Active Evaluation of Ranking Functions based on Graded Relevance

Christoph Sawade[1], Steffen Bickel[2], Timo von Oertzen[3], Tobias Scheffer[1], and Niels Landwehr[1]

[1] University of Potsdam, Department of Computer Science, August-Bebel-Strasse 89,
14482 Potsdam, Germany
{sawade,scheffer,landwehr}@cs.uni-potsdam.de
[2] Nokia gate5 GmbH, Invalidenstrasse 117, 10115 Berlin, Germany
steffen.bickel@nokia.com
[3] University of Virginia, Department of Psychology, Charlottesville, VA 22903
timo@virginia.edu

Abstract. Evaluating the quality of ranking functions is a core task in web search and other information retrieval domains. Because query distributions and item relevance change over time, ranking models often cannot be evaluated accurately on held-out training data. Instead, considerable effort is spent on manually labeling the relevance of query results for test queries in order to track ranking performance. We address the problem of estimating ranking performance as accurately as possible on a fixed labeling budget. Estimates are based on a set of most informative test queries selected by an active sampling distribution. Query labeling costs depend on the number of result items as well as item-specific attributes such as document length. We derive cost-optimal sampling distributions for the commonly used performance measures Discounted Cumulative Gain (DCG) and Expected Reciprocal Rank (ERR). Experiments on web search engine data illustrate significant reductions in labeling costs.

Keywords: Information Retrieval, Ranking, Active Evaluation

1 Introduction

This paper addresses the problem of estimating the performance of a given ranking function in terms of graded relevance measures such as Discounted Cumulative Gain [1] and Expected Reciprocal Rank [2]. In informational retrieval domains, ranking models often cannot be evaluated on held-out training data. For example, older training data might not represent the distribution of queries the model is currently exposed to, or the ranking model might be procured from a third party that does not provide any training data.

In practice, ranking performance is estimated by applying a given ranking model to a representative set of test queries and manually assessing the relevance of all retrieved items for each query. We study the problem of estimating ranking performance as accurately as possible on a fixed budget for labeling item relevance, or, equivalently, minimizing labeling costs for a given level of estimation accuracy. We also study the related problem of cost-efficiently comparing the ranking performance of two models; this is required, for instance, to evaluate the result of an index update.

We assume that drawing unlabeled data $x \sim p(x)$ from the distribution of queries that the model is exposed to is inexpensive, whereas obtaining relevance labels is costly. The standard approach to estimating ranking performance is to draw a sample of test queries from $p(x)$, obtain relevance labels, and compute the empirical performance. However, recent results on *active risk estimation* [3] and *active comparison* [4] indicate that estimation accuracy can be improved by drawing test examples from an appropriately engineered instrumental distribution $q(x)$ rather than $p(x)$, and correcting for the discrepancy between p and q by importance weighting.

In this paper, we study active estimates of ranking performance. Section 2 details the problem setting. A novel aspect of active estimation in a ranking setting is that labeling costs vary according to the number of items that are relevant for a query. Section 3 derives cost-optimal sampling distributions for the estimation of DCG and ERR. Section 4 discusses empirical sampling distributions in a pool-based setting. Naïve computation of the empirical distributions is exponential, we derive polynomial-time solutions by dynamic programming. Section 5 presents empirical results. Section 6 discusses related work, Section 7 concludes.

2 Problem Setting

Let $\mathcal{X}$ denote a space of queries, and $\mathcal{Z}$ denote a finite space of items. We study ranking functions

$$\mathbf{r} : x \mapsto \left(r_1(x), \ldots, r_{|\mathbf{r}(x)|}(x)\right)^\mathsf{T}$$

that, given a query $x \in \mathcal{X}$, return a list of $|\mathbf{r}(x)|$ items $r_i(x) \in \mathcal{Z}$ ordered by relevance. The number of items in a ranking $\mathbf{r}(x)$ can vary depending on the query and application domain from thousands (web search) to ten or fewer (mobile applications that have to present results on a small screen). Ranking performance of $\mathbf{r}$ is defined in terms of graded relevance labels $y_z \in \mathcal{Y}$ that represent the relevance of an item $z \in \mathcal{Z}$ for the query x, where $\mathcal{Y} \subset \mathbb{R}$ is a finite space of relevance labels with minimum zero (irrelevant) and maximum y_{max} (perfectly relevant). We summarize the graded relevance of all $z \in \mathcal{Z}$ in a label vector $\mathbf{y} \in \mathcal{Y}^{\mathcal{Z}}$ with components y_z for $z \in \mathcal{Z}$.

In order to evaluate the quality of a ranking $\mathbf{r}(x)$ for a single query x, we employ two commonly used ranking performance measures: *Discounted Cumulative*

Gain (DCG), given by

$$L_{dcg}\left(\mathbf{r}(x),\mathbf{y}\right)=\sum_{i=1}^{|\mathbf{r}(x)|}\ell_{dcg}\left(y_{r_i(x)},i\right) \tag{1}$$
$$\ell_{dcg}\left(y,i\right)=\frac{2^y-1}{\log_2(i+1)},$$

and *Expected Reciprocal Rank* (ERR), given by

$$L_{err}\left(\mathbf{r}(x),\mathbf{y}\right)=\sum_{i=1}^{|\mathbf{r}(x)|}\frac{1}{i}\ell_{err}\left(y_{r_i(x)}\right)\prod_{l=1}^{i-1}(1-\ell_{err}\left(y_{r_l(x)}\right)) \tag{2}$$
$$\ell_{err}\left(y\right)=\frac{2^y-1}{2^{y_{max}}}$$

as introduced by Järvelin and Kekäläinen [1] and Chapelle et. al [2], respectively.

DCG scores a ranking by summing over the relevance of all documents discounted by their position in the ranking. ERR is based on a probabilistic user model: the user scans a list of documents in the order defined by $\mathbf{r}(x)$ and chooses the first document that appears sufficiently relevant; the likelihood of choosing a document z is a function of its graded relevance score y_z. If s denotes the position of the chosen document in $\mathbf{r}(x)$, then $L_{err}\left(\mathbf{r}(x),\mathbf{y}\right)$ is the expectation of the reciprocal rank $1/s$ under the probabilistic user model. Both DCG and ERR discount relevance with ranking position, ranking quality is thus most strongly influenced by documents that are ranked highly. If $\mathbf{r}(x)$ includes many items, L_{dcg} and L_{err} are in practice often approximated by only labeling items up to a certain position in the ranking or a certain relevance threshold and ignoring the contribution of lower-ranked items.

