ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DEEP LEARNING BASED HYBRID RECOMMENDER SYSTEM

M.Sc. THESIS

Muhammet ÇAKIR

Department of Computer Engineering

Computer Engineering Programme

SEPTEMBER 2019

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DEEP LEARNING BASED HYBRID RECOMMENDER SYSTEM

M.Sc. THESIS

Muhammet ÇAKIR
(504161526)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Prof. Dr. Şule GÜNDÜZ ÖĞÜDÜCÜ

SEPTEMBER 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

DERiN ÖĞRENME TABANLI ÖNERİ SİSTEMİ

YÜKSEK LİSANS TEZİ

Muhammet ÇAKIR
(504161526)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Şule GÜNDÜZ ÖĞÜDÜCÜ

EYLÜL 2019

Muhammet ÇAKIR, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 504161526 successfully defended the thesis entitled "DEEP LEARNING BASED 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 :	**Asst. Prof. Dr. Ayşe TOSUN KUHN** Istanbul Technical University	.....
	Prof. Dr. Tunga GÜNGÖR Bosphorus University	.....
		.....

Date of Submission : **01 May 2019**
Date of Defense : **13 September 2019**

To my mother,

FOREWORD

This thesis was written as completion to the master of Computer Engineering at Istanbul Technical University. After graduating from the same department, I didn't hesitate to continue the master education here. While I was studying, I was working at the companies at the same. Lessons I took during the master education both improved my academic side and provided solving real life problems in business life.

I am grateful to my advisor Prof. Dr. Şule Gündüz Öğüdücü, and my colleague Research Assistant Resul Tugay. They supported me to not lose my motivation, and guided me in my researches. Also, I had an opportunity working for a project in collaboration with a big company, and this study was supported by ITU.

September 2019

Muhammet ÇAKIR
(Computer Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

RS	**:** Recommender System
MF	**:** Matrix Factorization
DNN	**:** Deep Neural Network
CF	**:** Collaborative Filtering
CBF	**:** Content-based Filtering
JR	**:** Job Recommendation

SYMBOLS

$\mathbf{p_u}$	**:** User latent vector
$\mathbf{q_i}$	**:** Item latent vector
a	**:** Activation function
L	**:** Loss function
W	**:** Weight matrix
b	**:** Bias vector
$\mathbf{y_{ui}}$, $\mathbf{\hat{y}_{ui}}$	**:** Actual and expected scores
HR	**:** Hit ratio
NDCG	**:** Normalized discounted cummulative gain

LIST OF TABLES

Page

LIST OF FIGURES

Page

DEEP LEARNING BASED HYBRID RECOMMENDER SYSTEM

SUMMARY

Recommender systems have been playing a vital and unavoidable role to overcome information overload with steadily increasing volume of online information. In recent years, deep learning has gained tremendous success on natural language processing, computer vision, and speech recognition. This success has triggered the adaptation of deep learning techniques on recommender systems by giving more opportunity to improve recommendation performance.

While there are several studies that purely focus on collaborative and content-based filtering approaches, we combine these two techniques under a hybrid system. Our hybrid system uses ID embeddings for collaborative filtering side under a matrix factorization technique, and content information such as categorical and continuous features of users and items are processed under a deep neural network architecture. It combines the strengths of non-linearity of DNN and linearity of MF to model user - item interaction.

In recommendation systems, recommendation tasks utilize explicit and/or implicit feedback of users. While explicit feedback directly reflects preferences of users through ratings, reviews, and like/unlikes, it is more challenging to use implicit feedback since user satisfication is not observed because of natural deficiency of negative feedback. While some types of datasets have explicit feedback that is easy to separate positive and negative feedback from each other, there is only implicit feedback for some recommendation problems. In this study, we specifically use a dataset from an e-recruitment website. Job recommendation task being a kind of reciprocal recommendation both provides job seekers to find the most proper job openings and helps recruiters to select the most suitable candidates. In job recommendation, job application information is considered as positive feedback, and negative feedback is randomly selected from unobserved interactions.

Our hybrid model obtains latent features from two different networks, and calculates a score by concatenating all feature vectors. The output of the proposed model can be thought as binary classification result since the interaction between users and items is taken as 1 for positive feedback or 0 for negative feedback, then binary cross entropy loss (log loss) is optimized. Also, top-k recommendation task is performed instead of rating prediction because of using implicit feedback. Hit Ratio (HR) and NDCG (Normalized Discounted Cummulative Gain) are used as evaluation metrices to realize top-k recommendation task. Detailed experiments on this job-site dataset show considerable improvements of the proposed hybrid model over the state of art models.

DERiN ÖĞRENME TABANLI ÖNERİ SİSTEMİ

ÖZET

Son yıllarda internet üzerindeki çevrimiçi bilgi hacminin sürekli olarak artmasıyla birlikte öneri sistemleri çok daha kritik ve kaçınılmaz bir role sahip olmaya başladı. Özellikle doğal dil işleme ve bilgisayarlı görü gibi metin ve resimdeki büyük miktardaki veriyi işleyip muazzam sonuçlar üreten derin öğrenme tabanlı yaklaşımlar öneri sistemlerinde de uygulanmaya başlandı. Bu tez kapsamında da derin öğrenme tekniklerinden yararlanılmıştır.

Öneri sistemleri alanında yapılan çalışmaların birçoğu ya işbirlikçi filtreleme ya da içerik-tabanlı öneri sistemleri üzerine yoğunlaşırken, bu tez kapsamında bu iki yaklaşımın beraber kullanıldığı hibrit bir öneri sistemi algoritması sunulmuştur. Bizim kurduğumuz hibrit öneri modelinin işbirlikçi filtreleme tarafında ID(kimlik) gömme tekniğinden yararlanılmıştır. Bu teknik kullanıcı ve ürünlerin ID'lerini alarak onları bir vektör şeklinde temsil eder. Kimlik gömme tekniğinde tüm kullanıcılar(ürünler) tek sıcak kodlama(one-hot encoding) işlemi yapılarak toplam kullanıcı sayısı kadar uzunlukta olan bir vektörle temsil edilirler. Daha sonra gömme tekniği kullanılarak önceden belirlenen kestirim faktörü (predictive factor)'nün boyutu kadar bir vektöre küçültülür. Kullanıcı ve ürünlerin ID'lerinden elde edilen vektörler klasik matris ayrıştırma tekniklerinin aksine daha esnek bir yapının yani daha fazla parametrenin kullanıldığı doğrusal ve doğrusal olmayan ayarlarla genişletilebilen bir matris ayrıştırma tekniği kullanılmıştır. Diğer taraftan kullanıcı ve ürünlerin kategorik özellikleri benzer gömme tekniği kullanılarak, sürekli değerler alan özellikler için ise normalizasyon işlemi yapılarak özellik vektörleri elde edilmiştir. Kullanıcı ve üründen ayrı ayrı gelen özellik vektörleri birleştirilerek oluşturulan derin sinir ağı modeline input olarak verilmiştir. Yapay sinir ağı modelinde çok katmanlı algılayıcılar kullanılmış, katmanlardaki ağırlık vektörleri ve yanlılık parametreleri ile birlikte kullanıcı ve ürün arasındaki etkileşim sağlanmıştır. Yapay sinir ağının çıktısı olan özellik vektörüyle matris ayrıştırma yönteminden gelen özellik vektörü hibrit sistemin etkileşim katmanında birleştirilerek sonuç elde edilmiştir. Bu hibrit sistem kullanıcı ve ürün etkileşimini modellemek için matris ayrıştırma işleminden gelen doğrusallık ile derin sinir ağının doğrusal olmayan özelliğinin güçlerini birleştirir.

