ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE
ENGINEERING AND TECHNOLOGY

DEEP HYBRID
RECOMMENDER SYSTEM

M.Sc. THESIS

Didem TÜRKER

Department of Computer Engineering

Computer Engineering Programme

MARCH 2020

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DEEP HYBRID RECOMMENDER SYSTEM

M.Sc. THESIS

Didem TÜRKER
(504161537)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Prof. Dr. Şule GÜNDÜZ ÖĞÜDÜCÜ

MARCH 2020

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

DERİN HİBRİT
ÖNERİ SİSTEMİ

YÜKSEK LİSANS TEZİ

Didem TÜRKER
(504161537)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Şule GÜNDÜZ ÖĞÜDÜCÜ

MART 2020

Didem TÜRKER, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 504161537 successfully defended the thesis entitled "DEEP HYBRID RECOMMENDER SYSTEM", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Şule GÜNDÜZ ÖĞÜDÜCÜ** Istanbul Technical University	....
Jury Members :	**Dr. Yusuf YASLAN** Istanbul Technical University	....
	Dr. Alper ÖZCAN Softtech Research Center	....

Date of Submission : 11 March 2020
Date of Defense : 25 March 2020

FOREWORD

I give thanks all the support I received from my advisor, Prof. Dr. Şule GÜNDÜZ ÖĞÜDÜCÜ, during the process of my Master's degree in general and thesis in particular.

I also do appreciate all the support I received from my family.

March 2020

Didem TÜRKER
Computer Engineer

TABLE OF CONTENTS

ABBREVIATIONS

MF	: Matrix Factorization
MLP	: Multi- Layer Perceptron
FM	: Factorization Machine
SVM	: Support Vector Machines
CNN	: Convolutional Neural Networks
RNN	: Recurrent Neural Networks
NCF	: Neural Collaborative Filtering
NCR	: Neural Collaborative Ranking
MSE	: Mean Squared Error
HR	: Hit Ratio
NCF with SI	: Neural Collaborative Filtering with Side Information
NCR with SI	: Neural Collaborative Ranking with Side Information
CMF	: Collective Matrix Factorization
CDL	: Collaborative Deep Learning

LIST OF TABLES

LIST OF FIGURES

DEEP HYBRID
RECOMMENDER SYSTEM

SUMMARY

With the increasing popularity of e-commerce platforms in recent years, recommendation systems have become highly popular. E-commerce platforms can offer items to the user with personalized information from large quantities of data that are often dirty and difficult to use. Not only e-commerce platforms, but also many social media and many platforms where users interact with items use recommendation systems. Recommendation systems also provide a quality user experience to users. In this way, users can easily reach the items according to their taste instead of getting lost among many products.

Traditional methods mainly based on the user-item interactions for recommendation. However user-item interactions mostly suffers from data sparsity problem. Data sparsity is the term used to describe the fact that not observing enough data. For example, recommender systems recommend thousands of products to hundreds of thousands of users, if you stored the data about user-product interaction in a matrix, it would be a huge amount of data consisting of lots of zeros. In addition, when a new user with no interaction with any item or a new item with no interaction with any user is included in the system, recommendation cannot be generated for that user or item using only user-item interactions. Therefore, beyond the user-item interactions, rich side information is a good source to increase the quality of recommendation. To mitigate the sparsity issue and improve the recommendation quality, we incorporate side information with user-item interactions. In recommender systems, explicit or implicit feedback of users is used. Implicit feedback (purchase/nonpurchase or click/nonclick etc.) represents opinion indirectly through analyzing user behaviour. Using implicit feedback is more challenging because of lack of negative feedback. But the high-quality explicit feedback, which provides direct input from users about their item interests, is the most functional input type to understand the user's exact response for an item. However, many systems do not have explicit feedback and it is difficult to collect this type of feedback. We have tested our framework with real world fashion retailer e-commerce data using implicit feedback. In our study, item purchasing is considered as positive feedback, and negative feedback is randomly selected from unobserved interactions. Consequently, lack of negative feedback of the implicit feedback was tried to be eliminated.

For long years, neural networks have significant role in many areas of computer science and have gained popularity in recommendation systems in recent years. Successful results have been obtained in understanding the complex and non-linear relationship between user-item interactions with neural networks.

In this study, artificial neural networks are used to increase the performance in understanding the complex and nonlinear relationship between user-item interactions.

Also, user-item interactions are combined with side information to solve the problem of data sparsity and improve the recommendation performance. In this study, two neural network architecture are proposed. In the first proposed model, our input is purchasing count of product. Mean Squared Error (MSE) is optimized and recommendation task is performed as rating prediction. Also MSE is used as evaluation metric of the first model. In the second proposed model, the output of the model is binary classification since input of this model is taken as 1 for positive feedback or 0 for negative feedback, then binary cross entropy loss is optimized. Top-k recommendation is made instead of rating prediction. Hit Ratio (HR) is used to evaluate second model.

DERİN HİBRİT
ÖNERİ SİSTEMİ

ÖZET

Son yıllarda e-ticaret platformlarının popülaritesinin artmasıyla birlikte öneri sistemleri oldukça önemli hale gelmiştir. Öneri sistemleri sayesinde e-ticaret platformları, kullanıcıya genellikle kirli ve işlemesi zor olan büyük miktardaki verileri kullanarak kişiselleştirilmiş öneriler sunmaktadır. Sadece e-ticaret platformları değil, aynı zamanda birçok sosyal medya platformu ve kullanıcıların sistemdeki nesneler ile etkileşime girdiği birçok platform öneri sistemlerini kullanmaktadır. Öneri sistemleri ayrıca kullanıcılara kaliteli bir kullanıcı deneyimi sağlamaktadır. Bu sayede kullanıcılar; birçok ürün arasında kaybolmak yerine, zevklerine uygun ürünlere kolayca ulaşabilmektedirler. Literatürde işbirlikçi filtreleme ve içerik tabanlı yöntemler gibi birçok öneri sistemi önerilmiştir. İşbirlikçi filtreleme, kullanıcıların ürünler ile olan etkileşimlerine (puanlama, satın alma vb.) göre birbirlerine benzer kullanıcıları belirleyip bu doğrultuda ürün önerileri sunmaktadır. İçerik tabanlı filtrelemede ise içeriklerin benzerliği kullanılarak öneride bulunulur. Ancak içerik tabanlı ve işbirlikçi filtreleme yöntemlerinin de bazı kısıtlamaları vardır. Bu nedenle işbirlikçi ve içerik tabanlı yaklaşımların birleştirilmesi ile hibrit (melez) yöntemler önerilmiştir.

Geleneksel öneri yöntemleri, temel olarak öneri için kullanıcı-ürün etkileşimlerine dayanmaktadır. Gerçek dünyadaki durumlarda, kullanıcı-ürün etkileşimleri genellikle seyrektir. Bu durum kullanıcı-ürün etkileşimi temelli yöntemlerin öğrenme performansında önemli ölçüde düşüşe neden olur. Ayrıca, herhangi bir ürünle etkileşimi olmayan yeni bir kullanıcı veya herhangi bir kullanıcıyla etkileşimi olmayan yeni bir ürün sisteme dahil edildiğinde, yalnızca kullanıcı-ürün etkileşimine dayalı öneri sistemleri bu kullanıcı veya ürün için öneri oluşturamamaktadır. Bu nedenle, kullanıcı-ürün etkileşimlerinin ötesinde, zengin ek bilgi, öneri kalitesini artırmak için iyi bir kaynaktır. Veri seyrekliği sorununu azaltmak ve öneri kalitesini iyileştirmek için ek bilgileri kullanıcı-ürün etkileşimleriyle birleştiriyoruz. Öneri sistemlerinde, kullanıcıların açık veya örtülü geri bildirimleri kullanılır. Örtülü geri bildirim (satın alma / almama veya tıklama / tıklamama vb.), kullanıcı davranışını analiz ederek dolaylı olarak kullanıcının ürün hakkındaki görüşünü temsil eder. Olumsuz geri bildirim eksikliği nedeniyle örtülü geri bildirim kullanmak daha zorlayıcıdır. Ancak, kullanıcıların ürün hakkındaki görüşlerini doğrudan belirten yüksek kaliteli açık geri bildirim, kullanıcının bir ürün için tam görüşünü anlamak için en işlevsel geribildirim türüdür. Açık geri bildirimleri ise toplamak zordur çünkü kullanıcı her zaman puanlama yapma, beğenip / beğenmeme veya yorum yapma gibi ürün hakkındaki görüşünü direkt yansıtan işlemleri yapmaya istekli olmayabilir. Kullanıcılar bu tarz işlemleri yük ve zaman kaybı olarak görebilmektedir. Çalışmamızda modellerimizi örtük geri bildirimler kullanarak, gerçek dünyadaki bir moda perakendecisine ait e-ticaret sitesi verileriyle test ettik. Verisini kullandığımız e-ticaret sitesi, amazon.com

gibi temel sarf malzemeleri satmadığı, sitede yer alan ürünler; moda, hava durumu, özel gün (yılbaşı, sevgililer günü vb.) vb. nedenler ile sıklıkla değiştiği için önerilen ürünü müşterinin sitede bulamaması gibi durumların önüne geçmek için spesifik ürün önermek yerine "Kadın Kazak", "Genç Pantolon" gibi müşterinin meyilli olduğu ürün gruplarının tahminine odaklanıyoruz. Veri kümemiz 544 adet e-ticaret sitesinde tanımlı ürün grubu ve 1.201.126 müşteriden oluşmaktadır. Bu veri bir yıl içerisinde gerçekleşmiş, 5,5M müşteri - ürün grubu satın alma ilişkisini içermektedir. Müşterilerin e-ticaret sitesinden siparişlerini verdikleri haftanın günleri ve ayları, müşteri tipi (personel, üye, üye olmadan sipariş veren vb.), e-posta ve sms izinleri, kredi kartı ve ödeme tipleri, sipariş verdikleri şehirler ve müşterilerin siparişlerini verdiği platformlar, modellerimize entegre ettiğimiz müşteriye ait ek bilgilerdir.

