "select the right information for the right people by comparing representations of content contained in the document to representations of content that the user is interested in" [6]. Content-based recommendation algorithms include term frequency–inverse document frequency (TF-IDF) and feature encoding. The purpose of our work is to compare and contrast different recommender

Manuscript submitted to ACM

system algorithms, including user-based and item-based collaborative filtering, TF-IDF, feature encoding, matrix factorization methods with stochastic gradient descent (SGD) and alternating-least-squares (ALS), and a hybrid recommender algorithm using accuracy and coverage as our metrics for comparison. We evaluate our models on movie preference data from the MovieLens dataset [5]. Our research hopes to determine which type of algorithm and, if applicable, what associated combination of parameter settings yields the highest quality recommendations.

2 RELATED WORK

One of the first recommender systems of the 1990s is Tapestry [3], an email filtering system that uses email annotations to gain insight into a user's preferences and filter out irrelevant emails. Tapestry uses a hybrid recommender system that incorporates collaborative filtering from user annotations as well as content-based filtering from emails' textual content. This work [3] is regarded as one of the first collaborative filtering research papers available. Another early example of published recommender systems research is [11]. GroupLens is a collaborative filtering recommender system that was designed to filter internet news articles based on received user feedback given on a scale of one to five. The GroupLens system uses weighted correlations between users who have already rated an article to create predictions [6], which is related to the similarity significance weighting in our experiments. The Tapestry and GroupLens systems were groundbreaking advances in collaborative filtering techniques and provide examples of algorithm parameters that can be changed, such as significance weight.

Multiple papers compare user-based and item-based algorithms, and components that can be adjusted in these algorithms, in ways that are relevant to our work. Herlocker et al. [6] tested variations of "similarity weight," which this paper refers to as similarity method in our research, as well as variations of similarity significance weight that we also tested. The findings of this paper emphasize the importance of using a significance weight to devalue less correlated users, as well as normalizing ratings for improved accuracy. Ning et al. [9] compare user and item-based algorithms and describe their implementations. Also discussed in [9] are threshold filtering and negative filtering for neighborhoods to improve prediction accuracy, which we implement via significance threshold. Ning et al. [9] caution that it is very difficult to isolate one "best" approach to collaborative filtering because of the variety in recommendation problems.

Matrix factorization collaborative filtering methods gained significant notoriety in the Netflix Prize competition. The team that won the "Progress Prize" in this competition to improve upon Netflix's recommender algorithm used factorization methods to improve accuracy by 8.43 percent [7]. Koren et al. [7] describe how their entry consisted of over 100 predictor sets, most of which were factorization models, because "they deliver accuracy superior to classical nearest-neighbors techniques." Zhou et al. [12] also found success in the Netflix Prize competition using matrix factorization methods. This team specifically used an alternating-least-squares approach with parallelization and achieved good results with fast runtime. While the dataset used in the Netflix Prize competition poses significantly more scalability concerns than the MovieLens dataset, these matrix factorization results show that this method is effective for media items such a movies.

Content-based methods originate from information retrieval and information filtering research [1]. One of the most popular content based methods is term frequency-inverse document frequency which involves encoding words in a vector space model to summarize the content of an item in the system [10]. Feature encoding uses features or tags for each item to create profiles and generate recommendations based on items with similar features. This method has been successful in a recommender systems scenario using word embedding techniques on textual features extracted from Wikipedia with tests run on MovieLens and DBbook data, as described by Musto et al. [8]. Word embedding serves the purpose of reducing the overall dimensions of a feature-item matrix while preserving as much "semantic nuance" of the

original matrix as possible [8]. The promising results achieved in this research on the MovieLens dataset indicate that this approach is viable for our work.

There are many ways to construct a hybrid recommender system that blends together different recommendation algorithms. Adomavicius and Tuzhilin [1] describe four ways hybrid systems can be constructed: "implementing collaborative and content-based methods separately and combining their predictions, incorporating some content-based characteristics into a collaborative approach, incorporating some collaborative characteristics into a content-based approach, and constructing a general unifying model that incorporates both content-based and collaborative characteristics." Burke [2] describes how adding a hybrid with collaborative filtering to their recommender system Entree improved performance, showing that combining recommender algorithms is an effective way to achieve peak performance.

