

AUTO-SCALED BAYESIAN BROWSING MODEL IN MASSIVE DATA

Liyun Ru¹, Anhui Wang², Yingying Wu³, Shaoping Ma¹

¹State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

²Sogou.com Corporation, Beijing 100084, China

³Graduate School of Arts and Sciences, Harvard University, Cambridge, MA 02138, USA
lyru@vip.sohu.com

Abstract: In the field of information retrieval, the method of building a click model by mining click logs to improve the effect of the search engine has been widely studied. And Bayesian Browsing Model (BBM), for the calculation of the operability and the effectiveness of the result, is widely used. However, when applied in engineering, especially for the situation of large scale data, this model will not perform properly. This problem is described analytically and shown by numerical experiments in this paper. For this problem, an auto-scaled BBM method is proposed. Experiments show that the new method solves the problem of original BBM, and have a better performance in terms of NDCG.

Keywords: Click model; Bayesian; Click logs

1 Introduction

With the development of internet and explosive growth of information on the web, search engines have become an important tool helping users to acquire information. How to return documents with ideal ranking for a query is always a challenging task for researchers in related fields. Previous studies [1-6] have developed many ranking optimizing algorithms for manually labeled training data. But these methods are vulnerable to the distribution and quality of labeled data. And the work of labeling is expensive and time consuming because of up-to-date characteristic which needs to update labels frequently. To avoid these drawbacks, more and more researchers have been interested in exploring click-through logs which record user interactions with search engines and can be collected at a low cost. With fresh and timely information, these click-through data is treated as important signal of user's preference. Previous studies [7-9] show that click-through data is informative but biased, which describes the phenomenon: a document with a higher rank is more likely to be examined and clicked than a lower ranked one even its relevance is not so good as the lower's. Researchers regard this phenomenon as position bias. To resolve this problem and get a more accurate estimate of relevance, many studies [7-13] have proposed several click models describing users' behavior. Cascade model [7] assumes that users always

examine ranked documents from top to bottom sequentially. Users click the document and end their search session as soon as they find it satisfies their needs. Obviously, this model is only suitable for single click situation but not for multi-click sessions. For this problem, DCM [8] proposes that a user will not necessarily end his search session after the first click he made, and will continue his search with a probability. Therefore DCM extends Cascade model from single click situation to multi-click. UBM [9] extends the assumption of DCM, assuming that the probability of ending search session depends on not only the last click but also the positions between these documents clicked. With the assumptions of UBM, BBM(Bayesian Browsing Model) [10, 12] extends its application to large scale data with incremental feature, which means the strategy needs only one pass through fresh click-through data to update features and parameters. With these advantages, BBM has been widely considered to be used in industry. However, it performs not as well as we have expected. With the increase in scale of sessions, numerical computation about posterior probability and related statistics may drop into disaster or lead to unstable values, which are shown by theoretical inference and numerical experiments. Then this paper presents auto-scaled BBM algorithm, which can automatically decide how many sessions used. Experiments with commercial search click-through data show that the new method really works and even outperforms original BBM in terms of NDCG.

2 Problem with BBM

2.1 Introduction of BBM

We first introduce some definitions and notations. Here a session is treated as a unit data used in the model, and it records the whole click behaviors within the top M results (usually M=10). We denote $\mathbf{C} = (C_1, C_2, \dots, C_M)$ as user's clicks vector, where $C_i=1$ if clicked and 0 otherwise. $\mathbf{E} = (E_1, E_2, \dots, E_M)$ is defined as users' examination vector with value 1 or 0. $\mathbf{S} = (S_1, S_2, \dots, S_M)$ is regarded as the user's satisfaction with the results. According to these denotations, $\mathbf{C}$ is what we can find directly from click-through data and the other two variable vectors are

unknown. For a given query search engine returns a set of ranked documents $\phi(query) = \{doc_i, 1 \leq i \leq N\}$, where N is the number of documents returned.

BBM inherits the assumptions of UBM, which proposes the relation of user's satisfaction with his behaviors as Figure 1.

