Balancing between Estimated Reward and Uncertainty during News Article Recommendation for ICML 2012 Exploration and Exploitation Challenge

Ku-Chun Chou r99922095@csie.ntu.edu.tw
Hsuan-Tien Lin htlin@csie.ntu.edu.tw
Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan

Abstract

Recommending relevant contents to users automatically in a web service is an important aspect that links with the income of many internet companies. The *ICML 2012 Exploration & Exploitation Workshop* holds an open challenge that aims at building state-of-the-art news article recommendation system on the Yahoo! platform. We propose an efficient scoring model that recommends the news article with the highest score during each user visit. The scoring model exploits by recommending the article with the highest estimated reward and explores articles with high reward potential by uncertainty measures. Three important aspects, global quality of articles, personal preference of users, and time effects are all considered in the scoring model. Furthermore, during the challenge, we adopt a systemic parameter tuning process to optimize the performance of the model. The tuned scoring model wins the first place of phase one of the challenge.

1. Introduction

The *ICML 2012 Exploration & Exploitation Challenge* aims at dynamically learning and predicting user preferences over news articles on the front page of Yahoo!. The prediction can then be used to recommend more preferable articles to users, which improves the overall user experience and increases the chance that users return in the future.

The dataset of the challenge is not visible to the participants. That is, participants are asked to submit and run their code on the challenge web site. The challenge consists of two phases. Phase 1 lasts three months, where the progress of each team is shown on a real-time scoreboard. After phase 1 ends, the best submission of each team is then evaluated on a larger and separate dataset for phase 2.

The paper summarizes the solution proposed by the National Taiwan University team, which won the phase 1 of the challenge. The solution balances between exploiting articles with high reward (i.e. preferred) and exploring articles with high uncertainty. The solution utilizes multiple components for estimating reward and uncertainty in order to capture different aspects of the challenge task. A systemic parameter-tuning step is included in the solution to gain better performance.

The paper is organized as follows. Section 2 describes the details of the challenge. The proposed solution is introduced in Section 3. Then, we discuss the parameter tuning step in Section 4. Finally, we show experimental results in Section 5 and conclude in Section 6.

2. Challenge Description

The challenge uses the Yahoo! R6b dataset, which is accessible only to the challenge participants at the time this paper is written. The dataset contains records from random traffic on the Today Module of Yahoo!, which means both the visitors and the recommended news article are selected randomly. The dataset consists of 30 million visits over a timespan of two weeks. For the i-th visit, the following five fields are logged:

1. a 136 dimensional boolean vector $\mathbf{x}_i$ that contains the features of the visiting user;

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2. the timestamp t_i of the visit;
3. a set of relevant news articles $\mathcal{A}_i$ that the system can recommend from;
4. a randomly recommended article $a_i^{data} \in \mathcal{A}_i$ during the visit;
5. a boolean value b_i indicating whether the recommended article a_i^{data} is clicked by the visiting user or not.

An algorithm $\mathbb{A}$ for the challenge shall conduct the following three steps on each visit i:

1. $\mathbb{A}$ observes feature $\mathbf{x}_i$ of the visitor on timestamp t_i along with a set of articles $\mathcal{A}_i$ to recommend from;
2. $\mathbb{A}$ recommends an article $a_i \in \mathcal{A}_i$ to the visitor;
3. $\mathbb{A}$ receives a click/non-click b_i from the visitor and tries to improve its strategy based on the record $(\mathbf{x}_i, t_i, a_i, b_i)$.

Each submission in the challenge is evaluated by the click-through rate (CTR), which is the proportion of received clicks in step 3. Note that step 3 can be executed only if the recommended article a_i of the algorithm $\mathbb{A}$ matches the recorded article a_i^{data} in the dataset. The evaluation procedure simulates an online environment with a huge offline dataset with sound theoretical guarantees (Li et al., 2011).

