공학석사학위논문

Time-varying Item Feature Conditional Variational Autoencoder for Collaborative filtering

협업 필터링을 위한 시간에 따라 변화하는 상품 특징 조건부 변분 자동 생성기

2019 년 8 월

서울대학교 대학원

산업공학과

김 지 영

Time-varying Item Feature Conditional Variational Autoencoder for Collaborative filtering

협업 필터링을 위한 시간에 따라 변화하는 상품 특징 조건부 변분 자동 생성기

지도교수 조 성 준

이 논문을 공학석사 학위논문으로 제출함

2019 년 7 월

서울대학교 대학원

산업공학과

김 지 영

김지영의 공학석사 학위논문을 인준함

2019 년 7 월

위 원 장 박 종 헌 (인)

부위원장 조 성 준 (인)

위 원 이 덕 주 (인)

Abstract

Time-varying Item Feature Conditional Variational Autoencoder for Collaborative filtering

Jeeyung Kim
Department of Industrial Engineering
The Graduate School
Seoul National University

We can assume that some factors that affect the user's decision to select products are several factors such as the time-invariant user's unique taste and the external trends or fashion that varies with time. Both mentioned factors should be considered in order to create a precise recommendation system, but the current recommendation system has the problem of making recommendations based only on the user's history without taking into account the timing of creating a recommendation.

Therefore, considering the timing of the recommendation made, this paper proposes a recommendation system that reflects inter-items trends of time-based bin. We focus on creating a model that could effectively combine the content-based recommender system with the context-based recommender system. Specifically, we use Conditional Variational Autoencoder to add a time dynamic item features to user-item implicit feedback data. In this case, distributed representation of items in the specific period is used as a condition that is added to input and latent variable of

VAE respectively. The distributed representation per periods can be extracted using LSTM. By putting a condition into VAE, a hybrid recommendation system can be created to reflect the item trend.

The model proposed in this paper differs from existing research in that it reflects the changing characteristics inherent in the product and utilizes it in the recommendation. In addition, we can get the additional effect of solving sparsity problem by using item feature to mitigate sparsity problem. The Movielens data (ml-1m) data and Amazon women's clothing dataset are used for the evaluation of the proposed model. The effectiveness of this model is verified by designing experimental methods to evaluate the recommended systems that reflect the time point of recommendation.

Keywords: Recommender system, Conditional variational autoencoder, LSTM

Student Number: 2017-24854

Contents

List of Tables

List of Figures

Chapter 1

Introduction

As the amount of information available to users on the websites has increased dramatically, recommender systems has become more important.

In this thesis, we consider the recommender system problem occurs in real-world industry. We start by describing the problem in Section 1.1.

1.1 Problem Description

Users consume a variety of contents such as videos, products, movies, music and so on thousands of times a day. The purpose of the recommender systems is to increase user consumption by recommending items that users would prefer among the vast amounts of content based on user history data or item feature data.

Recommendation systems can be classified into two types, content-based recommender system and context-based recommender system. The context-based recommender system, also known as collaborative filtering, is trained in accordance with user history data which consists of user-item rating or preference. On the other hand, the context-based recommender system uses the unique characteristics of item, for example, images of products or genre of movies. The hybrid recommender system, a

combination of these two systems, has advantage in terms of applying two different types of source data.

We assume that there are two factors that affect a user's choice when selecting an item. One is user's internal preference or taste, and the other is the trend changes of inter-item. For example, in e-commerce, clothes visual feature which recognized as "fashionable" change as time progresses. The elements of popular movies also change with the times according a situation or fashionable element of the times. In detail, the super-hero movies is popular in certain times, but suspenseful or swashbuckler movies could become more popular over time. Therefore, it is important to suggest items by considering when users choose them, and it is necessary to create a recommendation system that reflects the unique characteristics of the items at that time point.

However, there is a limitation in terms of model capacity by using the linear model to express this complex user-item interaction. In order to consider a nonlinear pattern, it is necessary to utilize the neural network, which is being studied a lot these days. Recently, the number of research applying neural networks to the collaborative filtering increased, which accordingly is followed by promising results. (see e.g. [6, 8, 19, 15, 7, 4]) In particular, content-based recommender systems mainly use dense data such as images and text, and area in which neural networks can improve performance. (see e.g. [2, 20, 19]) Among various structures of neural networks, auto-encoder structure is one of the main model for collaborative filtering. In the recent, the number of researches on applying variational autoencoder in recommender system also have increased.

In this paper, our main goal is to address the reasonable method of combining

user item history and according item time-variant feature. To achieve the goal, we propose a new structure that contains RNN(Recurrent Neural Network) which learn item time-variant pattern and Conditional VAE(Variational Autoencoder) which generate user-item interaction by using the result from recurrent neural network.

1.2 Research Motivation and Contribution

Our research motivations and main contributions of the thesis are as follow:

(a) We introduce a new hybrid recommendation system structure and corresponding training method which is able to combine temporal dynamics item features as the condition concatenated with each user-item matrix.

(b) We prove effectiveness of using using item feature which represents trends of the era in recommender system.

(c) We validate our method by showing outperforming empirical results on two types of large-scale datasets, Movielens 1-m and Amazon women's Clothing. One has movie-categories score features and the other has clothing visual features.

