ECONOMETRICS. EKONOMETRIA
Advances in Applied Data Analysis
Year 2020, Vol. 24, No. 3
ISSN 1507-3866; e-ISSN 2449-9994

IMPROVEMENT OF E-COMMERCE RECOMMENDATION SYSTEMS WITH DEEP HYBRID COLLABORATIVE FILTERING WITH CONTENT: A CASE STUDY

Filip Wójcik
Wroclaw University of Economics and Business,
Faculty of Management, Wrocław, Poland
e-mail: filip.wojcik@ue.wroc.pl

ORCID: 0000-0001-5938-7260

Michał Górnik
Wroclaw University of Economics and Business,
Faculty of Economics and Finance, Wrocław, Poland
e-mail: michal.gornik@ue.wroc.pl

ORCID: 0000-0002-3425-467X

Quote as: Wójcik, F., and Górnik, M. (2020). Improvement of e-commerce recommendation systems with Deep Hybrid Collaborative Filtering with content: A case study. *Econometrics. Ekonometria. Advances in Applied Data Analysis*, *24*(3).

DOI: 10.15611/eada.2020.3.03
JEL Classification: C45, C53, C55

Abstract: This paper presents a proposition to utilize flexible neural network architecture called Deep Hybrid Collaborative Filtering with Content (DHCF) as a product recommendation engine. Its main goal is to provide better shopping suggestions for customers on the e-commerce platform. The system was tested on 2018 Amazon Reviews Dataset, using repeated cross validation and compared with other approaches: collaborative filtering (CF) and deep collaborative filtering (DCF) in terms of mean squared error (MSE), mean absolute error (MAE) and mean absolute percentage error (MAPE). DCF and DHCF were proved to be significantly better than the CF. DHCF proved to be better than DCF in terms of MAE and MAPE, it also scored the best on separate test data. The significance of the differences was checked by means of a Friedman test, followed by post-hoc comparisons to control p-value. The experiment shows that DHCF can outperform other approaches considered in the study, with more robust scores.

Keywords: collaborative filtering, deep learning, content model, product recommendation.

1. Introduction

As the international digital market expanded in the late 20th century, targeted marketing and proper product placement became the most important tools for global companies willing to increase their customer base (Brynjolfsson and McAfee, 2014). The number of products offered by e-commerce platforms such as Amazon, eBay and Allegro, is so huge that intelligent systems suggesting next shopping items had to be adopted quickly in order to mitigate customer information-overflow (Jones, 2013).

The key challenge in shopping items recommendation is a proper modelling of users' preferences, which are not explicitly available to the system. Instead, they are visible only indirectly via ratings or reviews (Sarwar, Karypis, Konstan, and Reidl, 2001), which are often biased and not objective (De Myttenaere, Le Grand, Golden, Rossi, 2014). The other challenge is the so-called 'cold start problem', where a recommendation system cannot operate if there is no previous history for an item or user – e.g. after registration or adding new goods to the stock (Lam, Vu, Le, and Duong, 2008).

Over the years, several algorithms and approaches have been developed to help e-commerce platforms provide better personalized shopping recommendations. As technology and processing power evolved, new ideas for solving the problem emerged – ranging from matrix decomposition, through collaborative filtering, to neural networks. This paper presents a proposal to utilize flexible neural network architectures stacked on top of classic methods.

This study's main research question was: "can a properly tuned neural network trained on both raw customer ratings data and items characteristics achieve better accuracy than a classic collaborative filtering approach?" The hypothesis was that a hybrid neural network would perform better than a simple deep learning approach, and both of them will outperform collaborative filtering in terms of the Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) metrics.

This paper is organized as follows. Section 2 describes the existing approaches and previous research. Section 3 presents algorithms in the study and the experimental framework used to evaluate the quality of recommendations. Section 4 contains a discussion of the results and attempts to provide its more detailed interpretation. It also introduces possible method improvements. Finally, Section 5 presents the conclusions.

