

A Lasso-based Collaborative Filtering Recommendation Model

Hiep Xuan Huynh¹

College of Information and Communication Technology
Can Tho University (CTU)
Can Tho City, Vietnam

Vien Quang Dam²

Faculty of Information Technology
Can Tho Vocational College (CTVC)
Can Tho City, Vietnam

Long Van Nguyen³

Information & Communication Technology Department
Ministry of Public Security (MPS)
Hanoi, Vietnam

Nghia Quoc Phan^{4*}

Assessment Office
Tra Vinh University (TVU)
Tra Vinh Province, Vietnam

Abstract—This paper proposes a new approach to solve the problem of lack of information in rating data due to new users or new items, or there is too little rating data of the user for items of the collaborative filtering recommendation models (CFR models). In this approach, we consider the similarity between users or items based on the lasso regression to build the CFR models. In the commonly used CFR models, the recommendation results are built only based on the feedback matrix of users. The results of our model are predicted based on two similarity calculated values: (1) the similarity calculated value based on the rating matrix; (2) the similarity calculated value based on the prediction results of the Lasso regression. The experimental results of the proposed models on two popular datasets have been processed and integrated into the recommenderlab package showed that the suggested models have higher accuracy than the commonly used CFR models. This result confirms that Lasso regression helps to deal with the lack of information in the rating data problem of the CFR models.

Keywords—UBCF-LASSO; IBCF-LASSO; Lasso regression

I. INTRODUCTION

The recommendation models [1][2][7] are widely used in the fields of commerce, education, and entertainment. It saves users time when searching for needed information easily and quickly based on transaction data or rating data of users in the past. For example, Facebook displays ads related to keywords that users search for; YouTube automatically jumps to clips like the clips the user just watched. The Amazon sales site has been very successful in using the recommendation model [8]. Products are rated by users on a scale of 1 to 5 when they shop online. Transactional information and customer evaluations for products are collected to assist in increasing the accuracy of the recommender model. Therefore, every time a customer visits the website, they are always suggested products that are predicted from previously collected data.

The above example shows that the recommender system plays a very important role in e-commerce [4] and many areas of daily life around us. Therefore, building a good recommendation system to find products according to user requirements is the desire of not only researchers but also an

important investment in the development strategy of each company.

Many collaborative filtering recommendation models have been successfully applied in the field of online e-commerce [4][7][10]. It is also considered as one of the effective solutions to solve the problem of information explosion for online systems with a rapidly increasing number of users. These recommender models recommend users what products they need to buy based on the results of their ratings on the products. To predict products according to user preferences based on a rating matrix, collaborative filtering recommendation models perform calculations based on the efficient exploitation of statistical methods and data mining techniques [13]. However, collaborative filtering recommendation models still face objective problems that need further research and improvement. It is a matter of lack of information in rating data of users due to new users or new items, or too little rating data of users for items.

In this study, we propose a new method to improve the accuracy of collaborative filtering recommendation models due to the lack of user rating information (new users, new items, and sparse data) by looking at the correlation relationship between users or products based on Lasso regression. In particular, the proposed models are built based on two similarity methods: the similarity method is built based on Lasso regression [15] and the similarity method is built based on rating data [13].

This study is structured into six sections. Section one includes an overview of the recommendation models and research questions. Section two briefly reviews the linear Lasso regression. Section three presents the steps to calculate the similarity based on the linear Lasso regression. Section four proposes the collaborative filtering recommendation models based on similarity Lasso regression. Section five shows the experimental results of the proposed models on two popular datasets (MovieLens and Jester5k) integrated into the recommenderlab package [13]. The concluding section presents a summary of the results achieved.

*Corresponding Author.

II. COLLABORATIVE FILTERING

A recommendation system is based primarily on a set of users, a set of items, and a set of user ratings based on those data items; represented in a matrix (see Fig. 1).