1.3 Organization of the Thesis

The thesis is composed of 5 chapters. In Chapter 2, we review literatures related to the problem. In Chapter 3, we propose various solution approaches. In Chapter 4, results of computational experiments are presented. Finally, in Chapter 5, we give concluding remarks and possible future research directions of this thesis.

Chapter 2

Literature Review

2.1 Review on Neural Networks model

2.1.1 LSTM

Recurrent Neural Network(RNN) well-known neural networks which has directed cycle which is beneficial to represent sequential data. RNN ties the weights at each time step and condition the neural network on all previous input series.

$$h_t = \sigma(W^{hh} h_{t-1} + W^{hx} x_t) \tag{2.1}$$

$$\tilde{y} = softmax(W^s h_t) \tag{2.2}$$

$$L_j = -\sum_{j=1}^{N} y_j log\hat{y}_j \tag{2.3}$$

where $softmax = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$ for $i = 1, \cdots, N$ and $\mathbf{z} = (z_1, \cdots, z_N) \in \mathbb{R}^{\mathbb{N}}$.

LSTM(Long Short Term Memory) is one of the RNN model types, which has advantage of mitigating vanishing gradient problem by introducing cell state to RNN

state. The hidden state is modified to

$$\begin{aligned} f_t &= \sigma(W^{xh_f} x_t + W^{hh_f} h_{t-1}) \\ i_t &= \sigma(W^{xh_i} x_t + W^{hh_i} h_{t-1}) \\ o_t &= \sigma(W^{xh_o} x_t + W^{hh_o} h_{t-1}) \\ g_t &= \sigma(W^{xh_g} x_t + W^{hh_g} h_{t-1}) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot tanh(c_t) \end{aligned} \tag{2.4}$$

in LSTM.

RNN based models have been applied various types of sequential data such as text, voice and image data and have shown significant improvement.

2.1.2 Variational Autoencoder

Variational autoencoder is a type of the generative models using latent variable. Encoder which consists of neural network creates latent variable z in accordance with input x. Decoder which also consists of neural network reconstructs x with z that is created by the encoder. In other words, while the encoder acts as an abstraction of inputs and extracts features of inputs, the decoder reconstruct original data based on latent features. In order to train variational autoencoder, we have to assume that latent variable z has normal distribution, which allows us to estimate model parameters with maximum log likelihood. Also, reparameterization trick, a method for sampling noise in zero-mean Gaussian distribution, is used in order to train with backpropagation.

In equations below, p_θ represents encoder and q_ϕ means decoder.

$$
\begin{aligned}
logp_\theta(x) &= log \int p_\theta(x|z) \frac{p(z)}{q_\phi(z|x)} q_\phi(z|x) dz && (2.5)\\
&\geq \int log(p_\theta(x|z) \frac{p(z)}{q_\phi(z|x)} q_\phi(z|x) dz && (2.6)\\
&= \int log(p_\theta(x|z) \frac{p(z)}{q_\phi(z|x)}) q_\phi(z|x) dz && (2.7)\\
&= \int log(p_\theta(x|z) q_\phi(z|x) dz && (2.8)\\
&\qquad - \int log(\frac{q_\phi(z|x)}{p(z)}) q_\phi(z|x) dz
\end{aligned}
$$

As a result, objective function that we minimizes is

$$J(x) = -\mathbb{E}_{q_\phi(z|x)}[logp_\theta(x|z)] + D_{KL}[q_\phi(z|x)|p_\theta(z)] \tag{2.9}$$

2.1.3 Conditional Variational Autoencoder

Kingma et al.[9] introduces method of using label information as condition which add to input and latent variable. He shows that classification and generation performance of semi-supervised learning is improved by applying label information of image data. The objective function of conditional VAE is similar to that of VAE. The only thing that changes is conditioning all of the distributions with variable c.

The variational lower bound objective is in the following form:

$$J(x) = -\mathbb{E}_{q_\phi(z|x,c)}[logp_\theta(x|z,c)] + D_{KL}[q_\phi(z|x,c)|p_\theta(z|c)] \tag{2.10}$$

2.2 Review on recommender system

Collaborative filtering using encoder and decoder structure Wang et al. [18] use autoencoder structure in order to compress item-user matrix to latent space, and multiplys it with user vector to predict user-item rating. To train the model, ALS(Alternative Least Square) algorithm is used. Kuchaiev et al. [11] propose deep autoencoder effectiveness in recommender system. They validate that deep autoencoder models generalize better than shallow models, and non-linear activation function is crucial part for training deep models.

There are various of studies applying VAE to recommender system.(see e.g.[18, 17, 19]) Liang et al. [12] introduce to use VAE for recommender system and show state-of-the-art performance in implicit feedback data. The authors propose specific suggestions to apply VAE to recommender system. First, they suggest use multinomial likelihood rather than Gaussian and logistic likelihoods which have been mainly used in recommender system. They also validate effectiveness of KL-annealing in order to prevent suffering from underfitting with high-dimensional and sparse data.

Hybrid recommender system using VAE Many studies have applied VAE in recommender system for the purpose of using side information from items or users. (see e.g. [6, 8, 19, 15, 7]) Jhamb et al. [6] put attention structure to use with autoencoder for utilizing item and user side information. In addition, Karamanolakis et al. [8] use item features such as movie reviews to make distribution which replaces zero-mean Gaussian distribution which is used for sampling noise. Wang et al. [19] use movie plot data as item side information, they are used for user-item rating prediction while generating movie plot as part of multi-task learning.

