ADL Final Project Report

Abstract

The objective of our research is to predict users' interested topics and courses. We frist adopt methods that TA suggested such as ALS, BM25 to predict users' interested topics and courses. After that, We use bayesin personalized ranking(BPR) and k nearest neighbors(KNN) to determine courses and topics for recommendation in seen domain. Meanwhile, we tBertForContentSelection for topic prediction, innovational method query probability for course prediction in unseen user task.We observed that our new method didn't make massive improvement in performence.We concluded that the traditional IR method still get better preformence. However, we believed that our innovaiton will give new possibilities in information retrivel and NLP problem.

Introduction

Recommender systems (RS) are seeing significant in academic, economic and industry interest. It can be considered an important part of the e-commerce ecosystem, and reduce a large amount of data to a manageable amount and recommends it to the user according to his interests, desires, and **choices.It** (http://choices.It) is difficult to choose from thousands of options without the help or advice of someone who has prior knowledge of the product.Meanwhile, deep learning in RS is still poorly understood .We aim to build recommender systems base on not only newest deep learning method, but also the new possibility of old machine learning method.
In the seen users' part, we first tried out ALS, BPR and KNN. On top of that, since the order of both the recommended courses and topics matters, we tried to combine the different approaches and do trials of the rearrangement. The details of the rearrangement will be

disccussed in the Approach section. Aside from the rearrangement among CF and KNN, we adopted the NLP-based method. Details will be discussed in the next section. Regarding the Recommendation of topics, we intuitively recommend the topic based on the topics of the courses recommended. And if a topic has multiple occurance, it will be placed in priority. Somehow, We come up with another idea for topic recommendation. Why not we construct the customer-topic matrix? Nevertheless, we realize the idea and results will be presented in the experiment section.
In the unseen users' part, we try Bert for topic selection, BM25 and some innovative method for course recommending separately.Our goal is to find new method in old recommending task.Details will be discussed in the next section.

Related Work

seen

To recommend the course to seen users, we take advantage of their shopping records. The first step of the basic CF is to use this information to construct the customer-course matrix. Each cell a_{ij} represents wheter the $customer_i$ bought $course_j$ or not and follow the rules below.

$$a_{ij} = \begin{cases} 1 & \text{if the customer bought it} \\ 0 & \text{if the customer do not bought it} \end{cases}$$

For the next step, factorize the customer-course matrix into two low-dimension matrices. The first one(U) represents the user's preference and the last(V) represents the features of the courses. It's kind of the latent representation. What we want to do is to project a single customer's preference into k-dimensional vector space and so does the course' features. Therefore, we can somehow score the level of match between the customers and courses based on the inner product of the preference and features. The mathematical interpretation:

$$A_{n*m} = U_{n*k} * V'_{k*m}, a_{ij} = u'_i * v_j$$

where u_i and v_j are the row of the U and V. Because the reduction of dimension($k \ll m, n$), the complexity shrinks form $O(m * n)$ to $O((m + n) * k)$. In the traditional implicit feedback literatre, the records are limited and for those courses unbought we ensure that it has the base score and satisfy the formula: $b_{ij} = 1 + \alpha * a_{ij}$. Howerver, in our case, there is no difference between b_{ij} and a_{ij} so there is no need to tune the hyperparameter α. The final discussion of collaborative filtering is that how to coustuct U, V such that the product of U, V similar to A. We discuss the algorithm In this order: ALS, BPR, KNN.
In ALS, as the name states, we target to minimize the squared loss:

$$\min_{U,V} \sum_i \sum_j (a_{ij} - u'_i v_j)^2 + \lambda(\sum_i ||u_i||^2 + \sum_j ||v_j||^2)$$

giventh the L2 regularization parameter λ.
The iterative algorithm to update paramers:

- initialize U^0, V^0
- update V, U adatively:

$$V_{n+1} = \min_V \sum_i (a_{ij} - u_i^{n'} v_j)^2 + \lambda(\sum_i ||u_i^n||^2 + \sum_j ||v_j||^2)$$
$$U_{n+1} = \min_U \sum_j (a_{ij} - u'_i v_j^{n+1})^2 + \lambda(\sum_i ||u_i||^2 + \sum_j ||v_j^{n+1}||^2)$$

- repeat until converge
  As for BPR, it use the bayesian approach with
- prior probability $P(\Theta) = P(U, V) \sim N(0, \lambda_\Theta I)$

- likelihood:

$$\prod_{(u,m,n)\in D_S} ln\sigma(x_{\hat{u_i mn}}) - \lambda_\Theta ||\Theta||^2$$

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

$$x_{\hat{u_i mn}} = u_i' * v_m - u_i' * v_n$$

D_S is the data set of paris of observed and un-observed records for each u_i and the goal is to maximize the posterior probabiltiy which is the product of the prior and liklihood given the regularization parameter λ_Θ.
The iterative algorithm to update the paramers:

- initialize U^0, V^0
- draw (u_i, v_m, v_n) from D_S
- $u_i \leftarrow u_i + \alpha(\frac{e^{-x_{\hat{u_i mn}}}}{1+e^{-x_{\hat{u_i mn}}}} * (v_m - v_n) + \lambda_{u_i} u_i)$
- $v_m \leftarrow v_m + \alpha(\frac{e^{-x_{\hat{u_i mn}}}}{1+e^{-x_{\hat{u_i mn}}}} * u_i + \lambda_{v_m} v_m)$
- $v_n \leftarrow v_n + \alpha(\frac{e^{-x_{\hat{u_i mn}}}}{1+e^{-x_{\hat{u_i mn}}}} * -u_i + \lambda_{v_n} v_n)$
- repeat until converge
  At last, since we adopt the brute-force algorithm of KNN so the algorithm is simply calculating the Euclidean distance bewteen the users and the represented user vector is their own records of consumption. Notice that there will be some duplicated coureses among all the neighbors and also use the number of the occurance as weight to rank the output of the model. After try out the three above-mentioned methods we further do the experiments of combination of three based methods and NLP-based method to rearrange the results.

unseen

COURSE

for the two method we use:BM25 and innavation method
query probability
for BM25:
first compute weight of term i

$$IDF(q_t) = [log\frac{N - df_t + 0.5}{df_t + 0.5}]$$

df_t = document fruquency for term t
N = number of all document
Then we compute the relation score of term t and document d, BM25 believe that the relation between term frequency and document are not linear, which mean there are limitated relation between any term and documet. Therefore BM25 design score as below

$$S(q_t, d) = \frac{(k_1 + 1)tf_{td}}{K + tf_{td}}$$
$$K = k_1((1 - b) + b(\frac{L_d}{L_{ave}}))$$

L_d = length of document d
L_ave = average length of all document
k_1 = positive parameter to standardize range of term frequency in document
b = parameter in range 0 < b <1, to determine the weight of document length
Lastly, BM25 compute the weight between term i and query q

$$S(q_t, Q) = \frac{(k_3 + 1)tf_{td}}{k_3 + tf_{td}}$$

k_3 = positive parameter to adjust range of term frequency in query

The final score function is as below

$$score(q,d)=\sum_{t\in q}[log\frac{N}{df_t}]\frac{(k+1)tf_{td}}{k_1((1-b)+b(\frac{L_d}{L_{ave}}))+tf_{td}}\frac{(k_3+1)tf_{td}}{k_3+tf_{td}}$$

for Quert probability
First we assume that

$$P(d|q)=\prod_{t\in q}P(d|t)$$

However the method will give the combination that never seen in train data that probility=0, thus we need to smooth the probality.
First we try add one smoothing, add one in both numerator and denominator.
Second we try linear smoothing, let $\lambda=0.5$, then the smoothed probability $P(d|t)=\lambda P(d|t)+(1-\lambda)P(t|Q)$
$P(t|Q)$ = probability of term t given query Q in training data.

Approach

Seen

To recommend the course, besides from the 3 base methods we tried out different rearrangement approaches and the spirit is that we tried to use different approach's result to rearrange the base result. Take KNN_ALS for example, we first got the KNN recommendation and also the result of ALS and then we rearrange the KNN recommendation if the coures recommended by ALS as well. The order is determined by the coures's index in KNN and ALS result. The idea of KNN_ALS_BPR is similar. In the following interpretation, let's denote **ALS** as the ALS recommendation, **BPR** as the BPR recommendation and **KNN** as the KNN recommendation. Besides, we consider a single customer's recommendation rearrangement in the below algorithm.
The algorithm of 2 mixture(KNN_ALS):

- for each customer, get the **KNN**$_i$ and **ALS**$_i$ respectively
- create empty dictionary W and loop through **KNN**$_i$ as **KNN**$_{ij}$
- W[**KNN**$_{ij}$] = j
- if **KNN**$_{ij}$ in **ALS**$_i$, W[**KNN**$_{ij}$] += (index of **KNN**$_{ij}$ in **ALS**$_i$)
- else, W[**KNN**$_{ij}$] += length of **ALS**$_i$

The algorithm of 3 mixture(KNN_ALS_BPR):