Collaborative filtering [7][1][13] is the process of determining the missing values / ratings from a rating matrix. Let $U = \{u_1, u_2, \dots, u_m\}$ be a set of m users and $I = \{i_1, i_2, \dots, i_n\}$ be a set of n items obtained from a certain supermarket.

Ratings [13][7][1] are stored in a $m \times n$ user-item rating matrix $R = (r_{jk})$ where each row represents a user u_j with $1 \leq j \leq m$ and columns represent items i_k with $1 \leq k \leq n$. r_{jk} represents the rating of user u_j for item i_k . The values of r_{jk} is rated from 1 to q (for example $q = 5$) or missing. From this point of view recommender systems solve a regression problem to predict missing values [13] (see Fig. 1).

The first approach is to predict the rating value for a user – item pair. Suppose we have a dataset about the interest of m users (users) in n products (items), corresponding to an $m \times n$ matrix, where the element in row i , column j represents the price value the rating of the i th user on the j th item. Our work is to fill the empty values in the matrix, in other words, we will predict the user's rating value on the items that the user has not rated.

In fact, we don't really need to predict all user ratings on items to make recommendations to users. Instead, we just need to suggest k most suitable products (items) to the user or determine k users that best match the item. To find the most suitable k items for the user, we need to calculate the "distance/similarity/compatibility" to find the k neighbors (k – neighborhood) that best match the user u_a (see Fig. 2).

Collaborative filtering algorithms are also divided into memory-based and model-based [18].

A. User-based Collaborative Filtering

User-based recommendation uses the similarity of a group of users' purchasing or purchasing patterns to predict what items that user will buy or choose.

B. Item-based Collaborative Filtering

Item-based recommendation will use similarity in the purchase relationship of items to predict which items the user will buy or choose.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
u_1	?	4.0	4.0	2.0	1.0	2.0	?	?
u_2	3.0	?	?	?	5.0	1.0	?	?
u_3	3.0	?	?	3.0	2.0	2.0	?	3.0
u_4	4.0	?	?	2.0	1.0	1.0	2.0	4.0
u_5	1.0	1.0	?	?	?	?	?	1.0
u_6	?	1.0	?	?	1.0	1.0	?	1.0
	:	:	:	:	:	:	:	:
u_a	?	?	4.0	3.0	?	1.0	?	5.0
\hat{r}_a	3.5	4.0			1.3		2.0	

Fig. 1. User Rating Matrix (URM).

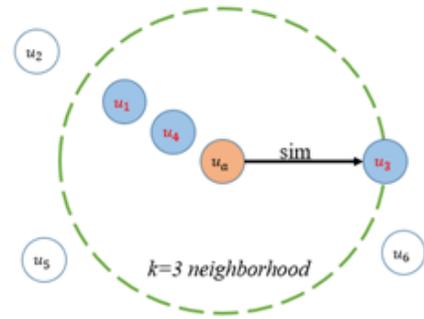


Fig. 2. k -neighborhood ($k=3$).

III. LASSO REGRESSION

Lasso Regression (Least Absolute Shrinkage Selection Operation) [15] [19] was first proposed by Robert Tibshirani in 1996.

$$\hat{\beta}^{lasso} = \operatorname{argmin}_{\beta} \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 \quad (1)$$

$$\text{subject to } \sum_{j=1}^p |\beta_j| \leq t \quad (2)$$

In Lagrangian form [19]:

$$\hat{\beta}^{lasso} = \operatorname{argmin}_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j| \right\} \quad (3)$$

This is a good method to narrow predictors by removing unimportant attributes based on the absolute value of the weights of the regression model by the following formula:

$$\hat{\beta}(\lambda) = \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij})^2 + \lambda \sum_{j=1}^p |\beta_j| \quad (4)$$

The purpose of the Lasso regression model is to minimize prediction errors [6][15]. In practice, the tuning parameter λ controls the strength of the penalty, when λ is large enough some of the coefficients β are exactly zero, in this way can reduce the dimensionality of the model. The larger the parameter λ , the more coefficients β shrink to zero.

