

Neural Choice by Elimination via Highway Networks

Truyen Tran^(✉), Dinh Phung, and Svetha Venkatesh

Centre for Pattern Recognition and Data Analytics, Deakin University,
Geelong, Australia
`truyen.tran@deakin.edu.au`

Abstract. We introduce *Neural Choice by Elimination*, a new framework that integrates deep neural networks into probabilistic sequential choice models for learning to rank. Given a set of items to choose from, the elimination strategy starts with the whole item set and iteratively eliminates the least worthy item in the remaining subset. We prove that the choice by elimination is equivalent to marginalizing out the random Gompertz latent utilities. Coupled with the choice model is the recently introduced Neural Highway Networks for approximating arbitrarily complex rank functions. We evaluate the proposed framework on a large-scale public dataset with over 425K items, drawn from the Yahoo! learning to rank challenge. It is demonstrated that the proposed method is competitive against state-of-the-art learning to rank methods.

1 Introduction

People often rank options when making choice. Ranking is central in many social and individual contexts, ranging from election [16], sports [17], information retrieval [22], question answering [1], to recommender systems [32]. We focus on a setting known as learning to rank (L2R) in which the system learns to choose and rank items (e.g. a set of documents, potential answers, or shopping items) in response to a query (e.g., keywords, a question, or an user).

Two main elements of a L2R system are rank model and rank function. One of the most promising rank models is listwise [22] where all items responding to the query are considered simultaneously. Most existing work in L2R focuses on designing listwise rank losses rather than formal models of *choice*, a crucial aspect of building preference-aware applications. One of a few exceptions is the Plackett-Luce model which originates from Luce’s axioms of choice [24] and is later used in the context of L2R under the name ListMLE [34]. The Plackett-Luce model offers a natural interpretation of making sequential choices. First, the probability of choosing an item is proportional to its worth. Second, once the most probable item is chosen, the next item will be picked from the remaining items in the same fashion.

However, the Plackett-Luce model suffers from two drawbacks. First, the model spends effort to separate items down the rank list, while only the first few items are usually important in practice. Thus the effort in making the right

ordering should be spent on more important items. Second, the Plackett-Luce is inadequate in explaining many competitive situations, e.g., sport tournaments and buying preferences, where the ranking process is reversed – worst items are eliminated first [33].

Addressing these drawbacks, we introduce a probabilistic sequential rank model termed *choice by elimination*. At each step, we remove one item and repeat until no item is left. The rank of items is then the reverse of the elimination order. This elimination process has an important property: Near the end of the process, only best items compete against the others. This is unlike the selection process in Plackett-Luce, where the best items are contrasted against all other alternatives. We may face difficulty in separating items of similarly high quality but can ignore irrelevant items effortlessly. The elimination model thus reflects more effort in ranking worthy items.

Once the ranking model has been specified, the next step in L2R is to design a rank function $f(x)$ of query-specific item attributes x [22]. We leverage the newly introduced *highway networks* [28] as a rank function approximator. Highway networks are a compact deep neural network architecture that enables passing information and gradient through hundreds of hidden layers between input features and the function output. The highway networks coupled with the proposed elimination model constitute a new framework termed *Neural Choice by Elimination (NCE)* illustrated in Fig. 1.

The framework is an alternative to the current state-of-the-arts in L2R which involve tree ensembles [4, 5, 12] trained with hand-crafted metric-aware

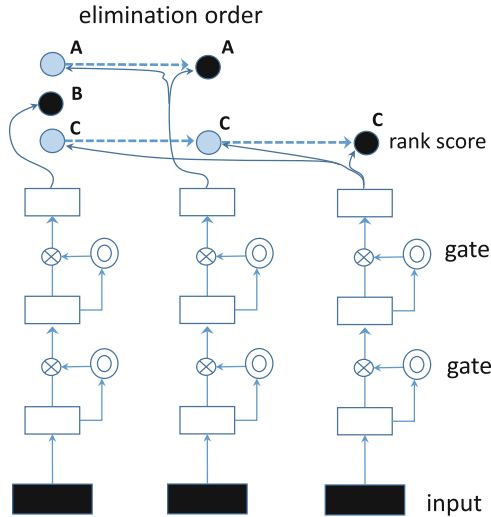


Fig. 1. *Neural choice by elimination* with 4-layer highway networks for ranking three items (A,B,C). Empty boxes represent hidden layers that share the same parameters (hence, recurrent). Double circles are gate that controls information flow. Dark filled circles represent a chosen item at each step – here the elimination order is (B,A,C).

