E-Commerce Item Recommendation Based on Field-aware Factorization Machine

Peng Yan
NetEase Youdao
Beijing, China
yanpeng@rd.netease.com

Xiaocong Zhou
Tsinghua University
Beijing, China
infinitezxc@gmail.com

Yitao Duan
NetEase Youdao
Beijing, China
duan@rd.netease.com

ABSTRACT

The RecSys 2015 contest [1] seeks the best solution to a top-N e-commerce item recommendation problem. This paper describes the team Random Walker's approach to this challenge, which won the 3rd place in the contest. Our solution consists of the following components. Firstly, we cast the top-N recommendation task into a binary classification problem and extract original features from the raw data. Secondly, we learn derived features using field-aware factorization machines (FFM) and gradient boosting decision tree (GBDT). Lastly, we train 2 FFM models with different feature sets and combine them by a non-linear weighted blending. This solution is the result of numerous tests and the scheme turns out to be effective. Our final solution achieved a score of 61075.2, ranking in the third place on the public leaderboard.

Categories and Subject Descriptors

H.2.8 [**DatabaseManagement**]: Database Applications - Data Mining

General Terms

Experimentation

Keywords

Field-aware Factorization Machine, Gradient Boosting Decision Tree, Ensemble, top-N recommendation

1. INTRODUCTION

The rapid growth of e-commerce requires the development of the right tools to facilitate various aspect of online transactions. Recommender systems that recommend items to users based on their past behavior can help users find what they need more efficiently and generate more revenues for the e-commerce vendors. RecSys Challenge 2015 [1] provides an opportunity for researchers and practitioners in the field of recommender systems to test their techniques in real-world data. This year's task is to predict whether a customer will purchase some item or not and, if he is buying, which items he will buy.

RecSys '15 Challenge, September 16-20, 2015, Vienna, Austria

DOI: http://dx.doi.org/10.1145/2813448.2813511.

The training data provided by RecSys 2015 challenge [1] comprises a sequence of clicks and buy events performed by some user during a typical session on an e-commerce website. Test data contains click events only. Each click event contains session id (user), click time, item id and item's category while each buy event contains session id, buy time, item id, item price and quantity of the item bought. Note that there is no other explicit identification for each user so we use session ids as user ids, i.e., we treat each session as belonging to a distinct user. There are a total of 33M clicks, 1.1M buys, 9M sessions and 53K items in the training data. The time span of these events is 6 months.

The contest takes the following score as the evaluation metric:

$$Score(Sl) = \sum_{s \in Sl} \begin{cases} \frac{|S_b|}{|S|} + \frac{|A_s \cap B_s|}{|A_s \cup B_s|} & \text{if } s \in S_b, \\ -\frac{|S_b|}{|S|} & \text{else.} \end{cases}$$

where the symbols are:

- Sl - sessions in submitted solution file.
- S - all sessions in the test set.
- s - session in the test set.
- S_b -sessions in test set which end with buy.
- A_s - predicted bought items in session s.
- B_s - actual bought items in session s.

The evaluation takes into consideration the ability to predict both whether the user will buy something and which items will be bought. Such information is of high value to an e-commerce vendor as it can indicate not only what items to suggest to the user but also how to encourage the user to become a buyer.

This paper describes the team Random Walker's approach to this challenge, which won the 3rd place in the contest. It is organized as follows. Section 2 provides an overview of our approach. Section 3 describes our feature learning method. Section 4 introduces an emerging classification model FFM and our improvement. Section 5 discusses model ensemble. Finally, we conclude in Section 6.

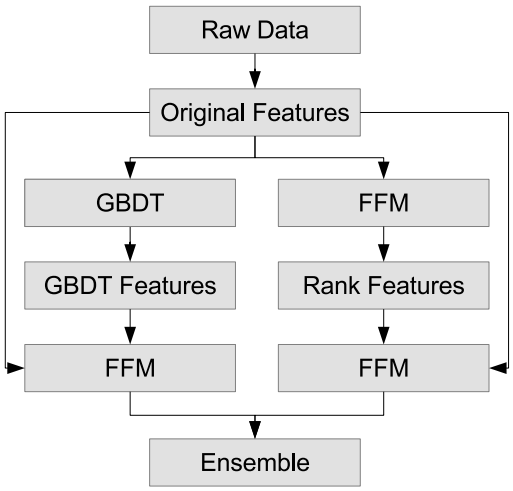

Figure 1: Overview of our approach.

2. OVERVIEW

The task of predicting whether and what a user will buy can be cast as a binary classification problem with a classifier that outputs a purchase probability. Each session-item pair provides a training instance and the classification models are trained to predict the probability for buying each item a user clicks. For each user, we rank the items by the probabilities that he will buy them. The top-N recommendations for a user is then generated via a threshold which is tuned using the evaluation score in cross validation. The same threshold is then used in the prediction on test data and generate the submission result.