3 THESIS

Recommender systems provide us with a way of sifting through the swaths of digital information, or informational overload, that people are faced with in the 21st century. Instead of spending valuable time, energy, and focus on trying to find good or relevant content, recommender systems can help users effectively find and consume content that they are either in search of or would take interest in.

Therefore, we strive to answer the question, what is the best recommender system configuration, in terms of the following attributes: the type of recommender system (collaborative filtering, content-based, hybrid), the type of recommendation algorithm (i.e. TF-IDF vs. FE), the method of calculating similarity between neighbors (if relevant), significance weighting (if relevant), and similarity threshold (if relevant). Since hybrid recommender systems are designed to combine the best elements of both collaborative and content-based filtering, our hypothesis is that a hybrid recommender system using Pearson similarity calculations with a significance weighting of $n/25$ and a hybrid weighting of 0.75 will likely have the best performance.

In order to measure performance, we will measure accuracy and coverage. To gauge accuracy, we will compute Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) for the 13 different variations of recommender systems. To gauge coverage, we will report each variation's percentage of users/items that can have recommendations calculated.

We will use numerous equations and formulae to create, test, and compare these recommender system variations. For collaborative filtering methods, we used one of two methods for calculating similarity - Euclidean distance or Pearson correlation. For calculating the Euclidean distance between two users' ratings for two items, we will use the general formula

$$d_{AB} = \frac{1}{(1 + \sqrt{(r_1 A - r_2 A)^2 + (r_1 B - r_2 B)^2})}$$

where r_1 and r_2 are different users and A and B are different items. We will calculate the Pearson correlation coefficient by using the general formula

$$r_{xy} = \frac{\sum((X_i - \bar{X}) \cdot (Y_i - \bar{Y}))}{\sqrt{\sum(X_i - \bar{X})^2} \cdot \sqrt{\sum(Y_i - \bar{Y})^2}}$$

where X and Y are different users and $\bar{X}$ and $\bar{Y}$ are the average ratings for similar items shared by both X and Y.

For collaborative filtering, specifically TF-IDF, we use cosine similarity which is calculated using the formula

$$\text{cosineSim}(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| * |\vec{b}|}$$

where $\vec{a}$ and $\vec{b}$ are term vectors that represent two different items. Essentially, this method compares the angle between vectors to determine their similarity or dissimilarity. For example, two vectors with an angle of 0° demonstrate maximum similarity. Likewise, 90° demonstrates no similarity or dissimilarity and 180° demonstrates maximum dissimilarity.

We decided to compute three different types of error metrics for comparing accuracy of the predictions. These are Mean Squared Error (MSE), Mean Absolute Error (MAE), and the Root Mean Square Error (RMSE). The Mean Squared Error is calculated by using the formula

$$MSE = \frac{1}{n} \cdot \sum_{i=1}^{n} (Y_i - \hat{Y})^2$$

where n is the number of ratings, Y_i is the actual rating, and $\hat{Y}$ is the predicted rating.

The Mean Absolute Error is calculated by using the formula

$$MAE = \frac{1}{n} \cdot \sum_{i=1}^{n} |(p_i - r_i)|$$

where p is the predicted rating, r is the actual rating, and n is the number of observations.

The final type of error that our group used is the Root Mean Square Error. RMSE can be calculated using the formula

$$RMSE = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} (p_i - r_i)^2}$$

where once again p is the predicted rating, r is the actual rating, and n is the number of observations.

Finally, we decided to calculate coverage. According to [6], coverage is "the measure of the percentage of items for which a recommendation system can provide predictions." For this metric, we calculated coverage by using the equation

$$\text{coverage} = \frac{\text{len(SE List)}}{100,000}$$

where 100, 000 is the number of ratings found within the data set and len(SE List) is the number of ratings being predicted given the item has an actual rating.

