Enlister: Baidu's Recommender System for the Biggest Chinese Q&A Website

Qiwen Liu
Baidu Inc.
Beijing, China
liuqiwen@baidu.com

Tianjian Chen
Baidu Inc.
Beijing, China
chentianjian@baidu.com

Jing Cai
Baidu Inc.
Beijing, China
caijing@baidu.com

Dianhai Yu
Baidu Inc.
Beijing, China
yudianhai@baidu.com

ABSTRACT

In this paper, we describe the concept & design of a real-time question RS (recommender system), the Enlister project, for the biggest Chinese Q&A (Questions and Answers) website and evaluate its performance on massive data from this real-world practice.

We demonstrate how we weigh in among different recommendation algorithms and optimization methods. To enhance recommendation accuracy and handling time-sensitive questions, we propose a large scale real-time RS based on the combination of machine learning algorithms and the stream computing technology. Considering of algorithm flexibility and performance, we use the maximum entropy model as the fundamental model design in the CTR (click-through rate) prediction of recommendation items. In the perspective of the Enlister system architecture, we illustrate how we divide and conquer massive data processing problem with a novel stream computing design which reduces the data process latency down to seconds.

Finally we analyze the online test result and prove our design concept by achieving a series of significant improvements.

Categories and Subject Descriptors

H3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Performance

Keywords

Machine Learning, Maximum entropy, Stream computing

1. INTRODUCTION

1.1 Background

Baidu Inc. is a leading internet service provider in China. As of January 2012, it ranked 5th in the Alexa global rankings and the 1st in China. One of Baidu's most popular services is the Baidu Knows, which offers a Q&A platform to its users for knowledge and experience sharing.

Today, there are over 400 million questions on the Baidu Knows website, of which more than 170 million questions were answered by users. Around 100 million users search for answers every day and over 12 new questions are posted online every second.

As the biggest Q&A website in China, the Baidu Knows create an eco-system of knowledge sharing between the website's users.

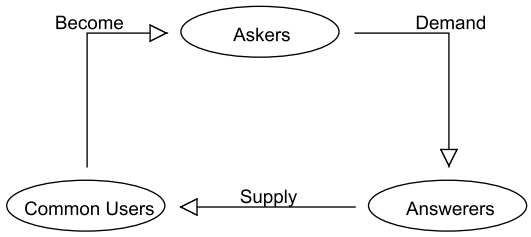

Figure 1: the Baidu Knows eco-system

As shown in Figure 1, typical common users just search in the Baidu Knows for answers with some related keywords. Some of these users will become the Askers to post new questions on the website. Then the users who know the answers to these questions post their answers back and become answerers in this community. As more questions have been answered, the search result quality of common users will be improved. Therefore, the fundamental concept of keeping this community healthy is stimulating the growth of answered questions.

As for the answerers, searching for unsolved questions is often too time-consuming. That's why a question RS is introduced to solve this problem by automatically presenting questions to potential answerers.

1.2 Application Description

We build an intelligent RS to provide the Baidu Knows users with questions that they may be willing to answer. This RS is also capable of tracking down the short-term variation of a user's intention.

After submitting a new answer to an unsolved question, a user will be redirected to a new web page where we put the question recommendation list there. The user could either click through one question to answer it or just leave this page if not interested. The quantitative analysis of the user reactions and recommender performance will be presented in Section 5.

1.3 Contributions

Our Contributions can be summarized as follows:

- We bring a large scale question RS, which is based on the machine learning technology, from fiction to a real internet service. Millions of users have benefitted from this approach.
- We illustrate that the stream computing technology could be very effective in the real-world RSs, which could address the time-sensitive issues explicitly.

RecSys'12, September 9–13, 2012, Dublin, Ireland.

- Inspired by the modern search engine ranking technology, we apply a machine learning based CTR prediction methodology to improve the recommendation accuracy.

For your convenience, we organize the remainder of this paper as follows. In Section 2, related work is introduced. The user model and the recommender algorithms are described in Section 3. In Section 4, we will have an overview of the Enlister system architecture. All the evaluations are placed in Section 5. Finally, we have a brief conclusion in Section 6.

2. RELATED WORK

The previous question RS of Baidu Knows is a typical content-based RS [1]. It models the relevance between a user and a question as the cosine similarity degree of the user's preference vector and the term vector extracted from the question. However, this approach has stuck in the mud of blurry optimization objective and wild computational complexity.

As in many other information retrieval [2] and search engine advertising application [3] [4], CTR prediction is introduced to improve the performance of the Q&A RSs [5] [6].

Compared with other RS that had applied CTR prediction, the Enlister has pushed the CTR prediction to the edge of real-time massive-data processing and has proven to be very successful in the real-time RS design.

