Personalized Search Ranking

Srividhya Balaji
sri_1007@tamu.edu
Texas A&M University

Ashwin Krishna
ashwinedu@tamu.edu
Texas A&M University

Abindu Dhar
abindhar@tamu.edu
Texas A&M University

Rishabh Singla
rishabh.singla@tamu.edu
Texas A&M University

ABSTRACT

With the rapid growth of the amount of information on the Internet, it has become important to help users alleviate the problem of information overload and select interesting and necessary information in web applications including web search, e-commerce search etc. We aim at providing improved Personalized search results to each user based on their activity and other similar users activity by leveraging the recent advances in graph embedding techniques. Extensive experiments on the CIKM cup e-commerce dataset demonstrated significant improvement achieved by our model over the baseline approach and the competition prize winners for query-full data.

KEYWORDS

ecommerce, graph embedding, personalized, feature extraction,ranking

1 INTRODUCTION

Current web search engines are quite effective in query search as they use various machine learning methods to process and rank information, however these results are usually based on the query, document features, but does not include methods to integrate user information to give a personalized search experience. What if we could build a personalized re-ranking application which could help a customer to find its intended results in a more effective way reducing the search time and increasing user experience. One of the application of this personalized re-ranking has a great impact on e-commerce product search.

Over the last few years, the popularity of internet and e-commerce have tremendously affected the overall market scenario and consumer's choices. With the increasing number of e-commerce websites, people have the freedom to research for the products they are looking for, compare their prices and then purchase things as per their interests. However, every platform often has such a wide variety of products and every product does offer numbers of choices based on brands, prices, colors etc. For example, Amazon the biggest online market offers a hundreds of million products followed by eBay, Apple, Walmart etc. As a customer, it becomes difficult to find relevant products without support of an effective personalized search engine. Such a personalized application not only saves search time and give them a great product search experience, but also brings a lot of revenue to online sellers. It bridges the gap between consumer and retailer benefiting both of them.

Inspired by a recent work [11] done in graph embedding techniques for information retrieval along with neural networks, we proposed to work on e-commerce product search data set to build a model for re-ranking product search results. Key challenges in the problem was with an anonymous data set to build a strong understanding in learning to rank models, search personalized algorithms, graph embedding techniques etc. We integrated product graphs along with basic item and user-based features which takes a lot of compute resources to implement different models to understand improvement at each step and ranked the products using Pairwise Ranking SVM approach.

Key contributions of our project work are as follows:

- We integrated Product Graph based embedding along with user-based feature embedding to propose an effective model for query full items for re-ranking
- With our evaluation method of calculating nDCG with the use of given iDCG, we could beat the CIKM CUP 2016 First Place model with a good margin.
- As compared to our baseline models, our best model result could produce NDCG score of 0.63095.

2 LITERATURE SURVEY

The learning to rank methods have received a lot of attention from the research community in information retrieval. SVM based learning to rank methods [5] have been widely covered in the literature, and clickthrough data based ranking methods [6], provided a base for our implementation. A survey of pairwise and listwise approaches in learning to rank [2] [1], covered the idea of encoding relevance in form of preference between pairs of documents for the same query, which led us to choose a pairwise approach for our ranking problem. The pairwise ranking model has been shown to perform significantly better than the traditional pointwise approach.

Specific to the competition, we referenced the paper [10] published by the winning team, to gain an understanding of feature engineering, query-item features and ensemble modelling.

Learning on graph-structured data [7] based on the variational auto-encoder helped us generate graph embeddings as another feature. Finally, a recent paper [11] that incorporated graph embedding based approach to this problem showed impressive results and motivated us to incorporate a graph based model.

3 DATASET DESCRIPTION

We used the E Commerce Product Search Dataset [3] released as part of the "CIKM CUP 2016 Track 2 : Personalized E-Commerce Search Challenge". This dataset was provided by DIGINETICA and its partners. The dataset included user sessions extracted from an e-commerce search engine logs, with anonymized user ids, hashed queries, hashed query terms, hashed product descriptions and meta-data, log-scaled prices, clicks, and purchases. The unique and challenging feature of the dataset was that it was completely anonymized with only meaningless hashed representations of users, search query tokens and product tokens.