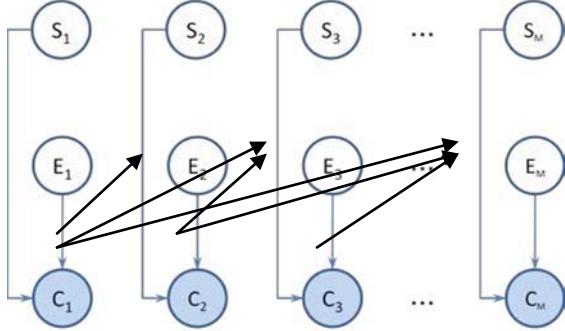


Figure 1 Relation of user's search behaviors.

According to BBM's definitions, we have the following:

$$P(E_i = 1 | C_{1:i-1}) = \beta_{r_i, d_i} \in [0, 1] \quad (1)$$

where i denotes the position of document user are examining. r_i is the position of last click when he examines the position i . d_i is the distance of current position between the last clicked. So $d_i = i - r_i$. $\beta = \{\beta_{r_i, d_i}\}$ are the parameters of BBM.

Let R_i be the probability of user's satisfaction S_i , then we can get the probability of next click:

$$P(C_i = 1 | E_i = 1, S_i) = R_i \in (0, 1) \quad (2)$$

Guo.eta [12] presents the posterior distribution of relevance as:

$$P(\mathbf{R} | \mathbf{C}) \propto p(\mathbf{R}) \prod_{k=1}^n P(C_k | \mathbf{R}) \quad (3)$$

where $\mathbf{R} = (R_1, R_2, R_3, \dots, R_n)$, $p(\mathbf{R})$ is the prior distribution of relevance, usually using Beta distribution. Because of the independence within documents for a query, we can get the posterior distribution of relevance for document j as this:

$$P(R_j | C_{1:n}) \propto R_j^{N_j} \prod_{(r,d) \in \varphi} (1 - \beta_{r,d} R_j)^{N_{j,r,d}} \quad (4)$$

where $N_j = \sum_{k=1}^n \sum_{i=1}^M I[(C_i^k = 1) \cap (\varphi_k(i) = j)]$ is the number of j -th document clicks.

$N_{r,d}^j = \sum_{k=1}^n \sum_{i=1}^M I[(C_i^k = 0) \cap (\varphi_k(i) = j) \cap (r_i^k = r) \cap (d_i^k = d)]$ denotes the number of situation that the j -th document has no click and a distance d from the last click position. And $\varphi_k(i)$ is the index of document ranked at i -th position of k -th session.

The parameters β of BBM can be estimated as following:

$$\beta_{r,d} = \min \left\{ 1, \frac{2N_{r,d}}{N_{r,d} + N_{r,d}'} \right\} \quad (5)$$

where $N_{r,d} = \sum_{k=1}^n \sum_{i=1}^M I[(C_i^k = 1) \cap (r_i^k = r) \cap (d_i^k = d)]$

$$N_{r,d}' = \sum_{k=1}^n \sum_{i=1}^M I[(C_i^k = 0) \cap (r_i^k = r) \cap (d_i^k = d)]$$

The posterior distribution $P(x)$ of relevance can be rewritten as the following form:

$$P(x) = C_0 * f(x) \quad (6)$$

$$f(x) = x^{e_0} * \prod_{i=1}^{M(M+1)/2} (1 - \lambda_i x)^{e_i} \quad (7)$$

where $e_0 = N$ is the number of clicks on the document considered. And $e_i = N_{r,d}$, $i = r * \frac{r-1}{2} + d$. In the same way, $\lambda_i = \beta_{r,d}$. C_0 is the normalized constant of distribution function $P(x)$, which has the main function part $f(x)$.