It is worth noticing in step 3 of the evaluation process, only the information of the article a_i recommended by algorithm $\mathbb{A}$ is visible to that algorithm. No other information is revealed for all other articles. This is a more realistic, and in general harder, setting of the real world environment than traditional supervised online learning and traditional reinforcement learning (Langford & Zhang, 2007; Lu et al., 2010).

In the challenge, all participants are asked to submit their code to a contest website for evaluation. The dataset is not released during the period of the challenge and any logging of the dataset is prohibited. Also, the meaning of features and information toward news articles are not provided. The offline-simulated CTR over all the visits is the only feedback.

Since the interest of the challenge is in real-time online recommendation, there is a 36-hour run time limitation to all submissions. This roughly translates to less then 5 ms from recommending a news article in step 2 from approximately 30 articles (size of relevant article set $\mathcal{A}_i$) and 50 ms for the updating algorithm $\mathbb{A}$ with record $(\mathbf{x}_i, t_i, a_i, b_i)$ in step 3.

Table 1. FTL VS ϵ-greedy

Model	Scoreboard CTR
FTL	0.0462
ϵ-greedy	0.0630

3. Our Approach

A straightforward approach is to recommend the article with the highest estimated CTR to the visitor. The approach, nevertheless, is sub-optimal. Imagine a scenario where there are two articles with a hindsight CTR of 0.1 and 0.9 respectively. For the first visit, we have no clue what to recommend. Then, a natural approach is to randomly recommend an article. With some luck, we may recommend the article with the 0.1 CTR and get a click from the visitor. Then, our system will keep recommending the article with the 0.1 CTR because its estimated CTR (1.0) is higher. This simple toy scenario demonstrates the dilemma in such a learning problem with partial information feedback. Thus, we actually need to explore less plausible articles by recommending them to the visitors in order to have a better modeling of the environment and also to minimize the effect of random influence.

We furthermore demonstrate the importance of exploration with two simple models. The first model is called follow the leader (FTL). FTL comes with two phases. The first phase is to randomly recommend articles to visitors and in the second phase always recommends the article with the highest CTR estimated from the first phase. The second model, ϵ-greedy, randomly recommends articles to the visitor with a probability of ϵ, and greedily recommends the article with the highest estimated CTR otherwise. The main difference between FTL and ϵ-greedy is the proportion of the dataset they explore on. FTL explores only the beginning of the dataset, while ϵ-greedy continuously explores over time. Table 1 shows the results of the two models on the challenge scoreboard. We can see that ϵ-greedy outperforms FTL. One possible explanation is that in real world situations, as well as the task of the challenge, the probability distribution of the CTR over the articles is not stationary. Thus, keeping some steps for exploration throughout the time shall be important.

With the importance of exploring throughout time and a need of real-time recommendation in mind, our proposed solution to the challenge is to build an efficient and dynamic scoring model for each article $a \in \mathcal{A}_i$. Then, we recommend the article with the highest score

during each visit:

$$a_i \leftarrow \underset{a \in \mathcal{A}_i}{\operatorname{argmax}}(r_a + c_a).$$

The scoring model consists of two parts. The first term r_a estimates the CTR of article a with the information we have gathered over time. The second term c_a measures how uncertain we are with our estimation.

The terms r_a and c_a automatically balance between exploitation and exploration. r_a governs over exploitation by favoring article with higher estimated CTR. On the other hand, c_a controls exploration by favoring articles that the model is more uncertain about. Then, the scoring model can recommend articles that are of high reward and/or high uncertainty. To properly balance the influence of the two terms, some parameter tuning (to be discussed in Section 4) is needed to adjust their numerical scales.

This kind of two-term scoring model is inspired by many successful previous works (Chu et al., 2011; Garivier & Cappé, 2011; Auer et al., 2002). The works demostrate promising theoretical guarantees and empirical performance when using a particular component in the scoring model such as linear regression with variance estimation. Our work, on the other hand, uses a mixture of multiple components to capture different aspects of the dataset of the challenge. While the use of multiple components makes our model harder to analyze in theory, we nevertheless observe its superior empirical performance within the challenge.