There are many advantages to using Lasso regression, first, it can provide a good prediction accuracy because shrinking and removing the coefficient can reduce the variance without significantly increasing the standard deviation. This is especially useful when we have a small number of observations and many attributes.

In addition, Lasso regression helps to increase the interpretability of the model by removing variables that are not related to the explanatory variable, which also avoids the overfitting of the model. So, Lasso regression is a good choice to build a recommendation model that avoids underfit or overfit when we choose too few or too many variables in the model.

IV. LASSO-BASED SIMILARITY

As described in the previous section, Lasso regression could build a good predictive model for large datasets by selecting important attributes and removing unimportant attributes in the dataset. This method can give good prediction

results on a sparse matrix, and it is suitable for overcoming the weakness of the lack of rating data of the collaborative filtering recommender model.

Therefore, the function to calculate the user similarity matrix based on Lasso regression is built as follows:

LUS (rating matrix; newdata)

Input: rating matrix, newdata
Output: URM (user result matrix)

Begin

Step 1: Building Lasso regression based on rating matrix
User_Lasso = Lasso (rating matrix);
Step 2: Using Lasso regression to build user similarity matrix
For each row of the rating matrix

Begin

Value = predict (User_Lasso, newdata)
URM = cbind (URM, Value)

End

Step 3: return (URM)

End

Like building a user similarity matrix, the function to calculate the item similarity matrix based on Lasso regression is built as follows:

LIS (rating matrix; newdata)

Input: rating matrix, newdata
Output: IRM (item result matrix)

Begin

Step 1: Building Lasso regression based on rating matrix
Item_Lasso = Lasso (rating matrix);
Step 2: Using Lasso regression to build item similarity matrix
For each column of the rating matrix

Begin

Value = predict (Item_Lasso, newdata)
IRM = cbind (IRM, Value)

End;

Step 3: return (IRM)

End;

V. RECOMMENDATION FRAMEWORK

This section presents the content of two proposed models based on Lasso regression: the UBCF-LASSO model and the IBCF-LASSO model. The UBCF-LASSO model is designed based on the user similarity matrix integrated between the Lasso-User-Similarity (LUS) matrix calculated by the linear regression Lasso and the user similarity matrix calculated from the rating data in the way of the traditional UBCF models [12][13]. Similar to the above approach, the IBCF-LASSO Model is designed based on the integrated item similarity matrix between the LIS matrix calculated by the linear regression Lasso and the item similarity matrix calculated from the rating data in the way of the traditional IBCF models [12][13].

A. UBCF-LASSO

The UBCF-LASSO model is designed with two input parameters: the user's rating matrix for the items: $R_{U \times I} = [r_{i,j}]$ with $U = \{u_1, u_2, \dots, u_n\}$ is n users; $I = \{i_1, i_2, \dots, i_m\}$ is m items, and u_a is a user who needs recommendation.

The UBCF-LASSO model has an overall block diagram design structure as follows.

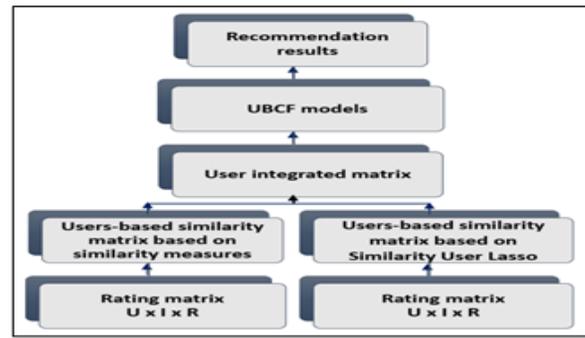


Fig. 3. UBCF-LASSO Model.