Bir kullanıcının tercihi genellikle iki yönlüdür; pozitif (kullanıcının tercih ettiği, beğendiği) ve negatif/nötr (kullanıcının sevmediği veya özel bir ilgi duymadığı). Amacımız kullanıcının beğenisine hitap edecek ürünleri önermek ise her iki tercih de bizler için eşit derecede öneme sahiptir. Müşteri-ürün grubu arasındaki satın alma işlemi pozitif geri bildirim olarak değerlendirilirken, negatif geri bildirimler gerçekleşmeyen etkileşimler içerisinden rastgele örneklenmektedir. Eğitim kümemizde yer alan her müşteri-ürün grubu satın alma işlemi için belirli sayıda rastgele olarak müşterinin satın almadığı ürün grupları arasından negatif örneklem seçilmektedir. Seçtiğimiz negatif örneklem kümesi de eğitim kümemize dahil edilir. Bu sayede örtük geribildirimin negatif geri bildirim eksikliğini gidermeye çalıştık.

Yapay sinir ağları, uzun yıllardır bilgisayar biliminin birçok alanında kullanılmaktadır ve son yıllarda öneri sistemlerinde popülerlik kazanmıştır. Yapay sinir ağları ile kullanıcı-ürün etkileşimleri arasındaki karmaşık ve doğrusal olmayan ilişkinin anlaşılmasında başarılı sonuçlar elde edilmiştir. Bu tez kapsamında, iki farklı yapay sinir ağı mimarisine etkili bir şekilde ek bilginin dahil edilmesi ile öneri performansının artırılmasına odaklanılmıştır. Modelimizin Genelleştirilmiş Matris Faktorizasyonu katmanında doğrusal olmayan matris faktorizasyonu gerçekleştirilmektedir. Çok Katmanlı Algılayıcı katmanındaki ağırlık vektörleri ve yanlılık parametreleri ile karmaşık, doğrusal olmayan kullanıcı-ürün etkileşimleri arası ilişkiler öğrenilmektedir. Son olarak Genelleştirilmiş Matris Faktorizasyonu katmanının çıktısı, Çok Katmanlı Algılayıcının son gizli katmanının çıktısı ve ek bilgilerin öğrenildiği yapay sinir ağının çıktısı modelin son gizli katmanında birleştirilmektedir. Bu tez kapsamında önermiş olduğumuz hibrit sistem, matris faktorizasyonunun sağladığı doğrusallık ile yapay sinir ağlarının doğrusal olmayan özelliğini birleştirerek öneri performansında iyileşme sağlamaktadır.

Bu tez kapsamında iki farklı derin hibrit öğrenme mimarisi sunulmaktadır. Mimarilerimizde kullandığımız ileri beslemeli yapay sinir ağı sayesinde kullanıcı-ürün etkileşimleri arasındaki doğrusal olmayan, karmaşık ilişkinin öğrenilmesindeki başarım artırılmaktadır. İşbirlikçi filtreleme işlemine ek bilgiyi de ekleyerek soğuk başlangıç problemi ve veri seyrekliği problemlerine çözüm sağlanmaktadır. Derin öğrenme ve ek bilginin güçlü yönlerinden faydalanarak işbirlikçi ve içerik tabanlı yaklaşımların kısıtlamalarının hafifletilmesini ve öneri performansının artırılmasını sağlamaktayız.

Öneri sistemlerini sınamak için yöntemlerden biri, kullanıcı-ürün matrisinin sıfırdan farklı olan kısımlarını belirli bir oranda rastgele silip, daha sonra öneri sistemi vasıtasıyla bu silinen kısımları geri tahminlemektir. Satın alınan ürün sayısını geri bildirim olarak kullandığımız Derin Hibrit İşbirlikçi Filtreleme modelimizin testi için veri kümesinin %0,5 kısmını rastgele siliyoruz. Daha sonra öneri performansını sınamak için ortalama kare hata değerini kullanıyoruz. İkili geri bildirim (satın aldı/almadı, 1/0) kullandığımız Derin hibrit İşbirlikçi Sıralama modelimizi sınamak için birisi dışarıda doğrulama yöntemini kullanıyoruz. Kullanıcıların %10 'ununu test için ayırıyoruz. Test için ayırdığımız kullanıcıların son satın aldıkları ürün grubunu test ögemiz olarak belirliyoruz, 99 adet kullanıcı ile etkileşmemiş ürün grubunu rastgele seçiyoruz. Ardından modelimiz test ögemizi 100 adet rastgele seçtiğimiz ürün grupları içerisinde sıralıyor. Sıralama performansını İsabet Oranı ile değerlendiyoruz.

Çalışmamız kapsamında Derin Hibrit İşbirlikçi Filtreleme ve Derin Hibrit İşbirlikçi Sıralama modellerimizin farklı parametre düzenleşimlerinde öneri başarımını değerlendirmek için gerçekleştirdiğimiz deneyler ve sonuçları, modellerimizin literatürde baz aldığımız modellere göre öneri başarım karşılaştırmaları sunulmaktadır. Yaptığımız deneyler ile yapay sinir ağlarının kullanıcı-ürün etkileşimlerindeki doğrusal olmayan ve karmaşık ilişkiyi yakalamanın etkin bir yolu olduğu, ek bilgileri modele dahil ederek hibrit bir yaklaşım uygulamanın, veri seyrekliği ve soğuk başlangıç problemlerine çözüm sağlayarak öneri kalitesini artırabileceği kanıtlanmıştır. Ek olarak literatürde yaygın olarak bilinen yöntemlere kıyasla öneri başarımında artış sağlandığı gösterilmiştir.

1. INTRODUCTION

The recommender systems have gain importance during the past few years, especially increasing competitions between e-commerce platforms have made recommendations systems more critical to take advantage of competition. Also social media services and e-commerce platforms provide large amount of data become available. Consequently, requirement of processing the massive data to provide useful personalized recommendations rises.

Some kind of approaches has been used to provide recommendation like Collaborative Filtering (CF) and Content Based Filtering (CBF).

Collaborative Filtering: In Collaborative Filtering approach, recommendation is based on the past user-item interactions. CF recommends items to users by the help of their interaction similarities [4]. For example, if one of the two similar users liked a movie, the other could be more probably to like that movie. In collaborative filtering approach, a user-item matrix which consists of ratings given by users for items is used. Users are matched by measuring the similarities of their past interactions to make recommendations.

Content-Based Filtering: This approach recommends items based on the similarity between the features of the items and/or a users [5, 6]. The features / content can be any data available about the item or user. For instance, if the item is a movie, than the content may be genre or director of that movie.

Hybrid Recommendation: Hybrid approaches combine different recommendation methods to improve recommendation performance. There are various ways to implement hybrid recommender systems: by using collaborative-based and content-based methods independently and then fusing them or by integrating the approaches into single model [7–9].

There are various algorithms and methods to make recommendations which can use past user-item interactions or content information; however content based and collaborative filtering methods suffer from some limitations.

Well known traditional recommendation methods are mostly based on user-item interaction (viewing, rating or purchasing) matrix. However, in real world cases the user-item interaction matrix usually suffers from sparsity issue, causing user-item interaction based methods to degrade significantly in learning performance. Furthermore, another restriction for user-item interaction based methods is how to offer recommendations when a new item or user that has no previous interaction emerges in the system. This problem called the "Cold Start Problem" in literature. To tackle the data sparsity and cold start problems, it is inevitable for user-item interaction based methods to use additional information related with the items or users, also called as the side information, therefore hybrid CF methods are more preferable in recent years. These problems represent the main limits to implement a well-worked recommendation system without any side information.

Among the various recommendation methods, Matrix Factorization (MF) [10, 11] is mostly preferred compared with the others, which uses a vector of latent features to state an item or a user. Matrix Factorization models an interaction between a user and an item as the dot product of latent vectors.

However, sometimes the dot product are not adequate to catch the complex relationship between user-item interaction data. Several studies have shown that deep learning is quite effective in capturing nonlinear complex relationships in user-item interaction. In recent years, deep learning has gained tremendous popularity by handling barriers of traditional recommendation models and improving recommendation performance [2, 3, 11, 12]. Some deep learning models take users' or items' contents as input. For instance, Zhang et al. [13] use textual and visual items' features and Liang et al. [14] use acoustic features from audio signals as semantic input for deep neural network. These works figure out the importance of using deep learning models for recommendation systems.

In this thesis, to overcome the challenges above and avoid the inevitable effects of them, we propose two different neural recommendation models with side information.

In this thesis we focus on improving the accuracy of a recommendation system developed for fashion industry where the data is sparse and the system has a prevalent cold start problem. To this purpose, a hybrid strategy has been built by the help of two recently proposed different frameworks, "Neural Collaborative Filtering Framework" and "Neural Collaborative Ranking Framework". In NCF framework, He, Xiangnan, et al. [2] offered a method which uses a MLP (multi-layer perceptron) to learn the function of user–item interaction and then they ensemble MF and MLP together. Song, Bo, et al. combined their classification method with the NCF framework and built their own neural collaborative ranking framework. Their neural network architecture evaluates the item pairs as less desirable or more in terms of user preference and they assume that the latent structure of latent factors will be captured effectively by the neural network [3]. These two papers focuses on implicit feedback of users [2, 3]. Implicit feedback represents opinion indirectly through analyzing user behaviour. Explicit feedback, which provides direct input from users about their item interests. For instance, watching a movie may not be enough to understand users' reaction, but users' rating about the movie reflects the exact opinion of them. In this thesis, we focus on both of these input types.

