Unbiased Comparative Evaluation of Ranking Functions

Tobias Schnabel
Cornell University, Ithaca, NY
tbs49@cornell.edu

Adith Swaminathan
Cornell University, Ithaca, NY
adith@cs.cornell.edu

Peter I. Frazier
Cornell University, Ithaca, NY
pf98@cornell.edu

Thorsten Joachims
Cornell University, Ithaca, NY
tj@cs.cornell.edu

ABSTRACT

Eliciting relevance judgments for ranking evaluation is labor-intensive and costly, motivating careful selection of which documents to judge. Unlike traditional approaches that make this selection deterministically, probabilistic sampling has shown intriguing promise since it enables the design of estimators that are provably unbiased even when reusing data with missing judgments. In this paper, we first unify and extend these sampling approaches by viewing the evaluation problem as a Monte Carlo estimation task that applies to a large number of common IR metrics. Drawing on the theoretical clarity that this view offers, we tackle three practical evaluation scenarios: comparing two systems, comparing k systems against a baseline, and ranking k systems. For each scenario, we derive an estimator and a variance-optimizing sampling distribution while retaining the strengths of sampling-based evaluation, including unbiasedness, reusability despite missing data, and ease of use in practice. In addition to the theoretical contribution, we empirically evaluate our methods against previously used sampling heuristics and find that they generally cut the number of required relevance judgments at least in half.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Importance Sampling, Evaluation, Crowd Sourcing, Pooling

1. INTRODUCTION

Offline evaluation of retrieval systems requires annotated test collections that take substantial effort and cost to amass. The most significant cost lies in eliciting relevance judgments for query-document pairs, and the size of realistic test collections makes it infeasible to annotate every query-document pair in the corpus. This has spurred research on intelligent choice of pairs to judge. Analogous annotation problems also exist in other domains, like machine translation or sequence tagging in natural language processing. Moreover, with the advent of crowd-sourced annotations for applications like image recognition and protein sequencing, the problem of judgment elicitation has become even more relevant.

Unfortunately, if the whole corpus is not judged, the missing judgments may bias the performance estimates. There are two broad approaches to addressing this problem. The first develops new evaluation measures that are robust to incompletely judged test collections [21]. For such measures, heuristics like pooling can then be employed effectively, but the evaluation measure may not precisely capture the notion of quality one requires. The other approach is to leave the design of the evaluation measure as unrestricted as possible, but instead design general evaluation methodologies and estimators that guarantee unbiased estimates even under missing data [2, 27, 18, 28]. We follow this second approach, specifically focusing on sampling approaches that possess the following desirable properties:

1. Unbiasedness. On average, estimates have no systematic error, and behave like we had judged all query-document pairs.
2. Reusability. Collecting judgments is labor-intensive and costly, and sampling-based approaches allow reuse of past data without introducing bias.
3. Statelessness. We often need to collect tens of thousands of judgments. Sampling is embarrassingly parallel and can be done in a single batch.
4. Sample Efficiency. Sampling distributions can be designed to optimize the number of judgments needed to confidently and accurately estimate a metric.

In this paper, we focus on three comparative evaluation scenarios that frequently arise in practice, and derive new estimators and principled sampling strategies that substantially improve sample efficiency while retaining the other desirable properties of sampling-based evaluation. In particular, we investigate the problem of estimating the performance difference between two systems, the problem of estimating k systems' performance relative to a baseline, and the problem of estimating a ranking of k systems. For all three scenarios, we propose importance-weighted estimators and their variance-optimizing sampling distributions, enabling the estimators to elicit relative performance differences much more efficiently than previous sampling approaches. We show that these estimators apply to any lin-

ICTIR '16 Delaware USA
© 2016 ACM. ISBN 123-4567-24-567/08/06. . . $15.00
DOI: 10.475/123_4

early decomposable performance metric (e.g., DCG, Precision@k), and that they are unbiased even with missing judgments. In addition to these theoretical arguments, empirical results show that our estimators and samplers can substantially reduce the number of required judgments compared to previously used sampling heuristics, making them a practical and effective alternative to pooling and heuristic sampling methods. Beyond these specific contributions, the paper more generally contributes a unified treatment of all prior sampling approaches in terms of Monte Carlo estimation and its theory, providing a rigorous basis for future research.

2. RELATED WORK

The Cranfield methodology [7] using pooling [24] is the most established approach to IR evaluation. Pooling aims to be fair to all submitted systems by exhaustively judging the top-ranked document from all systems, hoping for good coverage of all relevant documents for a query (implicitly through diversity in the submitted runs). However, pooling bias is a well known problem when re-using these pooled collections to evaluate novel systems that retrieve relevant but unjudged documents [31] – see [22] and the references therein for a detailed overview. Attempts have been made to correct this pooling bias, either using a small set of exhaustively judged queries [26] or using sample corrections that apply for MAE and Precision@k [4, 13].

Generally, however, the size of today's corpora has prevented complete judging of test collections, and this has driven research on evaluation strategies that are robust to missing judgments [6]. One approach to handling incomplete judgments is to define an IR metric that is robust to missing judgments, like Bpref [3], RankEff [1] and Rank-Biased Precision [16]. Sakai and Kando [21] provide an excellent review of these approaches. Another approach, and the one we build on in this paper, uses random sampling of ranked documents to construct a collection of judgments [8, 2, 27, 18, 28]. We unify all these sampling approaches by viewing them as Monte Carlo estimates of IR metrics and extend them to relative comparisons.

The idea of relative comparisons rather than absolute evaluation of IR measures has been studied before. Deterministic elicitation schemes have been proposed for differentiating two systems based on AP [5], and to rank multiple systems according to Rank-Biased Precision [15]. More recently, multi-armed bandit approaches have been studied to construct judgment pools [14]. These schemes suffer from the same bias that plagues pooling when comparing new systems. We extend the provably unbiased sampling approaches to these comparative scenarios, inheriting the improved sample efficiency of relative comparisons while yielding re-usable test collections. We anticipate future work that combines the simplicity of batch sampling with the sample efficiency of active learning and bandit algorithms to adaptively elicit judgments.