Fig. 3 shows the implementation steps of the UBCF-LASSO model. In the first step, the model builds a user similarity matrix from a rating matrix based on the similarity measures. In the second step, the model continues to build the user similarity matrix from the rating matrix based on the LUS. In the third step, the model builds the integration matrix by adding two user similarity matrices from the two steps above. In the last step, the model uses the integration matrix to predict the items to recommend to the user u_a (who needs recommendation).

B. IBCF-LASSO

The IBCF-LASSO model is designed similar to the UBCF-LASSO model with two input parameters: the user's rating matrix for the items: $R_{U \times I} = [r_{i,j}]$ with $U = \{u_1, u_2, \dots, u_n\}$ is n users; $I = \{i_1, i_2, \dots, i_m\}$ is m items, and u_a is a user who needs recommendation. However, when building similarity matrices, the model calculates similarity values based on item similarity.

The IBCF-LASSO model has an overall block diagram design structure as follows.

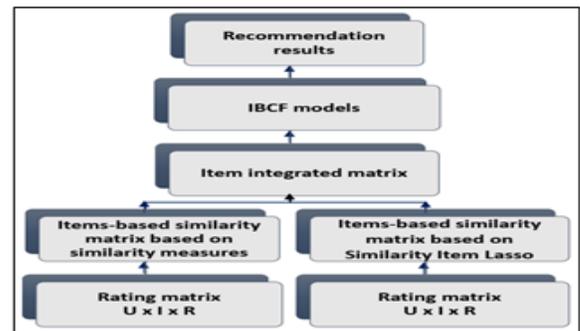


Fig. 4. IBCF-LASSO Model.

Fig. 4 shows the implementation steps of the IBCF-LASSO model. In the first step, the model builds an item similarity matrix from a rating matrix based on the similarity measures. In the second step, the model continues to build the item similarity matrix from the rating matrix based on the Lasso-Item-Similarity (LIS). In the third step, the model builds the integration matrix by adding two item similarity matrices from the two steps above. In the last step, the model uses the integration matrix to predict the items to recommend to the item i_a (item needs recommendation).

VI. EXPERIMENT

A. Datasets

The experimental part is deployed on two popular datasets for research on collaborative filtering models, which are the MovieLens dataset (100k) [3][13] and the Jester5K dataset (5k sample) [11][13]. These two datasets have been processed and integrated into the recommenderlab package [13].

The MovieLens dataset was collected through the MovieLens website during the seven-month period from September 19th, 1997 through April 22nd, 1998. The dataset contains about 100.000 ratings (1-5) from 943 users on 1664 movies. This dataset is stored in the sparse matrix format of the "realRatingMatrix" class. This matrix is similar in structure to the size of the dataset with rows equal to the number of users, columns equal to the number of movies, and nearly 7 percent of the cells of the matrix have rating values between 1 and 5 (the "null" value is "0").

The Jester5k is a dataset of the Jester Online Joke Recommendation System collected from April 1999 to May 2003. It contains a sample of 5,000 anonymous users who rated 100 jokes. This dataset is also stored in the sparse matrix format of the "realRatingMatrix" class. However, this dataset has two major differences from the MovieLens dataset. The first is that each user must have a rating for more than 30% of the total jokes. The second is that the rating value for jokes is a real number value between -10.00 and 10.00 (the "null" value is "99").

Both above datasets are randomly selected using the k-fold cross-validation technique (with k=5) [13][17]. This technique requires performing k times to evaluate the proposed models. In each evaluation, the models use one-fold as the testing set and the other k - 1 folds as the training set. This technique always makes sure that each tuple (row) has at least one occurrence in the testing set. The overall evaluation result of the proposed models is the average result of k times evaluations.

B. Tools

The experiments in this study were performed on the ARQAT tool developed in the R language by our research group [14][16]. In this ARQAT tool, we integrate the recommenderlab [13][16] and the glmnet [6][16] packages. The recommenderlab is a framework for developing and testing recommendation algorithms while the glmnet fits generalized linear and similar models via penalized maximum likelihood. This ARQAT tool also includes functions for experimental deployment such as: preprocessing of experimental data, calculating similarity matrices, installing recommendation models and methods of evaluating recommendation models.