Recommender system considering temporal dynamics Hidasi et al. [5] use

GRU(Gated Recurrent Unit) for modeling sequential pattern within session. Koren et al. [10] use temporal dynamics for SVD.(Singular Value Decomposition) The authors assume that the phenomenon that items' and users' average rating values fluctuate by time, and proposes the method including time drifting parameters. However, there is limit to the two studies in that they don't use item or user side information. He et al. [3] use amazon visual information. The paper first consider item feature dynamics, and reflect this features to build recommender system. However, there is limited model capacity because linear model is used.

Chapter 3

Solution Approaches

We consider only implicit feedback. Notations used in this paper are as follows.

Table 3.1: Notation

Notation	Explanation
U, I	user set, item set
I_u^+	the items for which user u had positive feedback
N_i^+	the number of items belong to I_u^+
N_u	the total number of users
v_t	deep CNN visual features of item at t timestamp
V	dimension of deep CNN visual features
s_t	relevance scores corresponding tags of item at t timestamp
S	dimension of relevance scores
$\hat{x}_{u,i}$	predicted preference of user u towards item i
P_n	item set which belongs to time-based bins n
Tr_u, V_u, Te_u	training/validation/test datasets
K	dimension of condition or RNN hidden vector
D_c	dimension of latent variable

3.1 Content Feature Encoder - Long Short Term Memory

We encode item contents features that users have selected with LSTM(Long Shot Term Memory). We expect LSTM can capture sequential pattern well even though in the case sequence length increase in case that user select a number of items.

$$h_t = \sigma(W^{hh} h_{t-1} + W^{hx} x_t) \tag{3.1}$$

In the above equations, x_t is according item feature, and each dataset has a different type of item feature. ($x_t = s_t$ or v_t) Each item was chosen or positively rated in the past by the same user. Item sets are sorted by time in order to be entered in LSTM. The last hidden vector h_T is considered as the result of encoded time-series contents features.

Our purpose to train LSTM is extracting time-dependent item features which not only differ from among time-based bins but also change gradually even within the same users. Therefore, we define objective function as cross entropy to maximize likelihood. Each class label is the time-based bin index which the users who consumed the items belongs to. Besides, the items feature which are the input of LSTM are chosen or rated only by users who belongs to the same bin. Experimental settting is elaborated in more detail in chapter 4. LSTM is pretrained to train whole model end-to-end in a stable manner.

3.2 User-Item matrix generator - Conditional Variational Autoencoder

3.2.1 Conditional Variational Autoencoder

It has already been proven in the previous paper that using variational autoencoder in the recommender system outperforms using autoencoder because variational autoencoder has beneficial when it comes to regularize better by using prior distribution and to contain data-points variance by getting mean and variance. [12] To utilize VAE structure in our research, we follow the experimental setting the previous paper suggested where KL-annealing and multinomial distribution is used.

Modified to consider the temporal dynamic content feature is that add the condition to either input and latent variable z in a manner conditional variational autoencoder. The encoded item features that are able to represent each period characteristic are used for condition. Instead of one-hot label data [9], we use content-encoded dense vector(C_{P_n}) which can be acquired after applying sigmoid function to affine transformation output of the last LSTM hidden vector. We expect that time-based bin trend or fashion information represented by encoded item features helps reconstructing user vector by predicting better what users prefer. The proposed equations are as follows.

$$
\begin{aligned}
log(p_\theta(x, C_{P_n})) &= log \int p_\theta(x, C_{P_n}|z) \frac{p(z)}{q_\phi(z|x, C_{P_n})} q_\phi(z|x, C_{P_n}) dz && (3.2)\\
&\geq \int log(p_\theta(x, C_{P_n}|z) \frac{p(z)}{q_\phi(z|x, C_{P_n})} q_\phi(z|x, C_{P_n}) dz && (3.3)\\
&= \int log(p_\theta(x|z, C_{P_n}) \frac{p(C_{P_n})p(z)}{q_\phi(z|x, C_{P_n})}) q_\phi(z|x, C_{P_n}) dz && (3.4)
\end{aligned}
$$

We call (3.4) as ELBO, which we want maximize in order to maximize $log(p_\theta(x, C_{P_n}))$, so that the objective function we want to minimize is

$$
L(x, C_{P_n}) = \underbrace{-\mathbb{E}_{q_\phi(z|x,C_{P_n})}[log(p_\theta(x|z, C_{P_n}))]}_{\text{reconstruction loss}} + \underbrace{D_{KL}(q_\phi(z|x, C_{P_n})|p(z|C_{P_n}))}_{\text{KL-Divergence}} \quad (3.5)
$$

In our setting, we can use

$$
L(x, C_{P_n}) = -\mathbb{E}_{q_\phi(z|x,C_{P_n})}[log(p_\theta(x|z, C_{P_n}))] + D_{KL}(q_\phi(z|x, C_{P_n})|p(z)) \quad (3.6)
$$

practically in that z and C_{P_n} is independent.