Öneri sistemlerinde öneri işlemini gerçekleştirmek için kullanıcının ürün hakkındaki açık ve/veya net olmayan yani zımni geri bildirimlerinden yararlanılır. Açık geri bildirim puan, yorum ve beğen/beğenme gibi işlemlerle kullanıcının ürün hakkındaki tercihini direkt olarak yansıtıyor iken, zımni geri bildirimde olumsuz geri bildirimi kestirmek mümkün olamadığı için kullanıcı memnuniyeti gözlemlenemez. Bu yüzden kullanması zor olmaktadır. Kitap, müzik ve film önerisi gibi çoğu öneri problemlerinde puan ve yorum gibi kullanıcının direkt memnuniyetini yansıtan geri bildirimler varken, çevrimiçi tanışma ve çevrimiçi işe alım siteleri gibi bazı problemlerde yalnızca zımni geri bildirimden yararlanmak durumunda kalınabiliyor. Bu tez çalışmasında, internet

üzerinden iş başvurularının yapıldığı websitesinden elde edilen veri kümesi üzerine yoğunlaşılmaktadır. İş öneri sistemleri, çevrimiçi tanışma siteleri gibi karşılıklı öneri sağlar. Hem iş arayanlara profillerine uygun olan iş ilanları listelenir hem de işe alım uzmanlarının kolaylıkla elde edebileceği iş ilanı için özgeçmişi en uygun adayların listesi çıkartılır. Yaygın olarak kullanılan öneri sistemlerinde film izleme, ürün satın alma ve ürüne tıklama gibi bilgiler kullanılıyor iken, bu çalışmada iş önerisi için iş başvuru bilgisi kullanılmıştır. İş başvuru bilgisine ek olarak, iş başvurusu yapan tüm kullanıcıların yaş, cinsiyet, askerlik durumu, çalışma durumu, yaşadığı şehir, bitirdiği üniversite ve departman kullanılıyor iken iş ilanı için işe alınacak aday sayısı, tecih edilen cinsiyet, pozisyon tipi, pozisyon seviyesi ve eğitim durumu bilgileri kullanılmaktadır. Yapılan iş başvuruları pozitif geri bildirim olarak düşünülürken, negatif geri bildirimler gözlemlenmeyen etkileşimler arasından rastgele seçilmektedir. Veriseti iki veya daha fazla iş başvurusu bulunan adaylar arasından seçilmektedir. Bu veri seti eğitim ve test kümesi olarak ikiye ayrılmaktadır. Öncelikle her adayın başvuruları başvuru gününe göre sıralanmaktadır, ve adayın son iş başvurusu test kümesine önceki iş başvuruları eğitim kümesine dahil edilmektedir. Eğitim kümesindeki her iş başvurusu pozitif geri bildirim olarak düşünülürken her pozitif geri bildirime karşılık belirli sayıda (m) rastgele olarak adayın başvurmadığı ilanlar arasından negatif örnek seçilmektedir. Önerilen model eğitim kümesindeki bu verilerle eğitildikten sonra test kümesindeki her bir kullanıcı(her bir kullanıcı için bir başvuru) için başvurabileceği ilanlar arasından rastgele bir ilan kümesi seçilerek test edilir.

Doğrudan geribildirimin olduğu öneri sistemlerinde kullanıcının sağladığı etkileşim genellikle ürüne bir puan verme şeklinde gerçekleştirilir. Bu yüzden bu sistemlerde derecelendirme tahmini yapılmaktadır. Kullanıcının ürünle etkileşiminin olup olmamasının pozitif ve negatif olarak değerlendirildiği öneri sistemlerinde ise daha çok top-k sıralama kullanılmaktadır. Öncelikle öneri yapılabilecek aday kümesi seçilmektedir ve daha sonra bu aday kümesi içerisinden en yakın k tanesi skora göre sıralanmaktadır. Bizim hibrit modelimiz iki farklı ağ üzerinden elde edilen gizli özellikleri birleştirerek skor üretir. Önerilen modelin çıktısı ikili sınıflandırma sonucu olarak düşünülebilir çünkü eğer kullanıcı ve ürün arasında bir etkileşim söz konusu ise pozitif geri bildirim olarak 1 alınır, ama herhangi bir etkileşim yoksa 0 olarak düşünülür. Hedef fonksiyon olarak ikili çapraz entropi kaybı optimize edilmektedir. top-k önerisinin kullanıldığı sistemlerde değerlendirme ölçütü olarak isabet oranı(hit ratio) ve normalleştirilmiş indirimli birikimli kazanç(normalized discounted cummulative gain) kullanılmaktadır. Bu çalışmada da bu iki değerlendirme ölçütü kullanılarak baz alınan metodlarla karşılaştırmalar yapılmıştır. Değerlendirme hem kullanıcı önerisi hem de ürün önerisi olmak üzere iki farklı kümeye ayrılmış veriler üzerinde yapılmıştır. Yapılan detaylı deneyler sonucunda önerilen hibrit model hem ürün önerisi için hem de kullanıcı önerisi için önemli bir gelişme kaydetmiştir. Ama değerlendirme ölçütlerinin sonuçları kullanıcı önerisinin ürün önerisinden daha düşük skor ürettiğini göstermektedir. Çünkü bir iş ilanı için seçilebilecek aday kümesinin boyutu, bir adayın başvurabileceği işlerin sayısından oldukça fazladır. Gelecekte Doğal Dil İşleme yaklaşımları kullanılarak kullanıcı ve ürünlerin metin tabanlı özellikleri üzerine çalışmalar yapılması hedeflenmektedir. Örneğin iş öneri sistemlerinde iş ilanı açıklaması, deneyim detayları ya da adayın vasıfları derin öğrenme yaklaşımının girdisi olan gizli özellik uzayını zenginleştirmektedir. Ayrıca top-k sıralama işlemi için kullanılan ve rastgele seçilen negatif örneklerin seçimi yerine öncelikle seçilebilecek adayların kümesi belirlenir ve bu küme için top-k

ürün (kullanıcı) listesi oluşturulur. Çünkü bir kullanıcı için rastgele seçilen ürün kümesindeki tüm elemanların hepsi aday profiliyle alakasız olabilir ve test kümesi için değerlendirme ölçütleri çok yüksek sonuçlar üretmesine rağmen kullanıcıya önerilen ürünler başarılı olmayabilir. Bu yüzden çeşitli kümeleme algoritmaları kullanılarak daha alakalı aday kümesi seçilmesi hedeflenmektedir.

1. INTRODUCTION

The number of products and services offered by companies has been increasing in parallel with the explosive growth of information online in recent years. The great number and variety of products offers customers more choices, and they cope with this large amount of information provied by websites and applications. At this point, recommender systems (RSs) are playing a more critical role to facilitate and accelerate decision-making process. The RS assists customers in offering products and services depending on their preferences, and past buying behaviours. Recommendation problem has various application areas ranging from e-commerce websites to online dating applications. RSs recommend products to users to buy in e-commerce websites like Amazon and Ebay, offer movies to watch in movie websites like Netflix, or suggest songs to listen in digital music streaming services like Spotify.

In this study, Job Recommendation (JR), a special type of RSs, will be adapted. It is different from traditional RSs with various aspects like providing mutual recommendation, and having a huge amount of auxiliary information. Unlike conventional RSs that only recommend items to users, both user and item preferences are evaluated to find the most appropriate match. In a recruitment website, jobs are recommended to candidates depending on the previous choices, and candidates are proposed to recruiters being interested in job posting as a reciprocal RS.

Recommendation lists are mostly generated by user-item interactions, user (item) properties, and some additional data. Recommendation models are mainly categorized into Collaborative Filtering (CF), Content Based Filtering (CBF), and Hybrid RSs based on the types of input data [2].

Collaborative Filtering: The recommendations in CF are done based on the behaviors of users. CF adopts the principle of sharing similar interests for the similar users or items [3], and recommends items to users on the basis of the interaction between them by calculating similarities.

Two types of CF techniques are used:

- **User-based:** It assumes that similar people have similar interests, and calculates the similarity between users.
- **Item-based:** It focuses on which items are more similar to items that user previously rates, and calculates the similarity between items.

Content-Based Filtering: This technique is based on the description or auxiliary information provided for items, and it exposes the similarity between items with their contexts. The properties of items are used as content of users for the recommendation task. [4, 5].

Hybrid Systems: This type of RSs fuses two or more recommendation techniques to generate better recommendation results [6, 7].