Figure 1 shows the overall structure of our approach which uses a concatenation of boosted decision trees and a couple of probabilistic sparse classifiers. Our solution consists of five components: feature extraction, feature learning via GBDT and FFM, individual model training, and model ensemble.

3. FEATURE EXTRACTION

Before introducing the models that we use, we first describe our feature extraction process. Our features are comprised of original features extracted directly from raw data and secondary features learned from the predictions of two classification models trained on the original features.

3.1 Original Features

Our task is to predict the probability of a user buying an item in the specified time. So we extract original features from three aspects including user, item and time.

3.1.1 Item features

Different item sales are very different. Item features describes the properties of the item itself. We select item id, category id, item price as features.

3.1.2 User features

User features describe a user's degree of preferences for online shopping in general and a given item in particular. If a person is willing to buy something on an e-commerce site he is likely to spend more time on it. So the total number of clicks and the entire time span of a session are used as indications of a user's general altitude towards online shopping. We also extract some features about the user's preference for a given item from the click sequence:

- Browsing time: time interval between the click of this item and the next one.
- Neighboring item: the previous and next item of current item.
- Click number: the number of current item's clicks within this session.
- Binary sequence: we convert the item sequence to binary sequence as follows. If an item is the exact one to be predicted we assign the value 1, otherwise we assign the value 0. For example, given a click sequence"a-b-c-d-c" (here each letter stands for an item) and the target item is c. We will get a binary click sequence of "00101" after binarization.
- Position: the position of this item in the whole click sequence.
- Items clicked: the total number of items that the user clicks.
- Categories clicked: the total number of categories whose items the user clicks.
- Clicks in category: the aggregated number of clicks in the category that the current item belongs to.
- Items in category: the number of items in the category that the current item belongs to.

3.1.3 Time features

We observe that the purchase probability changes a lot over time. To exploit this information, we extract month, day of week and hour as features from the time when user click the item. All these time features are used as categorical features directly.

3.1.4 Summary

To summarize, we list all the original features that we use in table 1. Please see previous text for their detailed descriptions and definitions.

3.2 Learned Features

In order to further improve prediction accuracy, we introduce secondary features learned from the original data. In particular, we train a Gradient Boosting Decision Tree (GBDT) and a Field-aware Factorization Machine (FFM) (to be explained later in Section 4) on the original features. Their prediction results provide higher level of abstraction for the information contained in the data and are used as additional features in subsequent classification models.

3.2.1 GBDT Features

Gradient boosting decision tree [3] is a powerful model for both regression and classification problems. The model consists of a number of individual decision trees and each tree is trained by the residual of previous trees. In our solution,

Table 1: Features used in our solution

Item features	Item id
	Category id
	Item price
User features	Total number of clicks
	Time span of a session
	Browsing time
	Neighboring items
	Click number
	Binary sequence
	Position
	Items clicked
	Categories clicked
	Clicks in category
	Items in category
Time features	Month
	Day of week
	Hour

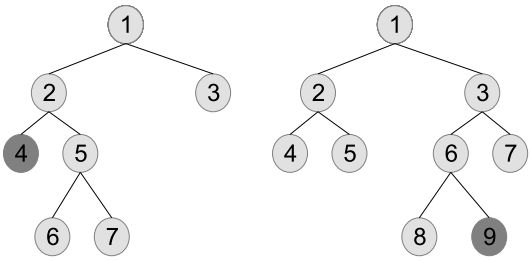

Figure 2: GBDT features example.

we train a GBDT model with log likelihood on training data as the loss function. An instance will fall on a leaf node of each tree.

Since boosting can be interpreted as an optimization process [2], the positions of an instance on each of the trees represents its relation to the cost function at various stages of the gradient descent process. This can be a very useful source of information. We use it in the following way: The ids of the trees are treated as *fields* (to be explained later in Section 4) and the indexes of the leaf nodes that an instance falls onto are treated as feature values. This method is similar to [4] but since we are using FFM, our method introduces the distinction between features and fields. Figure 2 shows an example of the GBDT feature extraction result. In this simplistic example, GBDT consists of two trees with depth three. If a sample falls on the fourth leaf of the first tree and the ninth leaf of the second tree, then we get 2 new features: tree1:4 tree2:9 where tree1 and tree2 are treated as fields.

3.2.2 Rank Features

If a user clicks more than one items, the probabilities of buying those items are not independent of each other. We have considered this information in the original features (Neighboring item), but that is not enough. To further exploit this information, we train an FFM model on the original features and get the predictions of each user's items. Then we sort the probabilities for each user and use its rank as a feature of the current item.

Table 2: Performance of different feature sets

Features	Evaluation score
Original features	58956
Original + GBDT features	60492
Original + RANK features	60333

In our empirical study, both types of learned features turned out to be very important and contributed significantly to improving the results. Table 2 shows the performance of different feature sets. For all these tests, the prediction model is FFM.