We note that the sampling approach we take here naturally incorporates noisy judgments, making it suitable for many tasks involving crowd-sourcing. Existing works have heuristically resolved noisy judgments as a pre-processing step before constructing the test collection [10, 19].

Ideas from Monte Carlo estimation and importance sampling [17] have been successfully applied in closely related problems like unbiased recommender evaluation [23, 12], although in those applications, the sampling distribution is typically not under the experimenter's control. Finally, a related problem to the judgment elicitation problem we study here is that of picking the most informative set of queries [9]. Our Monte Carlo formulation can offer a reasonable starting point to answer this question as well.

3. SAMPLING BASED EVALUATION

To support our novel sampling approaches to comparative evaluation, described in the following three sections, we first lay out a unified framework of Monte Carlo estimation for sampling-based evaluation of a single system, unifying existing IR work [2, 27, 18, 28] with the extensive literature and theory in Monte Carlo estimation.

3.1 Illustrative Example

Consider a retrieval system $S(\boldsymbol{x})$ that maps each input query $\boldsymbol{x}$ to a ranking $\boldsymbol{y}$. Given a set of $|\boldsymbol{X}|$ queries $\boldsymbol{X}$, we would like to estimate the average Discounted Cumulative Gain (DCG) with depth cut-off 100 of S on $\boldsymbol{X}$,

$$\begin{aligned} DCG@100(S) &= \frac{1}{|\boldsymbol{X}|}\sum_{\boldsymbol{x}\in\mathcal{X}}\sum_{r=1}^{100}\frac{\mathrm{rel}(\boldsymbol{x},S(\boldsymbol{x})_r)}{\log(1+r)} \\ &= \sum_{(\boldsymbol{x},r)}\frac{\mathrm{rel}(\boldsymbol{x},S(\boldsymbol{x})_r)}{|\boldsymbol{X}|\cdot\log(1+r)} \\ &= \sum_{(\boldsymbol{x},r)} V(\boldsymbol{x},r). \end{aligned}$$

$S(\boldsymbol{x})_r$ is the document at rank r in the ranking of S for query $\boldsymbol{x}$, and $\mathrm{rel}(\cdot)$ denotes its assessed relevance. The key insight behind sampling-based evaluation is the following: we do not need to know all summands $V(\boldsymbol{x},r)$ in this large sum to get a good estimate of $DCG@100(S)$. In particular, even if we just uniformly at random sample n query-document pairs $D=((\boldsymbol{x}_1,r_1),...,(\boldsymbol{x}_n,r_n))$ and elicit the value of $V(\boldsymbol{x},r)$ for those, the following average is a reasonable estimate of $DCG@100(S)$ even when $n<100\cdot|\boldsymbol{X}|$.

$$DCG@100(S)\approx\frac{100\cdot|\boldsymbol{X}|}{n}\sum_{(\boldsymbol{x}_i,r_i)\in D}V(\boldsymbol{x}_i,r_i).$$

But what are the quality guarantees we can give for such estimates? Is uniform sampling the best we can do? What about other performance measures? And what about statistical testing and comparisons of multiple systems? To address these questions, we now formalize sampling-based evaluation in the framework of Monte Carlo estimation.

3.2 Formalizing Evaluation

A ranking system $S(\boldsymbol{x})$ (e.g. search engine, recommendation system) maps an input $\boldsymbol{x}\in\mathcal{X}$ (e.g. query, user context) to a ranking $\boldsymbol{y}$. Each predicted ranking $\boldsymbol{y}=S(\boldsymbol{x})$ has a certain utility $U(\boldsymbol{x},\boldsymbol{y})$ for a given $\boldsymbol{x}$ which quantifies the quality of ranking $\boldsymbol{y}$ for $\boldsymbol{x}$. To aggregate quality over multiple $(\boldsymbol{x},\boldsymbol{y})$ pairs, virtually all evaluation approaches use the expected utility over the distribution $\mathrm{P}(\boldsymbol{x})$ as a summary of the overall quality of a system.

$$U(S)=\mathbb{E}_{\mathrm{P}(\boldsymbol{x})}[U(\boldsymbol{x},\boldsymbol{y})]=\int U(\boldsymbol{x},\boldsymbol{y})d\,\mathrm{P}(\boldsymbol{x}),$$

where, again, $\boldsymbol{y}=S(\boldsymbol{x})$ refers to the output of S given $\boldsymbol{x}$. Most often though, we wish to evaluate *multiple* systems, i.e., we have a number of systems $\mathcal{S}=\{S_1,\dots S_k\}$ which

