Multi-Perspective Modeling for Click Event Prediction

Tzu-Chun Lin
Computer & Information Science
Indiana University - Purdue University
Indianapolis
vansonlin@gmail.com

Xia Ning
Computer & Information Science
Indiana University - Purdue University
Indianapolis
xning@cs.iupui.edu

ABSTRACT

We present our solutions to the RecSys Challenge 2015. We propose a multi-perspective modeling scheme for click event prediction, which involves techniques from sophisticated feature engineering for both click sessions and clicked items, classification based on gradient boosting tree, semi-supervised learning that utilizes information from test data, multi-class classification for different categories of sessions and items, classifier-based feature fusion from multi-class classification and in the end classifier ensembles from multiple models. We demonstrate that our scheme is intuitive, flexible and powerful for the Challenge tasks. Our solution based on the scheme achieves a score of 49,517.2 in the Challenge.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms

Keywords

Feature engineering, classification, boosting, model ensembles

1. INTRODUCTION

User clicks represent a rich source of information, in which user preferences, behavior patterns and search intentions are implicitly embedded. It has been well recognized that such information is highly valuable particularly for online retailers to better understand their customers and make corresponding advertising and recommendations so as to increase revenue. Two out of many important questions with respect to user clicks and click sessions are whether a certain click session will result in a buying event, and which items will be bought then. These are the two challenge questions raised in RecSys Challenge 2015 [2].

We propose a multi-perspective modeling scheme to tackle the problems of predicting sessions with potential buying events and predicting potentially bought items in such sessions in the Challenge[1]. In specific, we propose to apply sophisticated feature engineering techniques on click sessions to predict potential buying events. The comprehensive features we generate for click sessions span a wide range of aspects that together thoroughly depict the click sessions that are typically limited in length and information content. We also propose to learn from clicked items collectively from multiple sessions to predict potential bought items. The item features we generate capture not only the item properties from a single session but also latent characteristics embedded from a large collection of sessions. Moreover, we take advantages of the multiple machine learning methodologies including semi-supervised learning [4], multi-class classification [3] and classifier ensembles [5] to further combine the above two modeling approaches and enhance the prediction performance. Our experimental results demonstrate good performance of the proposed scheme.

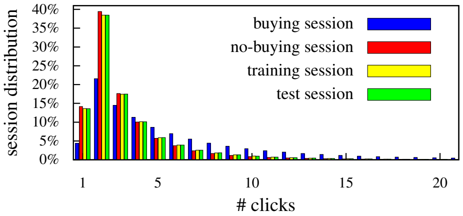

Figure 1: click distributions

2. CHALLENGE DESCRIPTION

2.1 Data Description

The Challenge provides two training datasets and one test dataset [2]. The first training dataset has 33,003,944 unique clicks from 9,249,729 sessions over 52,739 unique items spanning from 03/31/2014 to 09/29/2014. The other training dataset has all the sessions that are known to have a buying event, referred to as buying sessions. There are 509,696 buying sessions with 1,150,753 buying events over 19,949 items. All such sessions are included in the first training dataset. The sessions without buying events are referred to as no-buying sessions. The test dataset has a disjoint set of sessions without buying information, on which the final submission and evaluation will be. There are 2,312,432 sessions with 8,251,791 clicks over 42,155 items in the test dataset. There are 1,548 new items that only appear in the test dataset but not in the training dataset. In total, there are $m = 11,562,161$ sessions and $n = 54,287$ items given in the Challenge. Figure 1 presents the session distributions with respect to the number of clicks in a session. It demonstrates that buying sessions exhibit certain characteristics

RecSys'15 Challenge, September 16-20, 2015, Vienna, Austria
©2015 ACM. ISBN 978-1-4503-3665-9/15/09 ...$15.00
DOI: http://dx.doi.org/10.1145/2813448.2813520.

[1] Team name: LittleMaster, registered email address: vansonlin@gmail.com

(e.g., more clicks) and suggests that features that capture such characteristics will help predictions.