2. Previous work

Lately, machine learning has aroused great interest in the scientific community, however, recommendation engines as a knowledge-discovery technique have been analysed since the Web started to gain in popularity in the early 1990s (Schafer, Frankowski, Herlocker, and Sen, 2007). As improvements in artificial intelligence had not yet been made then, researchers tried to improve filtering systems with the use of human knowledge in the loop. Goldberg, Tapestry et al. (Goldberg, Nichols, Oki, and Terry, 1992) created and described a company internal mailing system that helped employees collaborate in order to select relevant and interesting documents. This was, however, implemented in a small group of people who knew one another – GroupLens (Resnick, Iacovou, Suchak, Bergstrom, and Riedl, 1994) tried to create a collaborative filtering system for selecting news in a huge stream of available articles. Another attempt to create an automated collaborative filtering system, in the music domain, was RINGO (Shardanand and Maes, 1995) – a web-accessible application that computed correlations between pairs of users and used them to provide recommendations.

An approach that showed recommendation engines from a slightly different angle focused on measuring the correlation between items rather than users, the recommendations were based on weighted sum and regression models (Sarwar et al., 2001). This was an answer to the challenging issue related to recommended systems, namely the ability to scale to large datasets whilst producing high-level recommendations.

Since there have been a number of advances in deep learning research, neural-net autoencoders were successfully tested against classic matrix factorization and simple neural networks (Sedhain, Menon, Sanner, and Xie, 2015) – their AutoRec showed representational and computational advantages over the aforementioned methods. This promising approach was continued in the work of (Strub, Gaudel, and Mary, 2016), where a hybrid approach is presented, the custom loss function was used to tackle the very common problem of missing data, and side information integration handled typical challenge with recommenders, i.e. the cold start. This approach was elaborated by Li and She (Li and She, 2017) with the creation of a collaborative variational autoencoder that is able to take multimedia data such as movie posters, as side information. Another deep-learning-based approach to collaborative filtering involves a special neural network architecture aiming at modelling the latent features of users and items (X. He et al., 2017). An extension to all neural-based ways is a hybrid model that integrates user and item data with the use of Stacked Denoising Auto-Encoders (Li and She, 2017).

Previous studies (X. He et al., 2017; Li and She, 2017; Sedhain et al., 2015; Strub, Gaudel, and Mary, 2016) show that recommendation engines improved significantly owing to the application of deep learning models. They helped to achieve better results as well as to incorporate side information among user ratings.

3. Methods

This section presents the algorithms and models used further in the study, as well as the overall training setup.

3.1. Collaborative Filtering (CF)

The classic collaborative filtering model is based on sparse ratings matrix $R_{(n\,customers \times n\,items)}$ with dimensionality equal to the number of users *x* the number of products. Each matrix entry corresponds to the rating (usually a positive integer) given to a product by a user (Jones, 2013).

Given a similarity metric between two arbitrary vectors (Sarwar et al., 2001):

$$sim(i,j) = \cos(\vec{\imath},\vec{\jmath}) = \frac{\vec{\imath} \cdot \vec{\jmath}}{\|\vec{\imath}\|_2 * \|\vec{\jmath}\|_2} \tag{1}$$

where: $\vec{\imath},\vec{\jmath}$ corresponds to two arbitrary vectors, representing customers or products.

the system aims at finding entries similar to a given element (user or item of interest) in matrix *R* to make recommendations. Once the neighbouring vectors are found, the weighted sum is computed in order to obtain a predicted rating (Sarwar et al., 2001):

$$P_{u,i} = \frac{\sum_{all\ similar\ items,\ N} s(i,N) * R_{u,N}}{\sum_{all\ similar\ items,\ N} (|s(i,N)|)} \tag{2}$$

where: *i* corresponds to a new item of interest, *u* – current user of interest, *s*(*i*, *N*) – similarity score between i and similar item *N*, $R_{u',N}$ – rating given by a user u to a similar item *N*.

The algorithm presented above can be used with various modifications, amongst which the matrix factorization approach has become the mainstream and the most popular one (X. He et al., 2018; Rendle, Freudenthaler, Gantner, and Schmidt--Thieme, 2012; Zhang et al., 2016).

Despite its popularity and simplicity, the CF model has serious limitations. Probably the most important one is the use of only a simple matrix operation (such as dot products) (X. He et al., 2018) or the lack of the possibility to include external product/user characteristics (X. He et al., 2017; Khattar, Kumar, Gupta, and Varma, 2018).