3. ALGORITHM

3.1 Overview

In our new version of RS, known as the Enlister system, the CTR metric is introduced to characterize the correlation between users and questions. It presents a closer view to the users' experience than the relevance score we used before.

In order to make more accurate predictions, delicate user models are constructed based on the data we collected from the users. On one hand, we expect these models to disclose the nature of our users' choices of answering questions. On the other hand, the models have to be simple enough to accommodate massive calculation and industrial adoption. Based on the user models, the click prediction models are trained according to the user click history and certain features from the user models. With the trained click model, we can easily predict the probability of a specific user clicking on a particular question. After we aggregate probability prediction results and generate a new recommendation list, diversity adjustments are introduced to avoid the potential monotonous problem of the question recommendation and pursue some limited but pragmatic novelty.

3.2 User Model

In the Enlister system, user model is built on both long-term status of user attributes and short-term variation of user interest.

3.2.1 User Attributes

Some user attributes are the basic information of a user, such as age, gender, education, expertise and other tags that one labeled on himself.

3.2.2 User Interest

User interest is essential in the user model data structure. In the Enlister system, three aspects of interest descriptions are generated from three different semantic abstractions:

1) Interest Term Vector

   A vector contains weights of terms, which implicate the correlation between user and the term on sematic level.

2) Related Questions

   Questions are browsed or answered by a user.

3) Abstract Interest Vector

   A vector contains weights of terms, which implicate the correlation between a user and an abstract concept. With the help of PLSA [7] technology, a conceptual topic model is trained from all the questions and answers on the Baidu Knows website which contains millions of question-answer pairs. When a user browse or reply to a question, we can calculate the distribution of conceptual topics to that question by the conceptual topic model mentioned before. Finally, with miner normalization efforts, we could get the abstract interest vector of a user to all questions that he has visited.

3.3 Click Prediction

In the Enlister system, the click model that we created is a kind of probabilistic classification model. It is a binary classification model to calculate the probability of a sample belonging to a class.

We will introduce more details about the click prediction in the following sections, including: 1) Sample Collection; 2) Feature Selection; 3) Classifier Algorithm.

3.3.1 Sample Collection

The samples are collected from the real online log of the original version RS. The positive samples are the questions that the user had examined and clicked. However, the negative samples are not questions that are unvisited by the user, but the questions we randomly choose from the question pool. As besides having no interest, there are many reasons may explain why people do not click on a question in the list. Our offline questionnaire investigation shows that most people did not click on a recommended question because they just did not notice it. Thus the probability that the randomly chosen questions coincide with the user's interest can be negligible, considering the huge size of our question pool.

3.3.2 Feature Selection

From our previous investigations, two factors are essential in the user's decision of following a question. One is the user's attributes. The other is the correlation degree between the user interest and a question.

3.3.2.1 User Attributes

As has been mentioned in Section 3.2.1, user attributes suggest the user's preference in the Q&A community. The basic user attributes contribute to many low level features in our model. For example, a woman who is over 30 with a PhD degree may have a tendency to receive more question recommendations than others who don't have any expertise. Other features from the statistics are also useful in prediction, such as the participation history of a user in the Q&A community. We have observed that the more active a user is, the more likely he will accept the recommended questions.

3.3.2.2 Correlation Degree

The major purpose of the user interest model, which we have described in Section 3.2.2, is to support the calculation of sematic correlation degree between user interest description and an input question. The correlation degree is a combined similarity degree of many natural language aspects consist to the user interest model. For the utilization of the interest term vector, we calculate the number of matched terms, cosine similarity and bm25 between the user interest term vectors and question vectors. Then the

matched terms and the semantic similarity are measured between the input question and the related questions in a user's model from both the term angle and the topic angle.

3.3.3 *Classification Algorithm*

In the Enlister design, two principles are crucial in the classifier selection. First, the classifier should be capable of performing a probabilistic classification, as we need not only the class label, but also the confidence degrees. Second, it needs to be a (generalized) linear classifier, which meets the requirement of online system response latency. Based on these two principles, we choose maximum entropy classifier. The probability P of a user u will click the question q can be calculated as follows:

$$P(c=1\mid u,q)=\frac{\exp(\sum_{i\in I}\partial_i P_i(c=1\mid u,q)-\partial_0)}{1+\exp(\sum_{i\in I}\partial_i P_i(c=1\mid u,q)-\partial_0)} \quad (3.1)$$

With $\partial_i\ (i\in I)$ are the features.

The global optimization solution is maximizing a logarithmic posteriori against the training set by some optimization methods, such as limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) and Stochastic Gradient Descent (SGD). We have tried these two methods in our experiment respectively. The result shows that the L-BFGS optimization is slightly better in this application scenario. After the model training process, we apply 10-fold cross validation to test it. The evaluation criteria include precision, recall etc. More details will be discussed in section 5.