2.2 Task Description

There are two tasks in the Challenge. The first task is to predict in the test dataset which sessions may end up with a buying event. The second task is to predict for those predicted buying sessions which items are going to be bought. The performance of the proposed methods is evaluated using the score defined as follows.

$$Score(\mathbf{Sl}) = \sum_{\forall s \in \mathbf{Sl}} \begin{cases} \frac{|S_b|}{|S|} + \frac{A_s \cap B_s}{A_s \cup B_s} & \text{if } s \in \mathbf{Sb} \\ -\frac{|S_b|}{|S|} & \text{otherwise,} \end{cases} \quad (1)$$

where $\mathbf{Sl}$ is the set of sessions in the submitted solutions, S is the test dataset, S_b is the set of true buying sessions in S, A_s is the set of predicted bought items in session s and B_s is the set of actual bought items in s. Higher scores indicate better performance.

3. RELATED WORK

The work related to the Challenge primarily includes click models and predictions from click sessions. For example, Sadikov *et al.* [7] model the user click behaviors as graphs and cluster the graphs to understand user behavior patterns. Hassan *et al.* [6] use features from click sessions to predict whether the session is successful (i.e., the query goal is successful) or not. Anastasakos *et al.* [1] use click-ad graph and collaborative filtering method to predict and recommend relevant advertisements.

4. METHODS

The fundamental approach we take to solve the two Challenge tasks relies on feature engineering, classification and boosting. For the two tasks, we first generate features to describe sessions and clicks. Then we apply Gradient Boosting Tree (GBT) classifiers[2] to train basic classification models. GBT classifiers have been demonstrated to achieve good performance in click models [6]. We further adapt other machine learning methodologies on the basic models, which will be presented later in Section 4.3. The given training set is large. Therefore, we rely on down sampling in our methods to generate training instances of smaller sizes that the models can produce results on in a reasonable time period.

4.1 Session Models for Task 1: M_s^1

The first task is to predict for each session whether there is going to be a buying event. We first build a basic supervised classification model for this task. This model is denoted as M_s^1.

4.1.1 Generating Training Instances

To generate training instances for M_s^1, we randomly select a set of 51,084 buying sessions (i.e., about 10% of all buying sessions), and a set of 67,372 no-buying sessions (i.e., relatively same size as buying sessions). We randomly split these samples so that a set of 40,763 of the buying sessions, denoted as $\mathcal{R}_s^+$ with label +1, and a set of 53,911 of the no-buying sessions, denoted as $\mathcal{R}_s^-$ with label -1, are used for model training, and the rest, denoted as $\mathcal{T}_s^+$ and $\mathcal{T}_s^-$, respectively, for evaluation.

4.1.2 Generating Session Features

We represent each session using 139 features. These features are categorized into 6 types that capture various aspects of the sessions. A detailed list of all the features is available in the supplementary materials[3].

Category 1: Statistics-based Session Features: The first type of session features captures the statistical information within a session (e.g., the number of items in a session, the number of clicks in a session). We generate 66 such features from each session.

Category 2: Similarity-based Session Features: The second type of session features is based on the pairwise item similarities within a session. We first construct a session-item matrix, denoted as C ($C \in R^{m \times n}$, $m = 11,562,161$, $n = 54,287$), in which each row represents a session, each column represents an item, and if an item is clicked within a session, the corresponding element in C is set to the number of clicks, otherwise 0. Then we construct an item-item click similarity matrix , denoted as S ($S \in R^{n \times n}$), by considering each column of C as a feature representation of an item and then calculating the cosine similarity of all the column pairs. Thus, for all the items in a session, we find their pairwise similarities and construct 8 similarity-based features.

Category 3: Transition-based Session Features: The third type of session features is based on the click transition probabilities. We first construct an item-item transition matrix by accumulating all the transitions between items from all the sessions. If an item j is clicked right after an item i, then there is a transition from item i to item j. We generate 8 transition-based features.

Category 4: Category-based Session Features: This set of 15 features captures the categories of the items in a session.

Category 5: Matrix-Factorization-based Session Features: Following the matrix factorization idea in recommender systems, we factorize the session-item matrix C into $C = PQ^{\mathsf{T}}$ ($P \in R^{m \times p}$, $Q \in R^{n \times p}$, p is the rank of C) using Singular Value Decomposition (SVD) and thus each row of P can be considered as a feature representation of a session in a certain space. The matrix C has rank 7 (i.e., $p = 7$). Thus, we have 7 matrix-factorization based features for each session.