In these days, Deep Learning (DL) has become more popular in many fields containing image processing [8], natural language processing [9], and RSs [10, 11] etc. DL has gained a significant success using textual, visual, and content information with various neural network architectures. It can be used in all recommendation models that are CF, CBF, and Hybrid RSs to boost existing solutions. DL techniques model users and items taking their properties into account, and takes these features as the input of neural network. For example, Liang *et al.* proposed a model that uses acoustic features from audio signals [12]. Zhang *et al.* utilize textual and visual content of items with embedding techniques, and achieves a semantic input for neural network [13]. Also, He *et al.* apply ID embeddings to process in DL unlike traditional approaches that use only a dot product of the hidden features [10]. In this study, we adapt ID embedding technique together with general properties of candidates and job postings to make better recommendation.

Matrix Factorization (MF) [14] is the most popular CF technique, and it discovers the hidden factors which is the main reason of why user interacts with item. Although Deep Neural Networks(DNNs) [15] have many application areas, it is also used to model ID embeddings and content information. We propose a hybrid architecture that combines MF and DNN model; DNN combines auxiliary information with ID embeddings of users and items into a collaborative approach as the first part of the

hybrid model, and secondly, the main framework fuses the hidden features coming from DNN and MF into a shared layer to make a prediction.

The main contributions of this study are as follows:

- A hybrid model is proposed by joining user and item properties with ID embeddings, and it combines the strengths of linearity of MF and non-linearity of DNN.
- Content information used in hybrid model also provides the framework to tackle cold-start problem.
- Reciprocal recommendation is applied with a real-world dataset from a job recruitment website in Turkey.

The rest of the thesis is organized as follows: Chapter 2 presents background studies, Chapter 3 offers our proposed framework, Chapter 4 shows our analyzes and experimental results, and lastly Chapter 5 discusses outcomes with future studies.

2. BACKGROUND

In this chapter, existing solutions for collaborative filtering that inspire Hybrid Recommender System (HRS) will be mentioned. These preliminaries include Matrix Factorization and Deep Neural Network that build the main structure of our model using user and item IDs.

2.1 Related Works

Many of the studies make use of CF approaches [14, 16, 17] in RSs, but also some techniques are recommended by CBF [18, 19] with increasing deep learning based approaches. CF approaches basically utilize feedbacks that are provided by users to evaluate items' quality, usability, satisfiability or many other properties depending on what the item represents on dataset. Feedbacks of users can be divided into two categories: firstly explicit feedback [20] directly shows user preferences on items, and secondly implicit feedback [21] indirectly represents user behaviors on items. While user reviews, and ratings for items are taken into account as explicit feedback, purchasing products, watching videos, and clicking items are considered as implicit feedback [10]. When a user is not pleased with a product, s/he can directly show negative attitude by giving low rating and/or providing unfavorable review about that product.

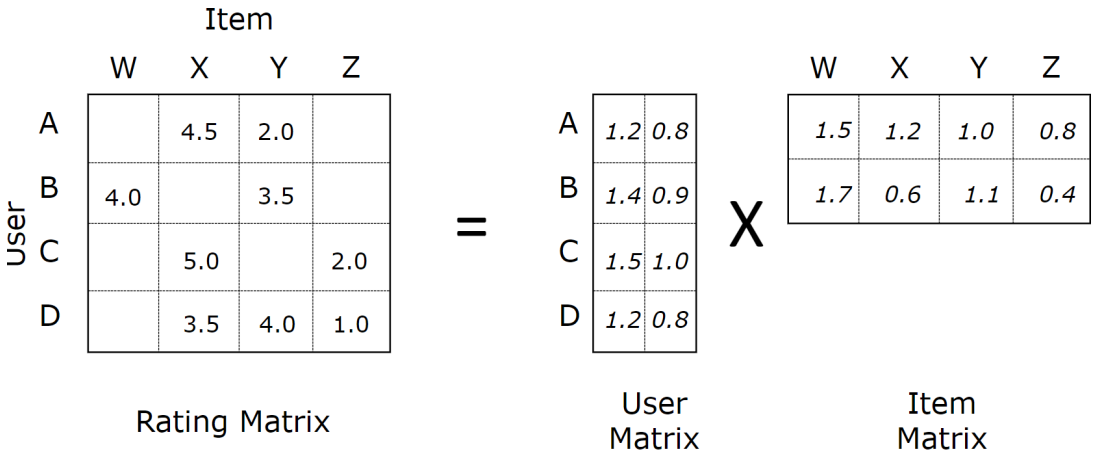


Figure 2.1 : Matrix Factorization Technique

CF approaches use the principle of sharing similar interests for the similar users or items [3], and MF is the most popular technique for CF. The original MF approach [14] was presented to model explicit or implicit feedbacks of users by mapping users and items to a latent factor space as shown in Figure 2.1. Let *m*, *n*, and *d* represent the number of users, the number of items, and the number of latent factor, respectively, Figure 2.1 shows that rating matrix generated by *m* x *n* is decomposed into two matrices: user matrix (*m* x *d*), and item matrix (*d* x *n*). Interactions (e.g. ratings) of users and items are calculated by the dot product of the latent factors after factorization operation. Many approaches have been introduced to improve MF like combining it with neighbor-based models [20], and extending it to factorization machines(FMs) [22] for generating a model using user and item features. FMs integrate the generality of Support Vector Machines(SVMs) into factorization models, and unlike SVMs, FMs can estimate parameters under high sparsity of dataset [22]. However, explicit ratings may not be existing for many of recommendation tasks every time, and users generally interact with items through implicit feedback [23]. As shown in Figure 2.2, two different rating matrices are constructed for both explicit and implicit feedback, but all implicit feedback instance(cell) is considered as the same (1), and MF techniques will not produce meaningful results because of the lack of negative instances In JRs, there is no explicit feedback like rating and like/dislike, but it is easy to collect implicit feedback such as job view and job application for content providers even being more challenging to use because of the natural lack of negative feedback.

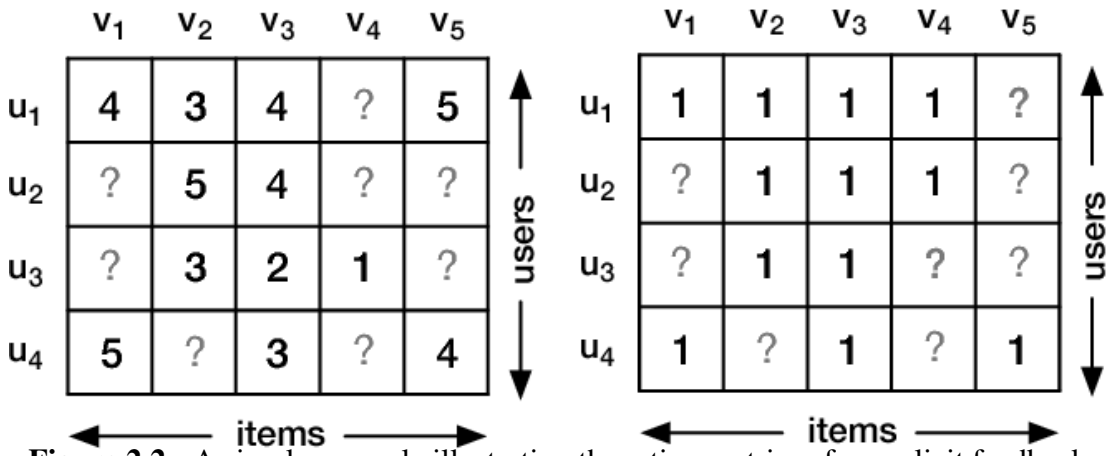


Figure 2.2 : A simple example illustrating the rating matrices for explicit feedback and implicit feedback [1]

Unlike CF approaches that utilize only user - item interactions, CBF approaches are primarily based on the description and / or auxiliary information of users and items, and indicates the similarity between items (users). Contents of videos, texts, and images can be evaluated as a wide variety of additional information [24]. In the past few years, many of the researches has utilized auxiliary information to advance the recommendation performance, and deep learning based approaches have came into question to process a large amount of information. They mostly adopted on DNNs as the main structure of the recommendation task to model auxiliary information using implicit feedback. This valuable information can be textual descriptions of items [11], audible features of musics [19], content of videos [15], and continuous - categorical features of both users and items [25]. While CF approaches suffer from the sparsity problem, CBF approaches can produce the better results when users and items have a rich content even if the interaction matrix is sparse. Many of the applications like video and music websites has no more information for users, and they have a limited amount of information such as video contents, audible features, and textual descriptions. Luckly, websites that includes job search and / or e-recruitment properties have too much information for user and item sides to apply DL tehniques.