We assume multinomial distribution for calculating reconstruction error like [12]

$$
reconstruction\, loss = -\sum_i x_{ui} log \pi_i(z_u, C_{P_n}) \quad (3.7)
$$

Also we use KL-annealing, which β increase from 0.0 to 0.2. The final objective function containing reconstruction error and KL-divergence is

$$
L_i = -\sum_i x_{ui} log \pi_i(z_u, C_{P_n}) + \beta \cdot D_{KL}(q_\phi(z|x_i)|p(z)) \quad (3.8)
$$

We use user-item click matrix as VAE input which consists of non-zero values($I \in I_u^+$) and zero values.($I \notin I_u^+$) Using content features corresponding items ($I \in I_u^+$), we make encoded item feature for each user from pretrained LSTM. In order to create items features containing characteristics of corresponding time-based bin, we enter items($I \in P_n \backslash P_k, (k \neq n)$) features to LSTM. That is, each user has their encoded item feature as condition. Also, we expect that users in the same time-based bins have similar condition value or similar tendency comparing with other users in different period. If none of the item lists consumed by the user were consumed solely in the corresponding bin, the condition was obtained by sampling from P_n. We add condition to both input and latent variable by simply concatenating together.

By using condition, the general trend or feature information related to each bin can be put into consideration when user-item matrix generated. It means that we use item temporal dynamic side information to predict user-item preference. Furthermore, We expect the model to focus more on elements other than temporal dynamic item characteristics when it is trained.

3.2.2 Attention Conditional Variational Autoencoder

We append an attention layer in the decoder part. Input and condition has attention layer respectively in order to generate attention score which will be applied to condition. After calculating element-wise multiplication between condition and attention

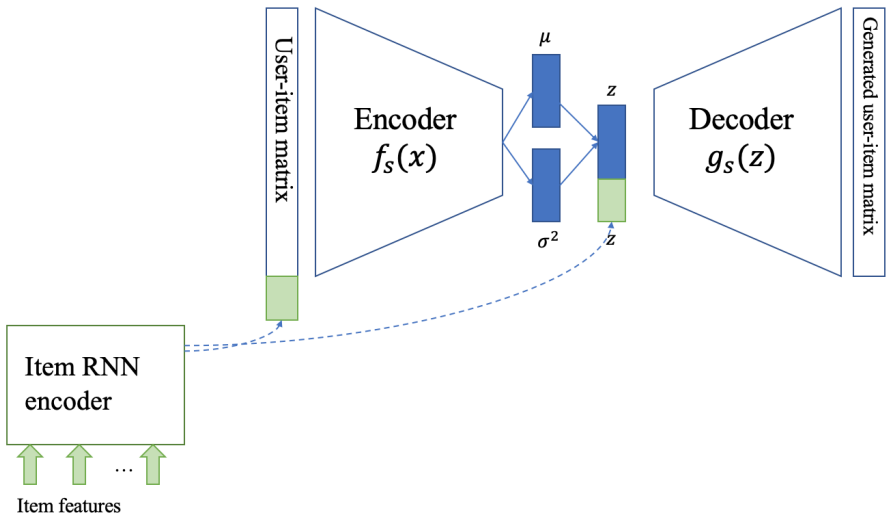

Figure 3.1: model structure

score, condition that attention layer applied to can be concatenated with the input.

$$attn = softmax(ELU(W_{ax}z + W_{ac}c_u)) \\ c_{uattn} = attn \odot c_u \tag{3.9}$$

where $W_{ax} \in \mathrm{R}^{D^c \times K}$, $W_{ac} \in \mathrm{R}^{K \times K}$, $ELU(x) = max(0, x) + min(0, \alpha * (exp(x)\text{-}1))$.

3.3 Process of Training

In this section, we elaborate training process of the proposed method. The described process can be adopted in real-world industry which deals with recommender systems.

1. **Pre-train Item-feature encoder LSTM** We pre-train LSTM by extracting the items which only were consumed in the specific period.

2. **Calibrate a condition per user from pre-trained LSTM** The feature of the item contents corresponding to the user's history is entered into pre-trained LSTM in order to obtain the last hidden vector of each user which will be used for condition distribution for VAE. We use items that were consumed during the specific time-based bin only.

3. **Concatenate the condition with each user input** The result of adding the condition to the user-item matrix is utilized as input of VAE. In detail, we use $(u; C_{P_n})$ as input to encoder part of VAE. Also, we add the same encoded item features to the latent variable z which is inferred from VAE encoder part. We use $(z; C_{P_n})$ as input to decoder part of VAE.

4. **Train the model with both VAE loss and RNN loss** We train the proposed model end-to-end by updating gradients and conducting backpropagation with both VAE loss(Reconstruction error + KL-divergence) and RNN loss.

Pseudo code for the training process is as follows.