The major statistics of the dataset are presented in Figure 1, eg. there are 184,047 products and 26,137 unique queries, average query length being 2.66 tokens and average length of product description being 5.12.
Figures 2 and 3 present a visualization of the distribution of some of the data. Figure 2 is a plot of ItemId and UserView, whereas Figure 3 is a plot of ItemID vs ItemClicks.

Statistics	Value	
# products	184047	
# unique queries	26137	
Vocabulary size	181194	
Length of queries	2.66 ± 1.77	
Length of product descriptions	5.12 ± 2.04	
# items per order	1.34 ± 0.96	
	Original	Chronological
# queries (train/test)	35615/16218	28380/3011
# search sessions	26880	21505
# browsing sessions	349607	242852
# clicks	37705	30160
# views	1235380	857008
# orders	13506	9130

Figure 1: Dataset Statistics

The problem statement was to use the data to re-rank the products returned to a user based on click and transaction data and using the state of the art learning to rank models to predict the relevance of products. We have to predict search

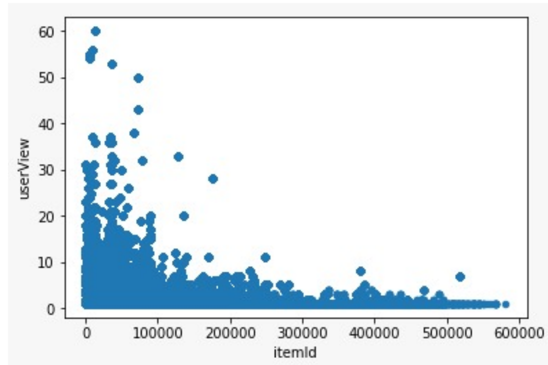

Figure 2: ItemID vs UserView

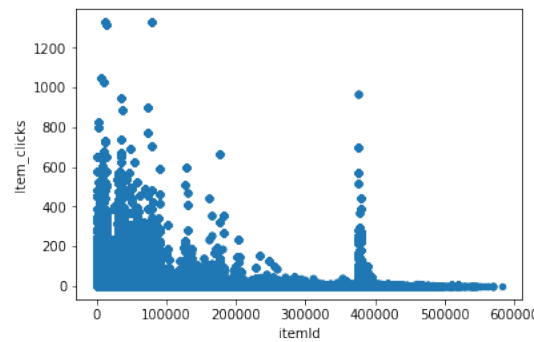

Figure 3: ItemID vs ItemClicks

relevance of products according to the personal shopping, search, and browsing preferences of the users.

4 PROPOSED SOLUTIONS

Our major goal was to combine the advantages of graph based features with other basic features of each product and query. The dataset was filtered to include only queries containing search tokens and non-null token IDs.Train Products were categorized with relevance label 2,1,0 based on clicked , viewed and no operation on them for the corresponding query. A positive product is one with relevance labels 2,1 and negative product is one with relevance label 0. A Pairwise ranking SVM was used to train the model. Given a query with a positive product and a negative product, the ranking loss is defined as [11],

$$L_{rank}(q, p_+, p_-) = h(score(q, p_+) - score(q, p_-)) \quad (1)$$

where $score(q, p)$ is the relevance score obtained for each query item pair from the model and h is the hinge loss of the rank SVM [9].

Various initial approaches were incorporated into the rank-wise SVM model. These include,

- **BASELINE MODEL :** Here, for a given query, the items were re-ranked by randomly shuffling their order without incorporating any specific features.
- **BASIC FEATURES :** About 10 global product, query, user features were extracted. Features include, item show count, item click count, item view count, item purchase count, number of users the product was shown to, number of users who clicked the product, number of users who purchased the product, click through rate, click value rate, item view rate. These basic features were computed for all products. For each query a positive product with its feature vector was fed along with another negative product with its feature vector to a Pairwise Ranking SVM to obtain the relevance score.
- **BASIC FEATURES + TOKEN EMBEDDINGS :** Here a 50 length token embedding was generated for each product token and query token using GENSIM's doc2Vec by creating a global space of all query tokens for query embedding and product tokens for generating product embeddings.
- **ADDITIONAL FEATURES :** Here, we additionally added a cosine feature, which is a cosine score between the query and product token embeddings. We also experimented with different combinations of features and token embedding sizes. This can be observed in the Section 5 of this paper.