Let $p(x,\mathbf{y})=p(x)p(\mathbf{y}|x)$ denote the joint distribution over queries $x\in\mathcal{X}$ and label vectors $\mathbf{y}\in\mathcal{Y}^{\mathcal{Z}}$ the model is exposed to. We assume that the individual relevance labels y_z for items z are drawn independently given a query x:

$$p(\mathbf{y}|x)=\prod_{z\in\mathcal{Z}}p(y_z|x,z). \tag{3}$$

This assumption is common in pointwise ranking approaches, *e.g.,* regression based ranking models [5,6]. The ranking performance of $\mathbf{r}$ with respect to $p(x,\mathbf{y})$ is given by

$$R[\mathbf{r}]=\iint L\left(\mathbf{r}(x),\mathbf{y}\right)p(x,\mathbf{y})\mathrm{d}x\,\mathrm{d}\mathbf{y}, \tag{4}$$

where $L\in\{L_{dcg},L_{err}\}$ denotes the performance measure under study. We use integrals for notational convenience, for discrete spaces the corresponding integral is replaced by a sum. If the context is clear, we refer to $R[\mathbf{r}]$ simply by R.

Since $p(x,\mathbf{y})$ is unknown, ranking performance is typically approximated by an empirical average

$$\hat{R}_n[\mathbf{r}]=\frac{1}{n}\sum_{j=1}^{n}L\left(\mathbf{r}(x_j),\mathbf{y}_j\right), \tag{5}$$

where a set of test queries $x_1, ..., x_n$ and graded relevance labels $\mathbf{y}_1, ..., \mathbf{y}_n$ are drawn *iid* from $p(x, \mathbf{y})$. The empirical performance $\hat{R}_n$ consistently estimates the true ranking performance; that is, $\hat{R}_n$ converges to R with $n \to \infty$.

Test queries x_i need not necessarily be drawn according to the input distribution p. When instances are drawn according to an instrumental distribution q, a consistent estimator can be defined as

$$\hat{R}_{n,q}[\mathbf{r}] = \left(\sum_{j=1}^{n} \frac{p(x_j)}{q(x_j)} \right)^{-1} \sum_{j=1}^{n} \frac{p(x_j)}{q(x_j)} L(\mathbf{r}(x_j), \mathbf{y}_j), \tag{6}$$

where $(x_j, \mathbf{y}_j)$ are drawn from $q(x)p(\mathbf{y}|x)$ and again $L \in \{L_{dcg}, L_{err}\}$. For certain choices of the sampling distribution q, $\hat{R}_{n,q}$ may be a more label-efficient estimator of the true performance R than $\hat{R}_n$ [3].

A crucial feature of ranking domains is that labeling costs for queries $x \in \mathcal{X}$ vary with the number of items $|\mathbf{r}(x)|$ returned and item-specific features such as the length of a document whose relevance has to be determined. We denote labeling costs for a query x by $\lambda(x)$, and assume that $\lambda(x)$ is bounded away from zero by $\lambda(x) \geq \epsilon > 0$. Our goal is to minimize the deviation of $\hat{R}_{n,q}$ from R under the constraint that expected overall labeling costs stay below a budget $\Lambda \in \mathbb{R}$:

$$(q^*, n^*) = \arg\min_{q,n} \mathbb{E}\left[\left(\hat{R}_{n,q} - R \right)^2 \right], \text{ s.t. } \mathbb{E}\left[\sum_{j=1}^{n} \lambda(x_j) \right] \leq \Lambda. \tag{7}$$

Note that Equation 7 represents a trade-off between labeling costs and informativeness of a test query: optimization over n implies that many inexpensive or few expensive queries could be chosen.

To estimate *relative* performance of two ranking functions $\mathbf{r}_1$ and $\mathbf{r}_2$, Equation 7 can be replaced by

$$(q^*, n^*) = \arg\min_{q,n} \mathbb{E}\left[\left(\hat{\Delta}_{n,q} - \Delta \right)^2 \right], \text{ s.t. } \mathbb{E}\left[\sum_{j=1}^{n} \lambda(x_j) \right] \leq \Lambda, \tag{8}$$

where $\hat{\Delta}_{n,q} = \hat{R}_{n,q}[\mathbf{r}_1] - \hat{R}_{n,q}[\mathbf{r}_2]$ and $\Delta = R[\mathbf{r}_1] - R[\mathbf{r}_2]$. In the next section, we derive sampling distributions q^* asymptotically solving Equations 7 and 8.

3 Asymptotically Optimal Sampling

A bias-variance decomposition [7] applied to Equation 7 results in

$$\mathbb{E}\left[\left(\hat{R}_{n,q} - R \right)^2 \right] = \left(\mathbb{E}\left[\hat{R}_{n,q} \right] - R \right)^2 + \operatorname{Var}\left[\hat{R}_{n,q} \right].$$

According to [8], Chapter 2.5.3, the squared bias term is of order $\frac{1}{n^2}$, while the variance is of order $\frac{1}{n}$. For large n, the expected deviation is thus dominated by the variance, and $\sigma_q^2 = \lim_{n\to\infty} n\,\mathrm{Var}[\hat{R}_{n,q}]$ exists. For large n, we can thus approximate

$$\mathbb{E}\left[\left(\hat{R}_{n,q} - R\right)^2\right] \approx \frac{1}{n}\sigma_q^2; \quad \mathbb{E}\left[\left(\hat{\Delta}_{n,q} - \Delta\right)^2\right] \approx \frac{1}{n}\tau_q^2,$$

where $\tau_q^2 = \lim_{n\to\infty} n\,\mathrm{Var}[\hat{\Delta}_{n,q}]$. Let $\delta(x,\mathbf{y}) = L(\mathbf{r}_1(x),\mathbf{y}) - L(\mathbf{r}_2(x),\mathbf{y})$ denote the performance difference of the two ranking models for a test query $(\mathbf{x}, y)$ and $L \in \{L_{dcg}, L_{err}\}$. The following theorem derives sampling distributions minimizing the quantities $\frac{1}{n}\sigma_q^2$ and $\frac{1}{n}\tau_q^2$, thereby approximately solving Problems 7 and 8.