4 EXPERIMENTAL DESIGN

For our trials, we will testing the performance of different recommender system variations on the MovieLens 100K dataset. Movie ens is a movie recommender system based on collaborative filtering techniques that is made and maintained by the University of Minnesota's GroupLens research lab. This particular dataset contains "100, 000 ratings $(1 - 5)$ from 943 users on 1682 movies", where "each user has rated at least 20 movies." [4]

For sake of simplicity and clarity, we have reduced the number of variations so that we will compare 17 recommender systems - 5 of which are content-based, 4 are collaborative filtering, and 8 are hybrid. More specifically there are 4 Term Frequency–Inverse Document Frequency (TF-IDF) variations, 1 Feature Encoding (FE) variation, 1 User-Based Collaborative Filtering variation (U-CF), 1 Item-Based Collaborative Filtering (I-CF) variation, 1 Matrix Factorization - Stochastic Gradient Descent (MF-SGD) variation, and 1 Matrix Factorization - Alternating-Least-Squares (MF-ALS) variation, and 8 Hybrid (HBR) variations.

- TF-IDF (similarity threshold > 0)
- TF-IDF (similarity threshold > 0.3)
- TF-IDF (similarity threshold > 0.5)
- TF-IDF (similarity threshold > 0.7)
- FE
- U-CF (Pearson, threshold=0, weight=n/25)
- I-CF (Pearson, threshold=0, weight=n/25)
- MF-SGD (K=20, alpha=.02, beta=.002)
- MF-ALS (f=200, Lambda=0.0001, iterations=20)
- HBR (Euclidean, hybrid weighting = 1, n = 100)
- HBR (Pearson, hybrid weighting = 1, n = 100)
- HBR (Euclidean, hybrid weighting = 0.75, n = 100)
- HBR (Pearson, hybrid weighting = 0.75, n = 100)
- HBR (Euclidean, hybrid weighting = 1, n = 1682)
- HBR (Pearson, hybrid weighting = 1, n = 1628)
- HBR (Euclidean, hybrid weighting = 0.75, n = 1682)
- HBR (Pearson, hybrid weighting = 0.75, n = 1682)

Our metrics to evaluate the performance of each recommender system variation will be accuracy and coverage. These two metrics will be taken into consideration to ensure that our best recommender system provides high-quality recommendations for a significant number of items. As discussed in the previous section, we have chosen Mean Square Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE) as our three accuracy metrics, and coverage will be calculated by taking the ratio of the number of ratings produced by a recommender system to the number of ratings found in the dataset.

In order to test and compare the performance of each recommender system variation, we will conduct experiments using leave-one-out-cross-validation (LOOCV). LOOCV enables us to estimate the performance of a recommender system variation on test data, or data other than its training data. In short, LOOCV iterates through the training data

and "leaves out" one data point from the training data to act as the test data, trains a model using the training data, uses the model to make a prediction about the test data, and computes accuracy metrics between the actual value and the predicted value of the test data. After all iterations have been run, we then compute the average of each accuracy statistic to describe the performance of the recommender system variation.

5 RESULTS

Our TFIDF results demonstrate that, across error metrics, both accuracy and coverage decrease as the sim threshed increases. Therefore, the best TFIDF model had a threshold of 0, yielding an MAE of .817, and we used this model to inform our hybrid recommender.

The hybrid model with the lowest MAE used a weight of 1, Pearson similarity, and 1682 neighbors, with a MAE of .777; the worst case used a weight of 1, Distance similarity, and 100 neighbors. Throughout all of the experiments, models with either Pearson similarity and/or 1682 neighbors did especially well. The coverage for all of the cases was remarkably high, with all of the coverage measurements exceeding 99%. Another key detail of our analysis was that, out of the parameters used in our hybrid model, we found similarity method in particular to be statistically significant. Similarity method had a t-stat of 2.427247 and a p-value of 0.093560068.