Metric	$u(\boldsymbol{x}, y)$	$\lambda(y \mid \boldsymbol{y})$
Prec@k	$\mathrm{rel}(\boldsymbol{x}, y) \in \{0,1\}$	$\mathbf{1}_{\mathrm{rank}(y) \leq k}/k$
DCG	$\mathrm{rel}(\boldsymbol{x}, y) \in [0, M]$	$1/\log(1+\mathrm{rank}\,(y))$
Gain@k	$\mathrm{rel}(\boldsymbol{x}, y) \in [0, M]$	$\mathbf{1}_{\mathrm{rank}(y) \leq k}/k$
MAE	$\lvert\mathrm{error}(\boldsymbol{x}, y)\rvert \in [0, M]$	$1/\lvert\boldsymbol{y}\rvert$
MSE	$(\mathrm{error}(\boldsymbol{x}, y))^2 \in [0, M]$	$1/\lvert\boldsymbol{y}\rvert$
RBP-p [16]	$\mathrm{rel}(\boldsymbol{x}, y) \in \{0,1\}$	$(1-p)/p^{\mathrm{rank}(y)}$
#ofSwaps	$\mathrm{swapped}(\boldsymbol{x}, y) \in \{0,1\}$	$1/\lvert\boldsymbol{y}\rvert$
wSwaps	$\mathrm{swapped}(\boldsymbol{x}, y) \in \{0,1\}$	$1/(\lvert\boldsymbol{y}\rvert \cdot \mathrm{rank}(\tilde{y_1})$ $\cdot\, \mathrm{rank}(\tilde{y_2}))$
AP	$\mathrm{rel}(\boldsymbol{x}, \tilde{y}_1) \cdot \mathrm{rel}(\boldsymbol{x}, \tilde{y}_2)$ $\in \{0,1\}$	$\mathbf{1}_{\mathrm{rank}(\tilde{y_2}) \leq \mathrm{rank}(\tilde{y}_1)}$ $/\,(R \cdot \mathrm{rank}(\tilde{y}_1))$

Table 1: A selection of popular metrics that can be written as expectations over single item judgments (top) or pairwise judgments (bottom). $R = \sum_{\tilde{y}} \mathrm{rel}(x, \tilde{y})$**.**

we wish to evaluate on a distribution of inputs $\mathrm{P}(\boldsymbol{x})$. Taking web-search as an example, $\mathcal{S}$ would be a collection of retrieval systems, $\mathrm{P}(\boldsymbol{x})$ the distribution of queries, and the utility of a ranking $U(\boldsymbol{x}, \boldsymbol{y})$ would be measured by some IR metric like Precision@10. We will now discuss how to obtain values of $U(\boldsymbol{x}, \boldsymbol{y})$.

3.3 Linearly Decomposable Metrics

Since assessing the utility $U(\boldsymbol{x}, \boldsymbol{y})$ of a complete ranking $\boldsymbol{y}$ is difficult for human assessors, the utility of $\boldsymbol{y}$ is typically aggregated from the utilities of the individual documents it contains. Formally, we assume U decomposes as a sum of the utilities of the parts

$$U(\boldsymbol{x}, \boldsymbol{y}) = \sum_{y \in \boldsymbol{y}} \lambda(y \mid \boldsymbol{y})\, u(\boldsymbol{x}, y). \tag{1}$$

The weights $\lambda(y \mid \boldsymbol{y}) \geq 0$ with which the utility of each document y of ranking $\boldsymbol{y}$ enters the overall utility is defined by the particular performance measure. The utilities $u(\boldsymbol{x}, y) \in \mathbb{R}^+$ refer to the individual utilities of each part y in $\boldsymbol{y}$. Taking Precision@k as an example, $u(\boldsymbol{x}, y) = \mathrm{rel}(\boldsymbol{x}, y) \in \{0,1\}$ denotes binary relevance of document $\boldsymbol{y}$ for query $\boldsymbol{x}$, and the weights $\lambda(y \mid \boldsymbol{y})$ are $1/k$ if y appears among the top k documents in the ranking $\boldsymbol{y}$, and zero otherwise.

The top half of Table 1 shows more examples of performance measures that are linear functions of the individual parts y of $\boldsymbol{y}$. The bottom half of Table 1 presents examples of performance measures whose natural decomposition of $\boldsymbol{y}$ is into pairs of variables, i.e., $y = (\tilde{y}_1, \tilde{y}_2)$. These examples demonstrate the wide applicability of the decomposition in Eq. (1). Furthermore, one can estimate normalized measures (e.g. AP, NDCG) by taking ratios of estimated U at the expense of a typically small bias [2, 28]. Other structured prediction tasks, like sequence labeling, parsing, and network prediction, use similar part-based performance measures, and much of what we discuss can be extended to evaluation problems where $\boldsymbol{y}$ is a more general structured object (e.g. sequence, parse tree).

3.4 Evaluation as Monte Carlo Estimation

Previous work has realized that one can use sampling over the documents y in the rankings $\boldsymbol{y} = S(\boldsymbol{x})$ to estimate Average Precision and NDCG [2, 27, 18]. However, the idea of sampling over the components y applies not only to these performance measures, but to any linearly decomposable performance measure that can be written in the form of Eq. (1). Making the connection to Monte Carlo methods, we start by defining the following distribution over the documents y of a ranking $\boldsymbol{y} = S(\boldsymbol{x})$,

$$\Pr(y \mid \boldsymbol{x}; S) = \frac{\lambda(y \mid S(\boldsymbol{x}))}{\sum_{y'} \lambda(y' \mid S(\boldsymbol{x}))}.$$

To simplify the exposition, we assume that the weights are scaled to sum to 1 (i.e., $\sum_{y'} \lambda(y' \mid S(\boldsymbol{x})) = 1$) for all systems S and inputs $\boldsymbol{x}$. We can now replace the sum over the components with its expectation,

$$\begin{aligned} U(S) &= \mathbb{E}_{\mathrm{P}(\boldsymbol{x})} \sum_{y \in S(\boldsymbol{x})} \lambda(y \mid S(\boldsymbol{x}))\, u(\boldsymbol{x}, y) \\ &= \mathbb{E}_{\mathrm{P}(\boldsymbol{x})} \mathbb{E}_{\Pr(y \mid \boldsymbol{x}; S)}[u(\boldsymbol{x}, y)]. \end{aligned} \tag{2}$$

This expectation can now be estimated via Monte Carlo, since we can sample from both $\mathrm{P}(\boldsymbol{x})$ and $\Pr(y|\boldsymbol{x}, S)$ without expensive utility assessments.