4. FIELD-AWARE FM

Factorization models have shown superior performance in estimating interactions between categorical variables of large domain. Factorization machines (FM) [7, 8] are a generic and effective approach for using factorization models to estimate variable interactions in machine learning tasks under very high sparsity. They can mimic most factorization models just by feature engineering and have been applied in many important applications such as recommender systems, click-through prediction, etc., where such interactions are important.

For a binary classification problem, let n be the dimensionality of feature space, $x \in \mathbb{R}^n$ the feature vector and $y \in \{0, 1\}$ the target, the FM model of order 2 is defined as

$$y(x) = \sigma(w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < v_i, v_j > x_i x_j)$$

where

$$w_i \in \mathbb{R}, v_i \in \mathbb{R}^k$$

are the model parameters and k is a hyperparameter that defines the dimensionality of the factorization. $< v_i, v_j >$ denotes the dot product of two vectors. σ is the logistic function $\sigma(z) = 1/(1 + exp(-z))$.

Field-aware factorization machine [5, 6] is an improved version of FM. With FFM, features are organized into *fields*. A field can be viewed as corresponding to a class of features. FFM learns a *different* set of latent factors for every pair of fields, i.e., each feature uses a different k-vector to interact with other features from different field. In addition, most FFM applications ignore the global bias w_0 and the linear terms. Formally, the model is defined as

$$y(x) = \sigma(\sum_{i=1}^{n} \sum_{j=i+1}^{n} < v_{i,f(j)}, v_{j,f(i)} > x_i x_j)$$

where $f(i)$ is the field feature i belongs to.

Compared with FM, FFM introduces more structured control and allows for more leverage in tweaking the interaction between features. The model has been used to win two recent click-through rate prediction competitions (Criteo's and Avazu's).

Table 3: Performance of two FFM models

Model	Evaluation score
Standard FFM	58782
FFM with first order features	58956

Table 4: The performance of model ensemble

Model	Evaluation score
FFM Model 1	60492
FFM Model 2	60333
Ensemble	61075

In our solution, we treat each feature, such as month, item id and each tree in GBDT features, as a field. In addition, our empirical results show that the linear terms (except for the global bias) also have some effect on the prediction precision. Thus in our solution, they are added back into the FFM model and our model is

$$y(x) = \sigma(\sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < v_{i,f(j)}, v_{j,f(i)} > x_i x_j)$$

All our FFM models use standard L2 regularization.

Table 3 shows the performance of two FFM models, standard FFM and FFM with the linear terms. Both models are trained on the original feature set. Our FFM with first order features performs slightly better. Therefore all FFM models used in our solution are FFMs with first order features.

5. MODEL ENSEMBLE

We blend our individual models by a simple non-linear weighted ensemble method. Let $p_1, p_2 \in [0, 1]$ be the predictions of two models, we select two weights $w_1, w_2 \in [0, 1]$ such $w_1 + w_2 = 1$. The final prediction is

$$p = \sigma(w_1 logit(p_1) + w_2 logit(p_2))$$

where

$$logit(p) = \log(\frac{p}{1-p})$$

is the inverse of the logistic function.

Table 4 shows our evaluation score on the public leaderboard. It is clear that the score improved after combining the two individual FFM models.

6. CONCLUSION

In this paper, we summarize our solution to the RecSys 2015 e-commerce recommendation challenge [1]. Our experiences show that feature transformation, FFM and ensemble are among effective techniques that won us, the Random Walker team, the third place on the public leaderboard of the contest. We will verify the validity of this method in other areas in the future.

7. ACKNOWLEDGMENTS

We thank the organizers of RecSys Challenge 2015 and YOOCHOOSE for providing the opportunity and resources for us to test our techniques and ideas. We learned a lot from the experience. We also thank Yu-Chin Juan, Wei-Sheng Chin, and Yong Zhuang from National Taiwan University for sharing their research in the optimization method of FFM model.

8. REFERENCES

[1] D. Ben-Shimon, A. Tsikinovsky, M. Friedman, B. Shapira, L. Rokach, and J. Hoerle. Recsys challenge 2015 and the yoochoose dataset. In *Proceedings of the 9th ACM conference on Recommender Systems*, RecSys '15. ACM, 2015.

[2] L. Breiman. Arcing the edge. Technical Report 486, Statistics Department, University of California, Berkeley, 1997.

[3] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[4] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, and J. Q. Candela. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, ADKDD'14, pages 5:1–5:9, New York, NY, USA, 2014. ACM.

[5] M. Jahrer, A. Töscher, J.-Y. Lee, J. Deng, H. Zhang, and J. Spoelstra. Ensemble of collaborative filtering and feature engineered models for click through rate prediction. In *18th ACM Int. Conference on Knowledge Discovery and Data Mining (KDD12), KDD Cup Workshop*, 2012.

[6] Y.-C. Juan. Libffm: A library for field-aware factorization machines. http://www.csie.ntu.edu.tw/ cjlin/libffm/, 2015.

[7] S. Rendle. Factorization machines. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 995–1000, Washington, DC, USA, 2010. IEEE Computer Society.

[8] S. Rendle. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22, May 2012.