Recent years, the number of hybrid approaches have been increasing to benefit both the properties of CF and CBF approaches, and it combines content-based and collaborative models [26, 27]. Hybrid approaches can be performed in three different ways. Firstly, collaborative and content-based tasks are separately applied, and their predictions are combined into an interaction layer. Secondly, some content-based features are joined into a collaborative approach, or opposite. Lastly, a general mixed model can be constructed to combine collaborative and content-based properties [2]. Figure 2.3 shows the basic architecture of a hybrid recommender system that combines CF and CBF recommendations. Each of these approaches has some advantages and disadvantages depending on data diversity and sparsity. For example, when a dataset is not sparse, and users or items have a limited amount of auxiliary information, to combine content-based features into a collaborative approach can make the model more strong. In this thesis, we will utilize two different approaches: firstly, we will combine content-based properties into a collaborative approach, then we will construct a general mixed model.

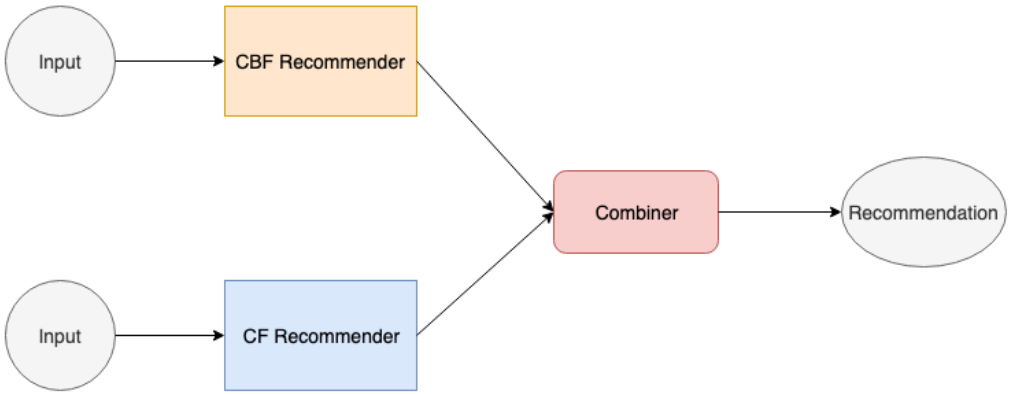


Figure 2.3 : Basic Architecture of Hybrid Recommender Systems

Moreover, there are no more studies for implementing JR task. Some studies [28, 29] especially focus on JR using a specific dataset. Lee *et al.* model the interactions for user-job title, user-job company, and user-job record, and make recommendation through graph analysis using AskStory dataset [29]. Wang *et al.* integrate user interactions into the recommendation task with a hierarchical graphical model using LinkedIn dataset [28]. Unlike these JR specific models, we purpose to build a framework that will be available for many applications such as movie, music, and book recommendation while making better recommendation for both job seekers and recruiters in JR. Also, it may not be reasonable to make reciprocal recommendation for many real-world applications since there is no need to recommend a user for an item like user recommendation for books, movies, and musics. However, some studies [30, 31] try to build a generalized framework for generating the list of the most suitable candidates. The elements of reciprocal recommendation can be two users having bilateral relations as traditional user $\times$ item interactions. Xia *et al.* bring a solution to online dating problem by finding out users' power of relationships analysing user-based and graph-based features [31]. Reciprocal recommenders, such as online dating and online recruiting, has a major challenge to satisfy the mutual preferences of two users or user $\times$ item, and graph-partitioning methods are used by representing the dataset as a bipartite graph [30]. While these works make a good job of graph analysis to find out the relationships of users, deep learning based approaches has been

disregarded to process a great amount of information existing in properties of users and items, textual explanation, and video/music content.

2.2 Preliminaries

2.2.1 Matrix Factorization

Matrix Factorization techniques come in sight after Netflix competition (2006), and Netflix announced a prize money of $1 million to participants who will improve its root mean square performance by 10% [14]. After the great improvement of recommendation performance, many of the studies in RSs have used MF techniques. MF technique allows to discover the latent features underlying the interactions between users and items. Each user and item can be associated with a real-valued vector of the latent features. Let p_u and q_i denote the latent vectors for user u and item i, respectively; MF is used to estimate an interaction $\hat{y}_{ui}$ as the dot product of p_u and q_i:

$$\hat{y}_{ui} = f(u,i|p_u,q_i) = \sum_{k=1}^{K} p_{uk}q_{ki} \tag{2.1}$$

where K is a parameter that represents the dimension of the latent space, p_{uk} and q_{ki} are used for user and item latent factor matrix elements, and f is a function that maps model parameters to the predicted score, respectively. MF models the interaction of user and item latent factors, and each dimension of the latent space is independent from each other. Dimensions of the latent space are linearly combined with the same weight, so MF can be considered as a linear model of latent factors.

2.2.2 Deep Neural Network

There are a variety of deep learning based approaches for collaborative filtering and content-based recommendation. In this part, Multi-layer Perceptron (MLP) that is a type of feedforward artificial neural network will be mentioned. User and item properties are integrated by concatenating them. Interaction between user and item latent features does not exclusively occur by a vector concatenation that is unsatisfactory for collaborative filtering modeling [10]. This problem is resolved by

adding some hidden layers after concatenating these latent vectors. Let p_u and q_i denote the latent vectors for user u and item i, respectively; DNN model is simply formulated as

$$o_1 = \begin{bmatrix} p_u \\ q_i \end{bmatrix}$$

$$\psi_2(o_1) = o_2 = \alpha_2(W_2^T o_1 + b_2),$$

$$\ldots$$

$$\psi_N(o_{N-1}) = o_N = \alpha_L(W_N^T o_{N-1} + b_N),$$

$$\hat{y}_{ui} = \alpha_{out}(h^T o_N) \tag{2.2}$$

where W_l and b_l specify the weight matrix and the bias vector, respectively. α_l is the activation function of the l-th layer of the model.

3. DEEPHYBRID MODEL

In this thesis, a hybrid recommendation model is presented to simultaneously capture different attributes that are user/item IDs for exposing the historical interactions and user/item content information to learn feature representations. This model is basically called DeepHybrid. First of all, the MF technique is used to get latent features of users and items using feedback of users to items. Secondly, latent feature representations of users and items are learnt by a deep learning based approach from both auxiliary information and ID embeddings. In this way, a hybrid recommendation architecture that combines MF and DNN under DeepHybrid is built.

3.1 Architecture of DeepHybrid Model

3.1.1 Features from MF

MF maps users and items to the hidden feature space for making a prediction for the target score. User and item properties are represented as p_u and q_i, respectively in this part of the study. An embedding vector is obtained by one-hot encoding of user (item), and it can be realized as the latent feature vector of user (item) [10]. v_u^U and v_i^I are two feature vectors that represent user u and item i which are transformed from the identities to the binarized sparse vectors with one-hot encoding. Let $P \in \mathbf{R}^{mxd}$, $Q \in \mathbf{R}^{nxd}$ be the latent factor matrices for users and items, respectively; the latent vector of user p_u, and the latent vector of item q_i are calculated as $P^T v_u^U$ and $Q^T v_i^I$. The output layer that calculates score of the model takes these hidden feature vectors as the input:

$$\hat{y}_{ui} = \alpha_{out}(h^T(p_u \odot q_i)) \tag{3.1}$$

where α_{out} denotes the activation function, $\odot$ represents the element-wise multiplication of vectors, and h defines edge weights (d-dimensional weight vector) for the vector calculated by multiplication of two hidden vectors for the output layer. As

opposed to the most widely used MF model, MF part of our framework will extend the collaborative approach by employing an activation function since a non-linear setting of the activation function may make MF model more expressive than linear MF models.