3.2. Deep Collaborative Filtering (DCF)

Deep collaborative filtering or neural collaborative filtering is a wide family of models using deep neural networks to extend the vector similarity or matrix factorization results described in the previous section (X. He et al., 2017; Khattar et al., 2018). Matrix factorization can be interpreted as a special case of a neural-network framework (X. He et al., 2017). Two matrices, the result of classic collaborative filtering decomposition, are followed by fully connected, dense layers, ending with a single output-neuron. In general, such a model can be described by equation (X. He et al., 2017):

$$\widehat{y_{ui}} = f\left(P^T V_u^U, Q^T v_i^l \middle| P, Q, \theta_f\right), \tag{3}$$

where: *P, Q* are latent factor matrices for users and items, θ_f indicated model parameters of the interaction function *f*.

The architecture of the model which scored the best results is presented in the Figure 1.

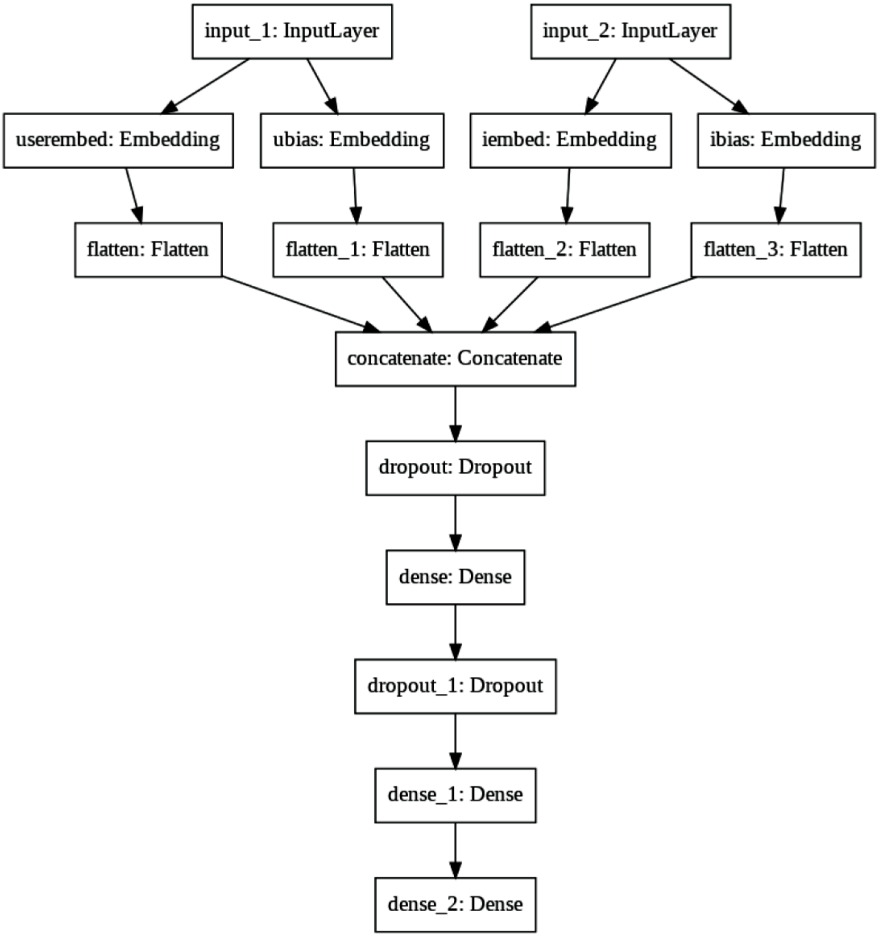

Fig. 1. Deep collaborative filtering model architecture

Source: own work.

Number of 'branches' is equal to two, as this neural network replicates the collaborative filtering approach, based only on customer-product relationships.

3.3. Deep Hybrid Collaborative Filtering (DHCF)

The last model in the study was proposed by the authors, hybrid deep collaborative filtering model with additional content-related features (DHCF), describing the properties of the products. It takes three inputs:

1. Customer identifier.
2. Product identifier.
3. Vector of numerically encoded product features.

Two matrices, the result of classic collaborative filtering decomposition, are followed by fully connected, dense layers. The same set of operations takes place for product content features – the matrix decomposition result is passed to a dense layer. Then, all three branches are concatenated and passed (now as a single matrix) to the next dense layer, ending with a single prediction unit. The architecture of the model which scored the best results is presented in the Figure 2.