3.4 Diversity Adjustment

Based on previous user research and eye-ball tests, we found the head part and the tail part of a list garnered most attention from the users.

In the Enlister, we use different filter algorithms for the head and the tail. For the head part, we apply a loose filtering algorithm, which only deletes some apparent duplication in the list. While for the tail part, we use a strict filtering algorithm to take out any questions that have noticeable semantic level similarity to each other in the list. In this way, the measurement leads to the novel appearance.

4. SYSTEM SETUP

The most important concept in the Enlister system design is real-time CTR prediction. The major data process can be described as follows.

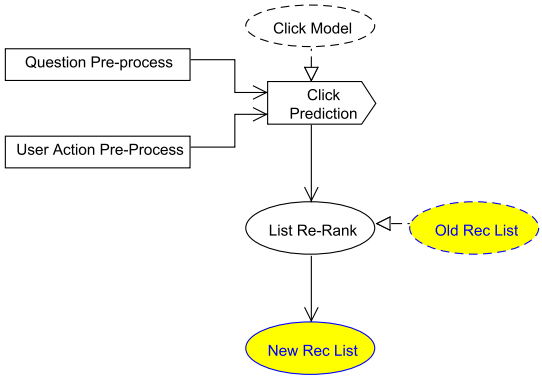

Figure 2: Stream computing model of Enlister

As shown in the Figure 2, the whole process is divided into 3 stages. The first stage includes 2 sections, one of which is used to extract the features of input questions and the other is used to generate user model from user actions in Baidu Knows online service system. The second stage is the CTR prediction where the user-action features, the input question features and the pre-trained click model are arranged together. With all the input parameters, we can calculate the probability of the user being interested in answering the input question. The last stage is the re-rank section. This is where the functionality of the list padding and diversity adjustment algorithm occurs in the whole data process.

For building the data processing flow, we construct multiple logic queues between processing nodes. The processing nodes are grouped into several node groups. Each group represents a simple logic section in the Figure 3, such as the pre-process section and the prediction section. If any group shows sign of lacking processing ability, we could just add nodes to that node group to solve the scalability problem [8].

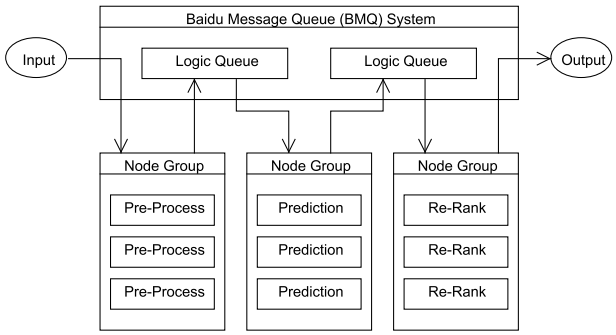

Figure 3: System architecture of Enlister

5. EXPERIMENT & EVALUATION

5.1 Evaluation Metrics

To evaluate the CTR model, we apply standard metrics in our experiments, such as precision, recall and accuracy.

As for a real online RS, users' reaction to the recommendation is a more convincing metric.

5.2 Experiment

5.2.1 *Sample Selection*

100,000 questions that had been viewed and clicked by users are selected from users' logs as positive sample, which involves 10 thousands users with 10 records per user on average. The negative samples could be built with two different types of data sources, the weak negative or the random negative. The weak negative samples are questions from real recommendation lists that are never clicked by users. The random negative samples are questions that chosen randomly from a large question pool. The data in Table 2 gives us the evaluation of different ways of negative samples selection. It is clear that using the random negative samples in training model is better in this case.

Table 2: Performance on different negative sample sets

	Accuracy	*Precision*	*Recall*
Weak negative sample	59.54%	64.16%	55.45%
Random negative sample	82.97%	91.73%	72.46%

5.2.2 *Sample Proportion*

The ratio of the positive and negative samples proportion is another issue to be settled. To find the proper ratio, we trained the model on datasets with different sample proportion settings. The results in Table 3 indicate that while the proportion ratio of

positive and negative samples is 50% to 50%, a better classifier performance is produced.

Table 3: Performance on different sample proportions

Proportion	*Accuracy*	*Precision*	*Recalls*
90%:10%	51.03%	50.52%	99.87%
70%:30%	87.58%	91.14%	84.35%
50%:50%	88.14%	95.07%	80.44%
30%:70%	86.41%	96.84%	75.28%
10%:90%	81.93%	98.37%	65.11%

5.2.3 Optimization Algorithm

As referred in Section 3, SGD and LBFGS algorithms are considered as the optimization algorithm in the maximum entropy model training. The data in Table 4 shows LBFGS is slightly better than SGD in this application. As a result, we choose the LBFGS algorithms for the online evaluation.