3.5 Importance Sampling Estimators

To obtain an unbiased sampling-based estimate of $U(S)$ in Eq. (2), one could simply sample queries and documents from $\mathrm{P}(\boldsymbol{x}) \cdot \Pr(y|\boldsymbol{x}, S)$ and average the results. However, this naive strategy has two drawbacks. First, to evaluate each new system S, it would sample documents from a new distribution, requiring additional expensive utility assessments. Second, there may be other sampling distributions that are statistically more efficient.

In principle, we can use any unbiased Monte Carlo technique to overcome these two drawbacks of naive sampling, and [28] have used stratified sampling. We deviate from their choice and focus on importance sampling for four reasons. First, importance sampling makes it straightforward to incorporate prior knowledge into the sampling distribution. This could be knowledge about the utility values $u(x, y)$ or about the systems being evaluated. Second, we can obtain confidence intervals with little additional overhead. Third, importance sampling offers a natural and simple way to reuse previously collected judgments for evaluating new systems. Finally, as we will show in this paper, the importance sampling framework extends naturally to scenarios involving concurrent evaluation of multiple systems, providing closed-form solutions that are easy to use in practice.

Central to importance sampling is the idea of defining a sampling distribution $Q(\boldsymbol{x}, y)$ that focuses on the regions of the space that are most important for accurate estimates. We consider the family of sampling distributions that first draws a sample $\boldsymbol{X}$ of $|\boldsymbol{X}|$ queries from $\mathrm{P}(\boldsymbol{x})$[1], and then samples query-document pairs with replacement from this set. This two-step sampling via $\boldsymbol{X}$ has the advantage that the assessment overhead of understanding a query can now be amortized over multiple judgments per query. Note that we may repeatedly sample the same query-document pair. See Section 3.7 for a discussion on whether to actually judge the same query-document pair more than once.

For a sample of n observations $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ drawn i.i.d. from a sampling distribution $Q(\boldsymbol{x}, y)$ and a target distribu-

[1] One may also consider sampling queries $\boldsymbol{x}$ from some other distribution than $\mathrm{P}(\boldsymbol{x})$, which may be beneficial if different types of queries contribute with different variability to the overall estimate [9, 20].

tion $\Pr(\boldsymbol{x}, y|S) = \Pr(y|\boldsymbol{x}, S)/|\boldsymbol{X}|$ for a given $\boldsymbol{X}$, the importance sampling estimator for Eq. (2) on $\boldsymbol{X}$ is

$$\hat{U}_n(S) = \frac{1}{n}\sum_{i=1}^{n} u(\boldsymbol{x}_i, y_i)\frac{\mathrm{P}(\boldsymbol{x}_i, y_i \mid S)}{Q(\boldsymbol{x}_i, y_i)}. \qquad (3)$$

Given any sampling distribution $Q(\boldsymbol{x}, y)$, applying this importance sampling estimator amounts to the following simple procedure:

1. Draw a sample $\boldsymbol{X}$ of $|\boldsymbol{X}|$ queries from $\mathrm{P}(\boldsymbol{x})$. Given a budget of n assessments,
   - draw query/document pair $(\boldsymbol{x}_i, y_i)$ from $Q(\boldsymbol{x}_i, y)$,
   - collect assessment $u(\boldsymbol{x}_i, y_i)$ and record $Q(\boldsymbol{x}_i, y_i)$.

   The result is a test collection

   $$\mathcal{D} = ((\boldsymbol{x}_i, y_i, u(\boldsymbol{x}_i, y_i), Q(\boldsymbol{x}_i, y_i))_{i=1}^{n}.$$

2. For systems $\mathcal{S} = \{S_1, \ldots S_k\}$, compute $\hat{U}_n(S_j)$ according to Eq. (3) using $\mathcal{D}$.

The following shows that this provides an unbiased estimate of $U(S)$ [17], if one ensures that $Q(\boldsymbol{x}, y)$ has sufficient support, i.e. $u(\boldsymbol{x}, y)P(\boldsymbol{x}, y|S) \neq 0 \Rightarrow Q(\boldsymbol{x}, y) > 0$.

$$\begin{aligned}
\mathbb{E}[\hat{U}_n(S)] &= \mathbb{E}_{\mathrm{P}(\boldsymbol{X})}\mathbb{E}_{Q(\boldsymbol{x}_i, y_i)}\left[\frac{1}{n}\sum_{i=1}^{n} u(\boldsymbol{x}_i, y_i)\frac{\mathrm{P}(\boldsymbol{x}_i, y_i \mid S)}{Q(\boldsymbol{x}_i, y_i)}\right] \\
&= \mathbb{E}_{\mathrm{P}(\boldsymbol{X})}\mathbb{E}_{Q(\boldsymbol{x}, y)}\left[u(\boldsymbol{x}, y)\frac{\mathrm{P}(\boldsymbol{x}, y \mid S)}{Q(\boldsymbol{x}, y)}\right] \\
&= \mathbb{E}_{\mathrm{P}(\boldsymbol{X})}\left[\sum_{\boldsymbol{x}, y} u(\boldsymbol{x}, y)\frac{\mathrm{P}(\boldsymbol{x}, y \mid S)}{Q(\boldsymbol{x}, y)}Q(\boldsymbol{x}, y)\right] \\
&= \mathbb{E}_{\mathrm{P}(\boldsymbol{X})}\left[\frac{1}{|\boldsymbol{X}|}\sum_{\boldsymbol{x} \in \boldsymbol{X}}\sum_{y} u(\boldsymbol{x}, y)\Pr(y \mid \boldsymbol{x}, S)\right] \\
&= \mathbb{E}_{\mathrm{P}(\boldsymbol{x})}\left[\sum_{y} u(\boldsymbol{x}, y)\Pr(y \mid \boldsymbol{x}, S)\right] \\
&= \mathbb{E}_{\mathrm{P}(\boldsymbol{x})}\mathbb{E}_{\Pr(y|\boldsymbol{x};S)}[u(\boldsymbol{x}, y)] \quad = \quad U(S).
\end{aligned}$$

Note that it is not necessary that any particular sample includes the assessment of all parts y of all system outputs $\boldsymbol{y} = S(\boldsymbol{x})$ to provide an unbiased estimate. For example, when estimating Precision@10, we may get far fewer than 10 judgments per query and still have an unbiased estimator.