Sim Threshold	MSE	RMSE	MAE	Coverage
0	1.048402187	1.023915127	0.816840615	0.99743
0.3	1.054067734	1.026678009	0.818359977	0.99716
0.5	1.081041089	1.039731258	0.824247656	0.99066
0.7	1.170108427	1.081715502	0.845008056	0.95204

Fig. 1. TFIDF test cases with varied sim thresholds.

Hybrid Weight	Sim Method	Neighbors	MSE	RMSE	MAE	Coverage
1	Pearson	1682	0.940487837	0.969787522	0.776553595	0.99999
0.75	Pearson	1682	0.952978108	0.976205976	0.781371906	0.99999
0.75	Pearson	100	0.952978108	0.976205976	0.781371906	0.99999
1	Distance	1682	1.009490792	1.00473419	0.803280806	1
0.75	Distance	1682	1.012471337	1.006216347	0.804357525	1
0.75	Distance	100	1.012471337	1.006216347	0.804357525	1
1	Pearson	100	1.023844245	1.011851889	0.807916349	0.99876
1	Distance	100	1.030666277	1.015217354	0.808368936	0.9981

Fig. 2. Hybrid recommender test cases, all with TF-IDF sim thresholds of 0. Results sorted by MAE.

The best hybrid model fares well in some aspects, specifically coverage. For example, the feature encoding model has an MAE of 0.817, significantly higher than the hybrid model. Feature encoding also has slightly worse coverage, at 99.7 percent as opposed to 99.9 percent for the hybrid.

However, there are still models with lower errors. Some alternating-least-squares models technically have lower MSEs, but they all have terrible distance measures and low counts, which likely indicate overfitting. The ALS model with the lowest distance and count has a clipped MSE of .75, which is much lower than the .94 MSE of the hybrid model. Similarly, the best stochastic gradient descent model, with K = 20, alpha = .02, and beta = .002, has an MSE of .363, which again we suspect may be due to overfitting.

Both user-based and item-based models have superior accuracy to the hybrid model but slightly worse coverage. The best user-based model in terms of coverage used a threshold of 0, weight of n/25, and Pearson similarity. The coverage of this model is 99.133% and its MAE is .72772. The best item-based model has a threshold of 0, a significance weight of n/25, and Pearson similarity. The coverage of this model is a slightly worse 96% and its MAE is .63174. While the MAE measures of these models is lower than the hybrid model, there is a trade-off between accuracy and a small drop in coverage.

6 DISCUSSION

Recommender	MAE	Coverage	AC
Hybrid	0.776554	0.99999	0.7765618
TFIDF	0.816841	0.99716	0.8191674
Feature Encoding	0.817784	0.99743	0.8198911
ALS	0.67648	1	0.67648
SGD	0.47438	0.99693	0.4758408
Item-Based	0.63174	0.96612	0.6538939
User-Based	0.72772	0.99133	0.7340845

Fig. 3. Overview of results for the best models in each of our recommender categories. AC is the MAE divided by coverage.

While the hybrid model did exceed in coverage, the primary purpose of its design, its accuracy was slightly disappointing. After analysis, the MAE of the hybrid model makes sense because it fills in the missing recommendations of the TFIDF model with item-based recommendations. It follows that the MAE of the hybrid model (.777) should be somewhere in between that of TFIDF (.817) and item-based (.632).

Choosing the "best" model out of our recommenders is difficult. In Figure 3, we can see the trade-off between accuracy and coverage laid out by the AC metric, which is MAE divided by coverage. As mentioned in the Results section, it is hard to judge the efficacy of ALS and SGD despite their fantastic AC scores because they very well may be overfitting the data. For comparison, the second-best SGD model, which had the second lowest distance measure, also had an unbelievably low MSE of .055. Moreover, leave-one-out cross validation was not conducted on these models. Instead, these error metrics were generated from the training data used to generate these models. It is difficult to compare these results to those of hybrid as there may be some bias in the SGD and ALS error metrics.