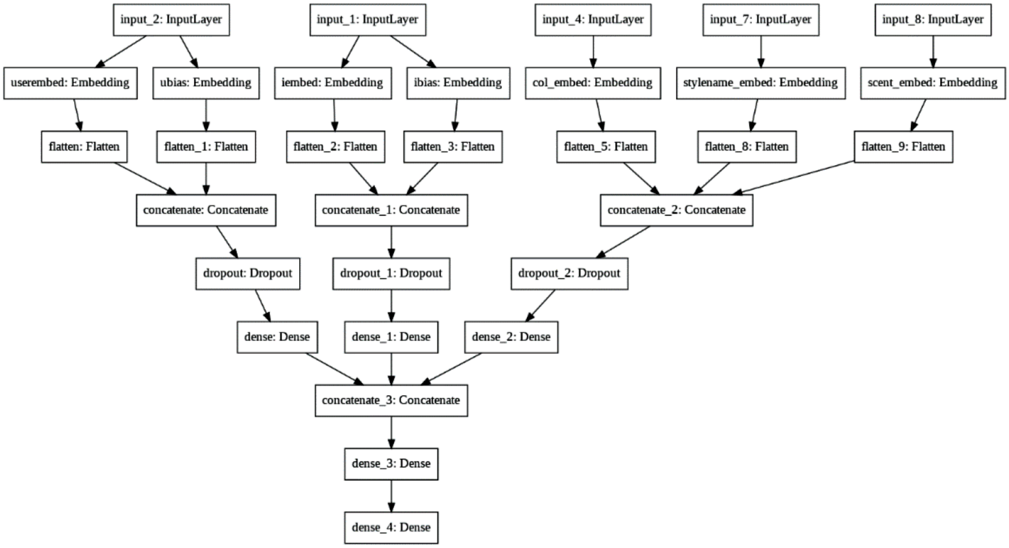

Fig. 2. Deep hybrid collaborative filtering with content model architecture

Source: own work.

The number of branches is not limited and should be adjusted according to the source data (e.g. number of relevant attributes which describe the product).

3.4. Dataset

The performance of the aforementioned algorithms was evaluated using the real-world dataset, containing customer reviews: "Amazon 2018 Reviews Dataset" (Ni, Li, and McAuley, 2020), which is an updated version of the previous edition of ”Amazon Reviews Dataset” (R. He and McAuley, 2016; McAuley, Targett, Shi, and Van Den Hengel, 2015). It was selected as a representative example due to its specific properties, namely:

1. Amazon is a global store, with a large international userbase, and therefore ratings are not given by a small group of geographically local customers.

2. Ratings are highly biased (see an analysis below), reflecting a general problem in recommendation systems (Adomavicius, Bockstedt, Shawn, and Zhang, 2014; Krishnan, Patel, Franklin, and Goldberg, 2014; Shani and Gunawardana, 2011).

The dataset consists of 5269 customer-review pairs coming from the products category: “All beauty” (fashion & cosmetics), with 991 customers and 85 unique products. The dataset ratings were expressed on a 5-star scale, with a single star being the lowest rating, and 5 stars being the highest.

The Amazon 2018 Reviews Dataset is representative in terms of a customer bias due to high rating skewness, which makes the task difficult especially for any prediction algorithms – more than 90% of ratings were maximum (5).

Apart from the customer identifier, product identifier and the rating, the dataset consists of the following additional, product-related information, all descriptive and categorical:

1. Size – 289 unique values.
2. Colour – 77 unique values.
3. Style name – 4 unique values.
4. Scent name – 9 unique values, in case of perfume, attribute describes the fragrance name.
5. Flavour – 4 unique values, describes special features of the product.
6. Design – 21 unique values, describes general design category of the product.

This additional information can be utilized by the proposed hybrid content-based neural network recommendation engine (DHCF) in order to make a better match between the customer and the product.

3.5. Data preprocessing and experimental setup

Each categorical attribute was encoded as a consecutive number so that it can be used by a neural embedding layer later during the training phase.

The source data was prepared for the experiment with the following steps:

1. Test set – 10% of the whole dataset was extracted as a test set, on which a final algorithms evaluation was performed. The ratings for this set were selected in such a way that at least three other ratings from the same customer are present in the train set, so the algorithm will be able to learn the regularities and latent factors which affect the customer’s choices. The test set consists of 527 customer-product-rating triplets.

2. Cross validation set – the remaining 4742 triplets were used to train algorithms using the ten times repeated ten-fold cross-validation method (Moore and Lee, 1994; Schaffer, 1993). During each cross-validation iteration, the cross-validation train subset consisted of 4267 customer-product-rating triplets and the validation subset consisted of 475 customer-product-rating triplets.