- for each customer, get the **KNN**$_i$, **ALS**$_i$ and **BPR**$_i$ respectively
- create empty dictionary W and loop through **KNN**$_i$ as **KNN**$_{ij}$
- W[**KNN**$_{ij}$] = j
- if **KNN**$_{ij}$ in **ALS**$_i$ and also **BPR**$_i$, W[**KNN**$_{ij}$] += (index of **KNN**$_{ij}$ in **ALS**$_i$ + index of **KNN**$_{ij}$ in **BPR**$_i$)
- else if **KNN**$_{ij}$ only in **ALS**$_i$, W[**KNN**$_{ij}$] += (index of **KNN**$_{ij}$ in **ALS**$_i$ + length of **BPR**$_i$)
- else if **KNN**$_{ij}$ only in **BPR**$_i$, W[**KNN**$_{ij}$] += (index of **KNN**$_{ij}$ in **BPR**$_i$ + length of **ALS**$_i$)
- else, W[**KNN**$_{ij}$] += (length of **BPR**$_i$ + length of **ALS**$_i$)

In addition to the CF and KNN rearrangement, we also consider the NLP-based rearrangement though the performance is terrible. We only apply this rearrangement scheme on ALS and the main idea is to rearrange according to the similarity calculated between customers' recreation and courese' introduction. The similarity criterion is the cosine similarity. The detail is that for each customer, we first apply the NLP tools, the word segmenter and and POS tagger, which provided by CKIP to extract all the noun from the recommended coures' introduction. Then use the pretrained model, `distiluse-base-multilingual-cased-v1`, to create the customers' category-wise recreations' embeddings and the set of embeddings of courese' introduction. Finally compute the similarity between each customers' recreations with every coureses recommended by the ALS.

Unseen

Topic

We regard this problem as a multiple choice problem, given the user's information, we need to choose a topic that he's most interested in.
So we use context selection to solve this problem. We concatenate the user's interest, job, gender, hobby into a string, and the list of topics as choices, and tell the model to choose one that the user would most likely to be interested in.
For ranking, we look at the logit of model output, and rank the topics with their logit value.

Course

We recommending course base on the relationship between user infomation and course introdution.
So we use BM25 to compute relation score between user and course, and use query probability to compute the probablity that user buying course under the user infomation. We concatenate the user's interest, job, gender, hobby into a string,and course introdution as document.For ranking, compute the course likely score buy above method, and rank top 50 courses that the user would most likely to be interested in.

Experiments

Seen

Course

- base

algorithm	validation	test
ALS	0.07454	0.04598
BPR	0.06386	0.03728
KNN	0.05734	0.03687

- rearrangement

algorithm	validation	test
KNN_ALS	0.07157	-
KNN_BPR	0.06085	-
ALS_KNN	0.07174	0.0458
ALS_BPR	0.07106	0.04295
BPR_ALS	0.06642	-
BPR_KNN	0.06404	-
KNN_ALS_BPR	0.07042	0.04055
ALS_KNN_BPR	0.06954	-
ALS_sim	0.04	-

Topic

- based on courses recommended

algorithm	validation	test
ALS	0.24260	0.26312
BPR	0.20731	0.20552
KNN	0.22904	0.22384
ALS_KNN	0.23840	0.25734
KNN_ALS_BPR	0.22197	0.22300

- based on customer-topic matrix

algorithm	validation	test
ALS	0.21194	0.24706
BPR	0.20650	-

Unseen

Topic

We tried two types of preprocess method

- Only use rank 1 topic as label
- Use all 4 ranks as label

We also tried two pretrained models:

- `bert-base-chinese`
- `hfl/chinese-roberta-wwm-ext`

The results are:

Preprocess \ Model	`hfl/chinese-roberta-wwm-ext`	`bert-base-chinese`
Rank 1	0.18031	0.16947
All Rank	0.13612	0.03291

Course

The results are:

algorithm	validation	test
QP(smooth: add 1)	0.0090	0.0071
QP(smooth: linear)	0.0073	0.0061
bm25	0.0457	0.0519

Discussion

Seen

We conclude the reason ALS outperforms BPR is that the assumption in BPR might fail in our case. BPR has posed strong assumption: customers perfer the observed than all the other non-observed and we think it might be violated easily and thus in our case, ALS perform better. Except for the ALS rearrangent, all the others are outperform the base methods which menas the order of the ALS should be relatively optimized. Besides, the NLP-based rearrangement is rather unsatisfactory and the problem might result from the poor embeddings and too much noise during similarity comparison.

Unseen

Topic

We think the reason why taking all the rank as label is bad for performance is that the four of them might cancel out each other’s gradient direction, which made the results worse.

Course

The reason why query probability have worse performance then BM25, is that the assumption that $P(d|q) = \prod_{t \in q} P(d|t)$ may not hold, or we need better smoothing method to balance the probability of combinaiton we never seen .

Conclusion

In this project, we aim to find innovational method apply on information retrivel.We tried bert for content selection, homemade method query probability and the mixture of multiple informaiton retrivel method.Althought didn't make massive improvement on performence, we believe that our attempt still give new possibilities in information retrivel and NLP problem.

Work Distribution

SEEN : 林子翔 葉秀軒

UNSEEN TOPIC : 陳旻浚

UNSEEN COURSE : 歐崇愷