C. MovieLens

1) *Accuracy based on model's predicted values:* The method of calculating model accuracy based on predicted values is a method of measuring the deviation between the predicted model's values and the user's actual rating values. In this study, we use three common measures: MSE (Mean Squared Error), RMSE (Root Mean Square Error), MAE

(Mean Absolute Error) [5][9][13] to compare the error value of the proposed models with the error value of traditional CFR models.

To ensure the accuracy of the results of comparing the models, we experiment on 4 models: UBCF-LASSO, IBCF-LASSO, UBCF [13], and IBCF [13] with the same training set and testing set.

The results of the comparison of error indexes (RMSE, MSE, MAE) of the four models are presented in TABLE I. This result shows that the model UBCF-LASSO (RMSE: 1.140918; MSE: 1.301695; MAE: 0.892930) has an error value lower than that of the UBCF model (RMSE: 1.146269; MSE: 1.313934; MAE: 0.902714) and the IBCF_LASSO model (RMSE: 1.320553; MSE: 1.743861; MAE: 0.944156) with a lower error value than the IBCF model (RMSE: 1.388423; MSE: 1.927719; MAE: 1.050939) on the MovieLens dataset.

2) *Accuracy based on model's recommendation results:* The method of evaluating the accuracy of the model based on the recommendation results is a method of determining the accuracy of the model by comparing the model's prediction results with the user choices on the testing set. This method uses the confusion matrix to calculate the values of the indicators: Precision, Recall, and F-measure [5][9][13]. Precision and Recall are two commonly used metrics used to evaluate recommender models.

To get the comparison results for this evaluation method, we run all four models: UBCF-LASSO, IBCF-LASSO, UBCF, IBCF on the same training set and testing set.

The results of comparing 3 indexes (Precision, Recall, and F-measure) of the 4 models are detailed in Fig. 5. This result shows that the UBCF-LASSO model has a higher Precision index than the UBCF model (Precision: 0.6 vs. Precision: 0.5) and the IBCF-LASSO model has a higher Precision index than the IBCF model (Precision: 0.5 vs. Precision: 0.4).

3) *Comparing Prec/Rec ratio of models:* As mentioned in section 2 above, Precision and Recall are two commonly used metrics to evaluate recommender models. However, in some cases when Precision and Recall are inversely proportional to each other, we can use a harmonious combination of Precision and Recall evaluating the overall efficiency of the model. Specifically.

In this study, we build a comparison chart based on the Precision/Recall ratios [13] of four models to better see the performance of the proposed models compared to other published models.

TABLE I. MODEL'S ERROR INDICATORS ON MOVIELENS

CRF models	Error indicators		
	RMSE	MSE	MAE
UBCF	1.146269	1.313934	0.902714
UBCF-LASSO	1.140918	1.301695	0.892930
IBCF	1.388423	1.927719	1.050939
IBCF-LASSO	1.320553	1.743861	0.944156

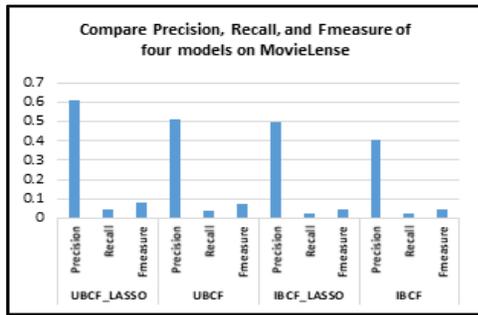


Fig. 5. Comparison Chart of Indicators: Precision, Recall, and F-measure of Four Models on MovieLense.

