Summary Paper: A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems

Sruthy Annie Santhosh, 312213
Universitat Hildesheim
Hildesheim, Germany
santhosh@uni-hildesheim.de

ABSTRACT

Recommendation systems play an important role in our everyday lives. In this paper, a summary of the important topics in the paper on Multi View Cross Domain User Modelling Recommendation System[1] are discussed. The authors of the above paper propose a novel recommendation system having better quality and scalability than other existing models. The proposed model is based on a deep neural network(DNN) framework and utilises a rich user and item feature set based on the content based approach. The problem of lack of recommendation quality for new users is also handled by modeling the user using browser history and search queries. In this paper, authors also combined the features from multiple domains, which was found to improve the quality. This leads to a multi view deep learning framework. There is also a mention of different data and dimensionality reduction techniques which help in making the model scalable. Experiments were also conducted on three domains like , Windows Appstore, News recommendation and Movie/Tv recommendation. These experiments also showed a great enhancement in the recommendation quality.

KEYWORDS

Recommendation Systems, Deep learning Neural Network, Multi-View, User Modeling, Dimensionality Reduction

ACM Reference Format:
Sruthy Annie Santhosh, 312213. 2022. Summary Paper: A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. In *Proceedings of Universitat Hildesheim.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 ABOUT PAPER

This summary is based on the paper written by two authors:

Ali Elkahky of the Department of Computer Science, Columbia University, New York and Yang Song, Xiaodong He at Microsoft Research, One Microsoft Way Redmond

The paper was published in WWW' 15, as part of the proceedings of the 24th International Conference on World Wide Web, May 2015. It has 589 citations on Google Scholar. Many further advancements in the field of Recommendation systems have been based on the idea of the model proposed in this paper[21–23].

Reference to the paper:
https://www.microsoft.com/enus/research/wpcontent/uploads/2016/02/frp1159-songA.pdf

2 INTRODUCTION

Due to the advancement of Internet in everyday tasks, the role of Recommendation systems have increased extensively. They are used in different domains like e-commerce, movie sites and social media to provide users with suggestions based on their preferences. Therefore this is an important field of research as there is always a need for better quality recommendation systems. The two main approaches in developing the same are Collaborative Filtering method and Content based method (CF) [4–6]. In this paper, the authors have focused on CF as the line of research as it can handle cold start problems better. Cold start problem occurs when we need to provide recommendations for new users whose previous history regarding the items is not available[3]. For better personalisation, this problem must be accommodated effectively. Currently CF handles this problem better.

Thus in this paper a new model is proposed based on the CF approach i.e, using rich user and item feature vectors. Here the model handles the cold start problem by creating a rich user vector from search queries.This model is an extension of the Deep Structured Semantic Model (DSSM) model which was introduced in the paper[2]. The DSSM model maps user and item features into a shared semantic space and recommends items having maximum similarity with each user. Here the model learns from users and item features from different domains (one user view and multiple data views) jointly and hence this model is called as Multi View Deep Neural Network (MVDNN). MVDNN also uses ranking based objective rather than minimizing root mean square error as it was found to be better[7] .

3 RELATED WORK

Recommendation Systems is an area which has been extensively studied and researched. As already mentioned in the above section there are mainly two approaches. In the collaborative filtering approach, we use an user's previous history in that domain to predict new items for recommendation. Hence this is not effective for new users. Matrix Factorisation is a popular method which uses item or user collaborative filtering by creating a user- item matrix based on the previous interactions[4, 5, 8].

Content based method on the other hand handles the cold start problem effectively as it extracts features from users and items and then recommends items to users based on the similarities between

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Universitat Hildesheim, 2022, Germany
© 2022 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

these features. For example if user A and user B liked movies in the same genre, then user A can be recommended a movie which user B liked and vice versa. Here we assume that users with similar interests tend to like items similar to ones they liked before. Many different methods have been studied in this area and can be seen in papers [3, 9].

There have also been studies on the combination of the two methods. The ability of CF method to handle cold start problem depends on the richness of the user vector. In paper [10] [26], user vector was constructed from search history while in [11], the authors used social profiles of users to model the vector.