From the remaining models, the AC score of the item-based recommender stands above the rest. However, its coverage is by far the worst out of all the algorithms tested. 3.4% of recommendations unable to be generated may not be acceptable for a movie recommendation system, regardless of how high quality the other 96.6% of the recommendations are. Conversely, while the hybrid model has near-perfect coverage, its MAE and AC are not the highest of the experiments, indicating consistent recommendations but not exceptional performance. The choice between the item-based and hybrid recommender comes down to a decision between overall accuracy or complete sets of recommendations being returned. For practical purposes, such as commercial use by Netflix or Amazon, high coverage that allows close to all users to receive recommendations would take priority.

A future fix to these problems could be to switch the functions of the TFIDF and item-based components of the hybrid recommender. In other words, the hybrid recommender would attempt item-based recommendation first, and

then use TFIDF to fill in the missing recommendations. Item-based is the most accurate model, so it may be sensible to try to use the results from that model first before using TFIDF, a higher-coverage model, as a safety net.

7 CONCLUSION

After conducting our hypothesis tests we can conclude that our data is statistically significant. P-value, F-statistic, and T-tests all indicated that the number n neighbors and the similarity method were significant variables in affecting our hybrid recommender. For our TFIDF and FE results, similarity threshold and significance weighting were also significant variables. Our results indicate that the use of Pearson correlation and a value of n equal to the full number of items in the dataset are correlated to lower MSE across our tests. Additionally, a similarity threshold of greater than 0 is optimal. For our hybrid system, Pearson correlation and a hybrid weight of 1 (augmenting all TFIDF values with item-based values) were the optimal parameters. However, our hypothesis testing indicated that many of our variables in a system with multiple parameters only slightly exceeded thresholds to be considered statistically significant. Changing one parameter to a different value was unlikely to result in a large change, but as all parameters moved to more optimal values a large increase in accuracy and coverage could be achieved.

Our hybrid system produced lower error than either TFIDF or FE systems and was more successful in achieving high or perfect coverage than either Matrix Factorization. This system especially showed promise with the ability to generate perfect or near perfect coverage, important for an end user who cares more that their recommendation is generated than if it is 5% less accurate than another prediction. Future work improving this system could include testing with other calculations of similarity, utilizing similarity thresholds close to zero but still larger than 0, or testing pipelining with another recommender.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin. 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (2005), 734–749. https://doi.org/10.1109/TKDE.2005.99

[2] Robin Burke. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction* 12 (11 2002). https://doi.org/10.1023/A:1021240730564

[3] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. 1992. Using Collaborative Filtering to Weave an Information Tapestry. *Commun. ACM* 35, 12 (Dec. 1992), 61–70. https://doi.org/10.1145/138859.138867

[4] GroupLens. [n.d.].

[5] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages. https://doi.org/10.1145/2827872

[6] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. 1999. An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Berkeley, California, USA) *(SIGIR '99)*. Association for Computing Machinery, New York, NY, USA, 230–237. https://doi.org/10.1145/312624.312682

[7] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37. https://doi.org/10.1109/MC.2009.263

[8] Cataldo Musto, Giovanni Semeraro, Marco de Gemmis, and Pasquale Lops. 2016. Learning Word Embeddings from Wikipedia for Content-Based Recommender Systems, Vol. 9626. 729–734. https://doi.org/10.1007/978-3-319-30671-1_60

[9] Xia Ning, Christian Desrosiers, and George Karypis. 2015. *A Comprehensive Survey of Neighborhood-Based Recommendation Methods.* Springer US, Boston, MA, 37–76. https://doi.org/10.1007/978-1-4899-7637-6_2

[10] M. Pazzani and Daniel Billsus. 2007. Content-Based Recommendation Systems. In *The Adaptive Web.*

[11] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work* (Chapel Hill, North Carolina, USA) *(CSCW '94)*. Association for Computing Machinery, New York, NY, USA, 175–186. https://doi.org/10.1145/192844.192905

[12] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. 2008. Large-Scale Parallel Collaborative Filtering for the Netflix Prize. In *Algorithmic Aspects in Information and Management*, Rudolf Fleischer and Jinhui Xu (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 337–348.