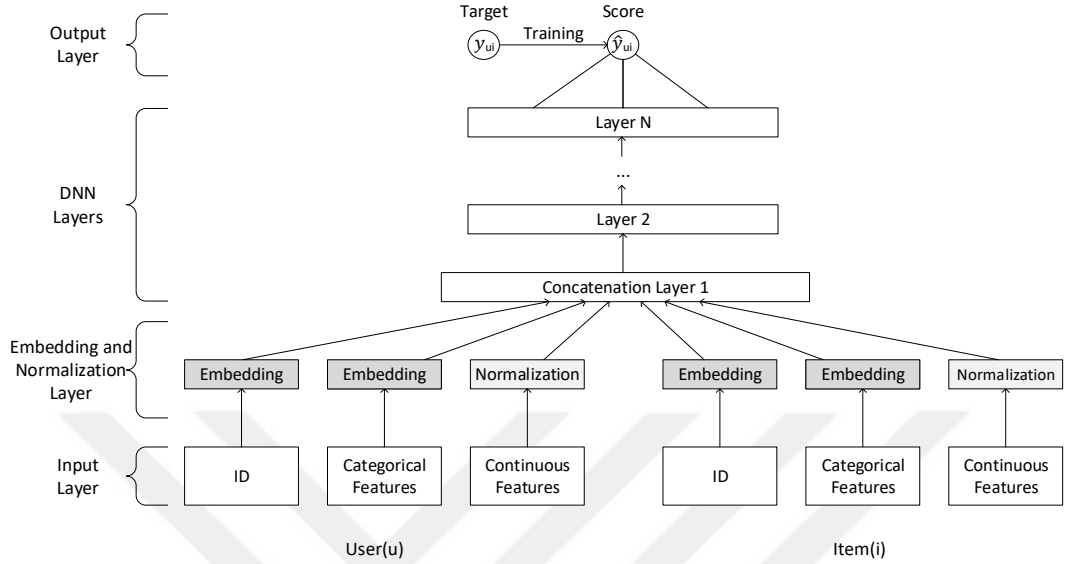


Figure 3.1 : Architecture of Deep Neural Network

3.1.2 Features from DNN

DNN model uses content information of users and items, and feature vectors obtained from these content properties and descriptions are concatenated to each other to be the input of the neural network model. Firstly, ID embedding technique generates the latent feature vector by creating the binarized sparse vector from the identity of user (item) with one-hot encoding, and multiplying this binarized vector by the latent factor matrix as similar to our MF technique. Secondly, user and item properties are used as inputs for the neural network framework in two different ways that are normalized value and embedding vector. These properties are represented as two classes; first one is categorical features that are embedded to d-dimensional vectors where d is the number of distinct values which is similar to ID embedding technique, and second one is continuous features that are normalized to [0, 1] by mapping a real-valued number. Equation 3.2 shows the calculation of normalized value for i-th quantity:

$$n_i = \frac{x_i - min(x)}{max(x) - min(x)} \tag{3.2}$$

where $x = (x_1, x_2, ..., x_n)$ show the n-dimensional data set and n_i is our $i_{th}normalizeddata$, respectively. In a job recommendation task, the number of people to hire can be seen as a continuous feature, and is normalized into [0, 1] in job posting, but position level is taken into account as a categorical feature because of having discrete values, and simply separated into limited groups like specialist, assistant specialist, and intern etc. Figure 3.1 shows the architecture of DNN that concatenates user and item properties together with IDs after embedding and normalization layer, then feedforwards this concatenated vector along with DNN layers. Let id_u, con_u, and cat_u define ID embedding, continuous, and categorical features of users, respectively; also id_i, con_i, and cat_i represent the same notations for items. User latent vector p_u and item latent vector q_i for DNN model is basically represented as Equation 3.3:

$$p_u = \begin{bmatrix} id_u \\ cat_u \\ con_u \end{bmatrix}, q_i = \begin{bmatrix} id_i \\ cat_i \\ con_i \end{bmatrix} \tag{3.3}$$

After the concatenation of embedding and normalization layer results, DNN processes this concatenated vector with many hidden layers as Equation 2.2. The model is easily adapted to learn the interaction between p_u and q_i by providing a high degree of non-linearity and flexibility unlike the traditional MF techniques since they generally apply a dot product between two latent vectors of users and items by standing out the linearity of the collaborative approach.

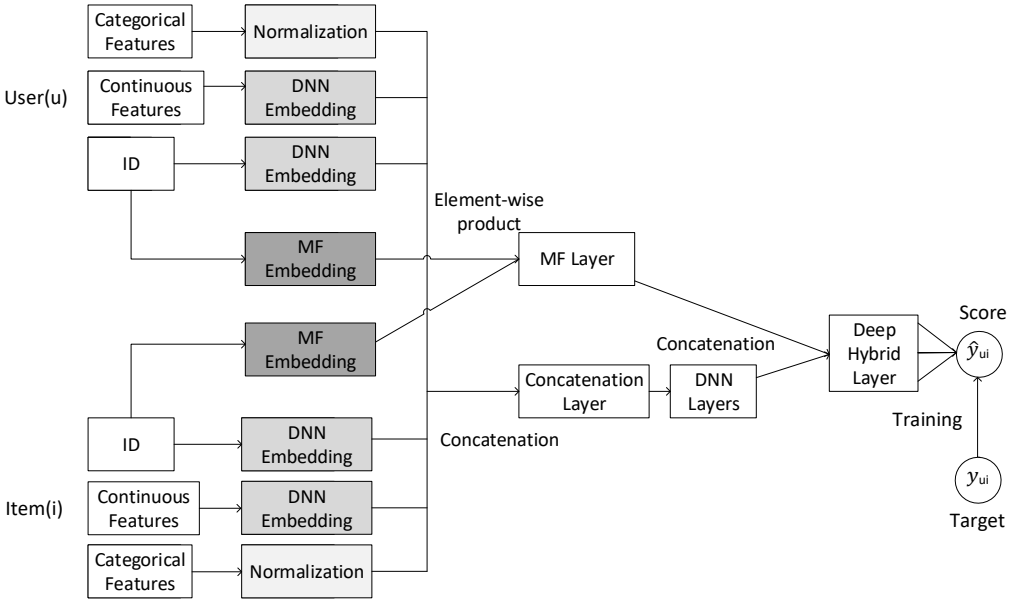


Figure 3.2 : Deep Hybrid Model

3.1.3 The shared layer

Two distinct feature mappings that are obtained from ID embeddings, categorical, continuous features mentioned in Sections 3.1.1 and 3.1.2 are fused under DeepHybrid as shown in Figure 3.2. The recommendation task of this framework is mainly to predict ranked list of items as the output of the model, and the predicted scores are calculated as equation 3.4. A top-n list is generated based on the nearest neighbors of the candidates by the ranking network [24]. Although the same embedding layer can be shared by MF and DNN models to embed user (item) IDs, MF and DNN frameworks employ different embedding layers, these embeddings are combined over the last layer of the models with concatenating to each other for the purpose of providing more flexibility to the main hybrid framework [10].

$$\psi^{MF} = p_u \odot q_i$$
$$\psi^{DNN} = \alpha_N(W_N^T(...\alpha_2(W_2^T \begin{bmatrix} p_u \\ q_i \end{bmatrix} + b_2)...)) + b_N$$
$$\hat{y}_{ui} = \alpha_{out}(h^T \begin{bmatrix} \psi^{MF} \\ \psi^{DNN} \end{bmatrix}) \tag{3.4}$$

3.2 Learning

Existing recommendation models in literature generally implements a regression with squared loss to learn model parameters, and this loss function is considered as an objective function. For example, a squared loss objective function accurately satisfies the rating prediction task since it is actually a regression problem. The rating values explicitly represent the negative and positive feedbacks of users like ranging from 1 to 5, but it may not be thought that 1, 2 values are negative, and 3, 4, 5 are positive feedbacks as an example. Unlike explicit feedbacks, the target value y_{ui} has binarized values as 0 and 1 depending on whether an interaction between user u and item i exists because of implicit data.