Algorithm 1 Algorithm 1: VAE-SGD Training collaborative filtering VAE with stochastic gradient descent

Input: Click matrix $X \in R^{N_U \times N_I^+}$, content feature c $\in R^{S,V}$, the number of time-based bins

Split users into the specific time-based bins based on the last time the user consumed an item

procedure TRAININGCFCVAE($\theta, \phi, \theta_{LSTM}$)

initialize θ, ϕ , and load θ_{LSTM} from pretrained LSTM

repeatSample a batch of users U

foreach $u \in U$ **do**

Sort items by timestamp and input sorted item features to pretrained LSTM to get encoded result

$C_{P_n} = \sigma(W^L \sigma(W^{hh} h_{T-1} + W^{hx} x_T))$

Sample $\epsilon \sim N(0, I_k)$ and compute z_u via reparametrization trick

Use $(x_u; C_{P_n}), (z_u; C_{P_n})$ to VAE

Compute gradient $\bigtriangledown_{\theta_{LSTM}}$ for LSTM

Compute gradient $\bigtriangledown_{\theta}$ and $\bigtriangledown_{\phi}$ for VAE

end for

Average noisy gradients from batch, and update $\theta, \phi, \theta_{LSTM}$

until convergence

return $\theta_{LSTM}, \theta, \phi$;

end procedure

Chapter 4

Experimental Procedure

Through experiments, we tried to show the two most important characteristics of our model. The first is to show that the proposed method had better recommendation performance than the existing recommender systems using VAE. Second, we try validating robust performance improvement regardless of content feature types.

4.1 Evaluation Datasets

We use two datasets of different applications. The movielens-1m dataset is from website providing users with movie recommender service. The dataset consists of user-movie ratings, and corresponding movie have genome data which is relevance score for 1128 tags. [16] Tags include movie atmosphere, subject, topic or genre. For example, tags consists of super hero, technology, touching and toys, etc.

The other dataset is user-product rating data from Amazon, one of the biggest online shop. There are various categories in the online shop, but we filter out only women's clothing items. Each item have image features which are extracted using Convolutional Neural Networks. In detail, they selected 5 CNN layers and 3 fully connected layers which is pre-trained with ImageNet dataset.[3, 13]

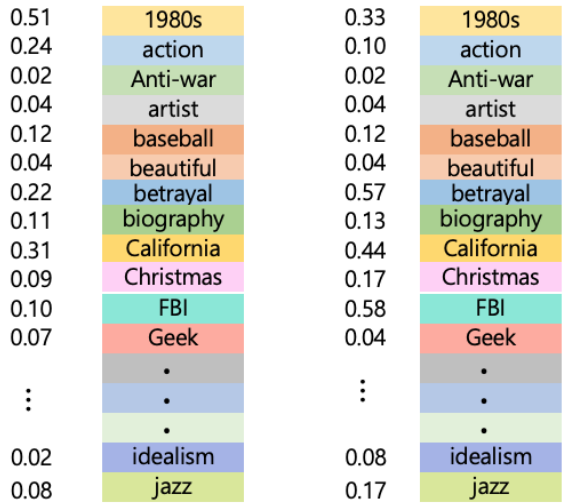

Figure 4.1: movie genome dataset

We transform the explicit data to implicit data by keeping ratings higher than three in order to interpret explicit rating data as implicit feedback. Also, we only keep users who have watched at least five movies. For Amazon women's clothing dataset, we keep items which have consumed by users more than 7 times. Table 4.2 summarizes the dimensions of two datasets after preprocessing.

Table 4.1: Description of test instances

Dataset	Movielens 1M	Amazon women's clothing
#users	6,034	19,563
#items	3,395	31,842
#interactions	573,351	137,181
sparsity	2.79%	0.022%
time span	Jan 2000 - Feb 2003	May 1996 - July 2014

Figure 4.2: Amazon women's clothing dataset

4.2 Experimental setting

We split users into each time-based bin based on the last time users consumed items. The number of users in each bin is constant, the number of interactions among the period can be different. We consider degree of finer resolution and the need to have enough user-item interaction per bin in terms of spliting timeline into bins. For both the movie rating data and the Amazon data, there are a wide variety of bin sizes that yield about similar accuracy. In our experimental setting, we split Movielens dataset into 3 bins, and Amazon dataset into 5 bins. While bin corresponds to about 1 year in Movielens dataset, bin corresponds to about 3.6 years in Amazon dataset. Users belong to one specific bin, but items can be belong to multiple bins because we train the model based on users.

When it comes to splitting train/validation/test sets, we sample each 15% users in each bin for validation and test sets. The rest users are used for training. We tag items which are consumed in the specific bin only to enter LSTM on the assumption that those items represents temporal characteristics for both pre-training and training. The dimension of item features differs depending on datasets. Movielens

1M dataset has 1128 dimension of tag relevance scores, on the other hand, Amazon dataset uses clothing visual feature of 4096 dimension.

In terms of structure of VAE part, it varies depending on the dataset used. For movie lens 1m dataset, we set 100 dimension as latent variable. Because the size of dataset is not large, the whole structure is [# of item $\rightarrow$ 100 $\rightarrow$ # of item]. For Amazon women's clothing dataset, the dimension of latent variable is 100 as same, but add two hidden layers whose dimension is 500. ([# of item $\rightarrow$ 500 $\rightarrow$ 100 $\rightarrow$ 500 $\rightarrow$ # of item]) We add condition to both input and latent variable, which has 50 dimension. We apply tanh activation at both the bottleneck layer as well as the output layer. The Kullback-Leibler annealing method is also set which increases linearly for 200,000 gradient updates. We apply dropout at the input layer with probability 0.5 which result in negative sampling because there are vast negative samples comparing with positive samples due to recommender system dataset. We train for 200 epochs with Adam optimizer and 100 batch-size users. Also, learning rate is 0.001 for both VAE and LSTM. We save the model showing the best validation NDCG@100 result for Amazon women's clothing dataset, NDCG@50 for movielens 1-m dataset and report test set metrics with it. The source code is made based on deep learning library Pytorch.