The below items are provided within the scope of this thesis.

– We propose deep neural networks based recommendation framework which combines side information of users and user-item interaction data. We combined the advantage of non-linearity of artificial neural networks with the advantage of linearity of matrix factorization.

– By effectively integrating the side information into our model, we significantly increase model performance. Using side information can improve the ability of detecting true interactions also the skill to correctly rank items according to user behaviors. We verify these two improvement on two different Neural recommender system; neural collaborative filtering framework [2] and neural collaborative ranking framework [3].

– We have tested our framework with real world fashion retailer e-commerce data using implicit feedback.

The thesis is structured as follows: Chapter 2 explains the referred methodologies; Chapter 3 contains our proposed methodologies and experimental results; Chapter 4 discusses method results and future works.

1.1 Purpose of Thesis

The purpose of this thesis is to investigate the effectiveness and robustness both of the deep learning models and adding side information in recommendation systems. We aimed at improving the performance of recommendation system by adding side information. In addition, by combining the well known matrix factorization method and artificial neural networks, we take advantage of the two methods.

For recommender systems, matrix factorization is a commonly used technique. The user-item interaction matrix which represents the users' own preferences for each item to make recommendation is utilized by Matrix Factorization. In real world cases, user-item interaction matrix suffers from sparsity problem because there exist large amount of users and items in data and it is impossible to react all of them with each other. Data sparsity problem is causing to decrease the recommendation quality of traditional methods significantly. In real world cases, when a new user with no preference is included in the system, recommender systems based on only user-item interaction cannot provide a recommendation for that user. This case called as the cold start problem. In this thesis, to overcome the cold start and data sparsity problem, we propose two different neural recommendation models with side information. By the help of overcoming these issues, the recommendation system can offer items to even the users who reacted with the products very first time. In this thesis, we aim at taking advantage of the side information to improve real world recommender systems, where the side information is convenient and available. Nevertheless, dot product may not be effective to detect complicated and nonlinear relations in user-item interactions. For capturing nonlinear complex relationships in data, we took advantage of deep learning. We aim to provide a baseline framework both using neural networks for recommendation and adding side information to the neural network system.

For building a general framework for both collaborative filtering and side information, we use feed-forward neural networks and train our deep learning model with pointwise loss and also pairwise learning.

2. RELATED WORKS

In this section, we examine the previous recommendation system studies which deal with side information. The side information could be unstructured texts for example reviews or descriptions; images like movie posters; and features or labels identifying the product or user's features. These type of side information can be gathered from various rating systems in different kind of fields and e-commerce platforms on a daily basis.

In order to overcome data sparsity and cold starts problems hybrid methods have gained popularity in recent decades. Singh and Gordon [15] in Collective Matrix Factorization (CMF) have combined side information into matrix factorization to learn effective latent factors. However, their method utilize the side information as regularizations. CMF factorizes the matrix of ratings together with other side information matrices by assuming user factors are the same in the matrices of ratings and side information. Such an approach can improve performance by adding side information but can introduce significant bias in sparse rating matrix and side information. Agarwal, Chen, and Long [16] discussed drawbacks of CMF in sparse datasets. In this thesis, we learn latent factors separately from the rating matrix and combine outputs of the Matrix Factorization layer and the MLP layer with the factors we learned from side information in the last prediction layer.

Sen et al. [17] estimated the ratings of users for items based on inferred preferences for labels. These labels are using to detect the similarity of items. By forecasting the label choices in terms of an item, the model in Gedikli et al.'s works [18, 19] enhance this work. The obstacle in the approaches above is that they educe the expectations for the user's own labels according to given information. However, the model is unsuccessful in predicting any label for an unrated item. The cold start problem represents a major drawback on the aforementioned works.

Deep learning is being preferred in recent works as one of efficient methods of recommender systems to learn effective representations [20]. In this way, integrating deep learning in recommendation systems improves to learn latent factors from both of item-user interaction matrix and rich additional side information. Moreover, in modeling users/items on the internet, where heterogeneous data is so usual, using deep learning is unavoidable to detect complex relations between user-item interactions. For example, when working with heterogeneous unstructured data which can be textual data (reviews [21], tweets [22] etc.) or image data (instagram posts, movie posters, product images etc.), CNNs and RNNs turn out to be more effective methods. Several studies in the literature have integrated side information such as social networks, user reviews or item content to make the performance of the recommendation better. From the belief that users should have close relevance with their followers in the social platforms, these works [23–25] integrate social networks for social recommendation.

MF is one of the most widely used successful methods, however the dot product can be insufficient to detect non-linear and complicated relations between user-item interactions. Some studies have made use of deep learning for the collaborative filtering task. Salakhutdinov, Mnih,and Hinton [26] in their research combines deep learning and CF, He, Xiangnan, et al. [27] proposed a neural collaborative filtering framework to replace the inner product with non-linear interaction function by a feed forward neural network and they achieve hopeful results. Nevertheless, they don't integrate side information. Only user-item interactions can not be enough data in real world cases. Thus, we examine the use of deep learning to extract significant information from both side information and user-item interactions, then combine these data to improve our prediction quality.

Even though Deep Learning Models achieve excellent performance compared to other model-based recommendation approaches, the works on how to effectively integrate side information into Deep Learning Models has not reached its full potential. These research issues, on the other hand, have been well studied in recent years for Latent Factor Models, which could provide inspiration for the development of side information for Deep Learning Models [28].

Strub, Mary and Gaudel in Hybrid Collaborative Filtering with Neural Networks [29] study introduce a collaborative filtering approach based on Stacked Denoising Autoencoders. Autoencoder is a simple unsupervised feed forward neural network that learns how to efficiently compress and encode data then learns how to reconstruct the data back from the reduced encoded representation to a representation that is as close to the original input as possible. Autoencoders generally, are used in the process of removing noise from an image, audio or a document. With this approach, they aim to predict ratings which are unrated from the users. For items that are not rated by the user, user element interactions are perceived as noise by the autoencoder. If you have correlated input data, the autoencoder method will work very well because the encoding operation relies on the correlated features to compress the data. Although in real world cases, users preferences change a lot so rating matrix is not much correlated also, the result is very dependent on the inputs. Therefore, as data sparsity increases, recommendation performance will decrease. Autoencoders, which are very successful in cases where data sparsity is low, have performance problems even if side information is added in every layer of network in real world recommendation systems where data sparsity is more than 95% also this method is troublesome due to a similar deficiency of CMF which factorizes together the item rating matrix and the additional side information.

Dong, Xin, et al. [12] used deep neural network method only to factorize latent factors from the rating matrix and the additional side information then they use dot product to predict ratings. However the dot product can be inadequate to identify non-linear and complicated user-item interaction relationships. In this thesis, we tackle the limitation by learning from the rating matrix and side information the interaction function using Deep Neural Network.

2.1 Preliminaries

2.1.1 Matrix Factorization

In 2006, Netflix started a contest to increase the state of its recommender performance. The first team that can increase on the Netflix algorithm's root-mean-square error by 10 percent or more than wins a $1 million prize. With this contest, after the significant improvement of recommendation performance, Matrix Factorization (MF) techniques have gained popularity [30]. Matrix factorization is utilized in several well performed latent factor models. Both users and items can be mapped to latent factor space from item-rating matrix by MF. MF decomposes the original rating matrix R into two low rank vectors U and V then user's rating or interest for an item is predicted by the inner product of U and V [30]. Illustration of Matrix Factorization can be seen in Figure 2.1.

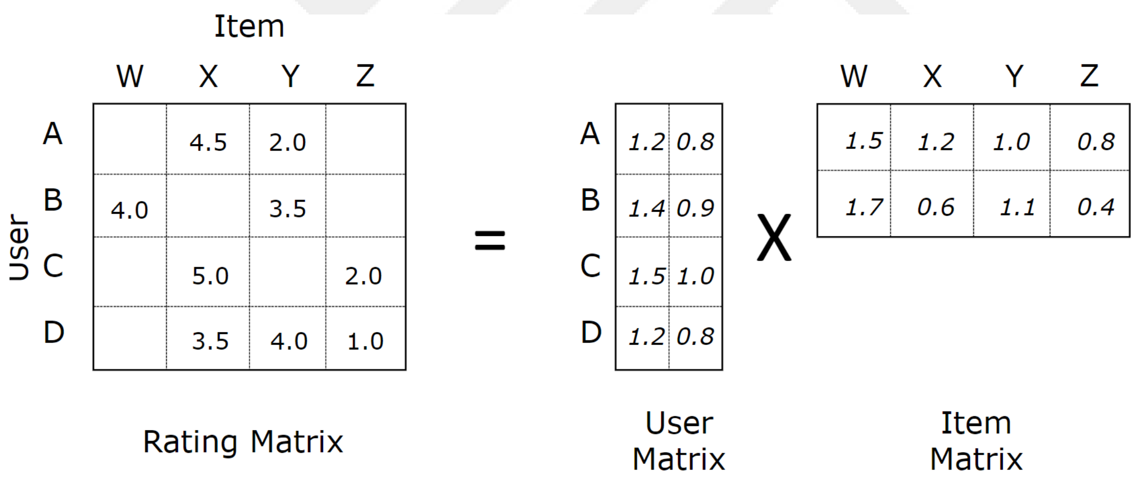


Figure 2.1 : Matrix Factorization

$$R = VU^T \tag{2.1}$$

Let r is rating defined for user u and item i, p is row of V for a user and q is the column of transpose(U) for a specific item i. So equation will become:

$$r_{ui} = p_u q_i \tag{2.2}$$

So the basic task is to make the mean square error minimum. MSE function can be represented as:

$$MSE = \frac{1}{|R|}(Y - \hat{Y})^2 = \frac{1}{|R|}\sum_{r_{ui}\varepsilon R}(r_{ui} - p_u q_i)^2 \tag{2.3}$$

As the main task is to minimize the error for this purpose it has been differentiated, hence new equation will become:

$$\frac{dMSE}{dq_i} = -2(Y - \hat{Y}) * p_u \tag{2.4}$$

$$\frac{dMSE}{dp_u} = -2(Y - \hat{Y}) * q_i \tag{2.5}$$

a is learning rate generally its value is very small. Hence the updated value for p and u will be now become:

$$\hat{p_u} = p_u + a\frac{dMSE}{dp_u} \tag{2.6}$$

$$\hat{q}_i = q_i + a\frac{dMSE}{dq_i} \tag{2.7}$$

However, by the use of dot product to predict complicated user–item interactions in the low-dimensional latent space can not be efficient to capture non-linear relations of user–item interactions.