In our framework, y_{ui} is categorized into two; 1 expresses that user u has a link with item i, and 0 represents the lack of relationship. The prediction score $\hat{y}_{ui}$ denotes the degree of the connection for user u over item i with implicit feedback. In this manner, it can be thought that recommendation task basically aims at solving binary classication

problem. The predicted score is restricted to the range [0,1] with logistic function as the activation function of the interaction layer to realize DeepHybrid. Let $\mathbf{X}$ be all interactions of users over items, y_{ui} and $\hat{y}_{ui}$ represent the target and the predicted scores, respectively;

$$L = \sum_{(u,i) \in X \cup X^{-}} y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui}) \tag{3.5}$$

where X and X^{-} show the set of observed and unobserved interactions in $\mathbf{X}$. X^{-} is generated by randomly selecting a specific number of samples through unobserved interactions for each positive sample while X directly indicates the set of positive instances based on only the interaction because of a lack of negative samples. This optimization function is called *binary cross-entropy loss* inspired by sigmoid activation function unlike *multi-class cross-entropy loss* that utilizes softmax activation function since the loss is computed by softmax function for every output vector component(class) independent from each other in multi-class classification problem.

4. EXPERIMENTS

4.1 Dataset

We focused on Job Recommendation in our experiments, and employed the dataset of the most widely used job hunting website in Turkey.

Kariyernet: Kariyernet dataset includes application information of users to items that means which job postings are applied by which applicants. Although the original data is very large, it is extremely sparse. In this study, we selected applications from users that have minimum 10 (>10) interactions, but it has still high sparsity *99.99%* since the number of users and items are close to each other unlike other recommendation tasks such as movie, music, and book recommendations. Table 4.1 shows the statistics of this dataset. In addition to application information, some resume and personal properties of candidates, and desired properties for jobs are supplied as:

- **Candidate properties:**
  - *Continuous features*: The age of candidates are considered as a continuous feature, and normalized to a real-valued number ranging from 0 to 1.
  - *Categorical features*: Military status, gender, city where s/he lives in, current working status, the name of university which is graduated from, and lastly the department in this university represent categorical features, and embedded into a vector in the embedding layer.
- **Job posting properties:** These properties are also processed after embedding and normalization operations like candidate properties.
  - *Continuous features*: The number of candidates to be hired stands for a continuous feature.
  - *Categorical features*: Position level, the type of position, preferred gender, and education level for candidates provide us categorical characteristics.

These categorical and continuous features of users and items are taken as the input in our DNN model shown in figure 4.1. The blue-colored boxes that are age and the number of people to hire represent continuous properties that are normalized before concatenation and white-colored boxes indicate categorical properties that enter the embedding layer instead of directly concatenating in the first hidden layer of the neural network.

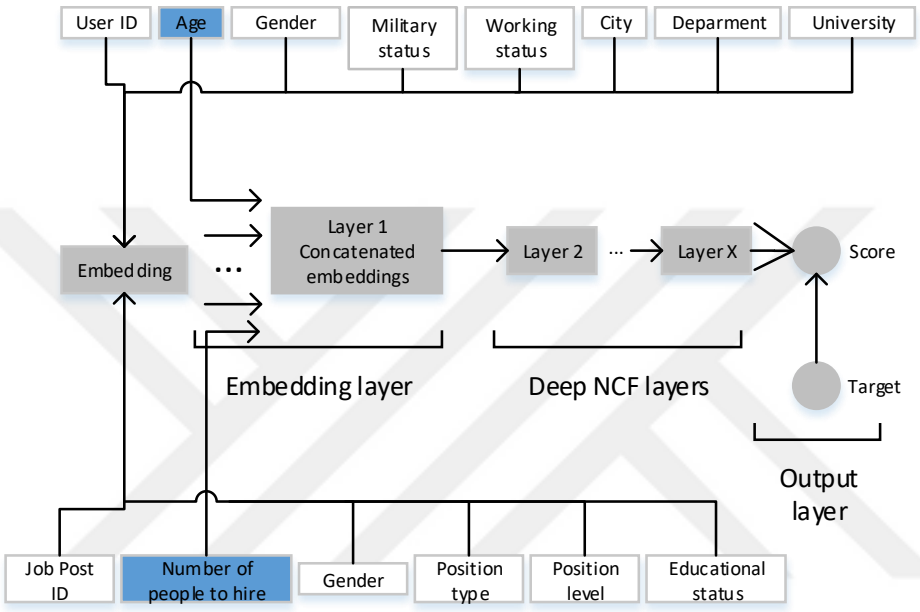


Figure 4.1 : Candidate and job posting inputs in Kariyernet

4.1.1 Preprocessing on dataset

Firstly, application data that includes IDs of candidates, job postings, companies, CV and application numbers, and the date of application are loaded, then job posting/candidate IDs and application dates are only used for this model. We numerate these IDs all over by getting counts of users and items, and items are ascendingly sorted by application dates for each user.

Secondly, candidate data that covers personal and resume properties, and job posting data that covers the desired properties for applications are processed as three ways; numerical categorical features are initially numerated like job posting/candidate IDs, string based categorical features are slugified to map integer values, and lastly continuous features are normalized to a range.

Application data is extremely sparse, so we only selected particular candidates that has minimum 10 interactions. Finally, all applications of candidates except the last one are treated for the training set. The last application that has the largest date is taken for the test set, and *99* negative samples are randomly selected from all remaining items for each element of the test set. Selection of negative samples for the test set provides us run the evaluation metrices. In this study, the model discovers how the score of the positive instance is higher than the negative instances, and the number of negative samples were determined based on the previous studies that use the same evaluation metrices. These operations belonging to item recommendation are also repeated for user recommendation in a similar way.

4.2 Experimental Settings

All preprocessing operations are implemented in Python programming language with specifically Pandas library. Our deep learning based model uses Keras [32] with Theano backend. For all implemented methods, hyper-parameters are tuned by sampling the last interaction of each user for the validation set, and the remaing data for training set. For each positive instance, negative instances are uniformly sampled from unobserved interactions since there is no negative feedback for implicit data as mentioned in section 4.1.1. In this study, we use pointwise log loss, and it provides us the flexible sampling ratio for negative instances unlike pairwise objective function. Sampling more than one negative instances is reasonable to achieve optimal performance, and three negative instances are selected in the training set, 99 negative instances are allocated for each user in the test set. Model parameters are initialized with a Gaussian distribution (0.01 standard deviation and 0 mean values), and optimized the model with binary cross entropy of equation 3.5, and mini-batch Adam optimizer [33]. The number of predictive factors for MF, and the output vector length of DNN model is selected from the set of [4, 8, 16, 32]. Moreover, continuous features are normalized into a number between 0 and 1, and categorical features are embedded to one-dimensional vector. The learning rate is selected from the values of [0.0001, 0.001, 0.01] to update the weights of neurons, and the batch size for each epoch is applied between the set of [100, 200, 400]. Consequently, our models almost show better performance with 200 batch size, 0.001 learning rate, and the predictive

Table 4.1 : Statistics of Kariyernet dataset

Candidates	Job postings	Applications	Sparsity
20283	16134	383434	99.99%

factor of 16 that represent the hidden factor for MF, and the output vector size of DNN model.