3.1. CTR Estimation Term

We include three components for the CTR estimation term r_a.

$$r_a = \alpha_1 \cdot \mu_a + \alpha_2 \cdot \frac{1}{\sqrt{t - t_a}} + \alpha_3 \cdot \mathbf{w}_a^T \mathbf{x}$$

All three components are calculated separately and then summed together, where $\alpha_{\{1,2,3\}}$ are tunable parameters that controls the mixture weights.

The first component μ_a keeps track of the naively estimated CTR of the articles from our observations using the feedback b_i that we have received.

The second component $\frac{1}{\sqrt{t-t_a}}$ biases the estimated CTR towards newer articles. Here t is the timestamp of the visit and t_a is the timestamp of the first appearance of article a. The component is motivated by an observation on the scoreboard using some baseline models. As shown in Table 2, always recommending the newest article is better than always recommending the oldest article by a CTR of 0.0246. Thus, the "freshness" of the articles is an important piece of information that can be used to bias the estimated CTR.

Table 2. Visitors tend to click on newer articles

Model	Scoreboard CTR
Always recommend oldest	0.0266
Random recommendation	0.0368
Always recommend newest	0.0512

The third component $\mathbf{w}_a^T \mathbf{x}$ tries to model the relation between different visitors from their feature vector $\mathbf{x}$ and their clicks b on the article a. For instance, teenagers are more likely to click on news article concerning with pop-music while businessmen are more likely to click on articles about the stock market.

We use a set of linear weights $\mathbf{w}_a$ to model the CTR on article a by taking a standard inner product with the user feature vector $\mathbf{x}$. The approach is proposed and is demonstrated to perform well in (Li et al., 2010) under a similar setting. With the click feedback from step 3 in the evaluation process, we can update the linear weights $\mathbf{w}_a$ with ridge linear regression

$$\mathbf{w}_a \leftarrow (\mathbf{X}_a^T \mathbf{X}_a + \mathbf{I})^{-1} \mathbf{b}_a,$$

where

$$\mathbf{X}_a = \begin{pmatrix} \mathbf{x}_1^a \\ \mathbf{x}_2^a \\ \vdots \\ \mathbf{x}_n^a \end{pmatrix}.$$

$\mathbf{x}_1^a$ to $\mathbf{x}_n^a$ are n visitor feature row vectors that is recorded for article a, and $b_a \in \{0, 1\}$ indicates a click from the click feedback. $\mathbf{I}$ is a respective identity matrix for regularization purpose.

3.2. Uncertainly Term

There are two components for measuring uncertainty.

$$c_a = \beta_1 \cdot \frac{1}{n_a} + \beta_2 \cdot \sqrt{\mathbf{x}^T (\mathbf{X}_a^T \mathbf{X}_a + \mathbf{I})^{-1} \mathbf{x}}$$

The values $\beta_{\{1,2\}}$ are tunable parameters like $\alpha_{\{1,2,3\}}$.

The first component $\frac{1}{n_a}$ models the uncertainty by considering the number of times n_a that an article a is recommended and has been given feedback on.

The second component $\sqrt{\mathbf{x}^T (\mathbf{X}_a^T \mathbf{X}_a + \mathbf{I})^{-1} \mathbf{x}}$ is the standard deviation of the CTR estimation component

$\mathbf{w}_a^T\mathbf{x}$ with respect of $\mathbf{x}$. This term is also used in the upper confidence bound of the popular linUCB model (Chu et al., 2011).

Both components will gradually decrease with a better understanding of the environment. The first component hints our model to explore articles that we have less information on. The second component guides our model to explore for articles that we have less information on with respect to the visitor.

3.3. Feature Engineering

In additional to the original 136 boolean features, we add three more features to enhance the performance.

The first feature records how many `true`'s there are in the total 136 boolean features. That is, the l_1 norm of the 136-dimensional vector when the boolean `true` is represented by 1 and `false` by 0. This feature allows the model to compensate for the length of the original feature vector and prevent a feature vector $\mathbf{x}$ with too many 1's from dominating the $\mathbf{w}^T\mathbf{x}$ part of the scoring function.