While single domain models are more common, there have been advances in cross domain recommendation models also. Using data from different domains was shown to provide better quality of recommendation, especially for users having less data in one domain[12]. Thus by sharing the user-item data across domains, a better user modeling can occur. Also the non-linear mapping of deep learning models provides a better representation of vectors which makes it easier to store and share information among domains[13].

4 DESCRIPTION OF DATASETS

Mainly four datasets were used in this experiment. All those four were collected from user logs of several Microsoft products between December 2013 and June 2014 from the English speaking users in countries like US, Canada and UK. Each of the datasets are as follows[1]:

- `User Features`: includes the user's search queries and clicked urls from Bing Web Vertical. The queries are normalized, stemmed and split into uni gram features while the URLs are shortened to domain level. Then TF-IDF scores are used to keep only the most popular features. These steps were done to reduce the dimension size. Overall length of the user feature vector is 3.5 million.

- `News Features`: includes the news article clicks of users from Bing news vertical. Each news item is represented by 3 parts. Firstly the title feature is encoded using letter trigram[2]. Then the top level category of the News is encoded as binary features. And lastly the named entities in each article are extracted by NLP parser and encoded using letter trigram. These steps resulted in a 100k feature vector.

- `App Features`: includes the app download histories of users from Windows Appstore logs.The title of each app is combined with its category and represented in trigram format. This resulted in a 50k feature vector for Apps set.

- `Movie/Tv Features`: includes the Movie/Tv view history of users from Xbox logs. The title and description of each item is combined and encoded using trigram format. The genre is represented using binary features. This resulted in a 50 K feature vector.
  In this study, the authors used the user-item pairs of joint users, i.e, common users between each item view and user view. The joint users were extracted using their IDs.

Type	DataSet	UserCnt	Feature Size	Joint Users
User View	Search	20M	3.5M	/
Item View	News	5M	100K	1.5M
	Apps	1M	50K	210K
	Movie/TV	60K	50K	16K

Table 1: Statistics of the four data sets used in this paper. The *Joint Users* column indicates the number of common users between each item view and the user view.

5 DSSM FOR USER MODELING IN RECOMMENDATION SYSTEMS

Since the model discussed in this paper is closely related to the DSSM model, it is essential to briefly discuss it first. DSSM was used to enhance the query - document selected matching in web search[2]. The input to the model is a high dimensional vector which consists of the raw count of terms in the query or the document. These inputs are passed through two neural networks (one for query and other for document) and then mapped into semantic vectors in a shared space. Here the activation function used is tanh.

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

Then their cosine similarities are found which are used for ranking each document for the corresponding query[].

$$cosine(y_q, y_d) = \frac{y_q^T * y_d}{||y_q|| * ||y_d||}$$

They are ranked by finding the posterior probability. DSSM also has a word hashing layer which transforms each word as a letter trigram vector.

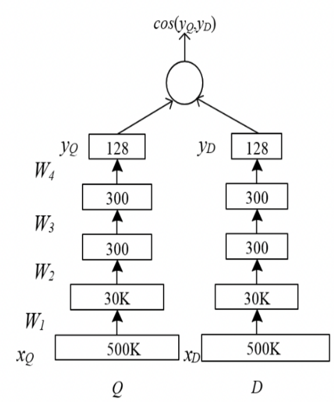

Figure 1: The illustration of the deep structured semantic model (DSSM).

6 MULTI VIEW DEEP NEURAL NETWORK

MultiView DNN is an extension of the DSSM model. Here different views are mapped into a shared view[1]. The user features are mapped to the user view which is the pivot view(X_u) and different auxiliary views are created for each item domain($X_1 \ldots X_v$). Thus this model will have 2 or more views(v+1 views). Each view will have its own input domain and non-linear mapping layers ($f_i(X_i, W_i)$). These layers transform the input X_i to a shared semantic space Y_i. Each sample of training data would have one user input and one active item input. Thus at any time only one item view would be active, rest all will be zero vectors. The mathematical representation of the objective is as follows:

$$p = \underset{W_u, W_1 \ldots W_v}{\arg\max} \sum_{j=1}^{N} \frac{e^{\alpha_a cos(Y_u, Y_{a,j})}}{\sum_{X' \epsilon R^{d_a}} e^{\alpha_a cos(Y_u, f_a(X', W_a))}}$$