losses [7, 21]. Unlike the tree ensembles where typically hundreds of trees are maintained, highway networks can be trained with *dropouts* [27] to produce an implicit ensemble with only one thin network. Hence we aim to establish that *deep neural networks are competitive in L2R*. While shallow neural networks have been used in ranking before [6], they were outperformed by tree ensembles [21]. Deep neural nets are compact and more powerful [3], but they have not been measured against tree-based ensembles for generic L2R problems. We empirically demonstrate the effectiveness of the proposed ideas on a large-scale public dataset from Yahoo! L2R challenge with totally 18.4 thousands queries and 425 thousands documents.

To summarize, our paper makes the following contributions: (i) introducing a new neural sequential choice model for learning to rank; and (ii) establishing that deep nets are scalable and competitive as rank function approximator in large-scale settings.

2 Background

2.1 Related Work

The elimination process has been found in multiple competitive situations such as multiple round contests and buying decisions [14]. *Choice by elimination* of distractors has been long studied in the psychological literature, since the pioneer work of Tversky [33]. These backward elimination models may offer better explanation than the forward selection when eliminating aspects are available [33]. However, existing studies are mostly on selecting a single best choice. Multiple sequential eliminations are much less studied [2]. Second, most prior work has been evaluated on a handful of items with several attributes, whereas we consider hundreds of thousands of items with thousands of attributes. Third, the cross-field connection with data mining has not been made. The link between choice models and Random Utility Theory has been well-studied since Thurstone in the 1920s, and is still an active topic [2, 30, 31]. Deep neural networks for L2R have been studied in the last two years [10, 11, 18, 25, 26]. Our work contributes a formal reasoning of human choices together with a newly introduced highway networks which are validated on large-scale public datasets against state-of-the-art methods.

2.2 Plackett-Luce

We now review Plackett-Luce model [24], a forward selection method in learning to rank, also known in the L2R literature as ListMLE [34]. Given a query and a set of response items $\mathcal{I} = (1, 2, \dots, N)$, the rank choice is an ordering of items $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$, where π_i is the index of item at rank i . For simplicity, assume that each item π_i is associated with a set of attributes, denoted as $x_{\pi_i} \in \mathbb{R}^p$. A rank function $f(x_{\pi_i})$ is defined on π_i and is independent of other items. We aim to characterize the rank permutation model $P(\boldsymbol{\pi})$.

Let us start from the classic probabilistic theory that any joint distribution of N variables can be factorized according to the chain-rule as follows

$$P(\boldsymbol{\pi}) = P(\pi_1) \prod_{i=2}^N P(\pi_i \mid \boldsymbol{\pi}_{1:i-1}) \quad (1)$$

where $\boldsymbol{\pi}_{1:i-1}$ is a shorthand for $(\pi_1, \pi_2, \dots, \pi_{i-1})$, and $P(\pi_i \mid \boldsymbol{\pi}_{1:i-1})$ is the probability that item π_i has rank i given all existing higher ranks $1, \dots, i-1$. The factorization can be interpreted as follows: choose the first item in the list with probability of $P(\pi_1)$, and choose the second item from the remaining items with probability of $P(\pi_2 \mid \pi_1)$, and so on. Luce's *axioms of choice* assert that an item is chosen with probability proportional to its *worth*. This translates to the following choice model:

$$P(\pi_i \mid \boldsymbol{\pi}_{1:i-1}) = \frac{\exp(f(x_{\pi_i}))}{\sum_{j=i}^N \exp(f(x_{\pi_j}))}$$

