ATTENTION-BASED COLLABORATIVE FILTERING

A. I. Romanov

Saratov State University, Russia
E-mail: romanow.lesha2013@yandex.ru

Attention mechanism invention was an important milestone in the development of the Natural Language Processing domain. It found many applications in different fields, like churn prediction, computer vision, speech recognition and so on. Many state-of-the-art models are based on attention mechanisms, especially in NLP. As this technique is very powerful, we decided to investigate its application in solving a collaborative filtering task. In this paper we propose a standard framework for developing a recommender system engine based on well known transformer architecture. We couldn't reproduce current state-of-the-art results on movielens datasets, but in our implementation attention based model achieves competitive scores on movielens 1M and movielens 10M datasets.

Introduction

Due to exponential growth of the web information, WEB applications face new challenges in providing the best user experience service. Nowadays recommender systems are getting to be core features of many WEB applications. It's widely popular in content streaming platforms, e-commerce web sites and even in financial services. There are three main types of recommenders systems:

- Collaborative filtering
- Content based
- Hybrid methods

Each of the methods has its own advantages and disadvantages. However, because it's relative simplicity of implementation, currently the most popular technique is collaborative filtering. Collaborative filtering assumes that the model learns users preferences based on it's previous historical interactions. Common output of collaborative filtering models is users embedding matrix U and items embeddings matrix V. Where embedding means: vector representation of the object. This framework is influenced by a cold start issue: more than half users and items have a very few interaction history and it leads to noisy predictions for them. To address this problem, usually additional users and items features may be utilized as a useful signal, which will improve prediction accuracy for cold objects. This kind of architecture is called a hybrid recommender system. In this paper we will not focus on content based and hybrid methods.

1. Problem statement

A collaborative filtering method can be represented as a matrix factorization problem. Given a log of users and items interactions history. Each interaction is represented by triplets: (u_i, i_j, r), where r - it's rating, which was given by user u_i to item i_j This log can be represented by interaction matrix R. Where each row is associated with the user, and each column is associated with the item. Each matrix cell will be a rating r. In most cases this matrix has around 98% of sparsity rate, which

means that most of the matrix elements will be empty. We have only partial information about the cells of this matrix, based on explicit or observed customer behavior. Explicit behavior may be a product rating given by a customer. Observed behavior tries to deduce how much a customer likes the product by implicit signals, for example, when a customer views a product, adds it to their cart, or purchases it. Our goal is to build a model that can predict the values of the empty cells of this interaction matrix. We try to approximate the interaction matrix as a product of two matrices of lower dimensions, user factors and item factors: $R = U \times V$. The scalar product of a row of matrix U and a column of matrix V gives a predicted item rating for the missing cells. Predictions for known items should be as close to the ground truth as possible. We fit those two matrices with known data using optimization algorithms.

2. Related work

Attention mechanisms have become an integral part of compelling sequence modeling and transduction models in various tasks, allowing modeling of dependencies without regard to their distance in the input or output sequences. Innovation of the transformer model was based on the assumption that it's not necessary to use recurrent neural networks in conjunction with attention to achieve state-of-the-art performance. New model architecture eschewing recurrence and insteadrelying entirely on an attention mechanism to draw global dependencies between input and output.

The Transformer allows for significantly more parallelization and can reach a new state of the art performance on a wide range of NLP tasks using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder.

Encoder: The encoder is composed of a stack of $N = 6$ identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position wise fully connected feed-forward network.