The objective is to find a non-linear mapping for user features ,Wu, that can transform them into a space that has the maximum similarity to all the different items the user liked in those different views/domains. This implies that the sum of similarities between pivot view and all other views must be maximised. Through this mapping, the domains which do not have enough information to learn a good mapping can also learn from the other domains having more data. This model was created under the assumption that the users who have similar tastes in one domain will have similar tastes in other domains as well. Thus by sharing the data across domains, the quality of recommendation can be improved. This was experimentally proved by the authors.

- `Input to the model`: High dimensional sparse vector(features of users and items)
- `Output of the model`: low dimensional dense feature vector in a shared semantic space.

Here each view can have arbitrary number of features. The first layer of the model is the hidden layer used for word hashing. This layer helps in reducing the dimension of the vectors. More information on the dimensionality reduction techniques used will be discussed in further sections(Section 8). After dimensionality reduction, the hashed features are projected through multiple non-linear mappings which finally transforms them into features in the semantic space. Then the cosine similarities are calculated between the user and each item. These calculations are same as those done in DSSM model.

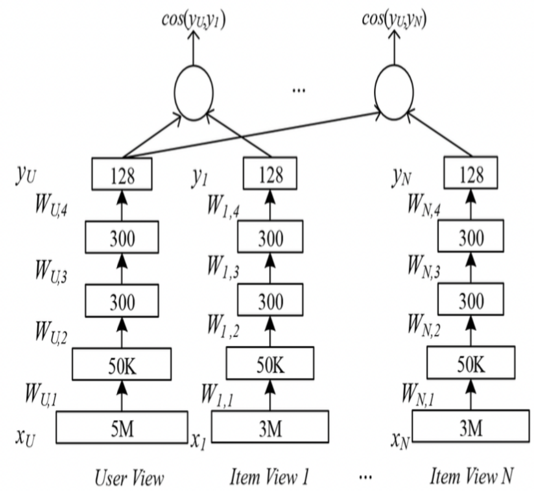

Figure 1: Illustration of the Multi View DNN model

Algorithm 1 Training Multi-View DNN

Input: $N = \#$ of view pairs, $M = \#$ of training iterations,
U_A = user view architecture,
$I_A = \{I_{A1}, \ldots I_{AN}\}$ item view architecture,
$U_D = \{U_{D1}, \ldots UDN\}$ user input files,
$I_D = \{I_{D1}, \ldots I_{DN}\}$ item input files,
$\mathbf{W_U}$ = user view weight matrix,
$\mathbf{W_I} = \{W_{I1}, \ldots W_{IN}\}$ item view weight matrices
Initialization
Initialize $\mathbf{W_U}$ and $\mathbf{W_I}$ using U_A and I_A
for $m = 1$ to M
  for $v = 1$ to N
    $T_U \leftarrow U_{Dv}$
    $T_I \leftarrow I_{Dv}$
    train $\mathbf{W_U}$ and $\mathbf{W_I}$ using T_U and T_I
  end for
end for
Output: $\mathbf{W_U}$ = final user weight matrix,
$\mathbf{W_I}$ = final set of item view weight matrices

6.1 Training

MVDNN is trained using Stochastic Gradient Descent algorithm. Each training sample contains one user feature and one item feature. Since all other item views are zero vectors, while taking the derivatives it will be zero. Hence it returns only two non-zero derivatives, $\frac{\partial p}{\partial w_u}, \frac{\partial p}{\partial w_i}$ which allows us to update the rules of DSSM easily. Here instead of query and document, the user and item vectors are substituted.

7 ADVANTAGES

Although MVDNN is an extension of DSSM, it has many features that make it superior.

- In DSSM, each view had the same size feature dimensions and inputs were processed using the same tri-gram representation. However there is no such constraint in the new

model, views can have arbitrary size of dimensions and any representation can be used like uni-gram, bi-gram, tri-gram etc. Removing this constraint allows for incorporating many categorical features like geospatial features, url domains and so on.
- DSSM can also be scaled to many different domains. This allows the model to learn from different domains and provide a better quality recommendation. In the experiment conducted by the authors 3 domains were considered- News, Movies and Apps.
- This model also has a better user modeling because of the rich variety of information in queries and history of the users across domains. This also helps give better recommendation for new as well as old users.