4.3 Evaluation Metric

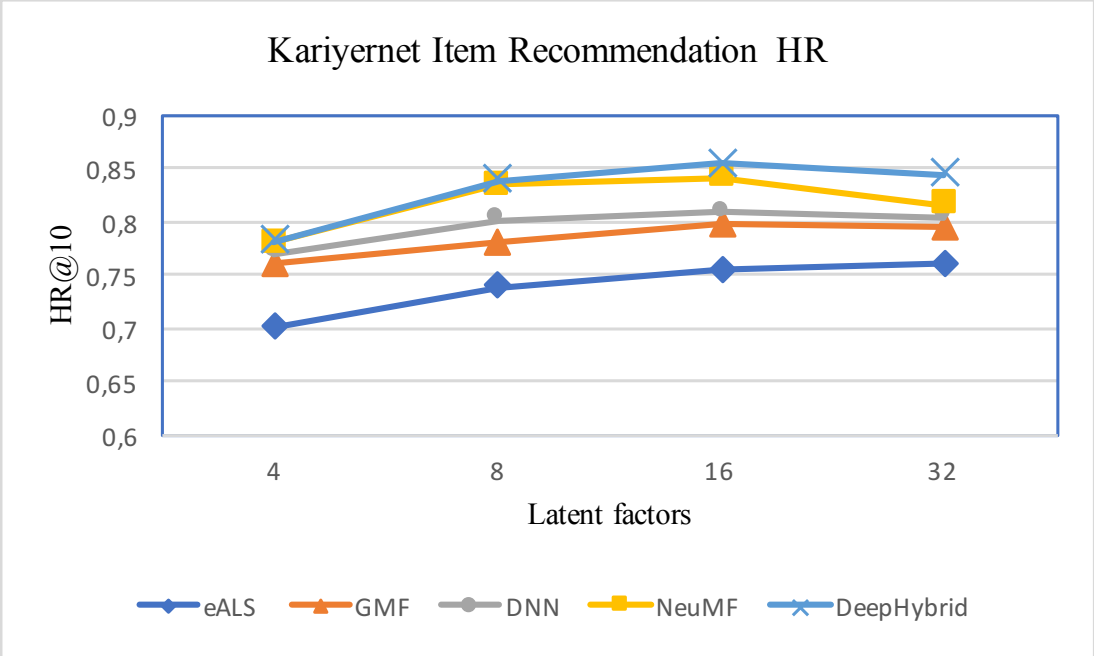
Leave-one-out cross validation strategy is used to evaluate both item and user recommendation. This methodology uses only one of the data for the test set, and all remaining data are treated as the training set [23, 34]. In this work, we sort the interactions for each user by interaction date. Although many of the studies focus on the rating prediction, we adopt on the ranking strategy that generates top-n list of items for each user or user list for each item. To evaluate model, *m* items are uniformly selected as negative instances for each positive instance of user in the test set, but this number is different from the number of negative samples in the training phase. He *et al.* employ *Hit Ratio* (HR) and *Normalized Discounted Cumulative Gain* (NDCG) evaluation metrices to evaluate the performance of top-n list [10]. HR evaluation metric only controls whether the positive item in the test set is available on top-n list of items, but the position of the positive instance is important while calculating the score in NDCG. The goal of top-n recommendation is to recommend a small set of *n* items from a large set of *m* items [35]. In this study, we randomly selected 99 items from unobserved entries for each user, then we ranked top-10 of these items in item recommendation.

4.3.1 Baseline models and performance evaluation

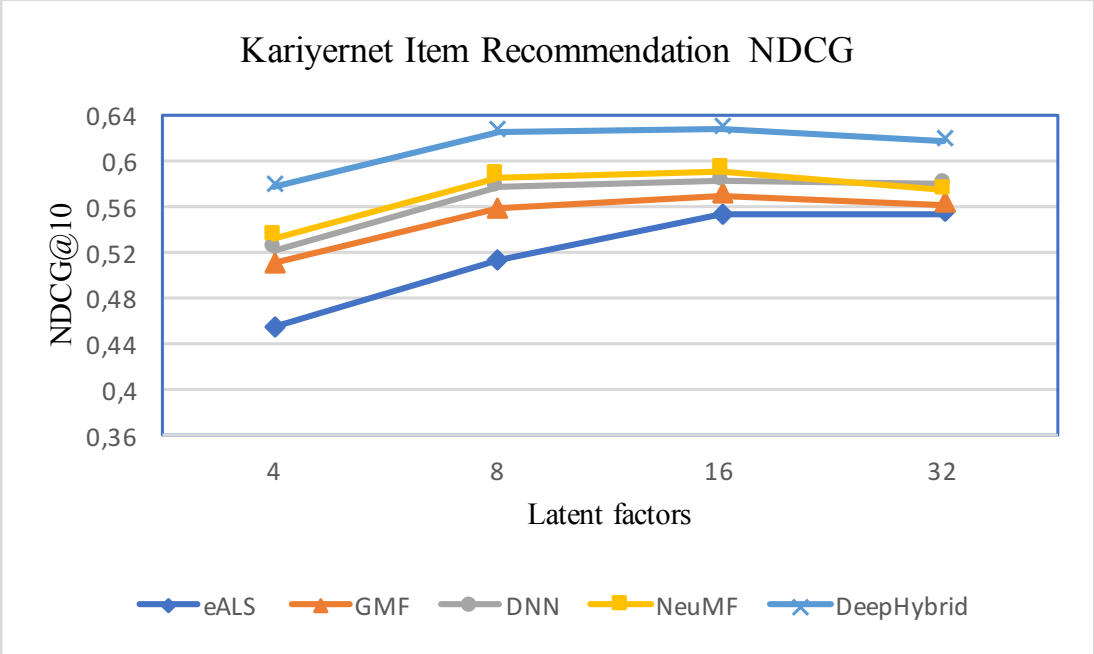
4.3.1.1 Baselines

We compared our proposed model with baseline models that were selected from the works calculating top-n list of items from the interactions of users and items.

- **eALS(CF):** eALS is a MF technique that is the most popular recommendation model for CF. In this study, it was used with implicit feedback. When we are



(a) Factors vs HR@10



(b) Factors vs NDCG@10

Figure 4.2 : Performance of HR@10 and NDCG@10 with the number of factors

creating matrix, all unobserved instances are treated as negative, but the model gives more weight to the popular instances [23].

- **GMF(CF):** This model uses ID embeddings of users and items. It gives a different perspective, and generalizes traditional MF models that utilize linear matrix multiplication. It employs element-wise product with a nonlinear activation function like sigmoid, tanh to predict the target score [10].

- **DNN(CBF)**: DNN is a neural network with more than two hidden layers. In our study, DNN model utilizes only content information of users and items while other baselines use ID embeddings. It normalizes continuous features, and categorical features are embedded in an embedding and a normalization layer. Lastly, these embedded vectors and normalized values of users and items are concatenated, then this concatenated vector is used as the input of the neural network, and the network with some hidden layers is run to learn the interaction between user and item latent features [15].

- **NeuMF(CF):** This model, designed for CF, uses two different techniques with ID embeddings. These techniques that are called GMF and MLP creates feature vectors, and they are fused under an interaction layer by concatenating. The result is optimized by *binary cross entropy loss* [10].

4.3.1.2 Performance evaluation

JR system is a type of reciprocal recommendation system, and we adopt on JR in this study. Therefore, while the most suitable candidates are selected for a job posting, job list can be also created for a candidate according as the similarity between jobs and candidate profile. In this thesis, we calculated recommendation results for both user-based and item-based. We repeated the experiments three times with HR and NDCG evaluation metrices, and the average results were recorded. The performance of baselines and our DeepHybrid model is tested on Kariyernet dataset, and the results are shown with HR@10 and NDCG@10 in Table 4.2.

In this thesis, the experimental results showed that DeepHybrid makes some improvements over the state-of-the-art methods using Kariyernet dataset. First of all, GMF reasonably outperforms eALS model even though both GMF and eALS models use only the interaction of users and items, and do not need any description or categorical information other than user and item identities. The main difference of these models is that while GMF has a non-linearity setting, eALS utilizes the linearity of MF, so it can be said that non-linearity feature is more meaningful than linear setting. Secondly, CBF contributes the enhancement of HR and NDCG evaluation metrices using DNN structure, so content information strongly learns the behaviour of the dataset. The last baseline model NeuMF joins two models with a linearity

Table 4.2 : Performance of DeepHybrid Model

	Item recommendation		User recommendation	
Methods	**HR@10**	**NDCG@10**	**HR@10**	**NDCG@10**
eALS	0.754	0.566	0.672	0.431
GMF	0.797	0.569	0.752	0.491
DNN	0.809	0.582	0.761	0.524
NeuMF	0.841	0.591	0.779	0.543
DeepHybrid	**0.855**	**0.628**	**0.795**	**0.581**

and a non-linearity setting, and it enables more flexibility than the other baseline models to learn the user − item interaction function. This fused model improved the recommendation performance in comparison with the other baseline models for both user and item recommendation. While DNN uses auxiliary information with CBF setting, other three baseline models that are accepted as CF approaches employ only user − item historical interaction. Lastly, our DeepHybrid model shows the best performance over all baseline models by integrating CBF and CF approaches with neural networks. This proposed approach concatenates categorical and continuous features with ID embeddings, and it makes the hidden feature vector of users and items more efficient. The experimental results also show that HR for DeepHybrid is advanced with ***1.7%*** ratio for both user and item recommendation over NeuMF, but DeepHybrid almost gains ***6%*** improvement in terms of NDCG score. As a result, these evaluation results show that resumes of candidate and job posting details allow the positive instance to be ahead in top-n list for Kariyernet dataset.

