Experiences with Using SVM-based Learning for Multi-Objective Ranking

Linh Thai Nguyen, Wai Gen Yee, Roger Liew
Technology Group
Orbitz Worldwide
Chicago, IL 60661
{lnguyen, rliew, wyee}@orbitz.com

Ophir Frieder
Department of Computer Science
Georgetown University
Washington, DC 20057
ophir@cs.georgetown.edu

ABSTRACT

We describe our experiences in applying learning-to-rank techniques to improving the quality of search results of an online hotel reservation system. The search result quality factors we use are average booking position and distribution of margin in top-ranked results. (We expect that total revenue will increase with these factors.) Our application of the SVMRank technique improves booking position by up to 25% and margin distribution by up to 14%.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services – *commercial services*, *web-based services*. H.2.8 [**Database Management**]: Database Applications – *data mining*.

General Terms

Algorithms, Performance, Economics, Experimentation.

Keywords

Learning to rank, retail search engine, revenue maximization, multiple objectives, hotel reservations.

1. INTRODUCTION

Unlike traditional search engines, which are designed to return the results that are most relevant to a query, retail search engines are designed to maximize revenue. This goal is realized by improving revenue per conversion as well as conversion rate.

Our goal is to identify results that are both relevant to a query and yield high margin. To this end, we apply a "learning to rank" (LTR) technique and adapt it to consider margin when producing the final ranked result set. The challenge is that the goals of query relevance and margin maximization are often mutually exclusive. Improving on one metric at the expense of the other may have a significant negative impact on total revenue.

Our data set is a click log from Orbitz's hotel search engine for the Chicago and New York markets. We use an adapted version of SVMRank [5] as our LTR technique. Offline experimental results show that, with some tuning, SVMRank has the potential to improve both relevance and margin in search results.

2. RELATED WORK

Liu describes three types of LTR techniques based on their types of training data [8]. *Point-wise* techniques require a relevance judgment for each result in a training set [3][7]. *Pair-wise* techniques require a relative relevance judgment for each pair of results [1][5]. *List-wise* techniques require relevance judgments for a list of results [2][9].

Among the many LTR techniques, we adopt the pair-wise SVMRank for several reasons. First, it is easy and cheap to collect pair-wise training data that represents customer relative preferences. (Building training data for either point-wise or list-wise techniques is much less practical.) Second, SVMRank is efficient in both learning and ranking, as it is a linear SVM. Other pair-wise LTR techniques, such as those based on neural networks, have a higher learning cost [1]. Third, SVMRank is guaranteed to find the global optimum, unlike other, non-convex techniques, such as ones based on neural networks. Finally, SVMRank results in a linear scoring function, which is easy to understand and tune. Complex, nonlinear functions may also over-fit training data and perform poorly when deployed.

3. MODEL

A query is defined by a location and a date range and retrieves a ranked list of results. Each result is a hotel that matches the query criteria and is defined by features, including:

- Price – The nightly room rate.
- Star rating – The star rating.
- Distance – The distance from the city center.
- Margin – The profit earned by the seller of the hotel.

We consider the first three features (among others) to be "buyer preferences" or "buyer features" because their values directly affect the result's relevance to the user. Margin, in contrast, is a "seller preference" or "seller feature" because its value is of greatest concern to the seller. (Note that seller features are generally not exposed to users.)

We assume that all features' values are fixed. We thus do not consider the case where prices and margins can be tuned based on some special knowledge of the market.

A result set is a ranked list of hotel information retrieved by a user query. Each time a hotel is displayed in an impression list, it is known as an *impression* for the hotel. Each time the user books a hotel, it is known as a *conversion* or a *booking*. We refer to the rank of the booked hotel in a result set as the *booking position* of the result set. The *conversion rate of a hotel* is the ratio of the number of conversions to the number of impressions for the hotel. The overall *conversion rate* is the ratio of the number of conversions to the number of result sets.

Margin (or *revenue*) is the profit margin associated with the booking of a hotel. The *margin distribution* of a result set is the sum of the margins of the top *N* results in the result set.