GRAPH EMBEDDING BASED APPROACH

We went further and explored a Graph based embedding approach in order to capture the intrinsic relationship between the features of different products and users. As shown in Figure 4, we construct a product graph by establishing links between each pair of products if they co-occur together in the same SERP / search session. The basic (10) features of each product are embedded within the node in the product graph. A sample graph between 4000 products can be seen as in Figure 5

The constructed product graph is given to a Graph Auto Encoder(GAE) to generate its corresponding graph embeddings [7]. Figure 7 shows the model of GAE. The encoder of the Graph Auto Encoder consists of a Graph Convolution Network (GCN). We altered the number of nodes in Hidden layers of GCN and obtained a 16 and 32 sized embedding vectors.

Figure 4: Product Graph

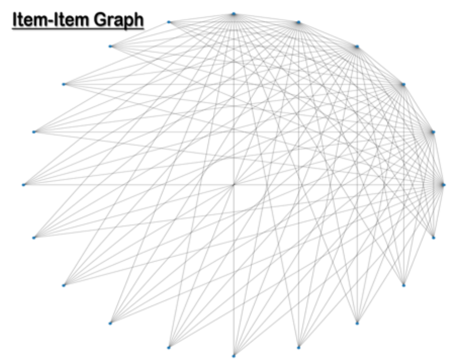

Figure 5: Constructed Product Graph for 4000 items

User Embeddings : Once we obtain the Graph Embedding of each product, we generate the user embedding of each user as $a = f(u, v, e)$, where, a is the user embedding, u is the aggregated(average) query token embedding of all his query search, v is the aggregated item token embedding of all the items that the user clicked or purchased in his complete search history, e is the aggregated graph embedding of those corresponding products. This user embedding is thus representative of the user behaviour from his complete search history.

Once these embeddings are obtained, we combine it with the query token embeddings and product token embeddings for each (query,item, user) triplet in order to obtain its corresponding relevance score. A block diagram of this model is illustrated in Figure 6.

5 EVALUATION & ANALYSIS OF RESULTS

The various models implemented have been tested based on the competitions' evaluation metric- nDCG score. The CIKM Cup [4] provides a solution set that consists of relevance labels and IDCG scores for each query-returned items pair. The evaluation metric provided by the competition computes a score as a weighted sum of nDCG scores from both query-less and query-full queries. For this project we report the nDCG of only query-full queries. We applied a progressive scheme, where each model applied is a modification on the previous. The most significant models applied and their