The second feature is a time feature which indicates the scaled timestamp with respect to the time span of the dataset, with the first timestamp scaled to 0 and the last timestamp scaled to 1.

The third feature is simply a constant value acting as a bias term.

The above three additional features is added to both $\mathbf{w}^T\mathbf{x}$ and $\sqrt{\mathbf{x}^T(\mathbf{X}^T\mathbf{X}+\mathbf{I})^{-1}\mathbf{x}}$ term (defined in Section 3.1 and 3.2). The average scoreboard CTR improvement compared to the baseline models without the added features is approximately 0.012.

4. Parameter Tuning

Ensemble learning has been shown to be a successful methodology in other learning problems (Chen et al., 2012; McKenzie et al., 2012). Ensemble learning combines multiple learning models to obtain a grand model that is superior in performance. Linear blending is one of the most popular ensemble learning methods due to its simplicity and efficiency.

The scoring model in Section 2 can be viewed as an ensemble model. It contains multiple components, each of which captures a different aspect of task. There are five parameters in our model, $\alpha_{\{1,2,3\}}$ and $\beta_{\{1,2\}}$, which can be viewed as blending weights.

We have studied two ensemble methods during the phase one of the challenge. The first one is grid search, which is a simple method which will be discussed in Section 4.1. The second one is a two-stage linear regression method that treats the estimated CTR term and the uncertainty term differently, as discussed in Section 4.2.

4.1. Grid Search

Grid search is a simple way for finding the weight of each component. It locates a sweet spot of the weights by searching a subset of the possible combination of weights exhaustively.

4.2. Linear Blending on Click Rate and Residuals

If we want to combine more components, grid search could become really slow and tedious. We have considered another more systematic/automatic way of finding the weights of each component for the scoring model in Section 2. Recall that the model is a combination of an estimated click rate term and an uncertainty term, where each term contains multiple components.

The estimated CTR term can be naturally obtained by conducting linear regression from the components to the click feedback, much like how we conduct linear regression from the feature vector $\mathbf{x}$ to the click feedback. Linear regression allows obtaining the weights $\alpha_{\{1,2,3\}}$ to accurately estimate the CTR itself.

We reuse linear regression in finding the weights $\beta_{\{1,2\}}$ for the uncertainty term. The idea here is to capture the uncertainty with the amount of mistakes made by the estimated CTR term. The mistakes are called the residuals, defined as the (b_t − estimated CTR). We propose to determine $\beta_{\{1,2\}}$ by conducting linear regression from the two uncertainty terms to the residual. The final ensemble method contains two stages, one for CTR estimation and one for uncertainty estimation, and are listed in Algorithm 1.

The intuitive explanation of the above method is that we would like the scoring model to give scores to different articles that matches their real CTR as closely as possible. If we can have the real CTR of each article, recommending the article with the highest CTR would be an optimal strategy. Nevertheless, in our models the estimated CTR may not be so accurate because of the partial feedback. Then, the residuals represent the inaccuracy and can be used in the uncertainty term to correct the estimated CTR.

Algorithm 1 Two-stage Linear Regression

Initialize: $\eta \in [0, 1]$:
parameter for balancing between learning and tuning
repeat
1) Get user feature $\mathbf{x}$ and timestamp t.
2) Get values $\omega_n, n = 1, 2, \cdots, N$ from N estimated CTR term component for each $a_i \in \mathcal{A}_i$.
3) Get values $\nu_m, m = 1, 2, \cdots, M$ from M uncertainty term component for each $a_i \in \mathcal{A}_i$.
4) Show the article with the max score.
5) Get click feedback b_i.
6) With η probability do 7-1) else do 7-2) and 7-3)
7-1) Update estimated CTR term and uncertainty term with b_i.
7-2) Update parameters for estimated CTR term with b_i and $\omega_{1,\cdots,n}$ with linear regression.
7-3) Update parameters for uncertainty term with residual $(\frac{1}{M}\sum_{m=1}^{M}\nu_m) - b_i$ and $\nu_{1,\cdots,m}$ with linear regression.
until no more visits