8 DIMENSION AND DATA REDUCTION

In practical applications, the model needs to handle huge amount of training data in high dimensional feature space as the user modeling will be rich. In order to handle these high dimensional feature vectors, the authors have proposed various dimension and data reduction techniques.

8.1 Top features

In this method, the top k most frequent features are selected. Here the assumption is that users can be described well enough using a small set of frequent features. The raw user features are preprocessed using TF-IDF scores and then selected. This resulted in 83k user features which is a great decrease from the initial size.

8.2 K Means

This is a clustering technique in which a number of clusters are created such that the sum of distances between each point and its nearest cluster is minimized[14]. Therefore similar features are grouped together and a feature vector Y is generated which has size equal to the number of clusters. In this study, the authors set the number clusters to 10K as useful patterns can be learned when there are more clusters. But this makes it computationally expensive. In this experiment the cloud based Map-Reduce implementation is used[15].

8.3 Local Sensitive Hashing

Here the data is projected onto a lower dimensional space using a random projection matrix while maintaining the pairwise cosine distance of the original feature space[16]. LSH hashes similar input items into "same" buckets with high probability. LSH is also expensive to run as it requires the generation of a transformative matrix. But in this experiment, the authors uses a pooling trick[17] to reduce the memory requirement to 10M.

8.4 Reduction in the number of training samples

As discussed before, the number of user-item pairs in the training data is huge. It is almost around 1 billion for the News data set. But this can cause GPU and memory issues. Hence the authors have compressed the training data such that there is only one training sample per user per view. This implies that user features are matched with item features having an average score of all the items liked by that particular user in that view. This reduces the number of training samples effectively. It is also shown through the experiment that even after approximation, the quality of recommendation remains high.

9 EXPERIMENTAL SET UP

Data Set	Training			Testing		
	Number Of unique users	Number of unique items	Number of training pairs	Number of new users	Number of test pairs for old users	Number of test pairs for new users
Apps Data	200K	55k	2.5M	1K	11K	2K
News Data	1.5M	5M	$> 1B$	5K	50K	10K
Xbox Data	16K	10K	45K	1K	10K	3K

Table 2: Details of the Train/Test breakdown for the three data sets used to evaluate our proposed approach.

The aim of the experiment in this paper is to evaluate the performance of the model on old users as well as new users who have no previous history in that domain but has some search and query history in other domains. The datasets are divided into training and test sets. Each user is assigned either to training set or test set according to the probability ratio of 0.9:0.1 . Then for each user in the test set, is further split into old users and new users in the ratio 0.8:0.2 . Among those users labelled as old, half are used for training while the rest are in testing. This process ensures that new users are actually encountered only while testing process.

In this experiment, the authors select 9 random user-item test pairs for each sample and add them into the testing data. The performance of the model depends on how well it ranks the correct pair. Here the authors used mainly two metrics for evaluation[18].

- `Mean Reciprocal Rank(MRR)`- computes the inverse of the rank of the correct item among all the items and takes the average score across the data set. Here only the rank of the first answer is considered.
- `Precision@1` - computes the percentage of times the model ranks the correct item as the top item. This metric considers every result obtained as relevant.

The quality of recommendation of the MVDNN model was compared with that of different baseline models as well as that of DSSM model.

9.1 Baseline Models

- `Standard SVD Matrix Decomposition` : This model is used as the standard baseline for collaborative filtering. Here user/item matrix is constructed and matrix decomposition is performed using SVD. This model is only suitable for small datasets like the Apps set.

- `Most Frequent Items` : This model is used as the standard baseline for content based approach. Hence it can handle new users as well. In this model frequency of each item is calculated and then the user-item pairs are ranked accordingly to find the most frequent ones.