According to above definitions, we can get the characteristic vector γ and parameter vector β :

$$\gamma = (e_0, e_1, e_2, \dots, e_{M*(M+1)/2})$$

$$\beta = (\lambda_1, \lambda_2, \dots, \lambda_{M*(M+1)/2})$$

2.2 The problem

With time flies, the number of users and their clicks increase, and the scale of sessions will be explosive. That process can be depicted as following limitation:

$$\begin{aligned} \lim_{t \rightarrow \infty} N_j &= \sum_{k=1}^{\infty} \sum_{i=1}^M I[(C_i^k = 1) \cap (\varphi_k(i) = j)] \\ &= \sum_{i=1}^M \sum_{k=1}^{\infty} I[(C_i^k = 1) \cap (\varphi_k(i) = j)] \\ &= +\infty \end{aligned} \quad (8)$$

Similarly, we have

$$\lim_{t \rightarrow \infty} N_{r,d}^j = +\infty \quad (9)$$

The relevance R_i of a query and document does not vary with the scale of sessions, which depend on the number of users and their search times. This means R_i is a constant in terms of the number of sessions. For $R_j \in (0, 1)$, we have

$$\begin{aligned} \lim_{t \rightarrow \infty} P(R_j | C_{1:n}) &\propto \lim_{t \rightarrow \infty} R_j^{N_j} \prod_{(r,d) \in \varphi} (1 - \beta_{r,d} R_j)^{N_{j,r,d}} \leq \\ &\lim_{t \rightarrow \infty} R_j^{N_j} = 0 \end{aligned} \quad (10)$$

which means the posterior distribution function $P(x)$ varies with changing scale of sessions and the main function part $f(x)$ has a tendency of zero. Therefore the statistics such as mean or variance will not be stable, which is unreasonable. And what's even worse is that the tendency will cause numerical computation disaster.

2.3 Numerical experiments

In the following parts, we set $M=5$, which means only the documents ranked at top 5 positions are taken into account. Parameter vector β has 15 values. Table I gives an example of parameter β , which is estimated by a month of commercial search engine user click-through logs. In order to show the problem with BBM, we carry out numerical experiments with simulated data and real search logs respectively.

Table 1 Values of β_0

	d=1	d=2	d=3	d=4	d=5
r=0	1	0.83	0.69	0.65	0.61
r=1	0.35	0.16	0.09	0.06	
r=2	0.38	0.20	0.12		
r=3	0.36	0.20			
r=4	0.34				

Assumptions: For updating features in BBM, the incremental part $\Delta\gamma$ is varying with some fluctuations around a constant, which can be depicted as following formula:

$$\Delta\gamma = \delta + \varepsilon \quad (11)$$

where δ is the constant vector, and ε belongs to Gaussian noise with distribution of $N(0, \sigma^2 I)$. And they share the dimension of γ . If we have got n-lth characteristic vector, the n-th feature can be updated with following formula:

$$\gamma_n = \gamma_{n-1} + \Delta\gamma \quad (12)$$

Obviously, this assumption is reasonable because it's a simplified version of BBM's incremental characteristic.

Experiments with simulated data: we produce initial feature vector γ_0 randomly, which has 16 values belonging to interval $[0, 100]$. And we take the incremental constant vector $\delta = \gamma_0$ and add Gaussian noise produced from $N(0, 0.01)$. Figure 2 is the result of this simulated experiment, which shows us: (a) main function $f(x)$ varies with scale n of sessions. Obviously, this function has a tendency of zero as n tends to be positive infinity; (b) normalized function also fluctuates with scale n of sessions; (c) mean and variance, most useful statistics, turn to be constant as n increases. But they become "NaN" at the case of $n=630$, which means they cannot be computed. (d) Normalized constant also tends to be positive infinity and throws computing error.

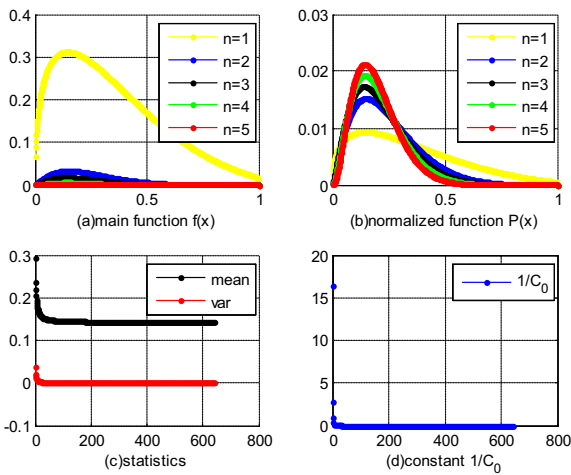


Figure 2 Simulated experiments

Experiment with real data: we select query "car" and document "http://auto.163.com" as a pair, and accumulate 60 day-units of click-through data from commercial search engine. Then we carry out the experiments by BBM with the result as Figure 3: (a)(b)

posterior distribution fluctuates with day-units. (c) mean and variance become "NaN" at the case of day-units=55 and throw out computing error. (d) Normalized constant also tends to be positive infinity and throws out computing error.