Theorem 1 (Optimal Sampling for Evaluation of a Ranking Function). *Let $L \in \{L_{dcg}, L_{err}\}$ and $\sigma_q^2 = \lim_{n\to\infty} n\,\mathrm{Var}[\hat{R}_{n,q}]$. The optimization problem*

$$(q^*, n^*) = \arg\min_{q,n} \frac{1}{n}\sigma_q \quad s.t. \;\; \mathbb{E}\left[\sum_{j=1}^{n} \lambda(x_j)\right] \leq \Lambda$$

is solved by

$$q^*(x) \propto \frac{p(x)}{\sqrt{\lambda(x)}}\sqrt{\int \left(L(\mathbf{r}(x),\mathbf{y}) - R\right)^2 p(\mathbf{y}|x)\mathrm{d}\mathbf{y}}, \quad n^* = \frac{\Lambda}{\int \lambda(x)q(x)\mathrm{d}x}. \tag{9}$$

Theorem 2 (Optimal Sampling for Comparison of Ranking Functions). *Let $L \in \{L_{dcg}, L_{err}\}$ and $\tau_q^2 = \lim_{n\to\infty} n\,\mathrm{Var}[\hat{\Delta}_{n,q}]$. The optimization problem*

$$(q^*, n^*) = \arg\min_{q,n} \frac{1}{n}\tau_q \quad s.t. \;\; \mathbb{E}\left[\sum_{j=1}^{n} \lambda(x_j)\right] \leq \Lambda$$

is solved by

$$q^*(x) \propto \frac{p(x)}{\sqrt{\lambda(x)}}\sqrt{\int \left(\delta(x,\mathbf{y}) - \Delta\right)^2 p(\mathbf{y}|x)\mathrm{d}\mathbf{y}}, \quad n^* = \frac{\Lambda}{\int \lambda(x)q(x)\mathrm{d}x}. \tag{10}$$

Before we prove Theorem 1 and Theorem 2, we state the following Lemma:

Lemma 1. *Let $a : \mathcal{X} \to \mathbb{R}$ and $\lambda : \mathcal{X} \to \mathbb{R}$ denote functions on the query space such that $\int \sqrt{a(x)}\mathrm{d}x$ exists and $\lambda(x) \geq \epsilon > 0$. The functional*

$$G[q] = \left(\int \frac{a(x)}{q(x)}\mathrm{d}x\right)\left(\int \lambda(x)q(x)\mathrm{d}x\right),$$

where $q(x)$ is a distribution over the query space $\mathcal{X}$, is minimized over q by setting

$$q(x) \propto \sqrt{\frac{a(x)}{\lambda(x)}}.$$

A proof is included in the appendix. We now prove Theorem 1 and Theorem 2, building on results of Sawade et al. [3,4].

Proof (Theorem 1 and Theorem 2). We first study the minimization of $\frac{1}{n}\sigma_q^2$ in Theorem 1. Since

$$\mathbb{E}\left[\sum_{j=1}^{n}\lambda(x_j)\right] = n\int\lambda(x)q(x)\mathrm{d}x,$$

the minimization problem can be reformulated as

$$\min_q\min_n\frac{1}{n}\sigma_q^2 \text{ s.t. } n \leq \frac{\Lambda}{\int\lambda(x)q(x)\mathrm{d}x}.$$

Clearly $n^* = \Lambda/\int\lambda(x)q(x)\mathrm{d}x$ solves the inner optimization. The remaining minimization over q is

$$q^* = \arg\min_q \sigma_q^2\int\lambda(x)q(x)\mathrm{d}x.$$

Lemma 1 in [3] implies

$$\sigma_q^2 = \iint\frac{p^2(x)}{q^2(x)}\left(L(\mathbf{r}(x),\mathbf{y})-R\right)^2 p(\mathbf{y}|x)q(x)\mathrm{d}x\,\mathrm{d}\mathbf{y}.$$

Setting $a(x) = p^2(x)\int\left(L(\mathbf{r}(x),\mathbf{y})-R\right)^2 p(\mathbf{y}|x)\mathrm{d}\mathbf{y}$ and applying Lemma 1 implies Equation 9. For the minimization of $\frac{1}{n}\tau_q^2$ in Theorem 2 we analogously derive

$$q^* = \arg\min_q \tau_q^2\int\lambda(x)q(x)\mathrm{d}x.$$

Lemma 3 in [4] implies

$$\tau_q^2 = \iint\frac{p(\mathbf{x})^2}{q(\mathbf{x})^2}\left(\delta(x,\mathbf{y})-\Delta\right)^2 p(y|\mathbf{x})q(\mathbf{x})\mathrm{d}y\,\mathrm{d}\mathbf{x}.$$

Setting $a(x) = p^2(x)\int\left(\delta(x,\mathbf{y}),\mathbf{y})-\Delta\right)^2 p(\mathbf{y}|x)\mathrm{d}\mathbf{y}$ and applying Lemma 1 implies Equation 10. □

4 Empirical Sampling Distribution

The sampling distributions prescribed by Theorem 1 and Theorem 2 depend on the unknown test distribution $p(x)$. We now turn towards a setting in which a pool D of m unlabeled queries is available. Queries from this pool can be sampled and then labeled at a cost. Drawing queries from the pool replaces generating them under the test distribution; that is, $p(x) = \frac{1}{m}$ for all $x \in D$.

The optimal sampling distribution also depends on the true conditional $p(\mathbf{y}|x) = \prod_{z\in\mathcal{Z}} p(y_z|x,z)$ (Equation 3). To implement the method, we approximate $p(y_z|x,z)$ by a model $p(y_z|x,z;\theta)$ of graded relevance. For the large class of pointwise ranking methods – that is, methods that produce a ranking by predicting graded relevance scores for query-document pairs and then sorting documents according to their score – such a model can typically be derived from the graded relevance predictor. Finally, the sampling distributions depend on the true performance $R[\mathbf{r}]$ as given by Equation 4, or $\Delta = R[\mathbf{r}_1] - R[\mathbf{r}_2]$. $R[\mathbf{r}]$ is replaced by an introspective performance $R_\theta[\mathbf{r}]$ calculated from Equation 4, where the integral over $\mathcal{X}$ is replaced by a sum over the pool, $p(x) = \frac{1}{m}$, and $p(\mathbf{y}|x) = \prod_{z\in\mathcal{Z}} p(y_z|x, z; \theta)$. The performance difference Δ is approximated by $\Delta_\theta = R_\theta[\mathbf{r}_1] - R_\theta[\mathbf{r}_2]$. Note that as long as $p(x) > 0$ implies $q(x) > 0$, the weighting factors ensure that such approximations do not introduce an asymptotic bias in our estimator (Equation 6).