Category 6: Time-based Session Features: We also consider when the sessions happen by constructing a set of time-based session features. In specific, we look at in which day of a week and in which hour of a day a session happens. We have 35 time-based session features.

4.1.3 Training a Binary Classifier

We train the basic GBT model M_s^1 on $\mathcal{R}_s^+$ and $\mathcal{R}_s^-$. We tune the parameters of M_s^1 by looking at the difference between true positive (TP) and false positive (FP) on the evaluation set $\mathcal{T}_s^+ \cup \mathcal{T}_s^-$. We apply a grid search over the three parameters of GBT (e.g., the number of estimators, learning rate and the maximum depth) to select the best parameters that maximize TP $-$ FP.

4.2 Item Models for Task 2: M_i^2

The second task is to predict for those sessions which may end up with a buying event, which items will be bought. We solve this task by building a supervised binary classification model, denoted as M_i^2, as follows.

4.2.1 Generating Training Instances

We generate training instances for M_i^2 from the known buying sessions. From all the buying sessions, we randomly select a set of 200K bought items, denoted as $\mathcal{R}_i^+$ with label +1, and a set of 200K clicked but not bought items, denoted as $\mathcal{R}_i^-$ with label -1, for model training, and similarly 200K bought items and 200K not bought items, denoted as $\mathcal{T}_i^+$ and $\mathcal{T}_i^-$, respectively, for evaluation.

[2] http://scikit-learn.org/stable/modules/ensemble.html

[3] https://goo.gl/EBLTtX

4.2.2 Generating Item Features

We represent each item using 27 features. The features are generated from buying sessions to capture the time spent on each item within a session, the number of clicks on each item, when the item is clicked and how the click patterns are, etc. A detailed list of such features is available in the supplementary materials. Unique features include the random-walk-based feature and the matrix-factorization-based features.

Random-Walk-based Item Feature: We introduce a feature based on random walk that captures the stationary distribution over items if we consider the click sequence in a session as a random walk among the items. In specific, for each session, we calculate and normalize the eigen vector of the click transition matrix corresponding to its eigen value 1. We use the probability of each item from the vector as a feature for that item, which represents the probability that the click sequence will end up at the item.

Matrix-Factorization-based Item Features: We introduce two sets of features based on matrix factorization, which capture the latent properties of items that are embedded across many sessions. In the session-item matrix factorization as in Section 4.1.2 (i.e., $C = PQ^{\mathsf{T}}$), each row of Q is used as a feature representation of an item and thus 7 features for each item. In addition, we factorize the item-item similarity matrix S as described in Section 4.1.2 into $S = HH^{\mathsf{T}}$ ($H \in R^{n \times k}$, $k \leq n$) using SVD and thus each row of H is used as a feature representation of an item in a certain space. The rank of S is 6 and thus there are 6 such features.

4.2.3 Training a Binary Classifier

We train the GBT model M_i^2 on $\mathcal{R}_i^+$ and $\mathcal{R}_i^-$. We apply an additional thresholding parameter, denoted as `pthrd`, on the probability predictions given by M_i^2 such that if the probabilities are higher than `pthrd`, the predicted labels will be considered as +1, otherwise -1. We evaluate the performance of M_i^2 using F1-score , which is the harmonic mean of precision and recall, on $\mathcal{T}_i^+ \cup \mathcal{T}_i^-$, and apply a grid search over the three parameters of M_i^2 and `pthrd` to select the best parameters that maximize the F1-score.

4.3 Improved Models for Task 1: $\mathrm{M}_s^{1,ml}$ & $\mathrm{M}_s^{1,ss}$

We take the idea of Task 2 modeling into Task 1 to enhance the performance on Task 1. In addition, we apply the idea of semi-supervised learning in Task 1. The final solution for Task 1 becomes an ensemble of solutions from multiple models.