- `Canonical Correlation Analysis (CCA)` : This is a multiview model which aims to find linear transformations for each view in the model[19]. In this model, the aim is to maximise the correlations between the transformations.This model differs from DSSM model in only two ways; it uses linear transformations instead of non-linear transformations and it aims to maximise the correlations under fixed variance constraints instead of maximising the rank of the correct pairs.
  In this experiment, the authors only used Top K method for dimensionality reduction as the others returned highly non sparse matrices which made the correlation matrix too dense.

- `Collaborative Topic Regression (CTR)` : This recommendation model combines Bayesian matrix factorisation with item features[20]. There are two inputs to a CTR model : a collaborative matrix and item features. This model handles new users as well. It matches users with items by maximising the reconstruction error of the matrix and using the item features to smooth over.

9.2 Method

The authors first ran three sets of experiment on Apps and News data set to train the Single view (SV) DNN model. Each set corresponds to one dimensionality reduction technique like SV-Top K, SV- Kmeans and SV- LSH. These experiments will be referred to as Type II.

Then three sets of experiments were conducted on MV DNN. In the first two sets only Apps and News data sets were combined and the techniques used were Top K and K means. The last set of experiments combined Apps, News and Movie/Tv datasets with Multi view DNN Top K method. The experiments on MV DNN will be referred to as Type III.

The baseline models were also run and these are referred to as Type I.

10 RESULTS

From the experiments we can see that the baseline models perform relatively bad. The results of all the experiments are shown in the tables.

Most frequent items baseline performs very poorly which shows that simple solutions will not work well for new users SVD baseline model is also found to be not good enough even for current users, showing that content based approach is better. CCA models also performed poorly, which indicates the importance of non-linear mapping and ranking based objective. The CTR model performed relatively well for current users but was not good enough for new users. These results are consistent across the different datasets for both metrics.

In the case of Single View DNN models, the performance depends on the dimensionality reduction technique employed. It was shown that the best method for both Apps and News data set was Top K method. Thus it can be concluded that a small set of important features are enough for user modeling and that Top K method captures the semantics effectively. From the table we can observe that the best SVDNN model outperformed the best Type I model CTR by 11% for all users and 36.7 % for new users(MRR).The same is the case for P@1 also.

For Multi View DNN models, it can be observed that the performance increases with the increase in the number of domains. By combining the News and Apps datasets, higher performance can be observed in both the domains for both metrics. In Apps dataset, there is 4% increase in all users and 7% increase in new users when combined with News set. Hence it can concluded that increasing the number views helps handle new user problem. On adding the Movies/Tv data set, the results are even more promising. There was a 6% increase overall and 8% increase in new users in the App set for MRR metric. There is a similar increase for Precision@1 metric also.

Thus by evaluating the performance of different recommendation models it can be concluded that, the best method is the Multi View DNN method using Top K features. This method performed 25.2% better than CTR for all users and 115% better for new users in the App set using P@1 metric. Also due to the large size of the News set, traditional models like CCA and SVD failed.

To further evaluate the performance of the MVDNN recommendation model, the authors conducted a single feature input experiment also. The best performing MVDNN with Top K features model was taken and was given an input of user features with only one domain name. Then the model was run to find the recommendations against the News and Apps sets. It was observed that the model provided good recommendations. When given the domain id babycenter.com, the recommendations given in News set were related to pregnancy and baby names and those given in App set were related to Pregnancy and preschool. This further shows the effectiveness of this model.

11 SCALABILITY

As discussed in the above section, the News dataset with over a billion entries were not handled effectively by traditional baseline models. The MV DNN model with deep learning framework was able to handle this due to its distributed training. The authors also compared the training times of various algorithms and its performance. It can be observed that for Apps dataset which is smaller, SVD and CCA finishes fast though its quality is poor. SVDNN finishes 100 training iterations in 33 hours, while CTR model takes a longer time.

SV DNN TopK and MV DNN Top K models maintain a sublinear training time with respect to the size of the dataset as SGD requires less epochs when more data is available. In this experiment, the number of iterations are manually specified as 100, which can be changed also.

12 DISCUSSION AND FUTURE SCOPE

In this section I would like to discuss about different aspects of the paper and its future scope.