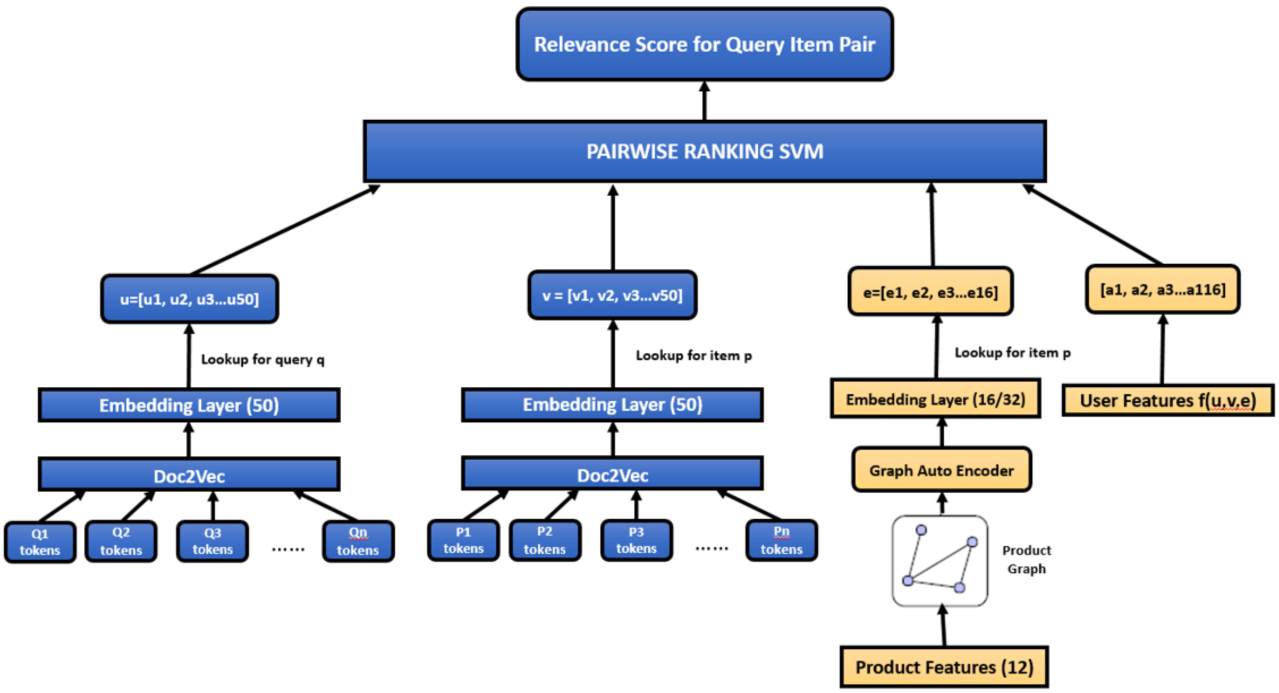

Figure 6: Model Architecture

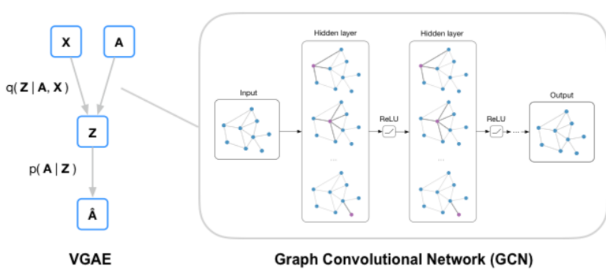

Figure 7: Variational Graph Auto Encoders.The model involves three components. The first is the inference model, where we learn the distribution of latent space $q(\mathrm{Z}|\mathrm{A}, \mathrm{X})$, and the parameters of the latent space (μ, σ). The second component is a 2-layer graph convolutional network (GCN) that is used to model the parameters of the latent space. The third component is the actual link-prediction model which learns the distribution $p(\mathrm{A}|\mathrm{Z})$.

respective scores are shown in Table 1. A naive approach would be to randomly shuffle the resultant items and present it to the user. We adopt this method to be our baseline for comparison. Models that used simple feature sets from elementary to basic incorporated with token based-embedding reported increasingly higher scores. However models that incorporated graph based-embedding noted higher nDCG scores. From these observations we can infer that graph based-embeddings are able to capture intrinsic components of our feature set and thus help build a better model.

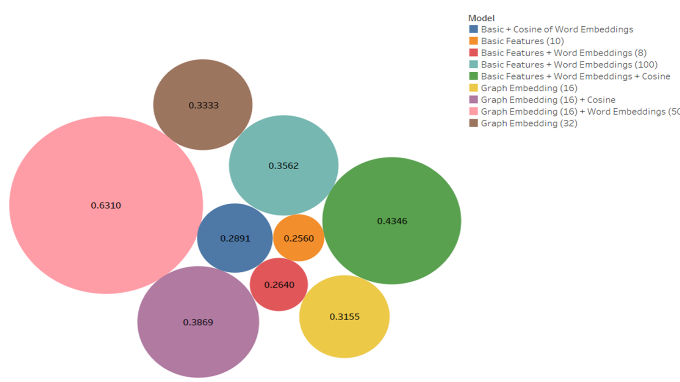

Figure 8: NDCG Distribution of Models: Depicts the models implemented proportional to their performance gain w.r.t the baseline

Our best model utilized a combination of graph embeddings and token embeddings to produce a nDCG score of 0.630, with a 177% improvement from the baseline model.