4.3.1 Multi-Level Item-Based Models: $\mathrm{M}_s^{1,ml}$

We take advantages of item-based modeling as in Section 4.2 to make predictions for sessions. The intuition is if there are items that are predicted as bought items in a session, the session could be predicted as a buying session. To implement this, we build a two-level model which encapsulates three item-based models on the first level, whose outputs are fused as features for a session-based model on the second level.

We first sample 10k bought items from buying sessions into a set denoted as $\mathcal{E}_i^+$ with label +1, and then a same number of items that are clicked but not bought from buying sessions into a set $\mathcal{E}_i^0$ with label 0, and a same number of items from no-buying sessions into a set $\mathcal{E}_i^-$ with label -1. Then we learn a GBT model $\mathrm{M}_i^{+/0}$ from $\mathcal{E}_i^+$ and $\mathcal{E}_i^0$, a GBT model $\mathrm{M}_i^{+/-}$ from $\mathcal{E}_i^+$ and $\mathcal{E}_i^-$, and a three-class GBT model $\mathrm{M}_i^{+/0/-}$ from $\mathcal{E}_i^+$, $\mathcal{E}_i^0$ and $\mathcal{E}_i^-$. We apply the three models on all the items from a different set of 40K sessions including 20K buying sessions and 20K no-buying sessions. We use four sets of probabilities generated for each session from the three models to generate session features for the next session-based model. These probabilities include the probability of belonging to class +1 based on $\mathrm{M}_i^{+/0}$, denoted as $p(+|\mathrm{M}_i^{+/0})$, the probability of belonging to class +1 based on $\mathrm{M}_i^{+/-}$, denoted as $p(+|\mathrm{M}_i^{+/-})$, and the probability of belonging to class +1 based on $\mathrm{M}_i^{+/0/-}$, denoted as $p(+|\mathrm{M}_i^{+/0/-})$. In addition, we calculate the probability of being from buying sessions based on $\mathrm{M}_i^{+/0/-}$, denoted as $p(+/0|\mathrm{M}_i^{+/0/-})$, by summing up the probabilities of belonging to class +1 and class 0 based on $\mathrm{M}_i^{+/0/-}$. Thus, each session will have four sets of probabilities $\{p(+|\mathrm{M}_i^{+/0})\}$, $\{p(+|\mathrm{M}_i^{+/-})\}$, $\{p(+|\mathrm{M}_i^{+/0/-})\}$ and $\{p(+/0|\mathrm{M}_i^{+/0/-})\}$ from all its items. We calculate the mean, and 0%, 10%, up to 100% percentiles, respectively, and thus in total 12 values (features) for each set of probabilities. For $\{p(+/0|\mathrm{M}_i^{+/0/-})\}$, we additionally include the second highest probability and the ratio between the first and second highest probabilities as the additional features. A detailed list of all the 50 features is available in the supplementary materials. We build a second-level GBT model, denoted as $\mathrm{M}_s^{1,ml}$, on the 50 features for Task 1.

4.3.2 Semi-Supervised Models: $\mathrm{M}_s^{1,ss}$

We adopt the idea of semi-supervised learning and use the basic model M_s^1 learned in Section 4.1 to learn another model, denoted as $\mathrm{M}_s^{1,ss}$, for Task 1. Semi-supervised learning is well known for its capacity of utilizing unlabeled data to improve model quality [4] We apply M_s^1 on the Challenge test sessions. From the test sessions we select the top 50K sessions that have the highest predicted probabilities of being buying sessions as the set of semi-positive training instances, denoted as $\mathcal{R}_s^{+0.5}$, and the top 50K sessions that have the highest predicted probabilities of being no-buying sessions as the set of semi-negative training instances, denoted as $\mathcal{R}_s^{-0.5}$. Then we train the GBT model $\mathrm{M}_s^{1,ss}$ on $\mathcal{R}_s^+ \cup \mathcal{R}_s^{+0.5}$ and $\mathcal{R}_s^- \cup \mathcal{R}_s^{-0.5}$ ($\mathcal{R}_s^+$ and $\mathcal{R}_s^-$ are used to train M_s^1).