During each cross-validation iteration, three prediction performance metrics were collected: mean squared error of prediction (MSE), mean absolute error of prediction (MAE) and mean absolute percentage error (MAPE). All metrics were expressed using the 'stars' scale. Mean squared error was used as the main monitoring metric (due to its useful mathematical properties, helpful in the backpropagation procedure (Goodfellow, Bengio, and Courville, 2016, Chapter 8), as it penalizes the biggest errors during training. MAE gives intuition on how big (on average) the model errors were (expressed in the same units as ratings), while MAPE shows the magnitude of the error relative to the original rating.

3.6. Prediction models

Three classes of modes were compared during this study, corresponding to the theoretical approaches described before.

The first one was a collaborative filtering (CF) model with internal matrix factorization. The number of latent factors was set to 32, based on model tuning results. For every cross-validation fold, it was trained on ten epochs using the Adam optimizer with learning rate set to $lr = 0.01$.

The number of latent factors was set at 32 for the second model (deep collaborative filtering DCF). The next two dense layers were added with 16 and 8 units accordingly, and their activation function was set to *tanh*. In order to avoid overfitting and to reduce residuals magnitude, dropout with rate = 0.2 was added after each layer. For every cross-validation fold, the model was trained on ten epochs, using the Adam optimizer with learning rate $lr = 0.01$.

Third model was deep hybrid collaborative (DHCF). The number of latent factors was set at 24 for customers and 16 for products. Out of all product attributes, colour (with latent factors set at 16), style name (with latent factors set at 4) and scent (with latent factors set at 8) proved to be informative. For customers and products, dense layers of 8 units were used, while for product features – 16 units. After concatenation, a 16-unit followed by a single unit dense layers were added. In order to avoid overfitting and to reduce residuals magnitude, dropout with rate = 0.2 was added after each dense layer, as well as weights l_2 regularization. For every cross-validation fold, the model was trained on ten epochs, using the Adam optimizer with learning rate $lr = 0.01$.

3.7. Results

The table below summarizes the training metrics for three model architectures, after a repeated ten-fold cross validation procedure. The values in each cell represent the mean value of the metric, while the values in brackets represent standard deviation.

Table 1. Training data metrics

Metric [stars]/Model	CF	DCF	DHCF
MSE	0.870 (0.4)	0.121 (**0.026**)	**0.095** (0.036)
MAE	0.347 (**0.07**)	0.108 (0.016)	**0.086** (0.013)
MAPE	0.082 (0.01)	0.037 (0.007)	**0.027** (**0.006**)

Bold text shows the lowest value in a row – for metric values (no bracket) as well as standard deviation.

Source: own work.

A thorough analysis indicates that all three metric values are the smallest for the DHCF model. Of three approaches considered in the study, the classic collaborative filtering model scored the worst results every time.

Each metric comparison can also be visualized using a boxplot, as presented below and such a visualization additionally shows a spread of the result metrics.

In order to assess the statistical significance of the differences, a non-parametric test was performed. Considering the fact that for repeated *k*-fold cross-validation samples are not independent (instead: selected on purpose to be in training or test subset without duplication) and measures are repeated *k*-times for multiple classifiers, the Friedman test was selected (Bouckaert and Frank, 2004; Demšar, 2006; García, Fernández, Luengo, and Herrera, 2009; Trawinski, Smetek, Telec, and Lasota, 2012). The test was conducted on a confidence level $\alpha = 0{,}05$.

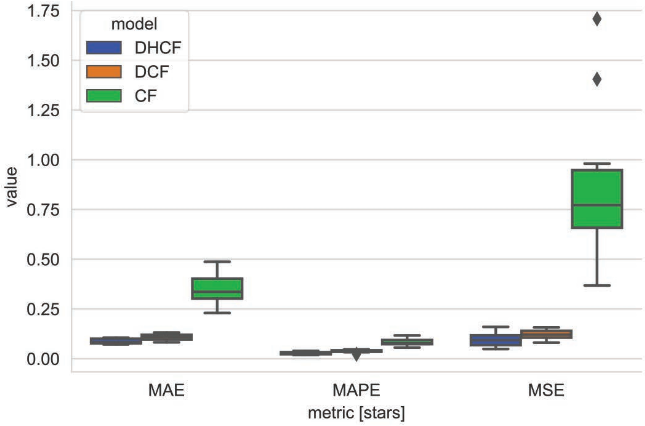

Fig. 3. Comparison of train error metrics

Source: own work.