Learning using maximizing likelihood minimizes the log-loss:

$$\ell_1(\boldsymbol{\pi}) = \sum_{i=1}^{N-1} \left(-f(x_{\pi_i}) + \log \sum_{j=i}^N \exp(f(x_{\pi_j})) \right) \quad (2)$$

3 Choice by Elimination

We note that the factorization in Eq. (1) is not unique. If we permute the indices of items, the factorization still holds. Here we derive a *reverse* Plackett-Luce model as follows

$$P(\boldsymbol{\pi}) = Q(\pi_N) \prod_{i=1}^{N-1} Q(\pi_i \mid \boldsymbol{\pi}_{i+1:N}) \quad (3)$$

where $Q(\pi_i \mid \boldsymbol{\pi}_{i+1:N})$ is the probability that item π_i receives rank i given all existing lower ranks $i+1, i+2, \dots, N$.

Since π_N is the most irrelevant item in the list, $Q(\pi_N)$ can be considered as the probability of *eliminating* the item. Thus the entire process is backward elimination: The next irrelevant item π_k is eliminated, given that more extraneous items ($\boldsymbol{\pi}_{i>k}$) have already been eliminated. It is reasonable to assume that *the probability of an item being eliminated is inversely proportional to its worth*. This suggests the following specification

$$Q(\pi_i \mid \boldsymbol{\pi}_{i+1:N}) = \frac{\exp(-f(x_{\pi_i}))}{\sum_{j=1}^i \exp(-f(x_{\pi_j}))}. \quad (4)$$

Note that, due to specific choices of conditional distributions, distributions in Eqs. (1, 3) are generally not the same. With this model, the log-loss has the following form:

$$\ell_2(\boldsymbol{\pi}) = \sum_{i=1}^N \left(f(x_{\pi_i}) + \log \sum_{j=1}^i \exp(-f(x_{\pi_j})) \right) \quad (5)$$

3.1 Derivation Using Random Utility Theory

Random Utility Theory [2, 29] offers an alternative that explains the ordering of items. Assume that there exists latent utilities $\{u_i\}$, one per item $\{\pi_i\}$. The ordering $P^*(\boldsymbol{\pi})$ is defined as $\Pr(u_1 \geq u_2 \geq \dots \geq u_N)$. Here we show that it is linked to *Gompertz distribution*. Let $u_j \geq 0$ denote the latent random utility of item π_j . Let $v_j = e^{bu_j}$, the Gompertz distribution has the PDF $P_j(u_j) = b\eta_j v_j \exp(-\eta_j v_j + \eta_j)$ and the CDF $F_j(u_j) = 1 - \exp(-\eta_j(v_j - 1))$, where $b > 0$ is the scale and $\eta > 0$ is the shape parameter.

At rank i , choosing the worst item π_i translates to ensuring $u_i \leq u_j$ for all $j < i$. The random utility theory states that probability of choosing π_i can be obtained by integrating out all latent utilities subject to the inequality constraints:

$$\begin{aligned} Q(\pi_i \mid \boldsymbol{\pi}_{i+1:N}) &= \int_0^{+\infty} P_i(u_i) \left[\int_{u_i}^{+\infty} \prod_{j < i} P_j(u_j) du_j \right] du_i \\ &= \int_0^{+\infty} P_i(u_i) \prod_{j < i} (1 - F_j(u_i)) du_i \\ &= \int_0^{+\infty} b_i \eta_{ii} v_i \exp \left(-v_i \sum_{j < i} \eta_{ij} + \sum_{j < i} \eta_{ij} \right) du_i \end{aligned}$$

Note that we have used η_{ij} instead of η_j since this location parameter is specific to rank i . Defining $\eta_{ij} = \exp(-f(x_{\pi_j}))$ and changing the variable from u_i to v_i gives us:

$$\begin{aligned} Q(\pi_i \mid \boldsymbol{\pi}_{i+1:N}) &= \int_1^{+\infty} \eta_{ii} \exp \left(-v_i \sum_{j < i} \eta_{ij} + \sum_{j < i} \eta_{ij} \right) dv_i \\ &= \frac{\eta_{ii} e^{\sum_{j < i} \eta_j}}{\sum_{j < i} \eta_{ij}} \int_1^{+\infty} \exp \left(-v_i \sum_{j < i} \eta_{ij} \right) dv_i \\ &= \frac{\eta_{ii}}{\sum_{j < i} \eta_{ij}} = \frac{\exp(-f(x_{\pi_i}))}{\sum_{j=1}^i \exp(-f(x_{\pi_j}))}. \end{aligned}$$

This resembles Eq. (4).