4.3 Evaluation Metrics

We use 3 different evaluation metrics to prove our performance, AUC, Recall and NDCG. Those are common metrics for evaluating Top-N recommendation which recommends to each consumer a small set of N items. Items in I_u^+, are expected to get higher rank than items not in I_u^+. If relevant items get higher score than

irrelevant, which means recommending items users are likely to like, high value in each metric can be obtained. We evaluate based on given a user-item pair (u, i). Users and corresponding items are divided into train/validation/test sets.

AUC (Area Under the ROC curve) measure is :

$$AUC = \frac{1}{|U|} \sum_{u} \frac{1}{|E(u)|} \sum_{(i,j) \in E(u)} \delta(\hat{x}_{u,i} > \hat{x}_{u,j}) \tag{4.1}$$

$$\delta(x) = \begin{cases} 1 & if, \quad x = True \\ 0 & otherwise \end{cases} \tag{4.2}$$

$$E(u) = \{(i, j) | i \in Te_u \wedge j \notin (Tr_u \cup V_u \cup Te_u)\} \tag{4.3}$$

NDCG (Normalized Discounted Cumulative Gain) measure is:

$$DCG@K(u, w) = \sum_{i=1}^{K} \frac{2^{\mathbb{I}[w(i) \in I_u]} - 1}{log(i + 1)} \tag{4.4}$$

where $w(i)$ is the item at rank i, and $\mathbb{I}[]$ is the indicator function because we use implicit feedback data. $NDCG@K$ is normalized value of $DCG@K$, which can be calculated divide DCG_p by ideal $IDCG_p$ for every position p.

Recall can be obtained as follows:

$$Recall@K(u, w) = \sum_{i=1}^{K} \frac{\mathbb{I}[w(i) \in I_u]}{|I_u|} \tag{4.5}$$

Recall is comparable with precision metric, but recall is more reasonable considering characteristic of recommender system which represented by sparse and large scale dataset.

4.4 Baseline

We compare results with the following standard collaborative filtering models, both linear(Matrix Factorization) and non-linear in the form of autoencoder(CFVAE). Also, we compare results with existing hybrid recommender system model(VAE-HPrior).

- Matrix Factorization [14]: Matrix Factorization is a standard linear model of the recommender system. We use SGD(Stochastic Gradient Descent) algorithm to train the model and Bayesian Personalized Ranking as optimization criterion which is known as useful method for faster and stable convergence and improved performance. The number of latent dimensions is 100. In addition, we use Adam optimizer and 0.0003 learning rate.

- CFVAE [12] : We use Variational autoencoder structure which is a non-linear model. We use exactly the same architecture as a VAE part of CFCVAE. In detail, we use multinomial likelihoods rather than Gaussian or logistic likelihoods. Also, KL-annealing method is applied to get better performance. We train the model with Adam optimizer for 200 epochs with 0.001 learning rate. It is existing state of the art method in recommender system.

- VAE-Hprior[8] : VAE-Hprior uses the same VAE structure, but modify normal

Gaussian distribution for noise sampling. We use Gaussian distribution which has mean and variance from item side information rather than using normal Gaussian distribution. The mean vector(t_u) equal to the element-wise average of user-history item feature and a diagonal covariance matrix S_u with diagonal values equal to the standard deviation of user-history item feature. The paper use word embedding for review, but in our experiment, product visual feature and movie genome data are applied as item feature to create distribution. The other hyper-parameters of variation autoencoder is identical as our setting.

4.5 Results and Discussion

Test results are shown in Table 4.2, 4.3, 4.4, 4.5. This results verify that the proposed method in this paper is more effective than existing methods.

First, we pretrain LSTM using each dataset. For Amazon women's clothing dataset, the number of period label is 5, and 3 for Movielens 1-m dataset. The pretraining results are as follow:

Table 4.2: pretrained RNN result

Dataset	accuracy	# of label(period)
movielens 1m	98.51%	3
Amazon women's clothing	96.8%	5

Pre-training LSTM with item features of time-based bin We pre-train LSTM using corresponding item features for each time-based bin. The task for pre-training is classifying time-based bins. We get the results of 98.51 % for Movielens dataset, and 96.8% for Amazon women's clothing dataset. We can confirm our proposed model validity of basic hypothesis that item features can contain time-variant characteristic or trends of each time period. Therefore, pretrained LSTM is able to be employed as time-dependent characteristic of side information encoder.

Hybrid recommender system vs. Context-based recommender system The proposed method significantly outperforms improved performance CFVAE which not apply information from item side in both Movielens 1-m and Amazon women's clothing dataset. We argue that the hybrid recommender system structure that we propose is appropriate for representing time dynamic content features

Table 4.3: Comparison between various baselines and our proposed methods using Amazon women's clothing dataset. This is the average of the results of ten time trials.

	$NDCG$@50	$NDCG$@100	$NDCG$@150
CFVAE	0.0122	0.0153	0.0167
HPrior-VAE	0.0117	0.0139	0.0156
CFCVAE-Attn	**0.0130**	**0.0155**	**0.0168**
CFCVAE	0.0118	0.0144	0.0156

Table 4.4: Comparison between various baselines and our proposed methods using Amazon women's clothing dataset. This is the average of the results of ten time trials.