A simple example of explicit and implicit feedback rating matrices can be seen in Figure 2.2. All implicit feedback instance is considered as 1 and MF techniques will not give expressive results because of the lack of negative instances. To overcome this, we determine a certain number of negative feedback sample sets for each positive feedback in our models.

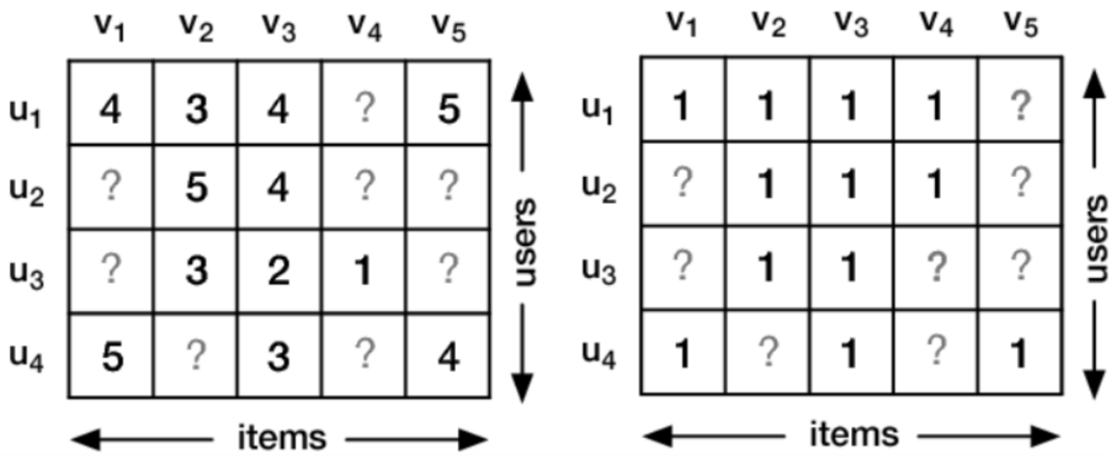


Figure 2.2 : A simple example illustrating the rating matrices for explicit feedback and implicit feedback [1]

2.1.2 Neural Collaborative Filtering

This paper is one of the studies that we have taken as a baseline while developing this work. In the Neural Collaborative Filtering paper, their goal is to tackle above MF limitations by exploring the deep learning for collaborative filtering. They combine MF and Multi-layer Perceptron (MLP) that is a class of feedforward artificial neural network under the Neural Collaborative Filtering framework; with this framework they take advantages of the strongest aspects of these two different techniques; linearity of MF and non-linearity of MLP. In this paper, user and item properties are joint by concatenating these vectors. However, vector concatenation is not enough for modelling user-item interactions for collaborative filtering. To learn the user-item interactions, they add hidden layers on the concatenated vector using MLP. Neural Collaborative Filtering architecture can be seen in Figure 2.3.

Their model has played a key role to guide the further future studies for the development of recommendation methods by using deep learning and they achieve promising results.

However, they only consider user-item interactions and use implicit feedback of users. And because of this their framework suffers from data sparsity and cold start problem. Because of their success to capture non-linear relations of user–item interactions, in this work we took this approach one step further, where we concatenate side information and also use explicit feedback. Besides ability of capturing non-linear

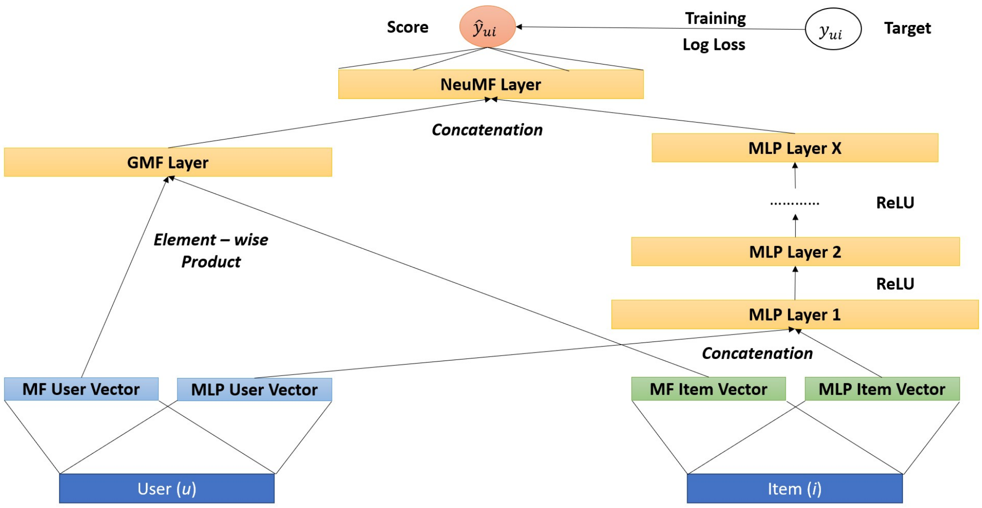


Figure 2.3 : Neural Matrix Factorization Model [2]

relations of user-item interactions, handling data sparsity and cold start problems are improved the quality of recommendation.

2.1.3 Neural Collaborative Ranking

If a user interact with an item Neural Collaborative Filtering framework labels this interaction as a positive instance and all other possible interactions as negative instances, therefore this hypothesis suffers from two limitations: the first of them caused by sparsity of training data, negative instances dominate the training data; secondly, assuming the uninteracted items as negative user feedbacks can fail the model. Because the user can be unaware about the item and didn't react with it yet. To overcome the problem above in Neural Collaborative Ranking paper uses pairwise learning to train a feed-forward neural network [3]. They don't consider that the uninteracted item interaction with user doesn't need to be negative, simply because they consider uninteracted items are less favored than interacted items . Except for the learning method, the entire main structure of the Neural Collaborative Ranking model is the same as the Neural Collaborative Filtering model.

In Figure 2.4 , Neural Collaborative Ranking model architecture can be seen. u is a user, i is an observed item, j has not been observed by u. Neural Bayesian Personalized Ranking (NBPR) and Deep Neural Collaborative Ranking (DNCR) layers learn different embeddings, and then combine the two models by concatenating their last hidden layer.

In this work, we combine side information with the user-item interaction data. And we aim to improve recommendation quality of Neural Pairwise Ranking (NeuPR) layer.

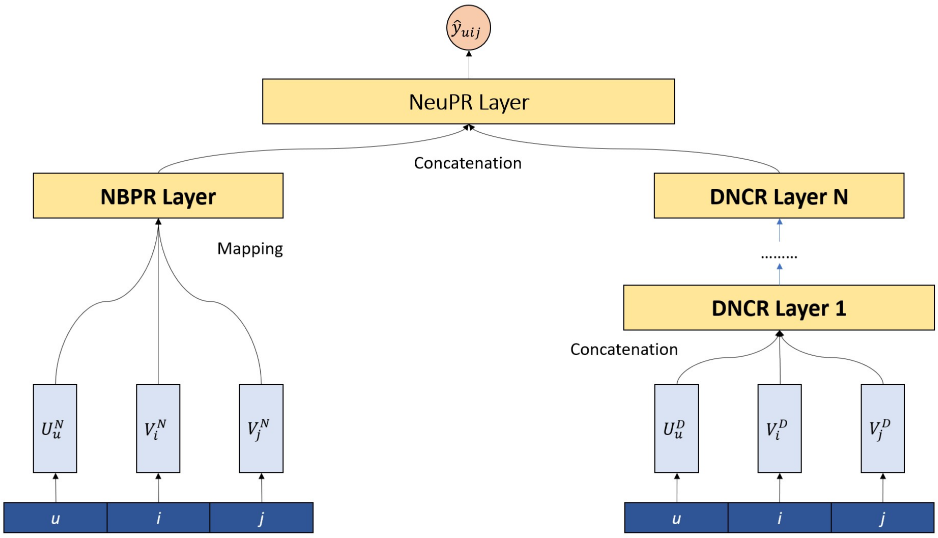


Figure 2.4 : Neural pairwise ranking model [3]

3. METHOD AND EXPERIMENTAL RESULTS

3.1 Dataset

We experimented our proposed method on a real-world problem: we applied our model to a fashion retailer e-commerce site data. This dataset consists of one year historical order data and 5.5M user-item interactions. We recommend product groups which represent categories like teenage trousers or women dress etc. rather than products to customers. This e-commerce historical order dataset consists of 544 product groups and 1,201,126 customers. The reason why we use a product group instead of a special product depends on the fact that the data we use belongs to a textile e-commerce company. Because the products on this e-commerce platform are not the basic consumables like the ones on amazon.com. The variety of products they offer to the users in the textile sector is very high, but the stock quantities are low. Therefore, if the product is recommended directly, the user may not find the appropriate size of the product in stock and there will be a need to train the model at very frequent intervals.