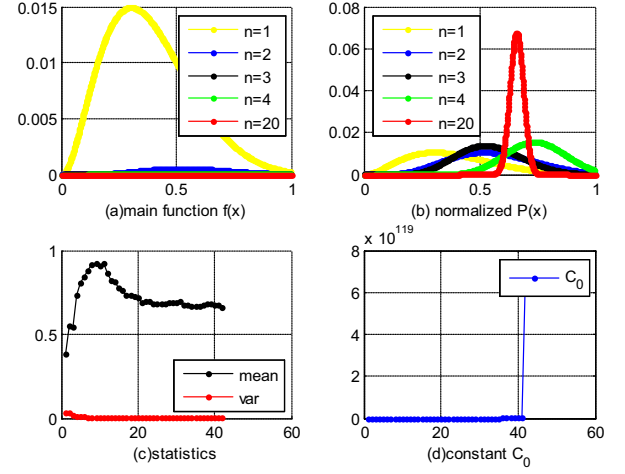


Figure 3 Real data experiments

According to the analysis of above experiments, we can see that BBM is not applicable to the massive data as we have expected, although it has incremental feature.

3 Auto-scaled BBM

As discussed above, BBM performs not as well as expected, especially for the massive scale of sessions. With the same assumptions as Eq. (11), we can find that the process of updating the features of BBM can be summarized as following formulas:

$$\Delta\gamma = \delta + \varepsilon$$

$$\gamma_n = \gamma_{n-1} + \Delta\gamma$$

So

$$\gamma_n = \gamma_0 + \sum_{k=0}^n (\delta + \varepsilon) \quad (13)$$

And we have

$$\varepsilon = \frac{\gamma_n - \gamma_0}{n} - \delta \in N(0, \sigma^2 I) \quad (14)$$

With the famous central-limit theorem, we can get

$$P\left(\left|\frac{\gamma_n}{n} - \delta\right| \geq \epsilon\right) \rightarrow 0, \text{ if } n \rightarrow \infty \quad (15)$$

which means $\delta \approx \frac{\gamma_n}{n}$ with high probability. So we can replace δ with $\frac{\gamma_n}{n}$ as the constant increment. More generally, n can be considered to be the scale of sessions. Then the constant increment can be computed by the average of accumulated features.

Considering the timeliness and stability requirement in ranking algorithm, we have to accumulate click-through data as much as possible which must be started from recently day. But computation requires the scale of data to be moderate. So how to decide the scale of data is a very important problem. According to this analysis, we set feature:

$$\gamma' = \alpha * \delta' \quad (16)$$

where $\alpha \in [1, n]$ is called transforming factor. And after this transformation, we notice that the parameters are the same as before.

$$\beta'_{r,d} = \min \left\{ 1, \frac{2N_{r,d} \frac{\alpha}{n}}{N_{r,d} \frac{\alpha}{n} + N'_{r,d} \frac{\alpha}{n}} \right\} = \min \left\{ 1, \frac{2N_{r,d}}{N_{r,d} + N'_{r,d}} \right\} = \beta_{r,d} \quad (17)$$

According to above analysis, this paper presents the auto-scaled Bayesian browsing model algorithm:

- 1) Set $T_0 = \epsilon > 0$, Estimate the β parameters
- 2) Accumulate BBM features γ_n from the nearest day-unit 0 to day-unit n . $\alpha = n$
- 3) $\gamma' = \alpha * \frac{\gamma_n}{n}$, and compute variance of posterior distribution BBM with parameter β and feature γ' . If the variance is small than T_0 , then stop. Otherwise return to step 2. If computation throws error, then Step 4
- 4) $\alpha = \alpha - 1$. Return to Step 3.
- 5) Statistics are computed as features for ranking.