While this multiview model shares the information between different views, it is not been clear how this happens. Also only a high level explanation of the model is given. Much details are not provided about its implementation and layers. Though the paper talks about various dimensionality reduction techniques, it has not been made clear when and where they are implemented. Also one

	Algorithm	All Users		New Users	
		MRR	P@1	MRR	P@1
I	Most Frequent	0.298	0.103	0.303	0.119
	CF	0.337	0.142	/	/
	CCA (TopK) [29]	0.295	0.105	0.295	0.104
	CTR [32]	0.448	0.277	0.319	0.142
II	SV- Kmeans	0.359	0.159	0.336	0.154
	SV-LSH	0.372	0.169	0.339	0.158
	SV-TopK	**0.497**	**0.315**	**0.436**	**0.268**
III	MV-Kmeans	0.362	0.16	0.339	0.156
	MV-TopK	0.517	0.335	0.466	0.297
	MV-TopK w/ Xbox	**0.527**	**0.347**	**0.473**	**0.306**

Table 3: Results for different algorithms on Windows Apps Data Set. Type *I* algorithms are baseline methods we compare with. Type *II* are single user-item view methods trained using the original DSSM framework. Type *III* are multi-view DNN models we proposed. The best performance is achieved by training a MV-DNN on all three user-item views with TopK as feature selection method.

	Algorithm	All Users		New Users	
		MRR	P@1	MRR	P@1
I	Most Frequent	0.301	0.111	0.305	0.111
	CTR [32]	0.427	0.215	0.276	0.123
II	SV-Kmeans	0.386	0.192	0.294	0.143
	SV-LSH	0.45	0.247	0.34	0.186
	SV-TopK	**0.486**	**0.286**	**0.358**	**0.208**
III	MV-Kmeans	0.391	0.194	0.296	0.145
	MV-TopK	0.494	0.303	0.368	0.222
	MV-TopK w/ Xbox	**0.503**	**0.321**	**0.398**	**0.245**

Table 4: Results for the News Data Set. Similarly, the best performance is achieved by our multi-view models. Note that due to the extreme big size of this data set (> 1B entries), traditional algorithms like CF (SVD) and CCA failed to handle it due to memory constraint.

User View with Single Domain ID Feature	Top Matched News	Top Matched Apps
barackobama.com	Obama to Delay Obamacare Again to Help Democrats Froma Harrop: Democrats should not run away from Obamacare Democratic Senator: I am willing to defy Obama Governor Jindal proposes Republican alternative to Obamacare	7 Minutes Fitter Relax Meditate Escape Sleep Sleep Tracker U.S. Constitution
spiegel.de	Nazi-Era Jerseys on View in World Cup Exhibit 2014 World Cup Day 3 Lessons: Colombia Fun In The Sun... Belgium Vs. Algeria World Cup 2014: Live Stream... Colombia vs. Ivory Coast: Tactical Preview ...	ESPN Cricinfo Golf News RSS Pulse News Dinamalar - Tamil News Paper

Figure 2: Results of single feature input experiment using domain id

fact to be considered is whether there is way to implement the model without these reduction techniques while maintaining its feasibility.

There was also no explanation given to the choice of the activation(why not ReLu?) and similarity functions used.More reasoning can be given on the usage of cosine similarity for measuring the similarity of features. In paper[21], co embedding method is used.

Algorithm	Data Set	
	Apps	News
CF	4 hours	OutOfMemory
CCA [29]	4 hours	OutOfMemory
CTR [32]	40 hours	120 hours
SV-TopK	33 hours	50 hours
MV-TopK	60 hours	
MV-TopK w/ Xbox	62 hours	

Figure 3: Training time of all the models in comparison

In the experiment, the number of iterations is set to be 100 while some data sets converge before that. Hence further analysis can be done to make it automatic stoppage. There was also no mention of validation set in the paper.

Further research in cross domain systems have found out that sparsity problems can be resolved by latent factor alignment[22].

While this model is based on the Content based approach, Collaborative approach can also be utilised using this model. Recently many papers have attempted to do the same[23].

More domains can also be added and more user information can be incorporated to user view as well. Scalability of the model can also be further evaluated without reducing the size of the data. Another point to note is the metrics used. This model can be evaluated on other metrics and results can be compared as well.