Having confirmed that there is a statistically significant difference between the algorithms, *post-hoc* tests were performed to assess the significance of specific pairwise differences. For that purpose, the Conover *post-hoc* test was used (Conover and Iman, 1979; Pereira, Afonso, and Medeiros, 2015) with the Bonferroni-Holm *p*-value correction to control the confidence level *α* at 0.05 (Armstrong, 2014; Holm, 1979). The results are summarized in the tables below:

Table 2. Friedman test results

Quality metric/Test	Friedman statistic	*p*-value	Significant at confidence level *α* = 0.05?
Train MSE	≈18.200	≈ 0.0001	true
train MAE	20	≈4.53 *e*–5	true
train MAPE	20	≈4.53 *e*–5	true

Source: own work.

All differences were significant, and hence detailed *post-hoc* tests followed to check each pair of models and metrics.

Table 3. *Post-hoc* test results

Model 1	Model 2	MSE *p*-val	MAE *p*-val	MAPE *p*-val
DHCF	DCF	0.055	**0.0023**	**0.002**
DHCF	CF	<0.001	<0.001	<0.001
DCF	CF	<0.001	<0.001	<0.001

Bold font indicates statistical significance.

Source: own work.

Two of the three metrics indicate that the DHCF model performs better on test data, compared to Deep Collaborative Filtering. Both DHCF and Deep Collaborative Filtering achieve significantly better results than the classic Collaborative Filtering approach.

The final comparison was conducted on a test dataset, separated from training data and not available to any of the algorithms in the study. The table below summarizes the results. The DHCF model scored the best results for all three metrics, although standard deviation of its percentage error residuals was higher than for the collaborative filtering.

Table 4. Test data metrics comparison

Model/Metric [stars]	Test MSE	Test MAE	Test MAPE
DHCF	**0.1698 (0.76)**	**0.1691 (0.375)**	**0.065 (0.24)**
DCF	0.5392 (2.29)	0.3592 (0.63)	0.139 (0.49)
CF	22.8573 (4.9)	4.7270 (0.71)	0.986 (0.02)

Bold font indicates the lowest value in the category. Numbers in brackets indicate standard deviation of residuals.

Source: own work.

Statistical significance for all the metrics was tested in a similar way as for the test data, i.e. using the Friedman test, followed by *post-hoc* pairwise comparisons, with significance level controlled using Holm's method. The results for all the pairs were proven to be statistically significant with p-values below $1e - 10$.

4. Discussion

The Deep Hybrid Collaborative Filtering with the Content (DHCF) model was proven to achieve the best results on the test data and for two of the three metrics in the test data, therefore the main research hypothesis was partially confirmed. No statistically significant difference was found for the Mean Squared Error (MSE) metric between the Deep Collaborative Filtering (DCF) model and DHCF, yet both performed statistically better than vanilla Collaborative Filtering (CF). In most of the comparisons, DHCF showed relatively low variance of residuals than other models.

These results are the consequence of at least two factors. Firstly, deep learning architectures stacked on top of vanilla collaborative filtering are better in terms of extracting and processing latent features, as they are considered to be *universal approximators* (Hornik, Stinchcombe, White, et al., 1989; Nielsen, 2015, Chapter 4) with better generalization abilities. Secondly, the DHCF model accepts external features such as content description, which provides additional information useful for matching customers with products. This capability makes it more flexible and extensible, compared to other approaches. The lack of statistical significance when comparing DCF and DHCF using the MSE metric can be attributed to the fact that two models belong to the same family of algorithms, with the latter being an extension of the former, and MSE over-emphasizes large errors (Qi, Du, Siniscalchi, Ma, and Lee, 2020). At the same time, the other metrics (MAE and MAPE) confirmed the statistical significance of differences in accuracy between DCF and DHCF.

One of the limitations of the presented research is that it was conducted using a single dataset. Although representative, it was recorded in a specific market environment and conditions. As a next step, a series of experiments could be performed using different datasets as inputs, varying in size, product, and customer characteristics. Any such experiment will require DHCF tuning and new architecture design, which might be an opportunity to verify the algorithm's ability to generalize. Alternative datasets might contain a different rating bias (the general tendency of customers towards positive or negative opinions), making the whole exercise harder for a model.