4 Experiments and analysis

From commercial search engine logs, we collected click-through data for nearly twenty days, which have almost 100 million queries with 0.67 billion sessions, containing click information, position, queries and documents. Each of the selected sessions has at least once clicks. In the experiments, we just considered the queries, which have clicks more than ten in all returned documents.

Experiment 1: 500 query-document pairs such as “car” and “http://auto.163.com” are selected randomly. With auto-scaled BBM, we computed statistics such as mean and variance, which are averaged with weight of query’s frequency. And we get the results as Figure 4, of which x-axis is the average number of sessions. The Figure shows us that auto-scaled BBM, the black curve, can meet the mean’s stable value of original BBM’s result, which is the red curve. In the experiment, we find the original BBM gets into trouble of computation when n reaches 65. That means we cannot get the features by numerical computing methods. However, auto-scaled BBM will never encounter this problem and work well. That means auto-scaled BBM can automatically decide the suitable scale of sessions and produce more stable statistics than the original one.

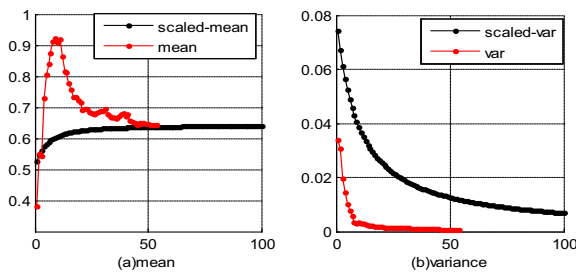


Figure 4 $T_0 = 0.01$, $n=980, \alpha=54$.

Experiment 2: It is known to all that queries can be classified into different kinds, mainly navigational and informational. From the given logs, 3602 queries are selected randomly and classified as navigational and informational according to [14]. After estimating parameters of different kind of queries, we compute posterior distribution with auto-scaled method and the original algorithm respectively. Statistics such as mean and variance are computed and used as features in our experiments. So there are three features for ranking: mean, variance, clicks. For query, we rank its documents by the value of mean*clicks/variance, which are regarded as score of relevance. Then we assign to the ranked documents expert’s labels: 0, 2, 3, 4 and 5, of which the bigger is the better. As to the evaluation method, we take the method of averaging the NDCG metrics at the top five positions respectively and get the results as Figure 5, which shows auto-scaled BBM, denoted by the red curves, and has a better performance than the original one in terms of NDCG.

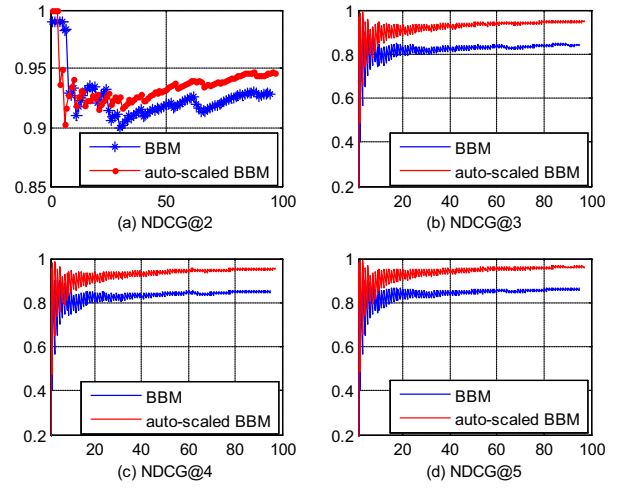


Figure 5 BBM vs. auto-scaled BBM $T_0 = 0.01$

5 Conclusions

Based on click logs with users’ behavior information, lots of click models are designed to improve ranking performance. Designed for large scale data, BBM has some trouble in application. In this paper, we present an auto-scaled BBM method, which can automatically decide the suitable scale of sessions and produce stable features such as mean, variance and so on. These features can be used for ranking algorithms such as ranking-svm or regression and so on. Experiments show that auto-scaled method solves the problem of original BBM and has a better performance.