During the time of its publication this paper provided a novel idea to combine different domains for recommendation systems.

13 CONCLUSION

Thus this paper proposes a new recommendation system that uses Deep learning framework to match user features to item features while combining data from multiple domains. This method was found to be of better recommendation quality due to its rich user features for both existing and new users. Scalability of the model was also discussed through various dimensionality reduction techniques.Various experiments were also conducted on real time data proving the same. Thus it was observed that this new model has better performance than other existing models by a large margin.

14 ACKNOWLEDGEMENT

All figures, tables and equations used in this paper are taken from the source paper :

Elkahky, Ali Mamdouh, Yang Song, and Xiaodong He. "A multi-view deep learning approach for cross domain user modeling in recommendation systems." In Proceedings of the 24th international conference on world wide web, pp. 278-288. 2015.

REFERENCES

[1] Elkahky, Ali Mamdouh, Yang Song, and Xiaodong He. "A multi-view deep learning approach for cross domain user modeling in recommendation systems." In Proceedings of the 24th international conference on world wide web, pp. 278-288. 2015.

[2] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In CIKM'13, pages 2333–2338.

[3] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In SIGIR'02, pages 253–260.

[4] Robert M Bell and Yehuda Koren. Improved neighborhood-based collaborative filtering. In KDD'13 CUP, 2007.

[5] Jasson DM Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In ICML'05, pages 713–719.

[6] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. Personalized news recommendation based on click behavior. In IUI'10, pages 31–40.

[7] Joonseok Lee, Samy Bengio, Seungyeon Kim, Guy Lebanon, and Yoram Singer. Local collaborative ranking. In WWW'14, pages 85–96.

[8] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In WWW'01, pages 285–295. ACM, 2001.

[9] Asela Gunawardana and Christopher Meek. Tied boltzmann machines for cold start recommendations. In RECSYS'08, pages 19–26.

[10] Yang Song, Hongning Wang, and Xiaodong He. Adapting deep ranknet for personalized search. In WSDM'14, pages 83–92.

[11] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. Twitter-based user modeling for news recommendations. In IJCAI'13, pages 2962–2966.

[12] Shaghayegh Sahebi and Peter Brusilovsky. Cross-domain collaborative recommendation in a cold-start context: The impact of user profile size on the quality of recommendation. In User Modeling, Adaptation, and Personalization, pages 289–295. Springer, 2013.

[13] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In NIPS'13, pages 2643–2651.

[14] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. Applied statistics, pages 100–108, 1979.

[15] Cheng Chu, Sang Kyun Kim, Yi-An Lin, YuanYuan Yu, Gary Bradski, Andrew Y Ng, and Kunle Olukotun. Map-reduce for machine learning on multicore. 2007.

[16] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In STOC'98, pages 604–613.

[17] Benjamin Van Durme and Ashwin Lall. Online generation of locality sensitive hash signatures. In ACL'10 Short Papers, pages 231–235.

[18] Thom, James, and Falk Scholer. "A comparison of evaluation measures given how users perform on search tasks." In Proceedings of the Twelfth Australasian Document Computing Symposium, pp. 100-103. RMIT University, 2007.

[19] Bruce Thompson. Canonical correlation analysis. Encyclopedia of statistics in behavioral science, 2005.

[20] Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In KDD'11, pages 448–456.

[21] Jia, Xiaodong Jing, Xiao-Yuan Zhu, Xiaoke Cai, Ziyun Chang-hui, Hu. (2022). Co-embedding: a semi-supervised multi-view representation learning approach. Neural Computing and Applications. 34. 10.1007/s00521-021-06599-y.

[22] Yu, Xu Hu, Qiang Li, Hui Du, Junwei Gao, Jia Sun, Lijun. (2022). Cross-domain recommendation based on latent factor alignment. Neural Computing and Applications. 34. 1-12. 10.1007/s00521-021-05737-w.

[23] Liu, Weiming Zheng, Xiaolin Hu, Mengling Chen, Chaochao. (2022). Collaborative Filtering with Attribution Alignment for Review-based Non-overlapped Cross Domain Recommendation.