Table 3. Scoreboard results

Model	Scoreboard CTR
$r_a = \alpha_1\mu_a$	0.0462
$r_a = \alpha_1\mu_a$ $c_a = \frac{\beta_1}{n_a}$	0.0872
$r_a = \alpha_3\mathbf{w}_a^T\mathbf{x}$ $c_a = \beta_2\sqrt{\mathbf{x}^T(\mathbf{D}_a^T\mathbf{D}_a + \mathbf{I})^{-1}\mathbf{x}}$	0.0758
$r_a = \alpha_1\mu_a + \alpha_3\mathbf{w}_a^T\mathbf{x}$ $c_a = \frac{\beta_1}{n_a} + \beta_2\sqrt{\mathbf{x}^T(\mathbf{D}_a^T\mathbf{D}_a + \mathbf{I})^{-1}\mathbf{x}}$	0.0892
$r_a = \alpha_1\mu_a + \alpha_3\mathbf{w}_a^T\mathbf{x}$ $c_a = \frac{\beta_1}{n_a} + \beta_2\sqrt{\mathbf{x}^T(\mathbf{D}_a^T\mathbf{D}_a + \mathbf{I})^{-1}\mathbf{x}}$ *(with automatic parameter tuning)*	0.0889
$r_a = \alpha_1\mu_a + \frac{\alpha_2}{\sqrt{t-t_a}} + \alpha_3\mathbf{w}_a^T\mathbf{x}$ $c_a = \frac{\beta_1}{n_a} + \beta_2\sqrt{\mathbf{x}^T(\mathbf{D}_a^T\mathbf{D}_a + \mathbf{I})^{-1}\mathbf{x}}$	0.0896
$r_a = \alpha_1\mu_a + \frac{\alpha_2}{\sqrt{t-t_a}} + \alpha_3\mathbf{w}_a^T\mathbf{x}$ $c_a = \frac{\beta_1}{n_a} + \beta_2\sqrt{\mathbf{x}^T(\mathbf{D}_a^T\mathbf{D}_a + \mathbf{I})^{-1}\mathbf{x}}$ *(with feature engineering)*	0.0905

5. Scoreboard Results

Next, we list the performance of different models that we have studied during the phase one of the challenge. They are combinations of different components described in Section 2 with grid search tuning. The results are listed in Table 3. In Table 3, r_a and c_a indicates the components used in the estimated CTR term and the uncertainty term, respectively, in the submitted model. For the seven listed model, the highest scoreboard CTR of each model is reported.

The first model $r_a = \alpha_1\mu_a$ is a baseline model which only considers the estimated CTR without using the features. The second model is a same model that considers a simple uncertainty term $c_a = \frac{\beta_1}{n_a}$ which greatly improves the performance by 0.0410.

The third model is the linUCB model which considers features in both estimated CTR and uncertainty term.

The fourth model is a mixture model of the second and third model. The fifth model is the same model as the fourth model, expect that the parameters in the forth model is tuned by grid search, and the fifth model is tuned by the method stated in Section 4.2. We can see that grid search outperforms our proposed systematic way of parameter tuning. Nevertheless, the proposed method may still be competitive if the budget on the number of submissions is limited.

The sixth model has added a time effect component $\frac{\alpha_2}{\sqrt{t-t_a}}$ in the estimated CTR term r_a. And lastly, the seventh model is the sixth model using three additional features stated in Section 3.3. The seventh model combines all the ideas described in this paper with grid search and is the model with the highest CTR during the phase one of the challenge.

6. Conclusion and Future Work

We presented a simple scoring model that predicts the preference of visitors over news articles on the Yahoo! website. This model won the phase one of *ICML 2012 Exploration & Exploitation Challenge*. The key of the model is to build multiple components to capture different aspects of CTR estimation and uncertainty using only the partial click feedback.