	Recall@10	*Recall*@50	*Recall*@100
CFVAE	0.0132	0.0307	**0.0481**
HPrior-VAE	0.0104	0.0294	0.0420
CFCVAE-Attn	**0.0139**	**0.0333**	0.0476
CFCVAE	0.0102	0.0280	0.0427

comparing to context-based recommender system considering only user-item interaction. This indicates that the model can be trained in the way to achieve higher evaluation value by focusing on other parts than the information contained in the time-dependent condition.

Time-variant item features vs. General item features When we compare the results with HPrior-VAE which apply general item features which are not split by time step, we can verify considering time-variant item features cause better performance than not. We can find higher scores of the proposed model than HPrior-VAE in most metrics in both datasets. We believe that adding the characteristics of items that change over time as conditions can make recommendations considering the trend of the era in which users belong, thus allowing more sophisticated recommendations. In addition, in the structural part of the model, using item side information

Table 4.5: AUC(Area Under ROC Curve) result for Amazon women's clothing dataset.

	AUC
CFVAE	0.6762
HPrior-VAE	0.6754
CFCVAE-Attn	0.6817
CFCVAE	**0.6833**

Table 4.6: Comparison between various baselines and our proposed methods using Ml-1M dataset. This is the average of the results of ten time trials.

Model	*Recall*@5	*Recall*@10	*Recall*@20	*NDCG*@5	*NDCG*@10	*NDCG*@50
CFVAE	0.4299	0.3808	**0.3421**	0.443	0.4046	**0.3519**
HPrior-VAE	0.4337	0.3649	0.3313	0.4337	0.3882	0.3468
MF	0.0277	0.0426	0.0881	0.0233	0.0480	0.0704
CFCVAE-Attn	0.4313	**0.3838**	0.3410	**0.4571**	**0.4150**	0.3506
CFCVAE	**0.4362**	0.3673	0.3261	0.4513	0.3974	0.3407

as condition in VAE proves more effective than creating distribution that samples noise with side information.

The effectiveness of attention layer Appending attention layer shows significant improvement of performance in both datasets. We demonstrate that using hidden vector as condition could be extensive amounts of information or unnecessary to some extent. By multiplying attention value with hidden vector, we expect it can be concatenated with input stably and efficiently.

The proposed method shows improved performance in different datasets comparing other existing methods which not apply time-variant characteristics of item features. When comparing absolute evaluation metrics between two datasets, values of Movielens 1-m dataset are much higher than those of Amazon dataset in that the number of items in Amazon datasets is 10 times larger than the number of items in

Table 4.7: AUC(Area Under ROC Curve) result for ML-1M dataset.

	AUC
CFVAE	0.8531
HPrior-VAE	**0.8538**
MF	0.7654
CFCVAE-Attn	0.8519
CFCVAE	0.8520

Movielens. Also, Amazon dataset is much sparser. However, notable improvement comparing with baseline is found in amazon women's clothing dataset rather than Movielens 1-m dataset. We pay attention to the possible reasons causing the results.

1. Visual item feature & Categorical score item feature Each dataset has different types of content features. Amazon women's clothing dataset has clothing visual features that is appropriate to apply neural network as many previous studies show. On the other hand, movie genome feature represents to content features in Movielens dataset whose type is categorical relevance score. The distinction between item features of datasets might lead to a different degree of improvement.

2. Scale and sparsity of datasest Besides, in case of Amazon women's clothing dataset, the sparsity of dataset is 0.022% which is much sparser than ML-1M dataset(2.4%). For sparser data, side information role can be more dominant. Also, the size of two datasets differs. Amazon women's clothing dataset is larger than ML-1M dataset. For large scale dataset, applying side information is more likely to aid to train pattern of various relationship among users and items. In real industrial world, we expect that various data may have a similar shape to the Amazon women's clothing data in the way that being sparse and large. Thus, we believe that the proposed method has sufficient potential in real world.

Chapter 5

Conclusion

5.1 Conclusion

In this thesis, we validate applying trend content feature to VAE is advantageous to predict user-item preference. This method can be applied in real-world in the way of using LSTM as content trend encoder and keeping the LSTM up to date. When recommending products to users, the encoded item features from trained LSTM based on user-history can be used as condition added to VAE.

5.2 Future Direction

We can extend our approaches to apply to other problems. We only use visual data and categorical relevance scores as content information. In addition to the above mentioned data types, other forms of data may be used such as text and voice. For example, user reviews can be adapted in the proposed method in the same manner of extracting time-variant characteristics contained in text reviews with LSTM.

There are several parameters that can be optimized. First, we use the specific period numbers for datasets we test several trials for period number. However, the number of time-based bins can be included in cost function to be optimized. Besides, we use 20 for sequence length for pretrain LSTM, but other sequence length can

replace it. Furthermore, while we use RNN structure in order to encode trend features within users, other neural network structure such as convolutional neural network can substitute RNN. Also, we have limitations and less performance comparing to denser data when we handling sparse data. Research needs to be done in the direction of obtaining high performance in even sparse data.

Bibliography

[1] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, *Fast and accurate deep network learning by exponential linear units (elus)*, arXiv preprint arXiv:1511.07289, (2015).

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

[3] R. He and J. McAuley, *Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering*, in proceedings of the 25th international conference on world wide web, International World Wide Web Conferences Steering Committee, 2016, pp. 507–517.