Also we use customer's side information. These features are converted into one hot encoded form. Customers' features include information of days and months when customers have completed their orders from e-commerce, types of customers in terms of their registration status, their permissions for mail and sms, credit card and payment types, order cities and platforms where the customers gave orders at.

Our experiments are apart from our recommendation performance comparisons, we performed with 1M data, which we randomly sampled from our 5,5M dataset. However, our latest performance comparisons show the results we have achieved over our entire 5,5M dataset.

Recommendation systems can offer items to users using different types of feedback. Explicit feedback takes into account direct preferences of the users about how they liked or disliked an item. In explicit feedback it is simple to distinct negative and positive feedback from each other but explicit feedback is challenging to collect.

Implicit feedback doesn't directly project the users' preferences but it acts as a representation of users' interest [10]. Browsing history, clicks and buyings, number of times a song is listened are examples of implicit feedback [31]. Unlike explicit feedback, in implicit feedback there is a lack of negative feedback [32].

There are good and bad aspects of both types of feedback. In order to avoid the negative effect of implicit feedback lack of negative feedback, we create negative feedback sample set of interactions that have not yet been observed for each positive feedback. We use number of purchased product groups by customers as the users' feedback in Neural Collaborative Filtering with Side Information method. Also, in Neural Collaborative Ranking with Side Information method we set feedback of user as 1, if a user bought an item from this product group, otherwise is 0 for that product group.

3.2 Proposed Method

In this work, we have utilize two publicly available deep learning models. We have built general framework for both side information and collaborative filtering.

Following Neural Collaborative Filtering framework, we concatenate side information with output of GMF layer and output of last hidden layer of MLP. And we use implicit feedback of users. Using feed-forward neural network can improve ability of learning non-linear relations of user-item interactions and combining side information can overcome cold start and data sparsity problems.

Both of the mentioned neural networks have embedding layers, embedding vectors can be seen as the latent vectors of user and item. Let p_u as the user latent vector and q_i as item latent vector. The mapping function of the first CF layer can be defined as:

$$\phi_1(p_u, q_i) = p_u \odot q_i \tag{3.1}$$

Where $\odot$ denotes the element-wise product of latent vectors. Next step is to project this output to the output layer of the model:

$$\hat{y}_{ui} = a_{out}(h^T(p_u \odot q_i)) \tag{3.2}$$

where a_{out} and h denote the activation function and edge weights of the output layer, respectively.

And for the last step, GMF embeddings, MLP embeddings and the output of the dense layer of neural network for learning side information are concatenated in last hidden layer. We use ReLU activation function on MLP Layers and don't use any activation function on NeuMF Layer. Figure 3.1 describes our propose method, the equation is given as follows:

$$\phi^{GMF} = p_u \odot q_i \tag{3.3}$$

$$\phi^{MLP} = a_L \left(W_L^T \left(a_{L-1} \left(...a_2 \left(W_2^T \left[\begin{pmatrix} p_u^M \\ q_i^M \end{pmatrix} \right] + b_2 \right) ... \right) \right) + b_L \right) \tag{3.4}$$

$$\phi^S = W_s + b \tag{3.5}$$

Where S is the side information of users. For learning side information, we use dense layer with four neurons. And side information is not so complex data thus we don't use any activation function on this layer.

$$\hat{y}_{ui} = \sigma \left(h^T \begin{bmatrix} \phi^{GMF} \\ \phi^{MLP} \\ \phi^S \end{bmatrix} \right) \tag{3.6}$$

To learn the parameters of our Neural Collaborative Filtering with Side Information model, we use the mean squared error loss function:

$$MSE = \frac{1}{|R|} \sum_{y_{ui} \varepsilon R} (y_{ui} - \hat{y}_{ui})^2 \tag{3.7}$$

Following Neural Collaborative Ranking framework, we concatenate side information with output of NBPR layer and output of DNCR layer. After that, the hidden layers are taken into the concatenated vector as input. In this framework, we use implicit feedback of user. However, while the count of products purchased in our first model is considered as feedback, in this model, the feedbacks are purchased / not purchased (1/0). Sigmoid function has been used as NeuPR layer's activation function in this

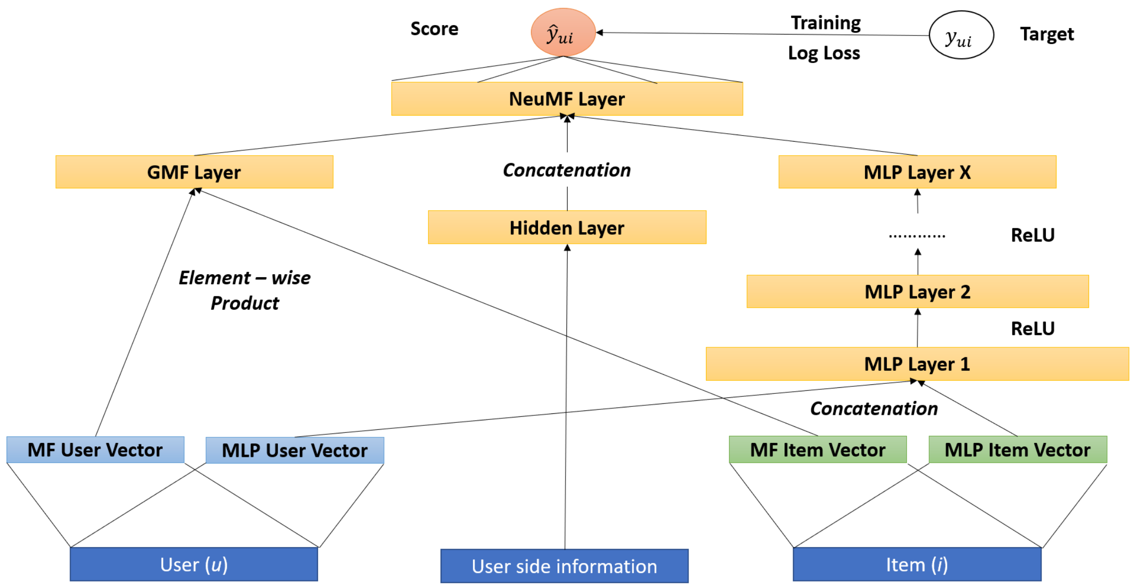


Figure 3.1 : Neural Matrix Factorization Model with Side Information

framework, because it is aimed to limit the output between [0,1]. For the hidden layers we use ReLU activation function. Figure 3.2 describes our proposed architecture of Neural Collaborative Ranking with Side Information. Neural Collaborative Ranking with Side Information represents a user's choice between two items, which one of the item is observed (interact) by the user and the other is not.

The last layer of the Neural Collaborative Ranking with Side Information is the prediction layer which maps previous layers' output to the predicted score $\hat{y}_{uij}$ explains the extent user u prefers item i to item j. Predicted score $\hat{y}_{uij}$ can be formulated as below:

$$\hat{y}_{uij} = f_{out}\left(f_N\left(...f_2(f_1\left(U_u, V_i, V_j, S_u\right)\right)\right) \tag{3.8}$$

f_{out} is the activation function of the prediction layer. In this case, it is the sigmoid function. To learn the parameters of our Neural Collaborative Ranking with Side Information model, we use binary cross-entropy loss function:

$$BCE = \frac{1}{|D_s|} \sum_{u,i,j\varepsilon D_s} y_{uij}.log\left(\hat{y}_{uij}\right) + \left(1 - y_{uij}\right).log\left(1 - \hat{y}_{uij}\right) \tag{3.9}$$

D_s indicate the triplets of the form (u, i, j), u is a user, i is an observed item, j has not been observed yet.

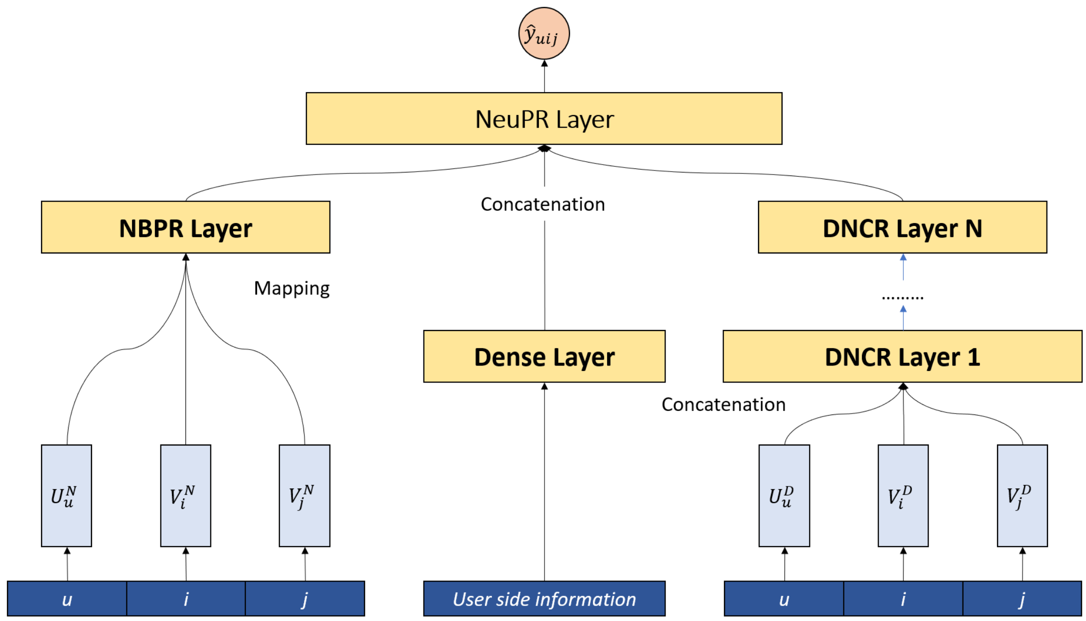


Figure 3.2 : Neural Collaborative Ranking Model with Side Information

3.3 Baseline Models

We compared the performance of our models with the following models:

PopRank: This is the non-personalized recommendation method in which products are ranked by number of interactions.