Decoder: The decoder is also composed of a stack of $N = 6$ identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack.

$$Attention(Q, K, V) = soft\max(\frac{QK^T}{\sqrt{d_k}})V$$

Where Q - is a matrix of queries, K - matrix of keys and V is a matrix of values. Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. With a single attention head, veraging inhibits this.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O$$

Where

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

and the projections are parameter matrices:

$$W_i^Q \in \mathrm{R}^{\mathrm{d}_{\mathrm{model}} \times d_k}, W_i^K \in \mathrm{R}^{\mathrm{d}_{\mathrm{model}} \times d_k}, W_i^V \in \mathrm{R}^{\mathrm{d}_{\mathrm{model}} \times d_k},$$

The Transformer, the first sequence transduction model based entirely on attention, replacing the recurrent layers most commonly used in encoder-decoder architectures with multi-headed self-attention. [1]

3. Proposed method

Interactions matrix can be represented as a bipartite graph (Figure 1):

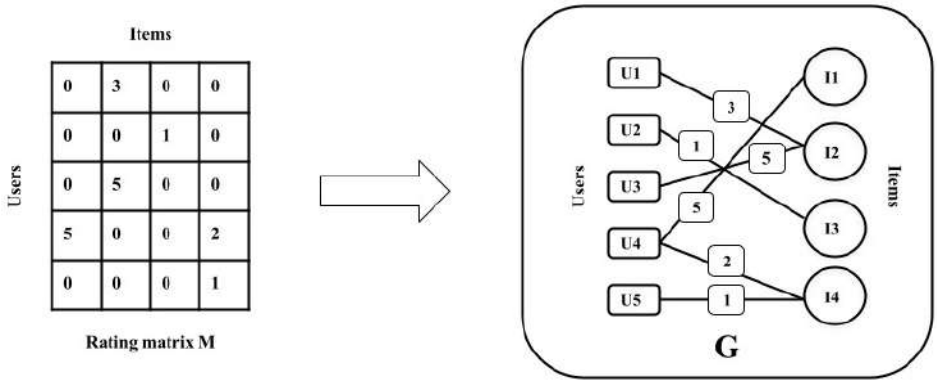

Fig. 2. User-Item interactions matrix representation as a bipartite graph

Graph Neural Networks aim to generalize neural networks to nonEuclidean domains such as graphs and manifolds. GNNs iteratively build representations of graphs through recursive neighborhood aggregation (or message passing), where each graph node gathers features from its neighbors to represent local graph structure. Transformers can be regarded as GNNs which use self-attention for neighborhood aggregation on fully-connected word graphs. [2]

We represent a user through the items, this user interacted with, analogically we represent the item - through the users, interacted with this item. Basically we represent the graph node through its neighbours.

Lets model interaction between User 4 (U_4) and Item 4 (I_4) . First we have to get U_4 representation. U_4 has 2 neighbors I_1 and I_4 (Figure 2),

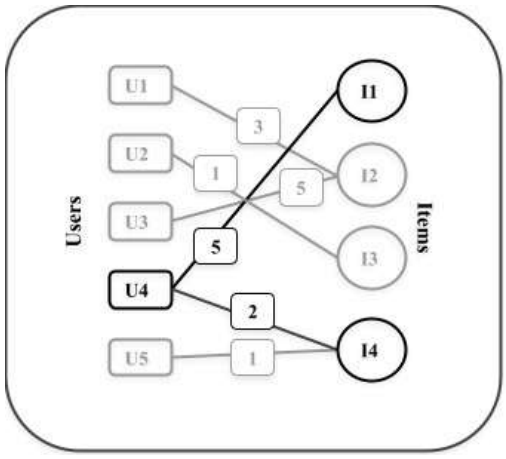

Fig. 3. User 4 graph representation

Inspired by NLP where each transformer input is a fully connected graph, which is one sentence, we will define U_4 representation as a sequence: $U_4 = \{I_1, I_4\}$. We don't include U_4 to the sequence intentionally, because users and items have a

different modality and embeddings should be optimized separately. These experiments we will leave for further work.