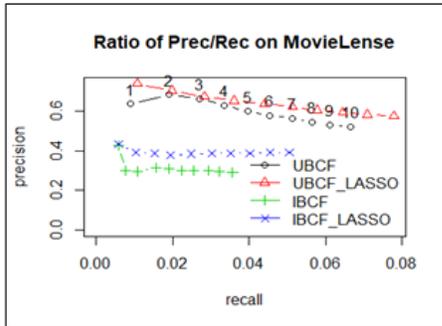


Fig. 6. Comparing the Prec / Rec Ratio of Four Models on MovieLense.

Fig. 6 shows that the UBCF-LASSO model has a better Precision/Recall ratio curve than the UBCF model and the IBCF-LASSO model also has a better Precision/Recall ratio curve than the IBCF model on the MovieLense. The result shows that Lasso regression has increased the accuracy of two models: UBCF-LASSO and IBCF-LASSO. This shows that Lasso regression is suitable to overcome the weakness of the lack of rating data of the CFR models.

D. Jester5K

1) *Accuracy based on model's predicted values:* In this section, the two proposed models continue to be evaluated for accuracy on the real number ranking dataset. The experimental content deployed on four models is like that deployed with the MovieLense dataset.

From the experimental results, we continue to calculate the error indexes (MSE, RMSE, MAE) of the two proposed models to compare with these indexes of the traditional CFR models.

The results of the comparison of error indexes of the four models are presented in TABLE II. This result shows that the model UBCF (RMSE: 5.380568; MSE: 28.950510; MAE: 4.243284) has an index of error that is higher than that of the UBCF-LASSO model (RMSE: 5.256894; MSE: 27.634931; MAE: 4.096698). Meanwhile, these indicators on both IBCF-LASSO and IBCF models are almost equal.

2) *Accuracy based on model's recommendation results:* In this evaluation, we continue to experimentally run four models on the Jester5K dataset and compare the accuracy indexes (Precision, Recall, and F-measure) of two proposed models with the accuracy indexes of two traditional CFR models.

The results of comparing the three above indicators of two proposed models and two traditional CFR models are presented in Fig. 7. This result shows that the accuracy indexes of the two proposed models are higher than those in the traditional CFR models. Especially, the precision value of the UBCF-LASSO model is higher than the precision value of the UBCF model (Precision: 0.898 vs. Precision: 0.698) and the precision value of the IBCF-LASSO model is higher than the precision value of the IBCF model (Precision: 0.647 vs. Precision: 0.447).

3) *Comparing Prec/Rec ratio of models:* Like the experimental part on the MovieLense dataset, we continue to build a comparison chart based on the Precision/Recall ratios [13] of four models to better see the performance of the proposed models compared to other published models. Fig. 8 shows that the two proposed models both have a higher Precision/Recall ratio curve than the two traditional CFR models. This result again shows that Lasso regression has increased the accuracy of the CFR models on the Jester5K dataset. This once again confirms that Lasso regression helps to deal with the lack of information in the rating data problem of the CFR models.

TABLE II. MODEL'S ERROR INDICATORS ON JESTER5K

CFR models	Error indicators		
	RMSE	MSE	MAE
UBCF	5.380568	28.950510	4.243284
UBCF-LASSO	5.256894	27.634931	4.096698
IBCF	5.844125	34.153799	4.410457
IBCF-LASSO	5.849720	34.219222	4.413152

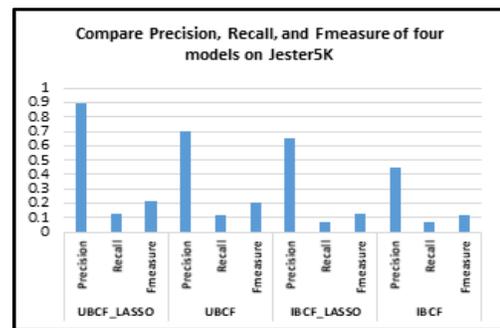


Fig. 7. Comparison Chart of Indicators: Precision, Recall, and Fmeasure of Four Models on Jester5K.