Remark: We note in passing that in case of plain Plackett-Luce, $P(\boldsymbol{\pi}) \equiv P^*(\boldsymbol{\pi})$ if and only if u_{π_j} is drawn from a Gumbel distribution of location $f(x_{\pi_j})$ and scale 1 for all $j = 1, 2, \dots, N$ [23, 35].

3.2 Linear Time Learning

We show that both the loss and its functional gradient can be computed in linear time. Let $\phi_i = \exp(-f(x_{\pi_i}))$, $Z_i = \sum_{j=1}^i \phi_j = \phi_i + Z_{i-1}$, and

$Y_i = \sum_{j=1}^i 1/Z_j$. The loss in Eq. (5) becomes $\ell_2(\boldsymbol{\pi}) = \sum_{i=1}^N (f(x_{\pi_i}) + \log Z_i)$ which can be evaluated in linear time. The functional gradient is reduced to $\partial_{f_k} \ell_2 = \gamma_k (1 - \phi_k Y_k)$, which is constant for each k .

4 Neural Highway Networks for Rank Function

Once the rank models have been specified, it remains to define the rank function $f(x)$. We propose the novel use of neural highway networks [28] to approximate $f(x)$, under the new ranking loss functions proposed in Eqs. (2, 5). Highway networks are powerful, compact function approximator that overcomes the major bottleneck of standard deep neural networks: with increasing depth, it is much harder to pass information and gradient between input and output. More specifically, highway networks are a special neural network of L nonlinear projection layers, of which $L - 1$ layers are recurrent. At the bottom, the data is projected into a K -dimensional space $z = g(b_H + W_X x)$, where g is an element-wise nonlinear transform. Subsequent layers are recursive same-dimensional projections:

$$z_{l+1} \leftarrow H(z_l) * T(z_l) + z_l * (1 - T(z_l))$$

where $H(z) = g(b_H + W_H z)$, $T(z) \in [0, 1]^K$ is an element-wise gating function, and $*$ denotes element-wise product. The gate $T(z)$ allows information to pass freely when $T(z) = 0$ and forces total nonlinear transforms when $T(z) = 1$. A typical implementation is $T(z) = \sigma(b_T + W_T z)$, where σ is the sigmoid function. At the top layer, we have $f(x) = \langle w, z_L \rangle$. Note that when the number of hidden layer is 1, this returns to the standard shallow neural network; and when $T(z) = 1$, the highway network becomes a recurrent network. Let $\boldsymbol{\theta} = (b_T, b_H, W_T, W_H, W_X, w)$ be model parameters, gradient-based learning proceeds as follows:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \frac{\mu}{N} \sum_{k=1}^N \frac{\partial \ell_{1,2}}{\partial f(x_{\pi_k})} \frac{\partial f(x_{\pi_k})}{\partial \boldsymbol{\theta}}$$

for learning rate $\mu > 0$, where $\frac{\partial \ell_{1,2}}{\partial f(x_{\pi_k})}$ are functional gradients derived from Eqs. (2, 5).

5 Experimental Evaluation

5.1 Data and Evaluation Metrics

We validate the proposed model using a large-scale Web dataset from the Yahoo! learning to rank challenge [8]. The dataset is split into a training set of 18,425 queries (425,821 documents), and a testing set of 1,520 queries (47,313 documents). We also prepare a smaller training subset, called Yahoo!-small, which has 1,568 queries and 47,314 documents. The Yahoo! datasets contain the

groundtruth relevance scores (from 0 for irrelevant to 4 for perfectly relevant). There are 519 pre-computed unique features for each query-document pair. We normalize the features across the whole training set to have mean 0 and standard deviation 1.