Acknowledgements

This work was supported by Natural Science Foundation (60903107, 61073071) and National High Technology Research and Development (863) Program (2011AA01A205) of China.

References

- [1] Hwanjo Yu, Youngdae Kim, and Seungwon Hwang. 2009. RV-SVM: An Efficient Method for Learning Ranking SVM. In Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD '09), Thanaruk Theeramunkong, Boonserm Kijirikul, Nick Cercone, and Tu-Bao Ho (Eds.). Springer-Verlag, Berlin, Heidelberg, 426-438.
- [2] Zheng Ye, Jimmy Xiangji Huang, and Hongfei Lin. 2011. Incorporating rich features to boost information retrieval performance: A SVM-regression based re-ranking approach. *Expert Syst. Appl.* 38, 6 (June 2011), 7569-7574.
- [3] Linh Thai Nguyen, Wai Gen Yee, Roger Liew, and Ophir Frieder. 2010. Experiences with using SVM-based learning for multi-objective ranking. In Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM '10). ACM, New York, NY, USA, 1917-1920.
- [4] Tao Qin, Tie-Yan Liu, Xu-Dong Zhang, De-Sheng Wang, Wen-Ying Xiong, and Hang Li. 2008. Learning to rank relational objects and its application to web search. In Proceedings of the 17th international conference on World Wide Web (WWW '08). ACM, New York, NY, USA, 407-416.
- [5] Yan Pan, Haixia Luo, Hongrui Qi, and Yong Tang. 2011. Transductive learning to rank using association rules. *Expert Syst. Appl.* 38, 10 (September 2011), 12839-12844.
- [6] Bing Xu, Tie-Jun Zhao, Jian-Wei Wu, and Cong-Hui Zhu. 2011. Automatically ranking reviews based on the ordinal regression model. In Proceedings of the Third international conference on Artificial intelligence and computational intelligence - Volume Part III (AICI'11), Hepu Deng, Duoqian Miao, Jingsheng Lei, and Fu Lee Wang (Eds.), Vol. Part III. Springer-Verlag, Berlin, Heidelberg, 126-134.
- [7] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In Proceedings of the international conference on Web search and web data mining (WSDM '08). ACM, New York, NY, USA, 87-94.
- [8] Fan Guo, Chao Liu, and Yi Min Wang. 2009. Efficient multiple-click models in web search. In Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM '09), Ricardo Baeza-Yates, Paolo Boldi, Berthier Ribeiro-Neto, and B. Barla Cambazoglu (Eds.). ACM, New York, NY, USA, 124-131.
- [9] Georges E. Dupret and Benjamin Piwowarski. 2008. A user browsing model to predict search engine click data from past observations. In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '08). ACM, New York, NY, USA, 331-338.
- [10] Chao Liu, Fan Guo, and Christos Faloutsos. 2009. BBM: bayesian browsing model from petabyte-scale data. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '09). ACM, New York, NY, USA, 537-546.
- [11] Chao Liu, Fan Guo, and Christos Faloutsos. 2010. Bayesian Browsing Model: Exact Inference of Document Relevance from Petabyte-Scale Data. *ACM Trans. Knowl. Discov. Data* 4, 4, Article 19 (October 2010), 26 pages.
- [12] Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Wang, and Christos Faloutsos. 2009. Click chain model in web search. In Proceedings of the 18th international conference on World wide web (WWW '09). ACM, New York, NY, USA, 11-20.
- [13] Feimin Zhong, Dong Wang, Gang Wang, Weizhu Chen, Yuchen Zhang, Zheng Chen, and Haixun Wang. 2010. Incorporating post-click behaviors into a click model. In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '10). ACM, New York, NY, USA, 355-362.
- [14] Yiqun Liu, Min Zhang, Liyun Ru, and Shaoping Ma. 2006. Automatic query type identification based on click through information. In Proceedings of the Third Asia conference on Information Retrieval Technology (AIRS'06), Hwee Tou Ng, Mun-Kew Leong, Min-Yen Kan, and Donghong Ji (Eds.). Springer-Verlag, Berlin, Heidelberg, 593-600.