Total revenue is total margin of all bookings. *Average margin* is total revenue divided by the number of bookings.

Impression data or *data set* refers to a set of result sets and their booking positions, excluding result sets with no bookings.

Our goal is to maximize total revenue. To do this, our approach must increase one or both of conversion rate and average margin.

We assume that conversion rate increases with the quality of the query results in terms of relevance. We also assume that users have a strong bias toward clicking (and, hence, booking) the top-ranked results. ("Position bias" is a well-known and accepted phenomenon (e.g., [4]) and also occurs in our data.)

We measure the ability of our ranking function to rank relevant results highly by average booking position and measure the ability of our ranking function to rank high-margin results highly by margin distribution. Hence, if we are able to improve either average booking position or margin distribution (or both) without hurting either, we expect that total revenue will increase. Although we do not quantify the change in revenue, we believe that there is a direct relationship between total revenue and improvement in either booking position or margin distribution.

4. PERFORMANCE EVALUATION

4.1 BASELINE RANKING APPROACH

For base case performance, we use a standard ranking function, where each hotel *j* is scored as a weighted sum of feature values:

$$S^i_{basic} = \sum_{j=1}^{n} w_j f^i_j \tag{1}$$

In the formula above, f^i_j is the value of hotel *i*'s feature *j* ($1 \leq j \leq n$) and w_j is the weight of this feature. Initial weights are manually tuned by a system administrator based on business goals (i.e., revenue maximization) and empirical evidence. Each result receives a score computed by this formula, and results are sorted in descending order of score. As will be shown, we achieve our goals to varying degrees by tuning these weights.

4.2 LEARNING TO RANK WITH SVM

Our training data consists of a random subset of Orbitz impression data from the last quarter of 2009 for the Chicago and New York markets – consisting of thousands of bookings. For each impression, we generate pair-wise *preference rules*, where a booked result is defined as being more relevant than others (i.e., unbooked results) in the result set. We apply SVMRank [5] to the impression data to yield weights for the ranking function described above in Equation 1.

We define as a *positive rule* any rule that relates the booked result to lower-ranked results. Simiarlly, we define as a *negative rule* any rule that relates the booked hotel to higher-ranked results. All negative rules are considered *rule violations*. The goal of SVMRank is to generate a model that minimizes the number of rule violations in the training set.

4.3 Metrics

Let $rank_i$ be the rank of the booked hotel in result set *i*. Let N^T be the number of result sets in the impression data. For each experiment, we use the following metrics to measure result quality in terms of relevance:

- Average booking position (ABP) – This is the average rank position of the booked hotel.

$$ABP = \frac{1}{N^T} \sum_{i=1}^{N^T} rank_i \tag{2}$$

- Mean reciprocal rank (MRR) – We measure the average of the inverse of the rank positions of the booked hotels.

$$MRR = \frac{1}{N^T} \sum_{i=1}^{N^T} \frac{1}{rank_i} \tag{3}$$

Ideally, ABP is as low as possible, with a minimum value of 1. MRR is ideally as high as possible, with a maximum value of 1.

ABP gives an unbiased view of result quality in the sense that it measures changes in booking position anywhere in a ranked list (unlike MRR). However, MRR captures the fact that users have a strong preference for higher-ranked results in practice.

We report *margin@N* for each impression set for *N* = 5 and *N* = 10 (denoted *m@*5 and *m@*10). As stated above, ABP, MRR, and margin are assumed to be related to conversion rate and average margin.

We report the percentage change in these metrics between baseline results (see Section 4.1) and experimental results. For consistency, we use positive percentage changes to indicate positive results. Hence, a *positive* percentage change in ABP actually refers to a *decrease* in rank value (i.e., closer to the top).

We use 10-fold cross validation to tune parameters.