Two evaluation metrics are employed. The Normalized Discount Cumulative Gain [19] is defined as:

$$NG@T = \frac{1}{N_{max}} \sum_{i=1}^T \frac{2^{r_{\pi_i}} - 1}{\log_2(1 + i)}$$

where r_{π_i} is the relevance of item at rank i . The other metric is Expected Reciprocal Rank (ERR) [9], which was used in the Yahoo! learning-to-rank challenge (2011):

$$ERR = \sum_i \frac{R(r_{\pi_i})}{i} \prod_{j>i} [1 - R(r_{\pi_j})], \text{ s.t. } R(r_{\pi_i}) = \frac{2^{r_{\pi_i}} - 1}{16}$$

Both metrics discount for the long list and place more emphasis on the top ranking. Finally, all metrics are averaged across all test queries.

5.2 Model Implementation

The highway nets are configured as follows. Unless stated otherwise, we use ReLU units for transformation $H(z)$. Parameters W_1 , W_H and W_T are initialized randomly from a small Gaussian. Gate bias b_T is initialized at -1 to encourage passing more information, as suggested in [28]. Transform bias b_H is initialized at 0. To prevent overfitting, dropout [27] is used, i.e., during training for each item in a mini-batch, input features and hidden units are randomly dropped with probabilities p_{vis} and p_{hid} , respectively. As dropouts may cause big jumps in gradient and parameters, we set max-norm per hidden unit to 1. The mini-batch size is 2 queries, the learning rate starts from 0.1, and is halved when there is no sign of improvement in training loss. Learning stops when learning rate falls below 10^{-4} . We fix the dropout rates as follows: (a) for small Yahoo! data, $p_{vis} = 0$, $p_{hid} = 0.3$ and $K = 10$; (b) for large Yahoo! data, $p_{vis} = 0$, $p_{hid} = 0.2$ and $K = 20$. Figure 2 shows the effect of dropouts on the NCE model on the small Yahoo! dataset.

5.3 Results

Performance of Choice Models. In this experiment we ask whether the new choice by elimination model has any advantage over existing state-of-the-arts (the forward selection method of Plackett-Luce [34] and the popular Rank SVM [20]). Table 1 reports results on the all datasets (Yahoo!-small, Yahoo!-large) on different sequential models. The NCE works better than Rank SVM and Plackett-Luce. Note that due to the large size, the differences are statistically significant. In fact, in the Yahoo! L2R challenge, the top 20 scores (out of more than 1,500 teams) differ only by 1.56 % in ERR, which is less than the difference between Plackett-Luce and choice by elimination (1.62 %).

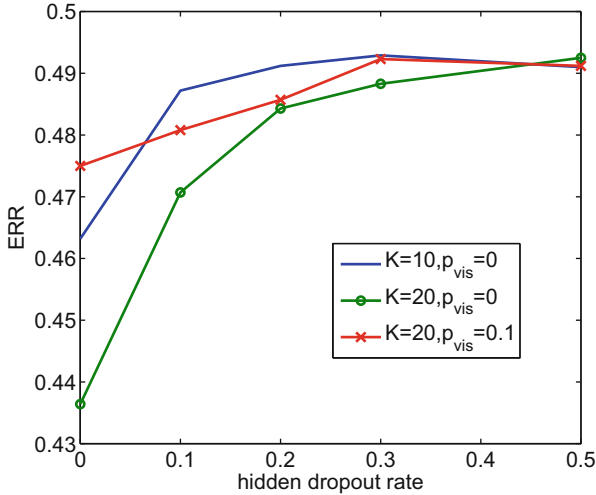


Fig. 2. Effect of dropouts on NCE performance of a 3-layer highway nets on Yahoo! small set. K is number of hidden units, p_{vis} is dropout rate for visible layer.

Table 1. Performance with linear rank functions.