Moreover, latent factor is one of the most important variables to evaluate model. This predictive factor is accounted as the output vector size of the last hidden layer for deep learning based approaches, but it stands for the latent factor in eALS and GMF models. Figure 4.2 shows the performance of HR@10 and NDCG@10 for each model and each latent factor values with [4, 8, 16, 32]. HR and NDCG evaluation metric values are lower for the latent factor 4 in comparison with 8, 16, 32 vector sizes that are very close to each other. We decided to compare models with the latent factor 16 based on the results of the experiments for Kariyernet dataset. These results show that lower factor sizes may not adequately learn the interaction function while larger factor sizes

cause overfitting of the model, so the optimal value is selected as 16 depending on the size of the dataset.

5. CONCLUSIONS AND RECOMMENDATIONS

In this thesis, we proposed a new model that uses deep learning approaches for recommendation task, and this model integrates two types of recommendation strategies that refer to hybrid recommender system. Firstly, it takes user − item interactions for CF. Secondly, CBF takes advantage of user and item properties from content information. MF and DNN models form the basis of DeepHybrid framework, and they are fused into a shared layer to propose top-n list of candidate items by combining the strengths of non-linearity of DNN and linearity of MF. Auxiliary content information that represents users and items enables the model to address the cold-start problem in CBF part. This approach focuses on implicit data, and all interactions are operated as positive instances, so lack of explicit feedback brings about uniformly sampling negative instances from unobserved entries.

In our experiments, we used Kariyernet dataset to evaluate the performance of the model, and this dataset used for JR enabled us reciprocal recommendation like online dating apps. Reciprocal job recommendation task both provides a candidate with the most suitable jobs and orders the fittest candidate list for a job posting. The performance of a ranked list is determined by *HR* and *NDCG* [36]. The results of these evaluation metrices indicate that user recommendation produces lower score than item recommendation since the candidate set of a job posting is larger than the number of jobs to be selected for a candidate.

We aim to work on text-based information of users and items using Natural Language Processing approaches in the future due to existing a big amount of information in text. For instance, explanation of a job posting, or experience details and characteristics of a candidate will enrich the hidden feature space that is the main supply of deep learning approaches. As a second future work, we will rank top-n list of items after candidate generation task. We are uniformly sampling items to be evaluated for users, and selecting top-n list among this randomly selected items in this work. However, it may recommend irrelevant items to users, or opposite although the higher evaluation

metric values because of easily finding the nearest items from random selection. This candidate generation task will utilize some clustering methods, and will decrease HR and NDCG values, but it will recommend more relevant items to users.

REFERENCES

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

[2] **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 & Data Engineering*, (6), 734–749.

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

[4] **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*, ACM, pp.195–204.

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

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

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

[8] **Wan, J.**, **Wang, D.**, **Hoi, S.C.H.**, **Wu, P.**, **Zhu, J.**, **Zhang, Y. and Li, J.** (2014). Deep learning for content-based image retrieval: A comprehensive study, *Proceedings of the 22nd ACM international conference on Multimedia*, ACM, pp.157–166.

[9] **Socher, R.**, **Bengio, Y. and Manning, C.D.** (2012). Deep learning for NLP (without magic), *Tutorial Abstracts of ACL 2012*, Association for Computational Linguistics, pp.5–5.

[10] **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.

[11] **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*, ACM, pp.1235–1244.

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

[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*, ACM, pp.353–362.

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

[15] **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.

[16] **Lee, D.D. and Seung, H.S.** (2001). Algorithms for non-negative matrix factorization, *Advances in neural information processing systems*, pp.556–562.

[17] **Mnih, A. and Salakhutdinov, R.R.** (2008). Probabilistic matrix factorization, *Advances in neural information processing systems*, pp.1257–1264.

[18] **Lops, P.**, **De Gemmis, M. and Semeraro, G.**, (2011). Content-based recommender systems: State of the art and trends, Recommender systems handbook, Springer, pp.73–105.

[19] **Van den Oord, A.**, **Dieleman, S. and Schrauwen, B.** (2013). Deep content-based music recommendation, *Advances in neural information processing systems*, pp.2643–2651.

[20] **Koren, Y.** (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model, *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp.426–434.

[21] **Hu, Y.**, **Koren, Y. and Volinsky, C.** (2008). Collaborative Filtering for Implicit Feedback Datasets., *ICDM*, volume 8, Citeseer, pp.263–272.

[22] **Rendle, S.** (2010). Factorization machines, *2010 IEEE International Conference on Data Mining*, IEEE, pp.995–1000.

[23] **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.

[24] **Zhang, S.**, **Yao, L.**, **Sun, A. and Tay, Y.** (2019). Deep learning based recommender system: A survey and new perspectives, *ACM Computing Surveys (CSUR)*, *52*(1), 5.

[25] **Cheng, H.T.**, **Koc, L.**, **Harmsen, J.**, **Shaked, T.**, **Chandra, T.**, **Aradhye, H.**, **Anderson, G.**, **Corrado, G.**, **Chai, W.**, **Ispir, M.** *et al.* (2016). Wide & deep learning for recommender systems, *Proceedings of the 1st workshop on deep learning for recommender systems*, ACM, pp.7–10.

[26] **Soboroff, I. and Nicholas, C.** (1999). Combining content and collaboration in text filtering, *Proceedings of the IJCAI*, volume 99, sn, pp.86–91.

[27] **Albadvi, A. and Shahbazi, M.** (2009). A hybrid recommendation technique based on product category attributes, *Expert Systems with Applications*, *36*(9), 11480–11488.

[28] **Wang, J.**, **Kenthapadi, K.**, **Rangadurai, K. and Hardtke, D.** (2017). Dionysius: A framework for modeling hierarchical user interactions in recommender systems, *arXiv preprint arXiv:1706.03849*.

[29] **Lee, Y.C.**, **Hong, J. and Kim, S.W.** (2016). Job recommendation in AskStory: experiences, methods, and evaluation, *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, ACM, pp.780–786.

[30] **Li, L. and Li, T.** (2012). MEET: a generalized framework for reciprocal recommender systems, *Proceedings of the 21st ACM international conference on Information and knowledge management*, ACM, pp.35–44.

[31] **Xia, P.**, **Liu, B.**, **Sun, Y. and Chen, C.** (2015). Reciprocal recommendation system for online dating, *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, ACM, pp.234–241.

[32] **Gulli, A. and Pal, S.** (2017). *Deep Learning with Keras*, Packt Publishing Ltd.

[33] **Kingma, D.P. and Ba, J.** (2014). Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*.

[34] **Rendle, S.**, **Freudenthaler, C.**, **Gantner, Z. and Schmidt-Thieme, L.** (2009). BPR: Bayesian personalized ranking from implicit feedback, *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, AUAI Press, pp.452–461.

[35] **Cremonesi, P.**, **Koren, Y. and Turrin, R.** (2010). Performance of recommender algorithms on top-n recommendation tasks, *Proceedings of the fourth ACM conference on Recommender systems*, ACM, pp.39–46.

[36] **He, X.**, **Chen, T.**, **Kan, M.Y. and Chen, X.** (2015). Trirank: Review-aware explainable recommendation by modeling aspects, *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, ACM, pp.1661–1670.

CURRICULUM VITAE

Name Surname: Muhammet Çakır

Place and Date of Birth: Çorum 1991

E-Mail: cakirmuha@itu.edu.tr

EDUCATION:

- **B.Sc.:** 2016, Istanbul Technical University, Faculty of Computer and Informatics, Computer Engineering
- **M.Sc.:** 2019, Istanbul Technical University, Faculty of Computer and Informatics, Computer Engineering Department

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2016-2017 Software Developer / Hitit Computer Services
- 2017-2018 Software Developer / Iugo Technology
- 2018-Present Software Developer / MaviUçak Bilişim

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- Çakır M., Gündüz Öğüdücü Ş., Tugay R., 2019. A Deep Hybrid Model for Recommendation Systems. *AIIA19 - The 18th International Conference of the Italian Association for Artificial Intelligence UNICAL*, November 19-22, 2019 Rende (CS), Italy.