4.4 Parameter Tuning and Results for User Preference Optimization

With SVMRank, there are several parameters to tune (e.g., parameters that control for over-fitting). We use standard parameter exploration techniques [10] and pick the parameters that minimize the following expression:

$$\frac{\log(1+pctimprove(ABP))}{2} + \frac{\log(1+pctimprove(MRR))}{2} + \log(1+pctimprove(m@5)) \tag{4}$$

In this expression, *pctimprove*() represents the improvement for the respective metric for a set of parameters. We use both MRR and ABP to represent the dual goals of improving ranking quality near the top of the list of results and improving overall ranking quality. The *margin*@5 is included to find high-margin solutions and is given an equal contribution as both ABP and MRR.

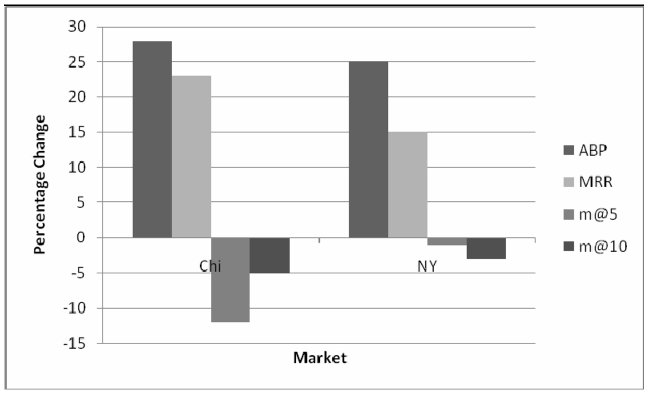

Figure 1. Results for buyer preferences.

SVMRank, by design, minimizes the number of rule violations (see Section 4.2) [6], which only indirectly satisfies our goals. Hence, we apply this additional objective. Similarly, we test a variety of numbers of positive and negative rules and pick the number that gives the best overall performance during training.

After applying SVMRank to our data set to the impression data, we were able to improve average booking position and MRR as shown in Figure 1. ABP (MRR) improves by at least 25% (15%) in both markets. The fact that both ABP and MRR improve suggest that the improvement occurred over all ranking positions.

Unfortunately, these improvements come at a price. *Margin@N* decreases in both markets. Chicago's *margin@5* decreased by 12%, while New York's decreased by 1%. This shows that users prefer less expensive hotels (there was also a drop in "price@*N*" on the order of 15%), which generally yield lower margins.

The large improvement in ABP and MRR are good signs. However, the negative values in the margin distributions in these results makes uncertain the effect of this model on total revenue, as discussed above. Hence, we search for a better solution with – one with no negative results.

4.5 CONSIDERING MARGIN

4.5.1 Weighted Sum of Buyer and Seller Features

Due to the large improvement in buyer preferences with the basic SVMRank results, we decided to trade off ABP/MRR performance for improved *margin@N* performance by varying the weight assigned to margin, f_M, using parameter α:

$$S_b^i = \alpha \sum\nolimits_{j \neq M} w_j f_j^i + (1-\alpha) f_M^i \qquad (5)$$

We pick the α value that maximizes Equation 4 and show the results in Figure 2.

We were able to improve margin distribution with a drop ranking quality (e.g., for Chicago, the percentage change in ABP drops from 28% to 23%). However, ranking quality is still positive compared with the base case. For Chicago, ranking quality is still good with 23% and 3% scores for ABP and MRR, respectively. *Margin@*5 and *margin@*10 are also both positive, but small at about 3% for both.

4.5.2 SVMRank Rules Engineering – Rule Deletion

We also tried improving *margin@N* performance by a process we refer to as *rules engineering* – by replicating or removing preference rules based on our objectives.