With these approximations, we arrive at the following empirical sampling distributions.

Derivation 1 *When relevance labels for individual items are independent given the query (Equation 3), and $p(y_z|x, z)$ is approximated by a model $p(y|x, z; \theta)$ of graded relevance, the sampling distributions minimizing $\frac{1}{n}\sigma_q^2$ and $\frac{1}{n}\tau_q^2$ in a pool-based setting resolve to*

$$q^*(x) \propto \frac{1}{\sqrt{\lambda(x)}} \sqrt{\mathbb{E}\left[\left(L(\mathbf{r}(x), \mathbf{y}) - R_\theta\right)^2 \middle| x; \theta\right]} \tag{11}$$

and

$$q^*(x) \propto \frac{1}{\sqrt{\lambda(x)}} \sqrt{\mathbb{E}\left[\left(\delta(x, \mathbf{y}) - \Delta_\theta\right)^2 \middle| x; \theta\right]}, \tag{12}$$

respectively. Here, for any function $g(x, \mathbf{y})$ of a query x and label vector $\mathbf{y}$,

$$\mathbb{E}\left[g(x, \mathbf{y}) | x; \theta\right] = \sum_{\mathbf{y}\in\mathcal{Y}^{\mathcal{Z}}} g(x, \mathbf{y}) \prod_{z\in\mathcal{Z}} p(y_z|x, z; \theta) \tag{13}$$

denotes expectation of $g(x, \mathbf{y})$ with respect to label vectors $\mathbf{y}$ generated according to $p(y_z|x, z, \theta)$.

We observe that for the evaluation of a single given ranking function $\mathbf{r}$ (Equation 11), the empirical sampling distribution gives preference to queries x with low costs $\lambda(x)$ and for which the expected ranking performance deviates strongly from the average expected ranking performance R_θ; the expectation is taken with respect to the available graded relevance model θ. For the comparison of two given ranking functions $\mathbf{r}_1$ and $\mathbf{r}_2$ (Equation 12), preference is given to queries x with low costs and for which the difference in performance $L(\mathbf{r}_1(x), \mathbf{y}) - L(\mathbf{r}_2(x), \mathbf{y})$ is expected to be high (note that Δ_θ will typically be small).

Computation of the empirical sampling distributions given by Equations 11 and 12 requires the computation of $\mathbb{E}\left[g(x, \mathbf{y}) | x; \theta\right]$, which is defined in terms of a sum over exponentially many relevance label vectors $\mathbf{y} \in \mathcal{Y}^{\mathcal{Z}}$. The following theorem states that the empirical sampling distributions can nevertheless be computed in polynomial time:

Algorithm 1 Active Estimation of Ranking Performance

```
input Ranking function $\mathbf{r}$ or pair of ranking functions $\mathbf{r}_1$, $\mathbf{r}_2$; graded relevance model
    $p(y_z|x, z; \theta)$; pool $D$, labeling budget $\Lambda$.
Compute sampling distribution $q^*$ (Derivation 1, Equation 11 or 12).
Initialize $n \leftarrow 0$.
Draw $x_1 \sim q^*(x)$ from $D$ with replacement.
while $\sum_{j=1}^{n+1} \lambda(x_j) \leq \Lambda$ do
    Obtain $\mathbf{y}_{n+1} \sim p(\mathbf{y}|x_{n+1})$ from human labeler (restrict to items in rankings).
    Update number of labeled instances $n \leftarrow n + 1$.
    Draw $x_{n+1} \sim q^*(x)$ from $D$ with replacement.
end while
Compute $\hat{R}_{n,q}[\mathbf{r}]$ or $\hat{\Delta}_{n,q} = \hat{R}_{n,q}[\mathbf{r}_1] - \hat{R}_{n,q}[\mathbf{r}_2]$ (Equation 6).
output $\hat{R}_{n,q}[\mathbf{r}]$ or $\hat{\Delta}_{n,q}$.
```

Theorem 3 (Polynomial-time computation of sampling distributions).

The sampling distribution given by Equation 11 can be computed in time

$$\mathcal{O}\left(|\mathcal{Y}||D| \max_x |\mathbf{r}(x)|\right) \quad \textit{for} \quad L \in \{L_{dcg}, L_{err}\}.$$

The sampling distribution given by Equation 12 can be computed in time

$$\mathcal{O}\left(|\mathcal{Y}||D| \max_x (|\mathbf{r}_1(x) \cup \mathbf{r}_2(x)|)\right) \quad \textit{for} \quad L = L_{dcg},$$
$$\mathcal{O}\left(|\mathcal{Y}||D| \max_x (|\mathbf{r}_1(x)| \cdot |\mathbf{r}_2(x)|)\right) \quad \textit{for} \quad L = L_{err}.$$

Polynomial-time solutions are derived by dynamic programming. Specifically, after substituting Equations 1 and 2 into Equations 11 and 12 and exploiting the independence assumption given by Equation 3, Equations 11 and 12 decompose into cumulative sums and products of expectations over individual item labels $y \in \mathcal{Y}$. These sums and products can be computed in polynomial time. A proof of Theorem 3 is included in the appendix.

Algorithm 1 summarizes the active estimation algorithm. It samples queries $x_1, ..., x_n$ with replacement from the pool according to the distribution prescribed by Derivation 1 and obtains relevance labels from a human labeler for all items included in $\mathbf{r}(x_i)$ or $\mathbf{r}_1(x_i) \cup \mathbf{r}_2(x_i)$ until the labeling budget Λ is exhausted. Note that queries can be drawn more than once; in the special case that the labeling process is deterministic, recurring labels can be looked up rather than be queried from the deterministic labeling oracle repeatedly. Hence, the actual labeling costs may stay below $\sum_{j=1}^{n} \lambda(x_j)$. In this case, the loop is continued until the labeling budget Λ is exhausted.

5 Empirical Studies

We compare active estimation of ranking performance (Algorithm 1, labeled *active*) to estimation based on a test sample drawn uniformly from the pool (Equation 5, labeled *passive*). Algorithm 1 requires a model $p(y_z|x, z; \theta)$ of graded relevance in order to compute the sampling distribution q^* from Derivation 1. If no such model is available, a uniform distribution $p(y_z|x, z; \theta) = \frac{1}{|\mathcal{Y}|}$ can be used instead (labeled *active*$_{uniD}$). To quantify the effect of modeling costs, we also study a variant of Algorithm 1 that assumes uniform costs $\lambda(x) = 1$ in Equations 11 and 12 (labeled *active*$_{uniC}$). This variant implements active risk estimation [3] and active comparison [4] for ranking; we have shown how the resulting sampling distributions can be computed in polynomial time (Derivation 1 and Theorem 3).