[4] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, *Neural collaborative filtering*, in Proceedings of the 26th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.

[5] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, *Session-based recommendations with recurrent neural networks*, arXiv preprint arXiv:1511.06939, (2015).

[6] Y. Jhamb, T. Ebesu, and Y. Fang, *Attentive contextual denoising autoencoder for recommendation*, in Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval, ACM, 2018, pp. 27–34.

[7] R. Jiang, S. Gowal, T. A. Mann, and D. J. Rezende, *Beyond greedy ranking: Slate optimization via list-cvae*, arXiv preprint arXiv:1803.01682, (2018).

[8] G. Karamanolakis, K. R. Cherian, A. R. Narayan, J. Yuan, D. Tang, and T. Jebara, *Item recommendation with variational autoencoders and heterogeneous priors*, in Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems, ACM, 2018, pp. 10–14.

[9] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, *Semi-supervised learning with deep generative models*, in Advances in neural information processing systems, 2014, pp. 3581–3589.

[10] Y. Koren, *Collaborative filtering with temporal dynamics*, in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 447–456.

[11] O. Kuchaiev and B. Ginsburg, *Training deep autoencoders for collaborative filtering*, arXiv preprint arXiv:1708.01715, (2017).

[12] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, *Variational autoencoders for collaborative filtering*, in Proceedings of the 2018 World Wide Web Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2018, pp. 689–698.

[13] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, *Image-based recommendations on styles and substitutes*, in Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2015, pp. 43–52.

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

[15] N. Sachdeva, G. Manco, E. Ritacco, and V. Pudi, *Sequential variational autoencoders for collaborative filtering*, in Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, ACM, 2019, pp. 600–608.

[16] J. Vig, S. Sen, and J. Riedl, *The tag genome: Encoding community knowledge to support novel interaction*, ACM Transactions on Interactive Intelligent Systems (TiiS), 2 (2012), p. 13.

[17] T. V. Vo and H. Soh, *Generation meets recommendation: proposing novel items for groups of users*, in Proceedings of the 12th ACM Conference on Recommender Systems, ACM, 2018, pp. 145–153.

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

[19] H. Wang, S. Xingjian, and D.-Y. Yeung, *Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks*, in Advances in Neural Information Processing Systems, 2016, pp. 415–423.

[20] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, and D. Zhang, *Irgan: A minimax game for unifying generative and discriminative information retrieval models*, in Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, ACM, 2017, pp. 515–524.

국문초록

사용자가 상품 선택을 결정할 때 영향을 미치는 요소 중에는 시기적 요소와는 무관한 사용자 고유의 취향과 시점에 따라 변화하는 상품의 유행과 같은 외부족 요소가 존재한다. 정밀한 추천시스템을 만들기 위해서는 이 두 요소를 모두 고려해야 하지만 기존의 추천 시스템은 추천을 해주는 그 당시의 시점을 고려하지 않고 사용자의 구매내역 데이터만을 기반으로 추천을 해준다는 문제점을 갖고 있다. 따라서, 본 논문에서는 추천이 진행되는 시점을 고려하여, 그 당시의 상품 유행 요소를 반영한 추천 시스템을 제안한다. 우리는 상품 내용 기반의 추천 시스템과 사용자 구매 내역 기반의 추천시스템을 효과적으로 결합할 수 있는 모델 생성에 초점을 두었다.

구체적으로, 사용자-상품 내포 피드백에 상품의 시간 기반 특징을 더하기 위해 조건부 변분 자동 생성기 모델을 활용한다. 이 때 입력 값과 잠재변수에 각각 더해주는 조건부로는 각 시점의 상품 특징을 나타내는 분산 표현을 사용한다. 각 시점에 해당하는 분산 표현은 LSTM을 사용하여 추출한다. LSTM에서 얻은 조건부와 사용자-상품 행렬을 연결시켜 변분 자동 생성기에 입력해줌으로써, 각 구간의 중요한 특징을 반영하는 하이브리드 추천시스템을 생성한다.

본 논문에서 제안하는 모델은 상품 고유의 변화하는 특성을 반영하여 추천에 활용한다는 점에서 기존 연구와 차이가 있다. 또한, 본 연구는 추천시스템 데이터의 섬김성 문제를 완화하는데에도 도움을 준다. 제안하는 모델의 평가를 위해 Movielens dataset(ml-1m) 데이터와 Amazon women's clothing 데이터를 사용하였고, 제안하는 방법에 알맞은 실험 과정을 설계하여 모델의 유효성을 확인했다.

주요어: 추천시스템, 조건부 변분 자동 생성기, Recurrent Neural Network

학번: 2017-24854

감사의 글

조성준 교수님을 비롯하여 2년동안 동거동락한 데이터 마이닝 연구실의 동기, 선후배 여러분들께 감사의 말씀을 전합니다. 돌이켜보면 힘든 날도 많았지만, 함께 연구하였기에 즐겁게 연구실 생활을 할 수 있었습니다. 더불어, 좋은 강의를 해주신 산업공학과 교수님들과 석사 졸업에 도움을 주셨던 과 사무실 선생님들께도 감사의 말씀을 드리고 싶습니다.

마지막으로, 힘들 때나 기쁠 때 저의 곁에 있어주셨던 가족 및 친구들과 졸업의 영광을 함께 누리고 싶습니다. 감사합니다.