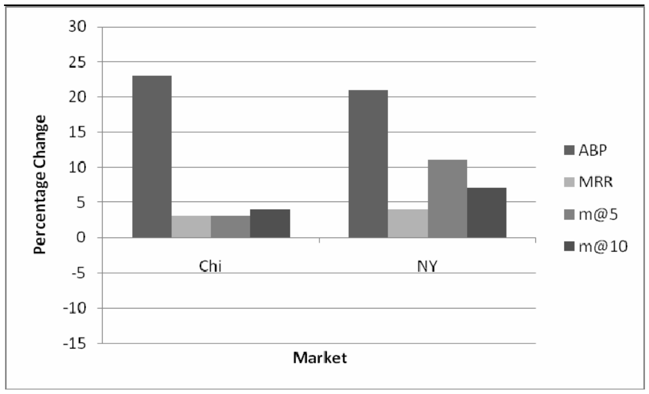

Figure 2. Weighted sum of buyer and seller preferences.

With *rule deletion*, we first create a set of input rules for SVMRank as described in Sections 4.4. We then delete a subset of these rules based on the following criterion:

Let *R* be a rule between results r_i and r_j: $r_i > r_j$. Let margin(r_i) denote the margin value of the hotel associated with r_i.

Keep *R* only if margin(r_i) > (1 - β)margin(r_j).

Hence, if β = 1, we keep all rules, and if β = 0, we only keep the rules that rank the results in descending order of margin. Note that β = 0 does not necessarily result in a ranking based on margin because of the contribution of the other features to the scoring.

4.5.3 SVMRank Rules Engineering – Rule Addition

With rule addition, we probabilistically replicate rules that reinforce the seller's preferences if the following criterion holds:

Let *R* be a rule in the rule set between results r_i and r_j: $r_i > r_j$. With probability 1 - β:

```
If margin(r_i) > margin(r_j),
  add an additional instance of R to the rule set.
Else,
  add an instance of ¬R (r_j > r_i) to the rule set.
```

The goal of rule addition is to reinforce rules that satisfy the margin requirement and to weaken rules that violate it.
Again, if β = 1, this technique reduces to the base case rule set. If β = 0, all eligible additional rules are added to the rule set to weaken margin violations.

4.5.4 Results

For our experimental results, we tune α from Equation 5, β from above to values that maximize Equation 4.

As shown in Figure 3, our results for rules engineering by deletion have mixed results. For the Chicago market, ABP and MRR improve by 23% and 14%, respectively. *Margin@*5, however decreases by 3%, while *margin@*10 increases by 1%.

The New York results are better overall. New York has ABP and MRR scores similar to those of Chicago at 24% and 10%, respectively, but its *margin@*5 and *margin@*10 are significantly better at 10% and 3%, respectively.

As shown in Figure 4, the results for rule addition are more consistent than those for rule deletion. Although the improvements to ABP and MRR are lower (there is no change in MRR for New York), the improvements in margin distribution are much greater than with rule deletion.

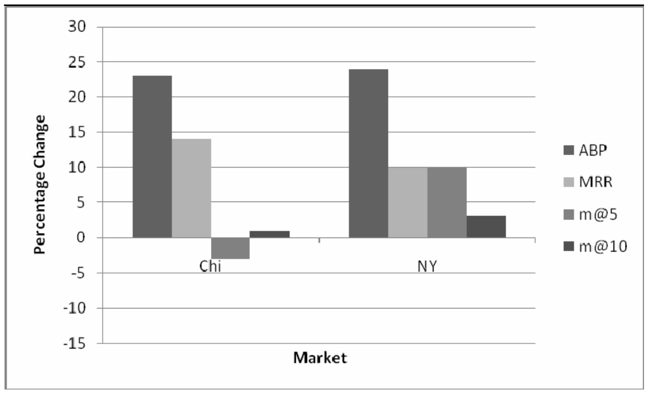

Figure 3. Rule deletion.

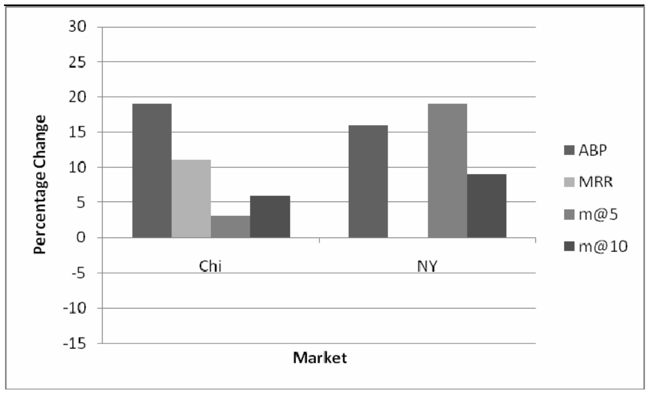

Figure 4. Rule addition.