Table 4: Performance optimizations on a small dataset

	Accuracy	*Precision*	*Recall*
LBFGS	88.08%	95.00%	81.93%
SGD	87.23%	93.42%	78.60%

5.3 Online Evaluation

Enlister was released to the Baidu Knows users and an online evaluation was conducted from Feb. 11th, 2012.

Table 5: Users' reaction to the recommendation

	Previous RS	*Enlister*	*Increase*	*Promotion*
Click	926875	1499940	573065	61.83%
Answer	129229	174622	45393	35.13%

In Table 5, the number of clicks/answers to recommended questions is given. As can be seen from the table, the total number of clicks/answers on the questions from the Enlister RS is increased by a big margin compared with the previous RS.

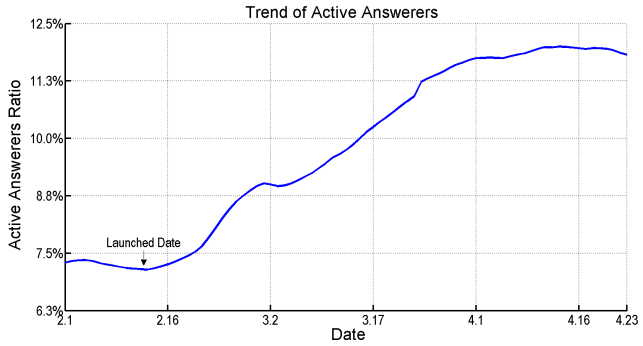

Figure 6: Trend of the active answerers after Enlister was launched

Essentially, as shown in Figure 6, the promotion of the active answerer number suggests a much favorable user experience, where the active answerer is defined as one who answered at least one question in the past 30 days.

6. CONCLUSION

In the Enlister project, we have successfully built a real-time RS that serves millions of users every day. The evaluation data shows that the algorithm and system design fit the recommendation scenario quite well. Great improvement had been made on the accuracy and time-sensitive issues. The number of active users had grown substantially after the system was officially launched.

The result data just illustrates the efficiency and performance of our current solution. Still, there are a lot of other recommendation algorithms and optimization methods to be considered in our future improvement. Two potential aspects are the timing of recommendation and the utilization of relationships between users.

Finally, we hope our work will inspire and encourage more people to build large scale recommender systems for helping users retrieve useful information.

7. ACKNOWLEDGMENTS

The authors would like to thank all the colleagues in Baidu Inc. who contributed to the Enlister project in various ways, especially Hao Tian, Jian Xian, Junyu Cai, Kai Chai, Xin Sun.

The authors are grateful to Dr. Evan Xiang for his comments on the early draft of this paper.

8. REFERENCES

[1] Michael J. Pazzani and Daniel Billsus. 2007. *Collaborative Filtering Recommender System*. Lecture Notes in Computer Science, 2007, volume 4321/2007, 291-324, DOI: 10.1007/978-3-540-72079-9_9

[2] Olivier Chapelle , Ya Zhang, *A dynamic bayesian network click model for web search ranking*, Proceedings of the 18th international conference on World wide web, April 20-24, 2009, Madrid, Spain

[3] M. Regelson and D. Fain. *Predicting click-through rate using keyword clusters*. Proceedings of the Second Workshop on Sponsored Search Auctions, 2006.

[4] Matthew Richardson , Ewa Dominowska , Robert Ragno, *Predicting clicks: estimating the click-through rate for new ads*, Proceedings of the 16th international conference on World Wide Web, May 08-12, 2007, Banff, Alberta, Canada [doi>10.1145/1242572.1242643]

[5] Xin Jin, Yanzan Zhou , Bamshad Mobasher. *A maximum entropy web recommendation system: combining collaborative and content features*. Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, August 21-24, 2005, Chicago, Illinois, USA [doi>10.1145/1081870.1081945]

[6] Yutaka Kabutoya, Tomoharu Iwata, Hisako Shiohara, Ko Fujimura. *Effective Question Recommendation using Multiple Features for Question Answering Communities*. IPSJ Transaction on Database (TOD), Vol.3, No. 4, 34-47, 2010

[7] Thomas Hofmann. *Probabilistic latent semantic indexing*. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*.50-57, August 15-19, 1999, Berkeley, California, United States [doi>10.1145/312624.312649]

[8] M. Stonebraker, U. Cetintemel, and S. Zdonik: *The 8 requirements of real-time stream processing*, SIGMOD, 2005, Baltimore, Maryland, USA