Model	nDCG	nDCG improvement %
Random Shuffle(Baseline)	0.22767	
Basic Features(10)	0.25595	12.42
Basic Features +Token Embeddings	0.26397	15.9441
Basic+ Cosine of Token Embeddings	0.28906	26.9644
Graph Embedding(16)	0.31546	38.5601
Graph Embeddings(32)	0.33333	46.4092
Basic Features + Token Embeddings(100)	0.35624	54.472
Graph Embeddings(16) + Cosine	0.38685	69.9168
Basic Features + Token Embeddings + Cosine	0.43456	90.8727
Graph Embeddings(16) +Token Embeddings(50)	0.63095	177

Table 1: Results

6 CONCLUSION

Graphs, such as social networks, word co-occurrence networks, and communication networks, occur naturally in various real-world applications. Analyzing them efficiently would yields great insight and help build better applications. Many approaches have been proposed to perform the analysis. Graph Based Embedding has been in the forefront gaining significant traction.[8] In this paper we leverage recent advances in graph embedding techniques to exploit graph structured data for feature extraction and apply it to the personalized search ranking problem. A data set provided by DIGINETICA which consisted of user sessions extracted from an e-commerce search engine log was used. Our models were designed with different approaches for feature extraction coupled with an implementation of rank SVM. After testing various combinations of features- graph based embeddings, token based embedding and rudimentary features, we see that the utilization of graph embeddings significantly bolstered our re ranking nDCG score. Thus we can ascertain that graph embeddings are able to determine convoluted traits of the feature set. The model which provided the best result, integrated product graph embedding along with token based embeddings to yield an nDCG score that was higher than the competitions first place at the time. Thus graph embedding can prove to be efficient for complex applications where the feature space can be represented as a convoluted graph.

7 FUTURE WORK

For further improvement in our work, we propose to integrate neural models along with our current feature set to re-rank product search results and compare the results with RankSVM. We also propose to build more personalized features based on sessions for each user. In this work, we use query full items, however this model could be extended for query less data points too. Here, we did use product graph, however we feel what if we can extract user query based graphs, which can be another useful information to improve our final nDCG score. We also wish to improve on user based features and extend this personalized Re-ranking model to web search too.

REFERENCES

[1] Christopher JC Burges. [n. d.]. From ranknet to lambdarank to lambdamart: An overview. ([n. d.]).

[2] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to Rank: From Pairwise Approach to Listwise Approach. (2007), 129–136. https://doi.org/10.1145/1273496.1273513

[3] competitions.codalab.org/competitions/11161#learn_the_details. 2016. CIKM Cup Track 2. (2016).

[4] http://cikm2016.cs.iupui.edu/cikm cup/. 2016. Personalized E-Commerce Search Challenge. (2016).

[5] Thorsten Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. (2002).

[6] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2017. Accurately Interpreting Clickthrough Data As Implicit Feedback. *SIGIR Forum* 51, 1 (Aug. 2017), 4–11. https://doi.org/10.1145/3130332.3130334

[7] Thomas N Kipf and Max Welling. 2016. Variational graph autoencoders. *arXiv preprint arXiv:1611.07308* (2016).

[8] Emilio Ferrara Palash Goyal. 2017. Graph Embedding Techniques, Applications, and Performance: A Survey. *Knowledge Based Systems, Volume 151, 1 July 2018, Pages 78-94, 2018* (2017).

[9] Jason DM Rennie and Nathan Srebro. 2005. Loss functions for preference levels: Regression with discrete ordered labels. In *Proceedings of the IJCAI multidisciplinary workshop on advances in preference handling*. Kluwer Norwell, MA, 180–186.

[10] Chen Wu, Ming Yan, and Luo Si. 2017. Ensemble Methods for Personalized E-Commerce Search Challenge at CIKM Cup 2016. *CoRR* abs/1708.04479 (2017). arXiv:1708.04479 http://arxiv.org/abs/1708.04479

[11] Yuan Zhang, Dong Wang, and Yan Zhang. 2019. Neural IR Meets Graph Embedding: A Ranking Model for Product Search. *arXiv preprint arXiv:1901.08286* (2019).