3.6 Designing the Sampling Distribution Q

What remains to be addressed is the question of which sampling distribution $Q(\boldsymbol{x}, y)$ to use for sampling the documents to be assessed for each query in $\boldsymbol{X}$. Since every $Q(\boldsymbol{x}, y)$ with full support ensures unbiasedness, the key criterion for choosing $Q(\boldsymbol{x}, y)$ is statistical efficiency [2]. Concretely, the variance $\boldsymbol{Var}_Q[\hat{U}_n(S)]$ of the estimator $\hat{U}_n(S)$ governs efficiency and is our measure of estimation quality,

$$\Sigma(Q) = \boldsymbol{Var}_Q\left[\hat{U}_n(S)\right].$$

We therefore wish to pick a distribution Q that minimizes $\Sigma(Q)$. Let z_Q be short-hand for $u(\boldsymbol{x}, y)\frac{\mathrm{P}(\boldsymbol{x}, y|S)}{Q(\boldsymbol{x}, y)}$, then

$$\Sigma(Q) = \frac{1}{n}\boldsymbol{Var}_Q\left[z_Q\right] = \frac{1}{n}\left(\mathbb{E}_Q[z_Q^2] - (\mathbb{E}_Q[z_Q])^2\right).$$

The first equality follows since $(\boldsymbol{x}_i, y_i)$ are i.i.d. and $\hat{U}_n(S)$ is a sample mean, and the second expands the definition of variance. Notice that our earlier proof of unbiasedness implies that $\mu = \mathbb{E}_Q[z_Q] = \sum_{\boldsymbol{x}, y} u(\boldsymbol{x}, y)\,\mathrm{P}(\boldsymbol{x}, y|S)$ is a constant independent of Q.

A key result for importance sampling is that the following Q^* is optimal for minimizing $\Sigma(Q)$ [11, 2]:

$$Q^*(\boldsymbol{x}, y) = u(\boldsymbol{x}, y) \cdot \mathrm{P}(\boldsymbol{x}, y \mid S)/\mu. \qquad (4)$$

To see this, observe that for any other sampling distribution Q different from Q^*,

$$\begin{aligned}
\boldsymbol{Var}_{Q^*}\left[z_{Q^*}\right] + \mu^2 &= \sum_{\boldsymbol{x}, y}\frac{(u(\boldsymbol{x}, y)\,\mathrm{P}(\boldsymbol{x}, y \mid S))^2}{u(\boldsymbol{x}, y) \cdot \mathrm{P}(\boldsymbol{x}, y \mid S)/\mu} \\
&= \mu\sum_{\boldsymbol{x}, y} u(\boldsymbol{x}, y)\,\mathrm{P}(\boldsymbol{x}, y \mid S) \\
&= \left(\mathbb{E}_Q\left[z_Q\right]\right)^2 \\
&\leq \mathbb{E}_Q\left[{z_Q}^2\right] = \boldsymbol{Var}_Q\left[z_Q\right] + \mu^2.
\end{aligned}$$

The first line is the definition of $\boldsymbol{Var}_{Q*}[z_{Q^*}^2]$, and the last line follows from Jensen's inequality.

Since we do not have access to the true $u(\boldsymbol{x}, y)$ in practice, one usually substitutes approximate utilities $\tilde{u}(\boldsymbol{x}, y)$ based on prior side information (e.g. the Okapi BM25 score of document y for query x) into Equation (4). Any $\tilde{u}(\boldsymbol{x}, y) > 0$ retains unbiasedness, and the better the estimates $\tilde{u}(\boldsymbol{x}, y)$, the better the efficiency of the estimator.

3.7 Practical Concerns

Using the importance sampling framework outlined above offers straightforward solutions to a number of matters of practical interest.

Noisy Utility Assessments. In practice, there is often no consensus on what the true utility $u(\boldsymbol{x}, y)$ is, and different assessors (and users) will have different opinions. An example is crowd-sourcing, where labels get consolidated from multiple noisy judges. Note that our framework naturally lends itself to these noisy settings, if we think of individual assessments $u(\boldsymbol{x}, y|a)$ as conditioned on the assessor a that is drawn from a distribution $\mathrm{P}(a)$ and define

$$v(\boldsymbol{x}, y) = \mathbb{E}_{\mathrm{P}(a)}[u(\boldsymbol{x}, y \mid a)].$$

Our estimator (3) stays the same, and remains unbiased by linearity of expectation. Also, the theoretical results for picking the optimal Q^* remain essentially unaltered. The only thing that changes is that we replace the true $u(\boldsymbol{x}, y)$ with the true expected utility $v(\boldsymbol{x}, y)$.

Reusing Existing Data. Given that any sampling distribution $Q(\boldsymbol{x}, y)$ with full support provides unbiased estimates, reusing old data $\mathcal{D} = ((\boldsymbol{x}_i, y_i, u(\boldsymbol{x}_i, y_i), Q(\boldsymbol{x}_i, y_i))_{i=1}^{n}$ is straightforward. To guarantee full support over all query-document pairs, we can use a mixture sampling distribution such as

$$Q(\boldsymbol{x}, y) \propto \tilde{u}(\boldsymbol{x}, y) \cdot P(\boldsymbol{x}, y \mid S) + \epsilon,$$

where ϵ is a small constant added to ensure Q has sufficiently heavy tails [29]. A larger ϵ will make the collected samples more reusable for any new system, but sacrifices statistical efficiency for evaluating the current S.