Rank model	Yahoo!-small			Yahoo!-large		
	ERR	NDCG@1	NDCG@5	ERR	NDCG@1	NDCG@5
Rank SVM	0.477	0.657	0.642	0.488	0.681	0.666
Plackett-Luce	0.489	0.683	0.652	0.495	0.699	0.671
Choice by elimination	0.497	0.697	0.664	0.503	0.709	0.680

Neural Nets Versus Gradient Boosting. We also compare the highway networks against the best performing method in the Yahoo! L2R challenge. Since it was consistently demonstrated that gradient boosting trees work best [8], we implement a sophisticated variant of Stochastic Gradient Tree Boosting (SGTB) [13] for comparison. In SGTB, at each iteration, regression trees are grown to fit a random subset of functional gradients $\frac{\partial \ell}{\partial f(x)}$. Grown trees are added to the ensemble with an adaptive learning rate which is halved whenever the loss fluctuates and does not decrease. At each tree node, a random subset of features is used to split the node, following [4]. Tree nodes are split at random as it leads to much faster tree growing without hurting performance [15]. The SGTB is configured as follows: number of trees is 300; learning rate starts at 0.1; a random subset of 50 % data is used to grow a tree; one third of features are randomly selected at each node; trees are grown until either the number of leaves reaches 512 or the node size is below 40.

Table 2 show performance scores of models trained under different losses on the Yahoo!-large dataset. The highway neural networks are consistently com-

Table 2. Comparing highway neural networks against stochastic gradient tree boosting (SGTB) on Yahoo!-large dataset.

Rank function	Plackett-Luce			Choice by elimination		
	ERR	NDCG@1	NDCG@5	ERR	NDCG@1	NDCG@5
SGTB	0.497	0.697	0.673	0.506	0.705	0.681
Neutral nets	0.501	0.705	0.688	0.509	0.719	0.697

petitive against the gradient tree boosting, the best performing rank function approximator in this challenge [8].

6 Conclusion

We have presented *Neural Choice by Elimination*, a new framework that integrates deep neural networks into a formal modeling of human behaviors in making sequential choices. Contrary to the standard Plackett-Luce model, where the most worthy items are iteratively selected, here the least worthy items are iteratively eliminated. Theoretically we show that choice by elimination is equivalent to sequentially marginalizing out Thurstonian random utilities that follow Gompertz distributions. Experimentally we establish that deep neural networks are competitive in the learning to rank domain, as demonstrated on a large-scale public dataset from the Yahoo! learning to rank challenge.

References

1. Agarwal, A., Raghavan, H., Subbian, K., Melville, P., Lawrence, R.D., Gondek, D.C., Fan, J.: Learning to rank for robust question answering. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 833–842. ACM (2012)
2. Azari, H., Parks, D., Xia, L.: Random utility theory for social choice. In: Advances in Neural Information Processing Systems, pp. 126–134 (2012)
3. Bengio, Y.: Learning deep architectures for AI. *Found. Trends Mach. Learn.* **2**(1), 1–127 (2009)
4. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
5. Breiman, L.: Bagging predictors. *Mach. Learn.* **24**(2), 123–140 (1996)
6. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: Proceedings of the 22nd International Conference on Machine Learning, p. 96. ACM (2005)
7. Burges, C.J., Svore, K.M., Bennett, P.N., Pastusiak, A., Qiang, W.: Learning to rank using an ensemble of lambda-gradient models. In: Yahoo! Learning to Rank Challenge, pp. 25–35 (2011)
8. Chapelle, O., Chang, Y.: Yahoo! learning to rank challenge overview. In: JMLR Workshop and Conference Proceedings, vol. 14, pp. 1–24 (2011)
9. Chapelle, O., Metlzer, D., Zhang, Y., Grinspan, P.: Expected reciprocal rank for graded relevance. In: CIKM, pp. 621–630. ACM (2009)