Experiments are performed on the Microsoft Learning to Rank data set MSLR-WEB30k [9]. It contains 31,531 queries, and a set of documents for each query whose relevance for the query has been determined by human labelers in the process of developing the Bing search engine. The resulting 3,771,125 query-document pairs are represented by 136 features widely used in the information retrieval community (such as query term statistics, page rank, and click counts). Relevance labels take values from 0 (irrelevant) to 4 (perfectly relevant).

The data are split into five folds. On one fold, we train ranking functions using different graded relevance models (details below). The remaining four folds serve as a pool of unlabeled test queries; we estimate (Section 5.1) or compare (Section 5.2) the performance of the ranking functions by drawing and labeling queries from this pool according to Algorithm 1 and the baselines discussed above. Test queries are drawn until a labeling budget Λ is exhausted. To quantify the human effort realistically, we model the labeling costs $\lambda(x)$ for a query x as proportional to a sum of costs incurred for labeling individual documents $z \in \mathbf{r}(x)$; labeling costs for a single document z are assumed to be logarithmic in the document length.

All evaluation techniques, both active and passive, can approximate L_{dcg} and L_{err} for a query x by requesting labels only for the first k documents in the ranking. The number of documents for which the MSLR-WEB30k data set provides labels varies over the queries at an average of 119 documents per query. In our experiments, we use all documents for which labels are provided for each query and for all evaluation methods under investigation.

The cost unit is chosen such that average labeling costs for a query are one. Figure 1 (left) shows the distribution of labeling costs $\lambda(x)$. All results are averaged over the five folds and 5,000 repetitions of the evaluation process. Error bars indicate the standard error.

5.1 Estimating Ranking Performance

Based on the outcome of the 2010 Yahoo ranking challenge [6,10], we choose a pointwise ranking approach and employ Random Forest regression [11] to train graded relevance models on query-document pairs. The ranking function is obtained by returning all documents associated with a query sorted according to their predicted graded relevance. We apply the approach from [12,6] to obtain the probability estimates $p(y_z|x, z; \theta)$ required by Algorithm 1 from the Random Forest model. As an alternative graded relevance model, we also study a MAP

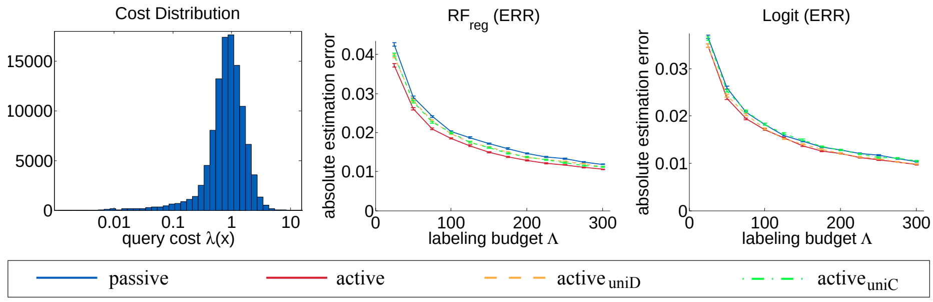

Fig. 1. Distribution of query labeling costs $\lambda(x)$ in the MSLR-WEB30k data set (left). Estimation error over Λ when evaluating Random Forest regression (center) and Ordered Logit (right). Error bars indicate the standard error.

version of Ordered Logit [13]; this model directly provides probability estimates $p(y_z|x, z; \theta)$. Half of the available training fold is used for model training, the other half is used as a validation set to tune hyperparameters of the respective ranking model. Throughout the experimental evaluation, we present results for the ERR measure; results for DCG are qualitatively similar and included in [14].

Figure 1 (center, right) shows absolute deviation between true ranking performance and estimated ranking performance as a function of the labeling budget Λ. True performance is taken to be the performance over all test queries. We observe that active estimation is more accurate than passive estimation; the labeling budget can be reduced from $\Lambda = 300$ by about 20% (Random Forest) and 10% (Ordered Logit).

5.2 Comparing Ranking Performance

We additionally train linear Ranking SVM [15] and the ordinal classification extension to Random Forests [12,6], and compare the resulting ranking functions to those of the Ordered Logit and Random Forest regression models. For the comparison of Random Forest vs. Ordered Logit both models provide us with estimates $p(y_z|x, z; \theta)$; in this case a mixture model is employed as proposed in [4]. We measure *model selection error*, defined as the fraction of experiments in which an evaluation method does not correctly identify the model with higher true performance. Figure 2 shows model selection error as a function of the available labeling budget for different pairwise comparisons. Active estimation more reliably identifies the model with higher ranking performance, saving between 30% and 55% of labeling effort compared to passive estimation. We observe that the gains of active versus passive estimation are not only due to differences in query costs: the baseline *active*$_{uniC}$, which does not take into account query costs for computing the sampling distribution, performs almost as well as *active*.

As a further comparative evaluation we simulate an index update. An outdated index with lower coverage is simulated by randomly removing 10% of all

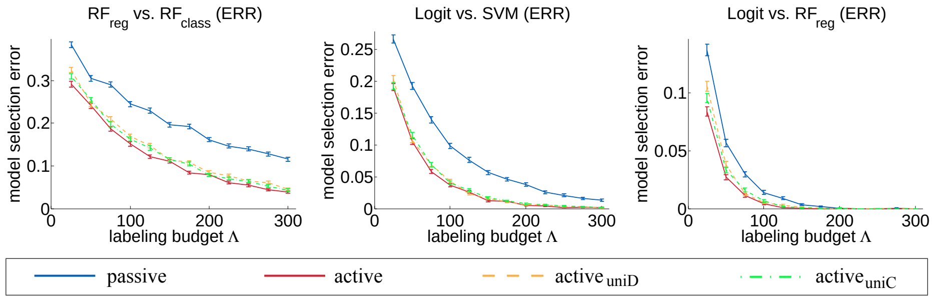

Fig. 2. Model selection error over Λ when comparing Random Forest regression vs. classification (left), and Ordered Logit vs. Ranking SVM (center) or Random Forest regression (right). Error bars indicate the standard error.

query-document pairs from each result list $\mathbf{r}(x)$ for all queries. Random Forest regression is employed as the ranking model. Active and passive estimation methods are applied to estimate the difference in performance between models based on the outdated and current index. Figure 3 (left) shows absolute deviation of estimated from true performance difference over labeling budget Λ. We observe that active estimation quantifies the impact of the index update more accurately than passive estimation, saving approximately 75% of labeling effort.