The model reveals that combining multiple components is a challenging issue that can be studied more carefully for building a successful system, especially when including many components. Both the theoretical and practical sides of ensemble learning for the online feedback system remain to be interesting research directions.

7. Acknowledgements

We thank the organizers for holding this interesting competition. We also thank the support from NTU CSIE Department, NSC 100-2628-E-002-010 and NTU 10R70839. Finally, we thanks the fruitful discussions from the Computational Learning Lab @ NTU.

References

Auer, Peter, Cesa-Bianchi, Nicolò, and Fischer, Paul. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256, May 2002. ISSN 0885-6125. doi: 10.1023/A:1013689704352. URL http://dx.doi.org/10.1023/A:1013689704352.

Chen, Po-Lung, Tsai, Chen-Tse, Chen, Yao-Nan, Chou, Ku-Chun, Li, Chun-Liang, Tsai, Cheng-Hao, Wu, Kuan-Wei, Chou, Yu-Cheng, Li, Chung-Yi, Lin, Wei-Shih, Yu, Shu-Hao, Chiu, Rong-Bing, Lin, Chieh-Yen, Wang, Chien-Chih, Wang, Po-Wei, Su, Wei-Lun, Wu, Chen-Hung, Kuo, Tsung-Ting, McKenzie, Todd G., Chang, Ya-Hsuan, Ferng, Chun-Sung, Ni, Chia-Mau, and Lin, Hsuan-Tien. A linear ensemble of individual and blended models for music rating prediction. *Journal of Machine Learning Research - Workshop and Conference Proceedings*, 18:21–60, 2012.

Chu, Wei, Li, Lihong, Reyzin, Lev, and Schapire, Robert E. Contextual bandits with linear payoff functions. *Journal of Machine Learning Research - Proceedings Track*, 15:208–214, 2011.

Garivier, Aurélien and Cappé, Olivier. The kl-ucb algorithm for bounded stochastic bandits and beyond. *Journal of Machine Learning Research - Proceedings Track*, 19:359–376, 2011.

Langford, John and Zhang, Tong. The epoch-greedy algorithm for multi-armed bandits with side information. In Platt, John C., Koller, Daphne, Singer, Yoram, and Roweis, Sam T. (eds.), *NIPS*. Curran Associates, Inc., 2007.

Li, Lihong, Chu, Wei, Langford, John, and Schapire, Robert E. A contextual-bandit approach to personalized news article recommendation. In Rappa, Michael, Jones, Paul, Freire, Juliana, and Chakrabarti, Soumen (eds.), *WWW*, pp. 661–670. ACM, 2010. ISBN 978-1-60558-799-8.

Li, Lihong, Chu, Wei, Langford, John, and Wang, Xuanhui. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In King, Irwin, Nejdl, Wolfgang, and Li, Hang (eds.), *WSDM*, pp. 297–306. ACM, 2011. ISBN 978-1-4503-0493-1.

Lu, Tyler, Pal, David, and Pal, Martin. Contextual multi-armed bandits. *Journal of Machine Learning Research - Workshop and Conference Proceedings*, 9:485–492, 2010.

McKenzie, Todd G., Ferng, Chun-Sung, Chen, Yao-Nan, Li, Chun-Liang, Tsai, Cheng-Hao, Wu, Kuan-Wei, Chang, Ya-Hsuan, Li, Chung-Yi, Lin, Wei-Shih, Yu, Shu-Hao, Lin, Chieh-Yen, Wang, Po-Wei, Ni, Chia-Mau, Su, Wei-Lun, Kuo, Tsung-Ting, Tsai, Chen-Tse, Chen, Po-Lung, Chiu, Rong-Bing, Chou, Ku-Chun, Chou, Yu-Cheng, Wang, Chien-Chih, Wu, Chen-Hung, and Lin, Hsuan-Tien. Novel models and ensemble techniques to discriminate favorite items from unrated ones for personalized music recommendation. *Journal of Machine Learning Research - Workshop and Conference Proceedings*, 18:101–135, 2012.