ALS [33]: ALS is a Matrix factorization method for item recommendation. This method is based on only user-item interactions.

Collaborative Deep Learning (CDL) [34]: It is a state-of-the-art Deep Learning based hybrid model, which couples denoising auto-encoders with matrix factorization. Side information is integrated in addition to user-item interactions.

Collective Matrix Factorization (CMF) [15]: They simultaneously factor several matrices, sharing parameters among factors when an entity participates in multiple relations. They combined side information into MF. CMF factorizes the user-item matrix and side information matrix together.

Neural Collaborative Filtering (NCF) [2]: It is the deep learning method for Collaborative Filtering. NCF replaces the user-item inner product with a neural architecture. This method combines MF and MLP to tackle MF limitations. This method is one of our base methods while building our framework. This method is based on only user-item interactions.

Neural Collaborative Ranking (NCR) [3]: It is the deep learning method that uses the pairwise loss function. In this work a neural network architecture is used to model a user's pairwise preference between items. This method is based on only user-item interactions.

3.4 Negative Sampling

A user's preference is generally two-way; positive (user preferred, liked) and negative / neutral (user dislikes or has no special interest). If our goal is to recommend products that will appeal to the user, both preferences are equally important for us. Since our dataset consists of user - product group purchase data, it does not contain negative sampling. In order to increase our recommendation success, we select a different number of negative sample sets from the product groups that the user does not interact with for each product group they interact with from our training data. Then we add our negative feedback sample set to our training data and train our model.

Model performance can be seen with different negative sample numbers in Figure 3.3 and Figure 3.4 . As can be clearly seen, a single negative sample is not sufficient for optimum recommendation performance. Figure 3.3 and Figure 3.4 additionally show that the hybrid methods that we have added side information have more successful recommendation performance than the methods that use user-item interaction only.

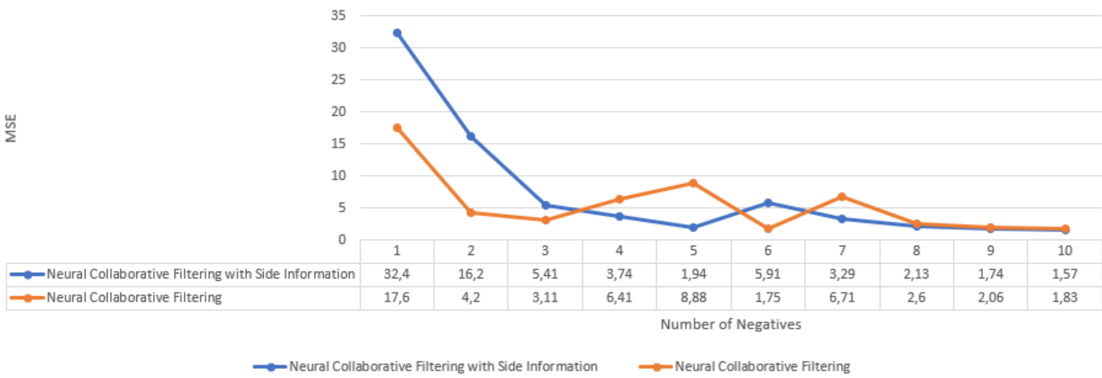


Figure 3.3 : MSE of NCF and NCF with side information methods according to the number of negative samples per positive instance

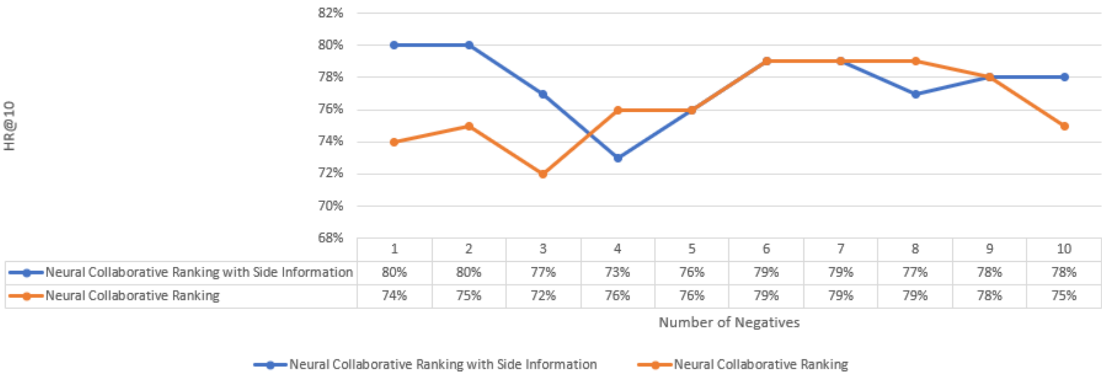


Figure 3.4 : HR@10 of NCR and NCR with side information methods according to the number of negative samples per positive instance

3.5 Evaluation Process

Our dataset consists of 5.5M user-item interactions. As can be seen in our negative sample experiments, a serious increase in performances has been observed up to 3 negative feedback samples. And since 3 negative sample did not seriously affect the model run length and RAM usage, 3 negative samples were chosen as optimal and experiments were carried out in this way. When we applied negative sampling, we have 22M negative and positive interactions. The Neural Collaborative Filtering with Side Information model takes the number of products purchased by the user as input. Recommendation task is predicting number of products purchased by the user. The recommendation performance is evaluated by the Mean Squared Error (MSE) function. For this evaluation, we randomly split 0.5% of our 22M data for testing. When we randomly split 0.5% of our dataset, we have 110K interactions to test our model. MSE is calculated for our test data:

$$MSE = \frac{1}{|R|}(Y - \hat{Y})^2 = \frac{1}{|R|}\sum_{r_{ui} \varepsilon R}(r_{ui} - p_u q_i)^2 \tag{3.10}$$

In Neural Collaborative Ranking with Side Information method top-k item recommendation is performed instead of rating prediction. In order to measure the accuracy of the system, the leave-one-out evaluation has been utilized. As a test item, we kept the latest item interaction per user, if there is a user have one order then we didn't add this interaction and user to the test data. However, the count of users in whole data is massive for testing so we randomly split 10% of users for testing.

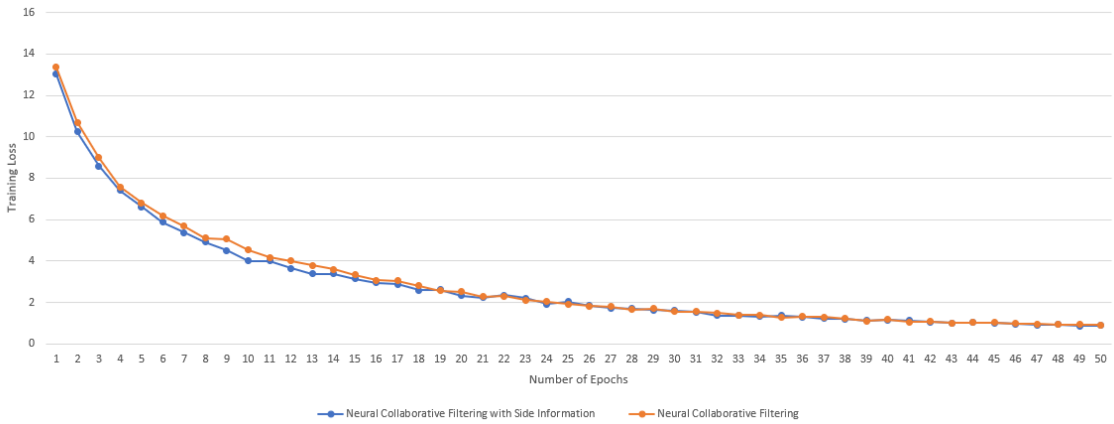


Figure 3.5 : Training Loss of NCF and NCF with side information methods according to the number of epochs

Then for these sample users, 99 different items are selected randomly that are not interacted by the user, then we ranked the test item among the non-interacted items that were sampled. The ranking performance is assessed by Hit Ratio (HR). HR@k is an evaluation metric which consider if the interacted item is in the top-k list or not. HR@k is used in our works as follows:

$$HR@k = \begin{cases} 1/k & \text{if } r_{test}(u,i)\ \varepsilon R_k \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

where $r_{test}(u,i)$ defines the interaction of user u to item i, and R_k defines the list of top-k recommended items for the user u.

3.6 Impact of Number of Epochs on Recommendation Performance

It can be seen in Figure 3.5 and Figure 3.6 that with more epochs, the training loss of models gradually decreases. The training losses of all models could not be visualized in a single figure, since the cost functions of the models are different from each other. It can be seen in Figure 3.8 and Figure 3.7 that with more epochs, the recommendation performance is improved. The most effective changes are occured in the first 10 epochs, and more epochs may overfit the model. While the training loss of models keeps decreasing after 10 epochs, their recommendation performance actually reduces. NCR with Side Information achieves lower training loss than NCR. The HR@10 also shows the similar trend that "NCR with Side Information" > "NCR" > "CDL".