Now let's defne representation of the I_4. This item has interactions with U_4 and U_5 (Figure. 3):

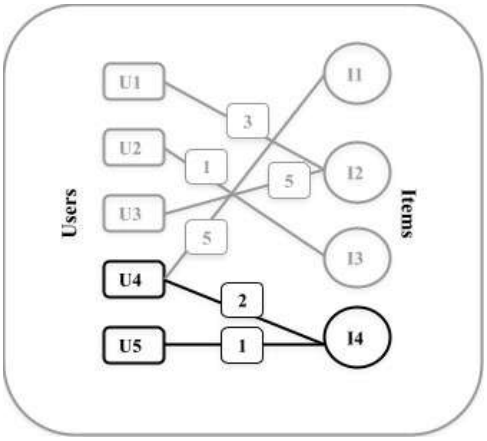

Fig. 4. Item 4 graph representation

So we got the representation of I_4. as a sequence $I_4=\{U_4,U_5\}$. Now the objective is to approximate the function $f(U_i,I_j)=r_{ij}$, for $i=4$ and $j=4$ it will be equivalent to $f(\{I_1,I_4\},\{U_4,U_5\})=2$.

We define a transformer input sample as a pair of two sequences: sequence of the item (all neighbor users of this item) and sequence of the user (all neighbor items of this user). The target will be to predict the rating 2. For better convergence of the regression model we will normalize the target to the scale of using [0,1] min-max transformation.

4. Experiments

We compared our method with the following baseline methods:

- **Alternative least squares.** Iterative method of matrix factorization. We used *implicit.als.AlternatingLeastSquares* framework implementation of the algorithm. We tuned model hyperparameters using a random search method.
- **LightFM.** Another implementation of matrix factorization. In contrast to the ALS algorithm, LightFM uses different optimization techniques, based on "*Adagrad"* or "*Adadelta"* optimizers. It's not an iterative method. The framework includes different versions of loss functions: "*bpr*", "*warp*", "*warp-kos*".
- **Matrix Factorization based on SGD.** We implemented vanilla matrix factorization. This implementation uses stochastic gradient descent as an optimization algorithm.
- **Neural collaborative filtering.** As a neural network based approach we used well known neural collaborative filtering with *L2* regularization and "*Adam*" optimizer.

For evaluation, we used Root Mean Squared Error (RMSE) on global random 90:10 split, which can be computed as follows:

$$D(\bar{r}) = \sqrt{\sum_i \sum_j I_{ij}(r_{ij} - \bar{r}_{ij})^2 / \sum_i \sum_j I_{ij}}$$

I_{ij} - indicates that entry (i, j) appears in the test set.

Summarized experiments results are presented in the table:

Algorithms comparison (RMSE)

Method	Movielens 100K	Movielens 1M	Movielens 10M
ALS	1.72	1.36	1.45
LightFM	3.23	2.82	3.07
Vanila MF	0.95	0.92	1.07
Neural CF	0.878	0.91	0.88
Transformer	1.04	0.91	0.83

Despite the proposed methods didn't outperforming current state-of-the-art methods, it still shows competitive scores on medium size dataset and best performance on the huge dataset compared to other reproduced methods. We will continue experiments with transformer based collaborative filtering. Because the training process is computationally expensive, we are limited in the hyper parameter tuning. For the further experiments we will customize the transformer model to achieve best performance on movielens datasets. An additional direction of research is the explainability power of self attention.

5. Conclusion

In this paper, we proposed an approach of training a state-of-the-art NLP technique to address a collaborative filtering task. We've shown how to achieve competitive results on basic RecSys movielens dataset. We believe that in any machine learning task it can be beneficial to look for ideas and inspiration in different machine learning domains, like in our case in natural language processing. We have to note that reported results are suboptimal, because as it pointed out in the paper [3], right models evaluation requires significant effort on hyperparameters tuning and experiments setup.

REFERENCES

1. *Vaswani A., Shazeer N., Parmar N.,* Attention Is All You Need: Conference on Neural // Information Processing Systems. 2017. P. 4-5.

2. *Velickovic P., Cucurull G., Casanova A.,* Graph attention networks // International Conference on Learning Representations. 2018. P. 2-3.

3. *Rendle S., Zhang L., Koren Y.* On the Difficulty of Evaluating Baselines // arXiv preprint arXiv: 2019.