4.3.3 Model Ensembles

To generate the final results for Task 1, we apply the model M_s^1 and the semi-supervised model $\mathrm{M}_s^{1,ss}$ on the Challenge test sessions. Each of the two models produces a probability, denoted as p^1 and $p^{1,ss}$, respectively, measuring the probability that a test session is a buying session. We linearly combine p^1 and $p^{1,ss}$ as $0.8p^1 + 0.2p^{1,ss}$ as the final probability for that session. We further apply a threshold 0.4 such that if a session has a combined probability that is higher than the threshold, it will be predicted as a buying session. Similarly, we apply the multi-level model $\mathrm{M}_s^{1,ml}$ from Section 4.3.1 on the test sessions and apply a threshold 0.45. The common sessions that are predicted as buying sessions based on M_s^1 and $\mathrm{M}_s^{1,ss}$ and based on $\mathrm{M}_s^{1,ml}$ are considered as the final predicted buying sessions.

5. EXPERIMENTAL RESULTS

5.1 Experiments for Task 1 Model M_s^1

Table 1 presents the results of various parameters for M_s^1. Based on $\mathrm{TF} - \mathrm{FP}$, the best performing parameters for M_s^1 are $\mathrm{nest} = 100$, $\mathrm{lr} = 0.1$ and $\mathrm{md} = 10$. Figure 2 shows the feature importance from the best M_s^1. The most important features for M_s^1 are the feature 69 (sum of all pairwise item similarities), and the feature 5 (average time spent on each click).

5.2 Experiments for Task 2 Model M_i^2

Table 2 presents the results of various parameters for M_i^2. Based on F-scores, the best performing parameters for M_i^2 are $\mathrm{nest} = 200$, $\mathrm{lr} = 0.1$, $\mathrm{md} = 5$ and $\texttt{pthrd} = 0.38$. Figure 3 shows the feature

Table 1: M_s^1 Model Performance for Task 1

nest	lr	md	TP	FP	precision	recall	F1	TP-FP
100	0.10	5	7197	3147	0.696	0.697	0.697	4050
100	**0.10**	**10**	**7108**	**2965**	**0.706**	**0.689**	**0.697**	**4143**
100	0.50	5	6865	2872	0.705	0.665	0.685	3993
100	0.50	10	6712	3244	0.674	0.650	0.662	3468

The column under "nest" represents the number of estimators for GBT. The column under "lr" represents the learning rate for GBT. The column under "md" represents the maximum depth for GBT. The best performance is in **bold**.

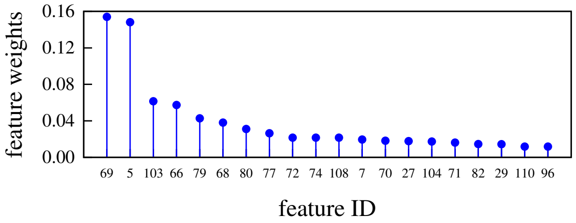

Figure 2: Feature Importance in M_s^1.

importance in the best M_i^2. The most important features for M_i^2 are the feature 12 (how many times the item has been bought by others), feature 4 (the percentage of time over the entire session spent on the item) and feature 13 (how many times the item has been clicked in buying sessions).

Table 2: M_i^2 Model Performance for Task 2

nest	lr	md	pthrd	precision	recall	F1
200	**0.10**	**5**	**0.38**	**0.710**	**0.834**	**0.767**
200	0.10	5	0.40	0.720	0.818	0.766
200	0.20	5	0.38	0.707	0.818	0.759
200	0.20	5	0.40	0.716	0.804	0.758

The column under "pthrd" represents the thresholding parameter for GBT. Other columns have identical meanings as those in Table 1, respectively. The best performance is in **bold**.

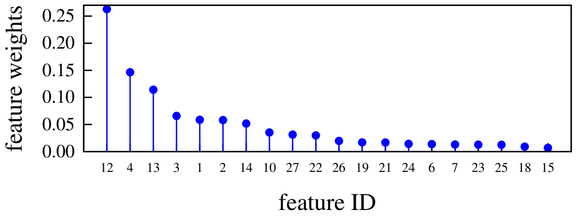

Figure 3: Feature Importance in M_i^2.