In addition, it is easy to see that not all data used in the estimator from Eq. (3) has to be collected using the same $Q(\boldsymbol{x}, y)$ or the same sample of queries $\boldsymbol{X}$, and that we can combine datasets that accumulate over time. As long as we keep track of Q for every $(\boldsymbol{x}_i, y_i)$ that was sampled, each sample may be drawn from a different $Q_i(\boldsymbol{x}_i, y_i)$, and we can use the mixture

$$Q(\boldsymbol{x}, y) = \frac{1}{n}\sum_{i=1}^{n} Q_i(\boldsymbol{x}, y), \tag{5}$$

in the denominator of our estimator. This is a direct application of the balance heuristic for Multiple Importance Sampling [17]. Furthermore, Equation (5) provides guidance on how to draw new samples given the distributions $Q_i(\boldsymbol{x}, y)$ of the existing $\mathcal{D}$, if we eventually want the overall data to be close to a particularly efficient distribution $Q^*(\boldsymbol{x}, y)$ for a new evaluation task.

Quantifying Evaluation Accuracy. An advantage of the sampling approach to evaluation is that it allows us to easily quantify the accuracy of the estimates on $\boldsymbol{X}$. For large sample sizes n, our estimate $\hat{U}_n(S)$ converges in distribution to a normal distribution. From the central limit theorem, we then obtain

$$\left[\hat{U}_n(S) - t_{\alpha/2}\frac{\hat{\sigma}_n(S)}{\sqrt{n}}, \hat{U}_n(S) + t_{\alpha/2}\frac{\hat{\sigma}_n(S)}{\sqrt{n}}\right] \tag{6}$$

as the approximate $1 - \alpha$ confidence interval. For example, a 95% confidence interval would be $\hat{U}_n(S) \pm 1.96\frac{\hat{\sigma}_n(S)}{\sqrt{n}}$. We can estimate $\hat{\sigma}_n(S)$ from the same samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ that we used to compute $\hat{U}_n(S)$ as follows:

$$\hat{\sigma}_n(S) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{u(\boldsymbol{x}_i, y_i)\,\mathrm{P}(\boldsymbol{x}_i, y_i \mid S)}{Q(\boldsymbol{x}_i, y_i)} - \hat{U}_n(S)\right)^2}.$$

Note that the rate of convergence of $\hat{U}_n(S)$ to its true value depends on the skewness of the distribution of z_Q, and we will empirically evaluate the quality of the confidence intervals in Section 7.2.

4. COMPARING TWO SYSTEMS

Up until now this paper has only considered the problem of estimating the performance of one system in isolation. In practice, however, we are typically much more interested in the relative performance of multiple systems. In the case of two systems S and S', we may be interested in measuring how much they differ in performance $\Delta(S, S') = U(S) - U(S')$. To this effect, we consider the estimator

$$\hat{\Delta}_n(S, S') = \frac{1}{n}\sum_{i=1}^{n} u(\boldsymbol{x}_i, y_i)\frac{\mathrm{P}(\boldsymbol{x}_i, y_i \mid S) - \mathrm{P}(\boldsymbol{x}_i, y_i \mid S')}{Q(\boldsymbol{x}_i, y_i)}. \tag{7}$$

Again, this estimator is unbiased, and it can be computed using data $\mathcal{D} = ((\boldsymbol{x}_i, y_i, u(\boldsymbol{x}_i, y_i), Q(\boldsymbol{x}_i, y_i))_{i=1}^{n}$ sampled from any Q with sufficient support. But what does the most efficient sampling distribution Q^* look like? Like in the single system case, the sample efficiency of the estimator is governed by its variance,

$$\Sigma(Q) = \boldsymbol{Var}_Q\left[\hat{\Delta}_n(S, S')\right]. \tag{8}$$

Analogous to $\hat{U}_n(S)$, $\hat{\Delta}_n(S, S')$ is the average of n i.i.d. random variables $z_Q = u(\boldsymbol{x}, y)\frac{\mathrm{P}(\boldsymbol{x}, y|S) - \mathrm{P}(\boldsymbol{x}, y|S')}{Q(\boldsymbol{x}, y)}$. So, $\Sigma(Q)$ is proportional to $\boldsymbol{Var}_Q[z_Q]$ and the only term that depends on Q is $\mathbb{E}_Q\left[(z_Q)^2\right]$. Following a similar argument as before, the optimal sampling distribution is

$$Q^*(\boldsymbol{x}, y) \propto u(\boldsymbol{x}, y) \cdot \left|P(\boldsymbol{x}, y|S) - P(\boldsymbol{x}, y \mid S')\right|. \tag{9}$$

$\mathbb{E}_{Q^*}[(z_{Q^*})^2] = \mathbb{E}_Q[|z_Q|]^2 \le \mathbb{E}_Q[(z_Q)^2]$ by Jensen's inequality, so Q^* from Equation (9) minimizes $\Sigma(Q)$. Unfortunately, we again cannot compute Q^* because it needs $u(\boldsymbol{x}, y)$, but we can substitute an approximate $\tilde{u}(\boldsymbol{x}, y)$ as before.

Note that this Q^* is very intuitive – items that have similar weights $P(\boldsymbol{x}, y|\cdot)$ in both systems will get sampled with a low probability, since they have negligible effect on the performance difference. This Q^* is different from the heuristic Q previously used in multi-system evaluation [28], where Q is simply the average of $P(\boldsymbol{x}, y|S)$ and $P(\boldsymbol{x}, y|S')$. In particular, this heuristic Q fails to recognize that documents at identical positions in both rankings contribute no information about the performance difference. In Section 7.3 we empirically compare against this heuristic Q.