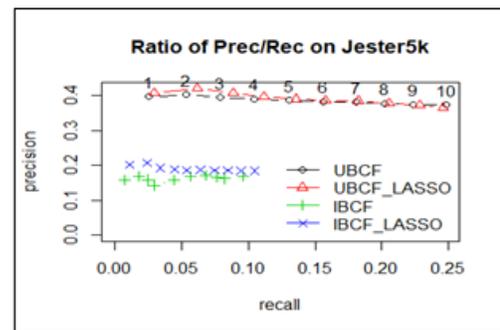


Fig. 8. Comparing the Prec / Rec Ratio of Four Models on Jester5K.

VII. CONCLUSION

Collaborative filtering model is one of the effective technical solutions to provide customer support on e-Commerce sites. Recommended collaborative filtering models are mainly based on user or product similarity to make recommendations to online customers from rating data. However, these models always face the problem of sparse data on e-commerce sites such as new customers, new products, or too little information about customer reviews of products.

In this approach, an integration matrix between Lasso regression similarity and rating data similarity is constructed in a way that is appropriate to make predictions for new users. Experiments on two popular datasets (MovieLense and Jester5K) suggest that the proposed models provide the recommendation result comparable to significantly better accuracy.

Furthermore, while the accuracy of the traditional CFR models is very dependent on rating data, our model is more accurate than the traditional CFR models even when the number of ratings of users is very small.

REFERENCES

- [1] C. C. Aggarwal, *Recommender systems: The textbook*, 1st ed., Springer International Publishing Switzerland, 2016.
- [2] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, 2013, pp. 109–132.
- [3] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, issue 4, 2015, pp. 1-19.
- [4] F. T. A. Hussien, A. M. S. Rahma, and H. B. A. Wahab, "Recommendation systems for e-commerce systems an overview," *Journal of Physics: Conference Series*, vol. 1897, IOP Publishing, 2021, pp. 1-14.
- [5] F. Zhang, T. Gong, V. E. Lee, G. Zhao, C. Rong, and G. Qu, "Fast algorithms to evaluate collaborative filtering recommender systems," *Knowledge-Based Systems*, vol. 96, 2016, pp. 96–103.
- [6] J. Friedman, T. Hastie, R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, issue 1, 2010, pp. 1–22.
- [7] G. Adomavicius and A. Tuzhilin, "Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, issue 6, 2005, pp. 734-749.
- [8] G. Linden, B. Smith, and J. York, "Amazon.com recommendations item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, issue 1, 2003, pp. 76-80.
- [9] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, 2004, pp. 5–53.
- [10] J. Tang, S. Wang, X. Hu, D. Yin, Y. Bi, Y. Chang, and H. Liu, "Recommendation with social dimensions," *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, 2016, pp. 251-257.
- [11] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste," *A constant time collaborative filtering algorithm*, *Information Retrieval*, vol. 4, 2001, pp. 133-151.
- [12] M. P. Robillard, W. Maalej, R. J. Walker, and T. Zimmermann, *Recommendation Systems in Software Engineering*, Springer, 2014.
- [13] M. Hahsler, *recommenderlab: A Framework for developing and testing recommendation algorithms*, Intelligent Data Analysis Lab, Southern Methodist University (IDA@SMU), 2022.
- [14] L. P. Phan, N. Q. Phan, K. M. Nguyen, Huynh, H. X. Huynh, and F. Guillet, "Interestingnesslab: a framework for developing and using objective interestingness measures," *Advances in Intelligent Systems and Computing*, vol. 538, Springer, 2017, pp. 302–311.
- [15] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 58, no. 1, 1996, pp. 267-288.
- [16] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to statistical learning – with applications in R*, 2nd ed., Springer Texts in Statistics, 2021.
- [17] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 2, pp. 1137–1143, 1995.
- [18] J.S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 1998, pp. 43–52.
- [19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning – Data Mining, Inference and Prediction*, 2nd ed., Springer Series in Statistics, 2008.