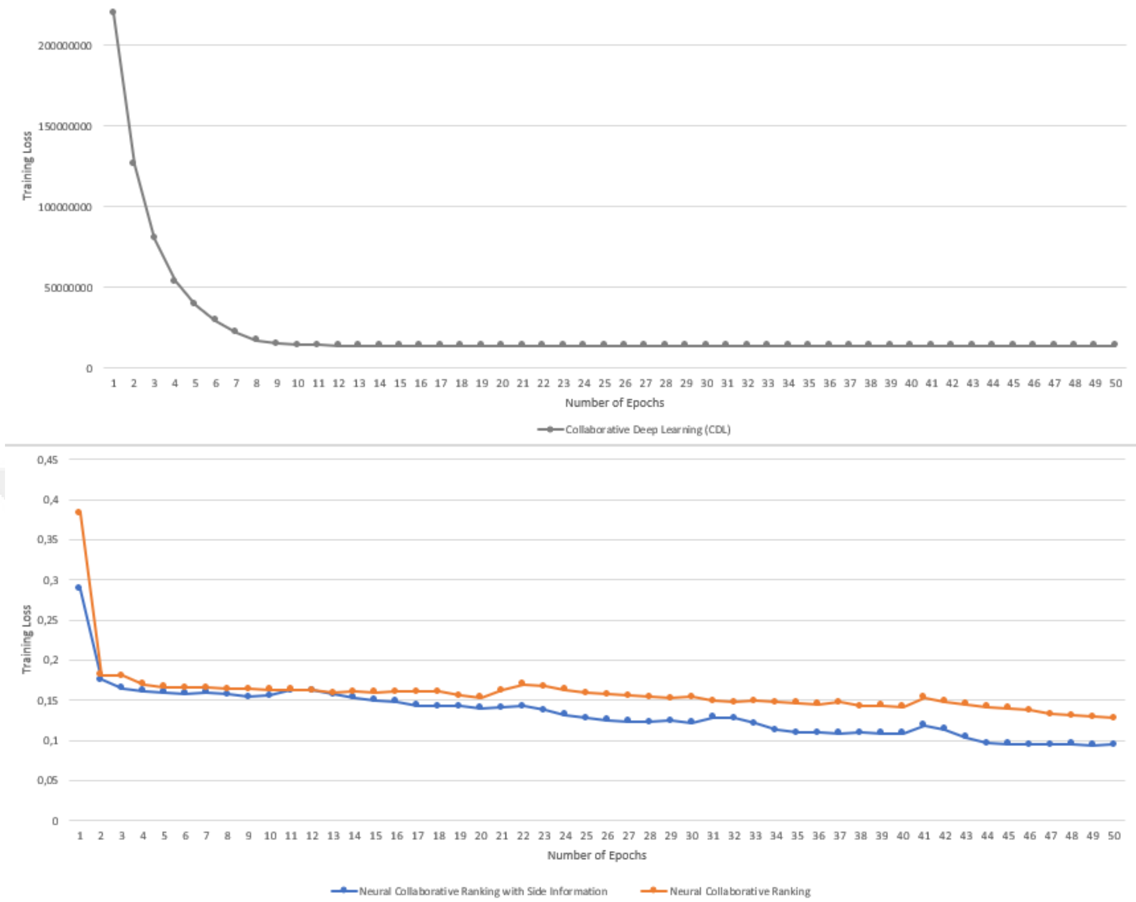


Figure 3.6 : Training Loss of NCR, NCR with side information and Collaborative Deep Learning (CDL) methods according to the number of epochs

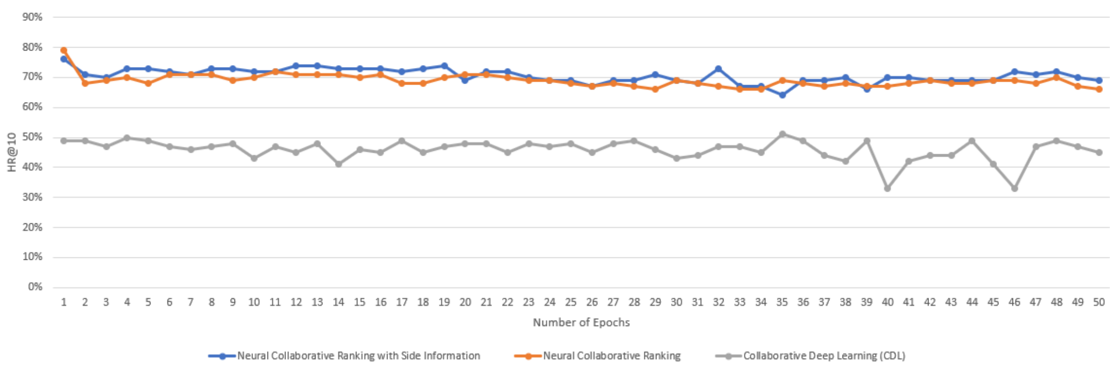


Figure 3.7 : HR@10 of NCR, NCR with side information and Collaborative Deep Learning (CDL) methods according to the number of epochs

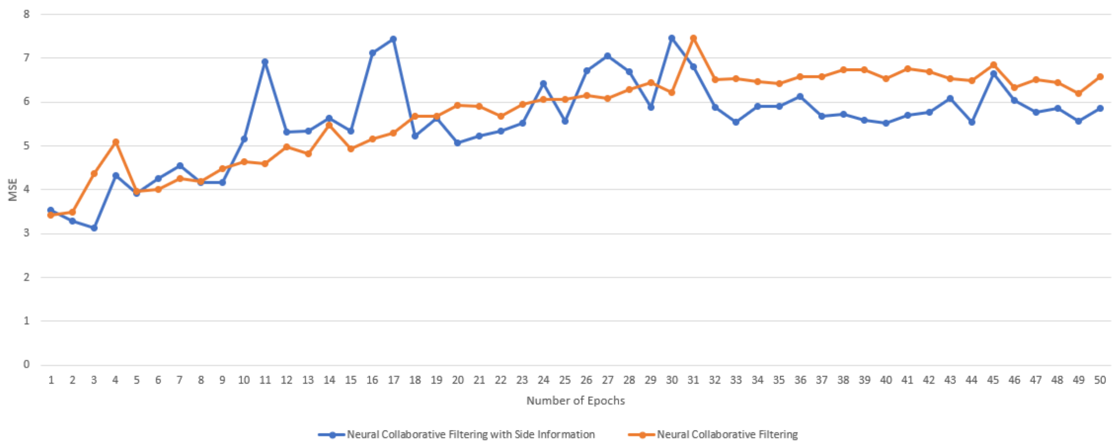


Figure 3.8 : MSE of NCF and NCF with side information methods according to the number of epochs

3.7 Experimental Settings and Performance Comparison

For preprocessing operations, we used Pandas library in Python. We also used Keras library in Python to implement our proposed deep learning methods. The single negative sample is not sufficient for optimum recommendation performance results, our models are implemented for negative sample numbers ranges from 1 to 10 and obtained the performance results in the 1M randomly selected dataset. While using our final comparison results, we used our 5,5 M dataset and 3 negative samples for each positive interaction. The learning rate is set to be as [0.0001, 0.001, 0.01] to update the neurons' weights. As a result our models show better recommendation performance with 0.001 learning rate. As we have seen in our experiments about impact of number of epochs ranging from 1 to 50, the biggest change occurs in the first 10 epochs. For this reason, we achieved our performance results in 10 epochs.

In our proposed methods, the batch size is set to 512 and we chose Adam optimizer. In Neural Collaborative Filtering with Side Information model, MF embedding size is equal to 8 and MLP embedding size is equal to 32. We employed four hidden layers which have sizes [64,32,16,8] for MLP. In Neural Collaborative Ranking with Side Information model, MF embedding size is equal to 8 and MLP embedding size is equal to 16. We employed four hidden layers which have sizes [32,32,16,8] for DNCR.

In this thesis, the performance of baselines and our NCF with Side Information model is tested on a real world fashion retailer e-commerce dataset, and results are shown with MSE in Table 3.1. ALS model doesn't have non-linearity settings. Other baseline models use deep learning architecture to capture non-linearity. For this reason, ALS has the lowest performance. Although CDL uses side information, it shows lower performance compared to NCF model which does not use side information. NCF has more flexibility to learn user-item interactions, because it fuses two models with a linearity features and non-linearity features. Our NCF model using the side information showed the best recommendation performance. Our NCF with side information method provides 29,96% performance increase compared to NCF method based on user-item interaction only.

In this thesis, the performance of baselines and our NCR with Side Information model is tested on a real world fashion retailer e-commerce dataset, and results are shown with HR@10 in Figure 3.9 and Table 3.2.

As can be seen in Table 3.2, PopRank is the model with the lowest performance as expected, as it does not offer personalized recommendations.

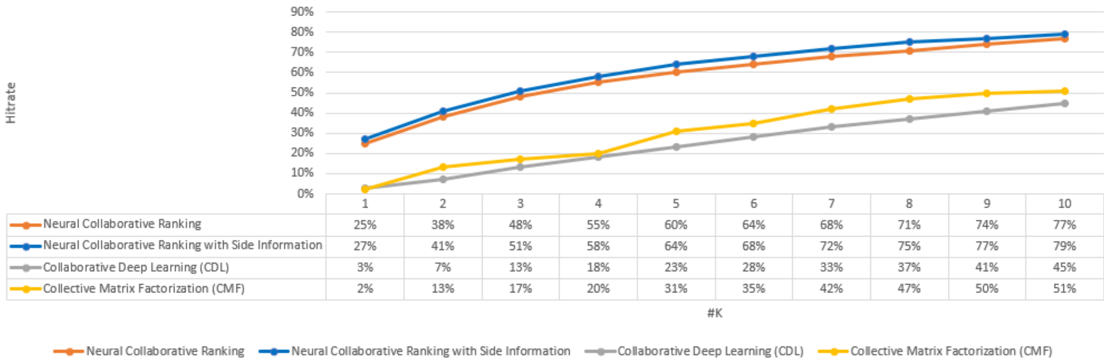
Table 3.1 : Performance Comparison in MSE

	Recommendation Methods			
	ALS	CDL	NCF	NCF with SI
	46,8434	4,4561	4,4438	**3,4192**

Table 3.2 : Performance Comparison in HR@10

	Recommendation Methods			
	PopRank	CDL	NCR	NCR with SI
	0,5618	0,6555	0,7866	**0,8160**

As can be seen in Figure 3.9, the CMF model, which does not use deep learning architecture, has the lowest recommendation performance due to the lacking of its non-linearity settings. The experimental results demonstrate the success of deep learning in capturing the nonlinear relationships of the user-item interaction matrix.