An important factor that makes the CF and classic approaches still popular is the possibility to interpret the model and its representation with clarity. A side effect of matrix factorization can be interpreted directly in terms of 'latent factors' – unobservable (hidden) characteristics shared between customers and products (X. He et al., 2018; Rendle, Freudenthaler, Gantner, and Schmidt-Thieme, 2012; Zhang et al., 2016). While the DCF and DHCF methods attempt to replicate such behaviour (by using 'embedding' neural network layer), higher-order interactions in subsequent layers make it much harder to extract such connections. A number of approaches, mostly based on simpler meta-models, have been designed to address this problem, e.g. SHAP (Lundberg and Lee, 2017) or LIME (Ribeiro, Singh, and Guestrin, 2016), but they are based on indirect *post-hoc* reasoning.

One should be aware that the DHCF architecture presented above is not the only possible one, and, for other problems (datasets), different configurations of customer-product-content layers might be required. The appropriate one can be found through experimentation and careful model selection, as there is no single configuration that could guarantee satisfactory results.

5. Conclusion

In this paper, three recommendation engine performance comparisons were conducted on real customer reviews datasets – Collaborative Filtering, Deep Collaborative Filtering, and Deep Hybrid Collaborative Filtering with Content (DHCF). Both models – Deep Collaborative Filtering and DHCF – should be seen not as a replacement or an alternative for vanilla Collaborative Filtering, but rather as its extensions. They are built on top of traditional architecture, providing new features and predictive abilities. Out of these two the DHCF model proved to be more flexible and extensible, which in consequence leads to statistically better generalization results.

Further studies could focus on testing the described model architectures on other customer reviews datasets belonging to different industries and product categories. Additionally, different content features might be used for that purpose, including natural language processing (such as parsing product descriptions or instructions).

References

Adomavicius, G., Bockstedt, J., Shawn, C., and Zhang, J. (2014). De-biasing user preference ratings in recommender systems. *CEUR Workshop Proceedings*.

Armstrong, R. A. (2014). When to use the Bonferroni correction. *Ophthalmic & Physiological Optics: The Journal of the British College of Ophthalmic Opticians (Optometrists)*, *34*(5), 502-508.

Bouckaert, R. R., and Frank, E. (2004). Evaluating the replicability of significance tests for comparing learning algorithms. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3056*, 3-12.

Brynjolfsson, E., and McAfee, A. (2014). *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*. New York, London: WW Norton & Company.

Conover, W. J., and Iman, R. L. (1979). On multiple-comparisons procedures. *Los Alamos Scientific Laboratory Tech. Rep. LA-7677-MS*, *1*(14).

De Myttenaere, A., Grand, B. Le, Golden, B., and Rossi, F. (2014). Reducing offline evaluation bias in recommendation systems. *ArXiv Preprint ArXiv:1407.0822*.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*(Jan), 1-30.

García, S., Fernández, A., Luengo, J., and Herrera, F. (2009). A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability. *Soft Computing*, *13*(10), 959-977.

Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, *35*(12), 61-70.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. Cambridge Massachusetts, London: MIT Press.

He, R., and McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *25th International World Wide Web Conference, WWW 2016*.

He, X., Du, X., Wang, X., Tian, F., Tang, J., and Chua, T.-S. (2018). Outer product-based neural collaborative filtering. *ArXiv Preprint ArXiv:1808.03912*.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. (2017). Neural collaborative filtering. *Proceedings of the 26th International Conference on World Wide Web*, 173-182.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 65-70.

Hornik, K., Stinchcombe, M., White, H., et al. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359-366.

Jones, M. T. (2013). Recommender systems. Part 1: Introduction to approaches and algorithms. *IBM DeveloperWorks*, *12*.

Khattar, D., Kumar, V., Gupta, M., and Varma, V. (2018). Neural Content-collaborative filtering for news recommendation. *NewsIR@ ECIR*, *2079*, 45-50.

Krishnan, S., Patel, J., Franklin, M. J., and Goldberg, K. (2014). A methodology for learning, analyzing, and mitigating social influence bias in recommender systems. *RecSys 2014 – Proceedings of the 8th ACM Conference on Recommender Systems*.

Lam, X. N., Vu, T., Le, T. D., and Duong, A. D. (2008). Addressing the cold-start problem in recommendation systems. *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, 208-211.