We finally simulate the incorporation of novel sources of training data by comparing a Random Forest model trained on 100,000 query-document pairs ($\mathbf{r}_1$) to a Random Forest model trained on between 120,000 and 200,000 query-document pairs ($\mathbf{r}_2$). The difference in performance between $\mathbf{r}_1$ and $\mathbf{r}_2$ is estimated using active and passive methods. Figure 3 (center) shows absolute deviation of estimated from true performance difference for models trained on 100,000 and 200,000 instances as a function of Λ. Active estimation quantifies the performance gain from additional training data more accurately, reducing labeling costs by approximately 45%. Figure 3 (right) shows estimation error as a function of the number of query-document pairs the model $\mathbf{r}_2$ is trained on for $\Lambda = 100$. Active estimation significantly reduces estimation error compared to passive estimation for all training set sizes.

6 Related Work

There has been significant interest in learning ranking functions from data in order to improve the relevance of search results [12,16,17,6]. This has partly been driven by the recent release of large-scale datasets derived from commercial search engines, such as the Microsoft Learning to Rank datasets [9] and the Yahoo Learning to Rank Challenge datasets [10].

In this paper, we have applied ideas from active risk estimation [3] and active comparison [4] to the problem of estimating ranking performance. Our problem

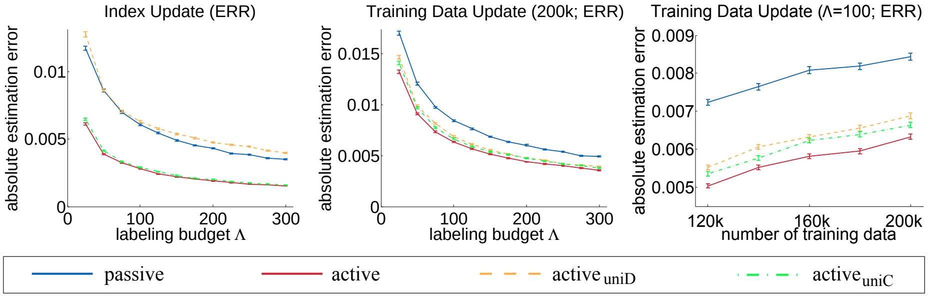

Fig. 3. Absolute estimation error over Λ for a simulated index update (left). Absolute estimation error comparing ranking functions trained on 100,000 vs. 200,000 query-document pairs over Λ (center), and over training set size of second model at $\Lambda = 100$ (right). Error bars indicate the standard error.

setting (Equations 7 and 8) generalizes the setting studied in active risk estimation and active comparison by allowing instance-specific labeling costs and constraining overall costs rather than the number of test instances that can be drawn. Applying the optimal sampling distributions derived by Sawade et al. [3,4] in a ranking setting leads to sums over exponentially many joint relevance label assignments (see Derivation 1). We have shown that they can be computed in polynomial time using dynamic programming (Theorem 3).

Besides sampling queries, it is also possible to sample subsets of documents to be labeled for a given query. Carterette et al. [18] use document sampling to decide which of two ranking functions achieves higher *precision at k*. Aslam et al. [19] use document sampling to obtain unbiased estimates of mean average precision and mean R-precision. Carterette and Smucker [20] study statistical significance testing from reduced document sets. Note that for the estimation of ERR studied in this paper, document sampling is not directly applicable because the discounting factor associated with a ranking position can only be determined if the relevance of all higher-ranked documents is known (Equation 2).

Active performance estimation can be considered a dual problem of active learning: in active learning, the goal of the selection process is to reduce the variance of predictions or model parameters; our approach reduces the variance of the performance estimate. Several active learning algorithms use importance weighting to compensate for the bias incurred by the instrumental distribution, for example in exponential family models [21] or SVMs [22].

7 Conclusions

We have studied the problem of estimating or comparing the performance of ranking functions as accurately as possible on a fixed budget for labeling item relevance. Theorems 1 and 2 derive sampling distributions that, when used to select test queries to be labeled from a given pool, asymptotically maximize the accuracy of the performance estimate. Theorem 3 shows that these optimal distributions can be computed efficiently.

Empirically, we observed that active estimates of ranking performance are more accurate than passive estimates. In different experimental settings – estimation of the performance of a single ranking model, comparison of different types of ranking models, simulated index updates – performing active estimation resulted in saved labeling efforts of between 10% and 75%.

Appendix

Proof of Lemma 1

We have to minimize the functional

$$\left(\int \frac{a(x)}{q(x)}\mathrm{d}x\right)\left(\int \lambda(x)q(x)\mathrm{d}x\right) \tag{14}$$

in terms of q under the constraints $\int q(x)\mathrm{d}x = 1$ and $q(x) > 0$. We first note that Objective 14 is invariant under multiplicative rescaling of $q(x)$, thus the constraint $\int q(x)\mathrm{d}x = 1$ can be dropped during optimization and enforced in the end by normalizing the unconstrained solution. We reformulate the problem as

$$\min_q C \int \frac{a(x)}{q(x)}\mathrm{d}x \quad \text{s.t.} \quad C = \int \lambda(x)q(x)\mathrm{d}x \tag{15}$$

which we solve using a Lagrange multiplier α by

$$\min_q C \int \frac{a(x)}{q(x)}\mathrm{d}x + \alpha\left(\int \lambda(x)q(x)\mathrm{d}x - C\right).$$

The optimal point for the constrained problem satisfies the Euler-Lagrange equation

$$\alpha\lambda(x) = C\frac{a(x)}{q(x)^2},$$

and therefore

$$q(x) = \sqrt{C\frac{a(x)}{\alpha\lambda(x)}}. \tag{16}$$

Resubstitution of Equation 16 into the constraint (Equation 15) leads to

$$C = \int \sqrt{C\frac{a(x)}{\alpha\lambda(x)}}\lambda(x)\mathrm{d}x, \tag{17}$$

solving for α we obtain

$$\alpha = \frac{\left(\int \sqrt{Ca(x)\lambda(x)}\mathrm{d}x\right)^2}{C^2}. \tag{18}$$

Finally, resubstitution of Equation 18 into Equation 16 proves the claim. □

Proof of Theorem 3

In order to show that the empirical sampling distributions given by Equations 11 and 12 can be computed efficiently, we have to show that Equation 13 can be computed efficiently. This can be done by suitable algebraic manipulation, exploiting the independence assumption given by Equation 3.