5. COMPARING MULTIPLE SYSTEMS TO A BASELINE

We now consider another evaluation use case that is frequently encountered in practice. We have a current production system S' and several new candidate systems $\mathcal{S} = \{S_1, \ldots S_k\}$. The goal of evaluation is to estimate by how much each candidate improves (or not) over the baseline S'.

We can formulate this goal in terms of k comparative evaluation problems $\Delta(S_i, S') = U(S_j) - U(S')$, and we want reliable estimates for all performance differences $\Delta(S_1, S')$, ..., $\Delta(S_k, S')$. We can use the estimator $\hat{\Delta}_n(S_j, S')$ from Equation (7) for each $\Delta(S_j, S')$, and the procedure for computing it is identical to Section 4. This is one of the strengths of the sampling approach: once a sampling distribution Q is designed, multiple systems and multiple differences can be concurrently evaluated from one batch of judgments using a unified, unbiased procedure.

But what is the optimal Q^* for this new use case? Since we are now considering k estimates in parallel, we consider the sum of estimator variances as our measure of estimation quality to optimize:

$$\Sigma(Q) = \sum_{j=1}^{k}\boldsymbol{Var}_Q\left[\hat{\Delta}_n(S_j, S')\right]. \tag{10}$$

We will show that the distribution minimizing $\Sigma(Q)$ is

$$Q^*(\boldsymbol{x}, y) \propto u(\boldsymbol{x}, y) \cdot \sqrt{\sum_{j=1}^{k}(P(\boldsymbol{x}, y \mid S_j) - P(\boldsymbol{x}, y \mid S'))^2}. \tag{11}$$

To see this, collect the terms of $\Sigma(Q)$ that depend on Q (up to a scaling constant n) and denote them as,

$$T(Q) = \mathbb{E}_Q\left[\left(\frac{u(\boldsymbol{x}, y)}{Q(\boldsymbol{x}, y)}\right)^2 \sum_{j=1}^{k}(P(\boldsymbol{x}, y \mid S_j) - P(\boldsymbol{x}, y \mid S'))^2\right].$$

We can then show that

$$T(Q^*) = \left[\sum_{\boldsymbol{x}, y} u(\boldsymbol{x}, y)\sqrt{\sum_{j=1}^{k}(P(\boldsymbol{x}, y \mid S_j) - P(\boldsymbol{x}, y \mid S'))^2}\right]^2$$

$$= \mathbb{E}_Q\left[\frac{u(\boldsymbol{x}, y)}{Q(\boldsymbol{x}, y)}\sqrt{\sum_{j=1}^{k}(P(\boldsymbol{x}, y \mid S_j) - P(\boldsymbol{x}, y \mid S'))^2}\right]^2$$

$$\le \mathbb{E}_Q\left[\left(\frac{u(\boldsymbol{x},y)}{Q(\boldsymbol{x},y)}\right)^2 \sum_{j=1}^{k}(P(\boldsymbol{x},y \mid S_j) - P(\boldsymbol{x},y \mid S'))^2\right]$$
$$= T(Q).$$

As before we use an approximate $\tilde{u}(\boldsymbol{x},y)$ for a computable sampling distribution in our experiments.

Our results here complement recent work in multidimensional importance sampling [30] which studies multidimensional $u(\boldsymbol{x},y)$ and a single target distribution $P(\boldsymbol{x},y|S)$. The choice of sum-of-variances as the $\Sigma(Q)$ objective yields a simple closed-form optimal sampling distribution Q^* for this evaluation scenario. Several other $\Sigma(Q)$ are possible (for instance, the maximum variance $\max_j \boldsymbol{Var}_Q[\hat{\Delta}_n(S_j,S')]$), however closed form Q^* that optimize these may not exist. We defer further study of different objectives characterizing estimation quality in these scenarios to future work.

6. RANKING MULTIPLE SYSTEMS

Finally, we consider the use-case of ranking a collection of systems $\mathcal{S} = \{S_1, \ldots S_k\}$ in order of their performance $\{U(S_1), \ldots, U(S_k)\}$. A first thought may be to estimate each $U(S_j)$ directly, which we call the absolute evaluation strategy. However, we can often do much better using comparative evaluations, since accurate ranking merely requires that we can estimate $\{U(S_1)+\delta, \ldots, U(S_k)+\delta\}$ up to some arbitrary constant δ.

Why should this comparative problem be more accurate than absolute evaluation? Imagine three systems, two of which are poor $(U(S_1), U(S_2) \simeq 0)$ while the third is good $(U(S_3) \gg 0)$. Consider the case where there are a lot of "easy" documents that all three system rank correctly, and that the large difference between S_3 and systems S_1/S_2 is due to a few "hard" documents where S_3 is superior. Furthermore, S_1 and S_2 may be small variants of the same system that produce almost identical rankings. Designing Q^* to optimize for the sum of absolute variances $\sum_{j=1}^{3} \boldsymbol{Var}_Q[\hat{U}_n(S_j)]$ would be ignorant to all this structure.

To design a more informed estimator and sampler for ranking, we propose to merely estimate each system's performance relative to some baseline system τ,

$$\Delta(S_1,\tau), \ldots \Delta(S_k,\tau), \tag{12}$$

and then rank the systems using the estimator $\hat{\Delta}(S_i,\tau)$ from Equation (7). We could then use the optimal sampling distribution Q^* from Section 5 to sample query-document pairs and collect judgments.

What remains to be shown is how to construct the optimal baseline system τ. Using

$$\Sigma(\tau) = \sum_{j=1}^{k} \boldsymbol{Var}_{Q^*}\left[\hat{\Delta}_n(S_j,\tau)\right]$$

as our measure of estimator quality analogous to the previous sections, we prove that the optimal τ corresponds to

$$P(\boldsymbol{x},y|\tau) = \frac{1}{k}\sum_{j=1}^{k} P(\boldsymbol{x},y \mid S_j)$$

for each query-document pair $(\boldsymbol{x},y)$.