Li, X., and She, J. (2017). Collaborative variational autoencoder for recommender systems. *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 305-314.

Lundberg, S. M., and Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*.

McAuley, J., Targett, C., Shi, Q., and Van Den Hengel, A. (2015). Image-based recommendations on styles and substitutes. *SIGIR 2015 – Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Moore, A. W., and Lee, M. S. (1994). Efficient algorithms for minimizing cross validation error. *Machine Learning Proceedings 1994*, 190-198.

Ni, J., Li, J., and McAuley, J. (2020). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*.

Nielsen, M. A. (2015). *Neural networks and deep learning* (Vol. 25). Determination Press USA.

Pereira, D. G., Afonso, A., and Medeiros, F. M. (2015). Overview of Friedman's Test and post-hoc analysis. *Communications in Statistics: Simulation and Computation*, *44*(10), 2636-2653.

Qi, J., Du, J., Siniscalchi, S. M., Ma, X., and Lee, C.-H. (2020). On mean absolute error for deep neural network based vector-to-vector regression. *IEEE Signal Processing Letters*.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2012). BPR: Bayesian personalized ranking from implicit feedback. *ArXiv Preprint ArXiv:1205.2618*.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW 1994*.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). Why should I trust you? Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135-1144.

Sarwar, B., Karypis, G., Konstan, J., and Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the tenth international conference on World Wide Web – WWW '01*, 285-295.

Schafer, J., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. *The Adaptive Web* (4321), 91-324.

Schaffer, C. (1993). Selecting a classification method by cross-validation. *Machine Learning*, *13*(1), 135-143.

Sedhain, S., Menon, A. K., Sanner, S., and Xie, L. (2015). Autorec: Autoencoders meet collaborative filtering. *Proceedings of the 24th international conference on World Wide Web*, 111-112.

Shani, G., and Gunawardana, A. (2011). Evaluating Recommendation Systems. In F. Ricci, L. Rokach, B. Shapira, and P. Kantor (Eds.), *Recommender systems handbook*. Boston, MA.: Springer. https://doi.org/10.1007/978-0-387-85820-3_8

Shardanand, U., and Maes, P. (1995). Social information filtering: algorithms for automating "word of mouth." *Proceedings of the SIGCHI conference on Human factors in computing systems*, 210-217.

Strub, F., Gaudel, R., and Mary, J. (2016). Hybrid recommender system based on autoencoders. *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 11-16.

Trawinski, B., Smetek, M., Telec, Z., and Lasota, T. (2012). Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. *International Journal of Applied Mathematics and Computer Science*, *22*(4), 867-881.

Zhang, H., Shen, F., Liu, W., He, X., Luan, H., and Chua, T.-S. (2016). Discrete collaborative filtering. *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 325-334.

WYKORZYSTANIE HYBRYDOWYCH GŁĘBOKICH SIECI NEURONOWYCH JAKO SYSTEMÓW REKOMENDACYJNYCH: STUDIUM PRZYPADKU

Streszczenie: W artykule zbadano innowacyjną architekturę sieci neuronowych zwaną Głębokim Hybrydowym Systemem Filtracji Kolaboratywnej (DHCF), mającą posłużyć jako system rekomendacji konsumenckich. Jego zadaniem jest sugerowanie produktów klientom platform *e-commerce*. System został przetestowany na zbiorze danych 2018 Amazon Reviews, z wykorzystaniem powtórzonej walidacji krzyżowej, i porównany z dwoma innymi podejściami: filtracją kolaboratywną (CF) oraz filtracją kolaboratywną z siecią neuronową (DCF). Do porównania wykorzystano metryki błędu średniokwadratowego (MSE), średniego błędu bezwzględnego (MAE) oraz średniego procentowego błędu bezwzględnego (MAPE). DCF i DHCF uzyskały wyniki istotnie lepsze niż CF, a dodatkowo DHCF uzyskał lepsze wyniki niż DCF pod względem MAE i MAPE. Istotność różnic sprawdzano testem Friedmana z porównaniami wielokrotnymi i kontrolą poziomu istotności. Eksperyment dowodzi, że DHCF uzyskuje lepsze i stabilniejsze wyniki niż pozostałe metody.

Słowa kluczowe: filtracja kolaboratywna, głębokie uczenie, model treści, rekomendacja produktów.