	1	2	3	4	5	6	7	8	9	10
Neural Collaborative Ranking	25%	38%	48%	55%	60%	64%	68%	71%	74%	77%
Neural Collaborative Ranking with Side Information	27%	41%	51%	58%	64%	68%	72%	75%	77%	79%
Collaborative Deep Learning (CDL)	3%	7%	13%	18%	23%	28%	33%	37%	41%	45%
Collective Matrix Factorization (CMF)	2%	13%	17%	20%	31%	35%	42%	47%	50%	51%

Figure 3.9 : Evaluation of Top-K item recommendation where K ranges from 1 to 10

Our NCR with side information method provides 3,73% performance increase compared to NCR method based on only user-item interaction.

4. CONCLUSIONS AND RECOMMENDATIONS

In this thesis, we discussed two neural network architectures for item recommendation. To solve data sparsity and cold start problems we build a general framework for both user-item interaction data and side information. We made our experiments on real-world e-commerce data. Thus we proved adding side information increase the quality of recommendation for both of these architectures. Also we proved the deep neural networks are effective way of capturing nonlinear and complex relation in large real world user-item interaction data. We studied with implicit feedbacks. Also we worked with pointwise and pairwise learnings. This work provides a baseline for research that will add side information to deep learning based recommendation methods.

In future, we will study with unstructured heterogeneous side information such as product reviews, tweets or product images. In addition, we plan to study with heterogeneous networks for this purpose.

REFERENCES

[1] **Li, Q.**, **Zheng, X. and Wu, X.** (2017). Collaborative autoencoder for recommender systems, *ArXiv e-prints*.

[2] **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*, International World Wide Web Conferences Steering Committee, pp.173–182.

[3] **Song, B.**, **Yang, X.**, **Cao, Y. and Xu, C.** (2018). Neural collaborative ranking, *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, ACM, pp.1353–1362.

[4] **Schafer, J.B.**, **Frankowski, D.**, **Herlocker, J. and Sen, S.**, (2007). Collaborative filtering recommender systems, The adaptive web, Springer, pp.291–324.

[5] **Mooney, R.J. and Roy, L.** (2000). Content-based book recommending using learning for text categorization, *Proceedings of the fifth ACM conference on Digital libraries*, pp.195–204.

[6] **Pazzani, M.J. and Billsus, D.**, (2007). Content-based recommendation systems, The adaptive web, Springer, pp.325–341.

[7] **Burke, R.** (2002). Hybrid recommender systems: Survey and experiments, *User modeling and user-adapted interaction*, *12*(4), 331–370.

[8] **Burke, R.**, (2007). Hybrid web recommender systems, The adaptive web, Springer, pp.377–408.

[9] **Adomavicius, G. and Tuzhilin, A.** (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE transactions on knowledge and data engineering*, *17*(6), 734–749.

[10] **He, X.**, **Zhang, H.**, **Kan, M.Y. and Chua, T.S.** (2016). Fast matrix factorization for online recommendation with implicit feedback, *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, ACM, pp.549–558.

[11] **Covington, P.**, **Adams, J. and Sargin, E.** (2016). Deep neural networks for youtube recommendations, *Proceedings of the 10th ACM conference on recommender systems*, ACM, pp.191–198.

[12] **Dong, X.**, **Yu, L.**, **Wu, Z.**, **Sun, Y.**, **Yuan, L. and Zhang, F.** (2017). A hybrid collaborative filtering model with deep structure for recommender systems, *Thirty-First AAAI Conference on Artificial Intelligence*.

[13] **Zhang, F.**, **Yuan, N.J.**, **Lian, D.**, **Xie, X. and Ma, W.Y.** (2016). Collaborative knowledge base embedding for recommender systems, *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp.353–362.

[14] **Liang, D.**, **Zhan, M. and Ellis, D.P.** (2015). Content-Aware Collaborative Music Recommendation Using Pre-trained Neural Networks., *ISMIR*, pp.295–301.

[15] **Singh, A.P. and Gordon, G.J.** (2008). Relational learning via collective matrix factorization, *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp.650–658.

[16] **Agarwal, D.**, **Chen, B.C. and Long, B.** (2011). Localized factor models for multi-context recommendation, *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp.609–617.

[17] **Sen, S.**, **Vig, J. and Riedl, J.** (2009). Tagommenders: connecting users to items through tags, *Proceedings of the 18th international conference on World wide web*, ACM, pp.671–680.

[18] **Gedikli, F. and Jannach, D.** (2010). Rating items by rating tags, *Proceedings of the 2010 Workshop on Recommender Systems and the Social Web at ACM RecSys*, pp.25–32.

[19] **Gedikli, F. and Jannach, D.** (2013). Improving recommendation accuracy based on item-specific tag preferences, *ACM Transactions on Intelligent Systems and Technology (TIST)*, *4*(1), 11.

[20] **Hinton, G.E. and Salakhutdinov, R.R.** (2006). Reducing the dimensionality of data with neural networks, *science*, *313*(5786), 504–507.

[21] **Zheng, L.**, **Noroozi, V. and Yu, P.S.** (2017). Joint deep modeling of users and items using reviews for recommendation, *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, ACM, pp.425–434.

[22] **Gong, Y. and Zhang, Q.** (2016). Hashtag Recommendation Using Attention-Based Convolutional Neural Network., *IJCAI*, pp.2782–2788.

[23] **Ma, H.**, **Yang, H.**, **Lyu, M.R. and King, I.** (2008). Sorec: social recommendation using probabilistic matrix factorization, *Proceedings of the 17th ACM conference on Information and knowledge management*, ACM, pp.931–940.

[24] **Ma, H.**, **Zhou, D.**, **Liu, C.**, **Lyu, M.R. and King, I.** (2011). Recommender systems with social regularization, *Proceedings of the fourth ACM international conference on Web search and data mining*, ACM, pp.287–296.

[25] **Yang, S.H.**, **Long, B.**, **Smola, A.**, **Sadagopan, N.**, **Zheng, Z. and Zha, H.** (2011). Like like alike: joint friendship and interest propagation in social networks, *Proceedings of the 20th international conference on World wide web*, ACM, pp.537–546.

[26] **Salakhutdinov, R.**, **Mnih, A. and Hinton, G.** (2007). Restricted Boltzmann machines for collaborative filtering, *Proceedings of the 24th international conference on Machine learning*, pp.791–798.

[27] **Gelli, F.**, **He, X.**, **Chen, T. and Chua, T.S.** (2017). How personality affects our likes: Towards a better understanding of actionable images, *Proceedings of the 25th ACM international conference on Multimedia*, pp.1828–1837.

[28] **Sun, Z.**, **Guo, Q.**, **Yang, J.**, **Fang, H.**, **Guo, G.**, **Zhang, J. and Burke, R.** (2019). Research commentary on recommendations with side information: A survey and research directions, *Electronic Commerce Research and Applications*, *37*, 100879.

[29] **Strub, F.**, **Mary, J. and Gaudel, R.** (2016). Hybrid collaborative filtering with neural networks, *CoRR Vol. abs/1603.00806*.

[30] **Koren, Y.**, **Bell, R. and Volinsky, C.** (2009). Matrix factorization techniques for recommender systems, *Computer*, (8), 30–37.

[31] **Hidasi, B. and Tikk, D.** (2012). Fast ALS-based tensor factorization for context-aware recommendation from implicit feedback, *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, pp.67–82.

[32] **Jawaheer, G.**, **Szomszor, M. and Kostkova, P.** (2010). Comparison of implicit and explicit feedback from an online music recommendation service, *proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems*, pp.47–51.

[33] **Pilászy, I.**, **Zibriczky, D. and Tikk, D.** (2010). Fast als-based matrix factorization for explicit and implicit feedback datasets, *Proceedings of the fourth ACM conference on Recommender systems*, pp.71–78.

[34] **Wang, H.**, **Wang, N. and Yeung, D.Y.** (2015). Collaborative deep learning for recommender systems, *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp.1235–1244.

CURRICULUM VITAE

Name Surname: Didem TÜRKER

Place and Date of Birth: Fatih, Turkey - 29/09/1994

E-Mail: didemturker94@gmail.com

EDUCATION:

- **B.Sc.:** 2017, Eskisehir Osmangazi University, Faculty of Computer and Informatics Engineering, Department of Computer Engineering
- **M.Sc.:** 2020, Istanbul Technical University, Faculty of Computer and Informatics Engineering, Department of Computer Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2017-2020 Business Intelligence Engineer / LC Waikiki Mağazacılık Hiz. Tic. A.Ş.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- Didem Turker and Recep Sezer and Sahin Isik and Kemal Özkan, Quality measurement of fruits and vegetables with near infrared camera. *25th Signal Processing and Communications Applications Conference, SIU 2017, Antalya, Turkey, May 15-18, 2017*,