10. Deng, L., He, X., Gao, J.: Deep stacking networks for information retrieval. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3153–3157. IEEE (2013)
11. Dong, Y., Huang, C., Liu, W.: RankCNN: when learning to rank encounters the pseudo preference feedback. *Comput. Stan. Interfaces* **36**(3), 554–562 (2014)
12. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. *Ann. Stat.* **29**(5), 1189–1232 (2001)
13. Friedman, J.H.: Stochastic gradient boosting. *Comput. Stat. Data Anal.* **38**(4), 367–378 (2002)
14. Fu, Q., Lu, J., Wang, Z.: ‘Reverse’ nested lottery contests. *J. Math. Econ.* **50**, 128–140 (2014)
15. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. *Mach. Learn.* **63**(1), 3–42 (2006)
16. Gormley, I.C., Murphy, T.B.: A mixture of experts model for rank data with applications in election studies. *Ann. Appl. Stat.* **2**(4), 1452–1477 (2008)
17. Henery, R.J.: Permutation probabilities as models for horse races. *J. R. Stat. Soc. Ser. B* **43**(1), 86–91 (1981)
18. Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, pp. 2333–2338. ACM (2013)
19. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst. (TOIS)* **20**(4), 446 (2002)
20. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 133–142. ACM, New York (2002)
21. Li, P., Burges, C., Wu, Q., Platt, J.C., Koller, D., Singer, Y., Roweis, S.: McRank: learning to rank using multiple classification and gradient boosting. In: Advances in Neural Information Processing Systems (2007)
22. Liu, T.-Y.: *Learning to Rank for Information Retrieval*. Springer, Berlin (2011)
23. McFadden, D.: Conditional logit analysis of qualitative choice behavior. In: *Frontiers in Econometrics*, pp. 105–142 (1973)
24. Plackett, R.L.: The analysis of permutations. *Appl. Stat.* **24**, 193–202 (1975)
25. Severyn, A., Moschitti, A.: Learning to rank short text pairs with convolutional deep neural networks. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 373–382. ACM (2015)
26. Song, Y., Wang, H., He, X.: Adapting deep ranknet for personalized search. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, pp. 83–92. ACM (2014)
27. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014)
28. Srivastava, R.K., Greff, K., Schmidhuber, J.: Training very deep networks (2015). arXiv preprint: [arXiv:1507.06228](https://arxiv.org/abs/1507.06228)
29. Thurstone, L.L.: A law of comparative judgment. *Psychol. Rev.* **34**(4), 273 (1927)
30. Tran, T., Phung, D., Venkatesh, S.: Thurstonian Boltzmann machines: learning from multiple inequalities. In: International Conference on Machine Learning (ICML), Atlanta, USA, 16–21 June 2013

31. Tran, T., Phung, D.Q., Venkatesh, S.: Sequential decision approach to ordinal preferences in recommender systems. In: Proceedings of the 26th AAAI Conference, Toronto, Ontario, Canada (2012)
32. Truyen, T., Phung, D.Q., Venkatesh, S.: Probabilistic models over ordered partitions with applications in document ranking and collaborative filtering. In: Proceedings of SIAM Conference on Data Mining (SDM), Mesa, Arizona, USA. SIAM (2011)
33. Tversky, A.: Elimination by aspects: a theory of choice. *Psychol. Rev.* **79**(4), 281 (1972)
34. Xia, F., Liu, T.Y., Wang, J., Zhang, W., Li, H.: Listwise approach to learning to rank: theory and algorithm. In Proceedings of the 25th International Conference on Machine Learning, pp. 1192–1199. ACM (2008)
35. Yellott, J.I.: The relationship between Luce’s choice axiom, Thurstone’s theory of comparative judgment, and the double exponential distribution. *J. Math. Psychol.* **15**(2), 109–144 (1977)

Trends and Applications in Knowledge Discovery and
Data Mining

PAKDD 2016 Workshops, BDM, MLSDA, PACC, WDMBF,
Auckland, New Zealand, April 19, 2016, Revised
Selected Papers

Cao, H.; Li, J.; Wang, R. (Eds.)

2016, XIII, 282 p. 84 illus., Softcover

ISBN: 978-3-319-42995-3