We now present the proof for the empirical distribution for absolute estimation (Equation 11) with $L = L_{err}$. The remaining cases can be found in [14]. It suffices to show that the intrinsic risk R_θ can be computed in time $\mathcal{O}(|\mathcal{Y}||D| \max_x |\mathbf{r}(x)|)$, and that for any $x \in \mathcal{X}$ the quantity $\mathbb{E}[(L(\mathbf{r}(x), \mathbf{y}) - R_\theta)^2 | x, \theta]$ can be computed in time $\mathcal{O}(|\mathcal{Y}||\mathbf{r}(x)|)$ given R_θ. We first note that for any $z \in \mathcal{Z}$, it holds that

$$\begin{aligned}
\mathbb{E}\left[\ell_{err}\left(y_z\right) \middle| x; \theta\right] &= \sum_{\mathbf{y} \in \mathcal{Y}^{\mathcal{Z}}} \ell_{err}\left(y_z\right) \prod_{z' \in \mathcal{Z}} p(y_{z'} | x, z'; \theta) \\
&= \sum_{y_z} \sum_{\mathbf{y} \in \mathcal{Y}^{\mathcal{Z} \setminus \{z\}}} \ell_{err}\left(y_z\right) \prod_{z' \in \mathcal{Z}} p(y_{z'} | x, z'; \theta) \\
&= \sum_{y_z} \ell_{err}\left(y_z\right) p(y_z | x, z; \theta) \sum_{\mathbf{y} \in \mathcal{Y}^{\mathcal{Z} \setminus \{z\}}} \prod_{z' \in \mathcal{Z} \setminus \{z\}} p(y_{z'} | x, z'; \theta) \\
&= \sum_{y_z} \ell_{err}\left(y_z\right) p(y_z | x, z; \theta), \qquad (19)
\end{aligned}$$

where $\mathbf{y} \in \mathcal{Y}^{\mathcal{Z} \setminus \{z\}}$ is a vector of relevance labels $y_{z'}$ for all $z' \in \mathcal{Z} \setminus \{z\}$. The quantity $\mathbb{E}\left[\ell_{err}\left(y_z, i\right) \middle| x; \theta\right]$ can thus be computed in time $\mathcal{O}(|\mathcal{Y}|)$. Furthermore, for $L = L_{err}$ it holds that

$$\begin{aligned}
R_\theta &= \sum_{x \in D} \frac{1}{|D|} \sum_{\mathbf{y} \in \mathcal{Y}^{\mathcal{Z}}} \sum_{i=1}^{|\mathbf{r}(x)|} \frac{1}{i} \ell_{err}\left(y_{r_i(x)}\right) \prod_{l=1}^{i-1} (1 - \ell_{err}\left(y_{r_l(x)}\right)) \prod_{z \in \mathcal{Z}} p(y_z | x, z; \theta) \\
&= \sum_{x \in D} \frac{1}{|D|} \sum_{i=1}^{|\mathbf{r}(x)|} \frac{1}{i} \mathbb{E}\left[\ell_{err}\left(y_{r_i(x)}\right) \prod_{l=1}^{i-1} (1 - \ell_{err}\left(y_{r_l(x)}\right)) \middle| x; \theta\right] \\
&= \sum_{x \in D} \frac{1}{|D|} \sum_{i=1}^{|\mathbf{r}(x)|} \frac{1}{i} \mathbb{E}\left[\ell_{err}\left(y_{r_i(x)}\right) \middle| x; \theta\right] \prod_{l=1}^{i-1} \left(1 - \mathbb{E}\left[\ell_{err}\left(y_{r_l(x)}\right) \middle| x; \theta\right]\right). \qquad (20)
\end{aligned}$$

Equation 20 can now be computed in time $\mathcal{O}(|\mathcal{Y}||D| \max_x |\mathbf{r}(x)|)$: for a given $x \in D$, we can compute the cumulative products $\prod_{l=1}^{i-1} \left(1 - \mathbb{E}\left[\ell_{err}\left(y_{r_l(x)}\right) \middle| x; \theta\right]\right)$ for $i = 1, ..., |\mathbf{r}(x)|$ in time $\mathcal{O}(|\mathcal{Y}||\mathbf{r}(x)|)$. We start by precomputing the cumulative products for all $x \in D$ and $i = 1, ..., |\mathbf{r}(x)|$ in time $\mathcal{O}(|\mathcal{Y}||D| \max_x |\mathbf{r}(x)|)$. Given precomputed cumulative products, the final summation over $x \in D$ and i can be carried out in time $\mathcal{O}(|D| \max_x |\mathbf{r}(x)|)$. We now turn towards the quantity

$$\mathbb{E}[(L(\mathbf{r}(x), \mathbf{y}) - R_\theta)^2 | x, \theta].$$

Let $\bar{\ell}_i = \frac{1}{i} \ell_{err}\left(y_i\right) \prod_{k=1}^{i-1} (1 - \ell_{err}\left(y_l\right))$. We derive

$$\begin{aligned}
&\mathbb{E}\left[\left(L\left(\mathbf{r}(x),\mathbf{y}\right)-R_\theta\right)^2\middle| x;\theta\right]\\
&=\mathbb{E}\left[\sum_{i=1}^{|\mathbf{r}(x)|}\bar{\ell}_i^2+2\sum_{i=1}^{|\mathbf{r}(x)|}\sum_{l=i+1}^{|\mathbf{r}(x)|}\bar{\ell}_i\bar{\ell}_l-2R_\theta\sum_{i=1}^{|\mathbf{r}(x)|}\bar{\ell}_i+R_\theta^2\right] \qquad (21)\\
&=\sum_{i=1}^{|\mathbf{r}(x)|}\left(\mathbb{E}\left[\bar{\ell}_i^2\middle| x;\theta\right]+2\sum_{l=i+1}^{|\mathbf{r}(x)|}\mathbb{E}\left[\bar{\ell}_i\bar{\ell}_l\middle| x;\theta\right]-2R_\theta\mathbb{E}\left[\bar{\ell}_i\middle| x;\theta\right]\right)+R_\theta^2. \qquad (22)
\end{aligned}$$