5.3 Experiments for Task 1 Model $\mathrm{M}_s^{1,ml}$

Table 3 presents the results of various parameters for $\mathrm{M}_s^{1,ml}$. Based on TF $-$ FP, the best performing parameters for $\mathrm{M}_s^{1,ml}$ are nest $=$ 200, lr $=$ 0.1 and md $=$ 3. Figure 4 shows the feature importance in the best $\mathrm{M}_s^{1,ml}$. The most important features for $\mathrm{M}_s^{1,ml}$ are the feature 39 (ratio of the second highest to the highest probability of $\{p(+/0|\mathrm{M}_i^{+/0/-})\}$), feature 23 (90% percentile of $\{p(+|\mathrm{M}_i^{+/-})\}$) and feature 12 (100% percentile of $\{p(+|\mathrm{M}_i^{+/0})\}$).

Table 3: $\mathrm{M}_s^{1,ml}$ Model Performance for Task 1

nest	lr	md	TP	FP	precision	recall	F1	TP-FP
100	0.1	2	15751	6483	0.708	0.788	0.746	9268
100	**0.1**	**3**	**15835**	**6561**	**0.707**	**0.792**	**0.747**	**9274**
100	0.3	2	16403	7250	0.693	0.820	0.751	9153
100	0.3	3	15767	6569	0.706	0.788	0.745	9198

The columns have identical meanings as those in Table 1, respectively. The best performance is in **bold**.

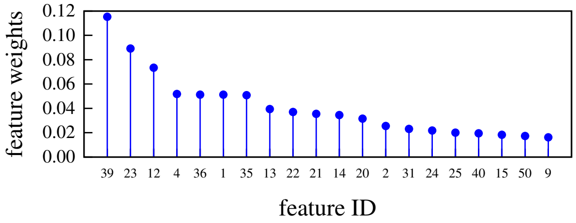

Figure 4: Feature Importance in $\mathrm{M}_s^{1,ml}$.

5.4 Final Submission Results

Table 4 presents the scores from various methods. The best score is from the ensemble of model M_s^1, $\mathrm{M}_s^{1,ss}$ and $\mathrm{M}_s^{1,ml}$ for Task 1, and M_i^2 for Task 2.

Table 4: Scores of submissions

Task 1	M_s^1	M_s^1	$\mathrm{M}_s^{1,ml}$	M_s^1+$\mathrm{M}_s^{1,ss}$	M_s^1+$\mathrm{M}_s^{1,ss}$+$\mathrm{M}_s^{1,ml}$
Task 2	$\mathrm{M}_i^{2,cf}$	M_i^2	M_i^2	M_i^2	M_i^2
#sbm	464,984	584,043	705,154	531,770	564,967
score	38,603.1	46,792.6	48,266.8	49,243.7	**49,517.2**

The row of "#sbm" represents the number of submitted sessions. $\mathrm{M}_i^{2,cf}$ is a simple model for Task 2 which predicts items that have been bought by others as the bought items. The best score is in **bold**.

6. CONCLUSIONS

We present our multi-perspective modeling scheme to the RecSys Challenge 2015. Our modeling scheme involves techniques from feature engineering, classification based on gradient boosting tree, semi-supervised learning, multi-class classification, classifier-based feature fusion and in the end classifier ensembles. Our modeling scheme is not only powerful for the Challenge, but also generalizable to similar problems.

7. REFERENCES

[1] T. Anastasakos, D. Hillard, S. Kshetramade, and H. Raghavan. A collaborative filtering approach to ad recommendation using the query-ad click graph. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 1927–1930, New York, NY, USA, 2009. ACM.

[2] D. Ben-Shimon, T. A., 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*. ACM, September 2015.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-supervised learning*. Adaptive computation and machine learning. MIT Press, Cambridge (Mass.), 2006.

[5] T. G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, pages 1–15, London, UK, 2000. Springer-Verlag.

[6] A. Hassan, R. Jones, and K. L. Klinkner. Beyond dcg: User behavior as a predictor of a successful search. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, pages 221–230, New York, NY, USA, 2010. ACM.

[7] E. Sadikov, J. Madhavan, L. Wang, and A. Halevy. Clustering query refinements by user intent. In *Proceedings of the 19th International Conference on World Wide Web*, pages 841–850, New York, NY, USA, 2010. ACM.