Overall, our techniques improve on ABP more than MRR. This is likely a sign of how difficult it is to improve the rank position of results already ranked near the top of a list. Margin is also difficult to improve on, likely due to users' preferences for more economical hotel rooms. Fortunately, the data strongly suggests that it is possible to improve rankings in a way that improves total revenue.

5. DISCUSSION

Judging from the results above, the ordering of the ranking techniques from most to least effective is:

1. Rules addition
2. Weighted sum
3. Rules deletion

We quantify the ordering of these three results by their scores with Equation 4.

The fact that the rules engineering results performed both best and worst suggests the importance of picking the right set of preference rules for training SVMRank. Intuitively, to the learning process, rules addition makes more data available, while rules deletion does the reverse. This may explain their overall performances. Also, the superior performance of rules engineering suggests the importance of building a good model before fine-tune performance using techniques like weighted sum.

In practice, both rules engineering and tuning the contribution of margin are important to retail search engine performance. Rules engineering will boost expected performance offline and tuning the weight of margin to rank score allows control over total revenue.

6. CONCLUSION

Our goal is to share our experiences with learning how to rank to satisfy both buyer and seller objectives using SVMRank on real hotel booking data. Our business objective is to maximize revenue and our approach is to improve ranking quality and distribution of margin over the top-ranked results.

We try three techniques: a weighted sum score, rules engineering by rule deletion, and rules engineering by rule addition.

Our results indicate that it is possible to improve both ranking quality and margin distribution. In Chicago, for example, we were able to improve average booking position by 19% and margin@5 by 3%. To this end, rules addition was the most effective technique, giving us the idea that fine-grained approaches may be a better way of approaching this problem.

For future work, we will consider other ways of creating and selecting rules for input into the learning system. We will also explore ways of creating a pairwise LTR technique that directly optimizes for the objective of total revenue.

7. REFERENCES

[1] Burges, C. J. C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. 2005. Learning to Rank Using Gradient Descent. In *Proc. ICML*, 2005.

[2] Burges, C. J. C., Ragno, R., and Le, Q. V. 2007. Learning to Rank with Nonsmooth Cost Functions. Advances in Neural Information Processing Systems 19, 2007.

[3] Crammer, K. and Singer, Y. Pranking with Ranking. Advances in Neural Information Processing Systems 14, 2001.

[4] Agichtein, E, Brill, E., Dumais, S., and Ragno, R. 2006. Learning user interaction models for predicting web search result preferences. In *Proc. ACM SIGIR*, 2006.

[5] Joachims, T. 2002. Optimizing search engines using clickthrough data. In *Proc. ACM SIGKDD*. 2002.

[6] Joachims, T. 2005. A Support Vector Method for Multivariate Performance Measures. Reasoning about naming systems. In *Proc. ICML*, 2005.

[7] Li, P., Burges, C. J. C., and Wu, Q. 2008. Learning to Rank Using Classification and Gradient Boosting. Advances in Neural Information Processing Systems 20, 2008.

[8] Liu, T.-Y. 2009. Learning to Rank for Information Retrieval. Foundations and Trends in Information Retrieval, Vol. 3, No. 3, pp. 225-331, 2009.

[9] Moon, T., Smola, A., Chang, Y. and Zheng, Z. 2010. IntervalRank – Isotonic Regression with Listwise and Pairwise Constraints. In *Proc. WSDM*, 2010.

[10] Witten, I. and Franke, E. 2005. Data Mining: Practical Machine Learning Tools and Techniques. 2nd ed. Morgan Kaufmann. June, 2005.