We expand the square of sums twice in Equation 21. Equation 22 follows from the independence assumption (Equation 3). We note that for $l > i$ the following decomposition holds:

$$\bar{\ell}_l=\frac{1}{l}\ell_{err}\left(y_l\right)\left(\prod_{k=1}^{i-1}(1-\ell_{err}\left(y_k\right))\right)\left(1-\ell_{err}\left(y_i\right)\right)\left(\prod_{k=i+1}^{l-1}\left(1-\ell_{err}\left(y_k\right)\right)\right).$$

Thus, Equation 22 can be expressed as

$$\begin{aligned}
&\sum_{i=1}^{|\mathbf{r}(x)|}\left(\frac{1}{i^2}\mathbb{E}\left[\ell_{err}\left(y_i\right)^2\middle| x;\theta\right]\prod_{l=1}^{i-1}\mathbb{E}\left[(1-\ell_{err}\left(y_l\right))^2\middle| x;\theta\right]\right.\\
&\quad+2\frac{1}{i}\mathbb{E}\left[\ell_{err}\left(y_i\right)\left(1-\ell_{err}\left(y_i\right)\right)\middle| x;\theta\right]\left(\prod_{l=1}^{i-1}\mathbb{E}\left[\left(1-\ell_{err}\left(y_l\right)\right)^2\middle| x;\theta\right]\right)\\
&\quad\quad\cdot\left(\prod_{l=i+1}^{|\mathbf{r}(x)|}\mathbb{E}\left[\left(1-\ell_{err}\left(y_l\right)\right)\middle| x;\theta\right]\right)\sum_{k=i+1}^{|\mathbf{r}(x)|}\frac{\mathbb{E}\left[\ell_{err}\left(y_k\right)\middle| x;\theta\right]}{k\prod_{l=k}^{|\mathbf{r}(x)|}\mathbb{E}\left[\left(1-\ell_{err}\left(y_l\right)\right)\middle| x;\theta\right]}\\
&\quad\left.-2R_\theta\frac{1}{i}\mathbb{E}\left[\ell_{err}\left(y_i\right)\middle| x;\theta\right]\prod_{l=1}^{i-1}\mathbb{E}\left[\left(1-\ell_{err}\left(y_l\right)\right)\middle| x;\theta\right]\right)+R_\theta^2 \qquad (23)
\end{aligned}$$

Equation 23 can be evaluated in time $\mathcal{O}(|\mathcal{Y}||\mathbf{r}(x)|)$ as follows. We start by precomputing all cumulative products in time $\mathcal{O}(|\mathcal{Y}||\mathbf{r}(x)|)$ as shown above. The cumulative sums of the form $\sum_{k=i+1}^{|\mathbf{r}(x)|}...$ for $i = |\mathbf{r}(x)|-1,...,1$ can then be computed in overall time $\mathcal{O}(|\mathcal{Y}||\mathbf{r}(x)|)$. Given these precomputed quantities, the outer summation can then be carried out in time $\mathcal{O}(|\mathcal{Y}||\mathbf{r}(x)|)$ as well.

References

1. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems **20**(4) (2002) 422–446
2. Chapelle, O., Metzler, D., Zhang, Y., Grinspan, P.: Expected reciprocal rank for graded relevance. In: Proceeding of the Conference on Information and Knowledge Management. (2009)

3. Sawade, C., Bickel, S., Scheffer, T.: Active risk estimation. In: Proceedings of the 27th International Conference on Machine Learning. (2010)
4. Sawade, C., Landwehr, N., Scheffer, T.: Active comparison of prediction models. Unpublished Manuscript.
5. Cossock, D., Zhang, T.: Statistical analysis of Bayes optimal subset ranking. IEEE Transactions on Information Theory **54**(11) (2008) 5140–5154
6. Mohan, A., Chen, Z., Weinberger, K.: Web-search ranking with initialized gradient boosted regression trees. In: JMLR: Workshop and Conference Proceedings. Volume 14. (2011) 77–89
7. Geman, S., Bienenstock, E., Doursat, R.: Neural networks and the bias/variance dilemma. Neural Computation **4** (1992) 1–58
8. Liu, J.: Monte carlo strategies in scientific computing. Springer (2001)
9. Microsoft Research: Microsoft learning to rank datasets. http://research.microsoft.com/en-us/projects/mslr/. Released June 16 (2010)
10. Chapelle, O., Chang, Y.: Yahoo! Learning to rank challenge overview. JMLR: Workshop and Conference Proceedings **14** (2011) 1–24
11. Breiman, L.: Random forests. Machine learning **45**(1) (2001) 5–32
12. Li, P., Burges, C., Wu, Q.: Learning to rank using classification and gradient boosting. In: Advances in Neural Information Processing Systems. (2007)
13. McCullagh, P.: Regression models for ordinal data. Journal of the Royal Statistical Society. Series B (Methodological) **42**(2) (1980) 109–142
14. Sawade, C., Bickel, S., von Oertzen, T., Scheffer, T., Landwehr, N.: Active evaluation of ranking functions based on graded relevance. Technical report, University of Potsdam (2012)
15. Herbrich, R., Graepel, T., Obermayer, K.: Large margin rank boundaries for ordinal regression. Advances in Large Margin Classifiers (2000) 115–132
16. Zheng, Z., Zha, H., Zhang, T., Chapelle, O., Chen, K., Sun, G.: A general boosting method and its application to learning ranking functions for web search. In: Advances in Neural Information Processing Systems. (2007)
17. Burges, C.: RankNet to LambdaRank to LambdaMART: An overview. Technical Report MSR-TR-2010-82, Microsoft Research (2010)
18. Carterette, B., Allan, J., Sitaraman, R.: Minimal test collections for retrieval evaluation. In: Proceedings of the 29th SIGIR Conference on Research and Development in Information Retrieval. (2006)
19. Aslam, J., Pavlu, V., Yilmaz, E.: A statistical method for system evaluation using incomplete judgments. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval. (2006)
20. Carterette, B., Smucker, M.: Hypothesis testing with incomplete relevance judgments. In: Proceedings of the 16th ACM Conference on Information and Knowledge Management. (2007)
21. Bach, F.: Active learning for misspecified generalized linear models. In: Advances in Neural Information Processing Systems. (2007)
22. Beygelzimer, A., Dasgupta, S., Langford, J.: Importance weighted active learning. In: Proceedings of the International Conference on Machine Learning. (2009)