Proof. From the results of Section 5, we know that for any system τ, the optimal sampling distribution $Q^*(x,y) \propto u(\boldsymbol{x},y) \cdot \sqrt{\sum_{j=1}^{k}(P(\boldsymbol{x},y|S_j) - P(\boldsymbol{x},y|\tau))^2}$. If we plug in this Q^* into $\Sigma(\tau)$ and simplify (note that all terms of the variance depend on τ),

$$\Sigma(\tau) = \left[\sum_{\boldsymbol{x},y} u(x,y)\sqrt{\sum_{j=1}^{k}(P(\boldsymbol{x},y \mid S_j) - P(\boldsymbol{x},y \mid \tau))^2}\right]^2 - \sum_{j=1}^{k}\left[\sum_{\boldsymbol{x},y} u(\boldsymbol{x},y)\{P(\boldsymbol{x},y \mid S_j) - P(\boldsymbol{x},y \mid \tau)\}\right]^2.$$

Let each $P(\boldsymbol{x},y|\tau)$ be a variable $\tau_{\boldsymbol{x},y}$ which we minimize over. $\Sigma(\tau)$ is convex in each $\tau_{\boldsymbol{x},y}$. A minimum requires that $\partial\Sigma(\tau)/\partial\tau_{\boldsymbol{x},y} = 0$, subject to $\tau_{\boldsymbol{x},y} \ge 0$ for all $(\boldsymbol{x},y)$ and $\sum_{\boldsymbol{x},y}\tau_{\boldsymbol{x},y} = 1$. To begin, $\partial\Sigma(\tau)/\partial\tau_{\boldsymbol{x}_i,y_i} = 0$ yields that $\forall(\boldsymbol{x}_i,y_i)$:

$$\sum_{\boldsymbol{x},y} u(\boldsymbol{x},y)\left[\sum_{j=1}^{k} P(\boldsymbol{x},y|S_j) - \tau_{\boldsymbol{x},y}\right] + \alpha\sum_{j=1}^{k} P(\boldsymbol{x}_i,y_i|S_j) - \tau_{\boldsymbol{x}_i,y_i} = 0$$
$$\text{where} \quad \alpha = \frac{\sum_{\boldsymbol{x},y} u(\boldsymbol{x},y)\sqrt{\sum_{j=1}^{k}(P(\boldsymbol{x},y \mid S_j) - \tau_{\boldsymbol{x},y})^2}}{\sqrt{\sum_{j=1}^{k}(P(\boldsymbol{x}_i,y_i \mid S_j) - \tau_{\boldsymbol{x}_i,y_i})^2}}.$$

Observe that $\tau_{\boldsymbol{x}_i,y_i} = \frac{1}{k}\sum_{j=1}^{k} P(\boldsymbol{x}_i,y_i|S_j)$ satisfies all the equations simultaneously, which completes the proof. □

Putting everything together, the optimal sampling distribution is

$$Q^*(\boldsymbol{x},y) \propto u(\boldsymbol{x},y)\sqrt{\sum_{j=1}^{k}\left(P(\boldsymbol{x},y|S_j) - \frac{1}{k}\sum_{i=1}^{k} P(\boldsymbol{x},y|S_i)\right)^2}, \tag{13}$$

and we can again use an approximate $\tilde{u}(\boldsymbol{x},y)$ in practice.

7. EXPERIMENTS

The following experiments evaluate to what extent the theoretical contributions developed in this paper impact evaluation accuracy empirically. We compare sampling-based approaches only in this section, recognizing that deterministic approaches do not deliver the same guarantees as laid out in the introduction. To study effects in isolation, we first explore our estimator and sampling distribution design principles on the problem of single-system evaluation. We then in turn consider the three comparative multi-system evaluation problems.

7.1 Datasets and Experiment Setup

We use two datasets for our experiments, *SYNTH* and *TREC*, which give us different levels of experimental control and different application scenarios. The SYNTH dataset is designed to resemble judgments as they occur in a recommender system. To create SYNTH, we generated a sample $\boldsymbol{X}$ of 6000 users (i.e. queries) and 2000 items (i.e. documents). The ground truth judgments $u(x,y) \in \{0,...4\}$ for each user/item pair were drawn from a categorical distribution whose parameters were drawn from a Dirichlet distribution with hyper-parameters $\alpha = (.54, .25, .175, .03, .005)$, so as to give high ratings a low probability. Based on this data, we created ranking systems S_j of different quality in the following way. S_{OPT} denotes the perfect ranking system where each user ranking $\boldsymbol{y}$ is sorted according to the true relevances $u(x,y)$. The $S_{SHIFT-m}$ ranking system shifts all rankings of S_{OPT} down by m entries; el-

	True $U(S)$	Shallow Pool	Deep Pool	Sampling $\hat{U}(S)$
OPT	284.40	16.98	284.48 ± 1.69	284.54 ± 1.22
REV-75	277.63	10.00	277.79 ± 1.75	277.58 ± 1.07
REV-150	271.32	8.51	271.21 ± 1.59	271.18 ± 0.97
SHIFT-5	274.94	4.72	275.03 ± 1.69	274.73 ± 1.10
SHIFT-7	269.87	0.00	269.90 ± 1.75	269.98 ± 1.12
mds08a3	6.96	1.93	7.32 ± 5.05	6.86 ± 0.76
nttd8ale	9.01	2.41	8.79 ± 5.30	8.97 ± 0.87
weaver2	7.48	1.94	8.40 ± 6.59	7.53 ± 0.83

Table 2: Mean and standard deviation of DCG estimates across 100 trials for all systems in SYNTH (top) and three randomly chosen systems in TREC (bottom). The first column shows the true $U(S)$ that ShallowPool, DeepPool, and $\hat{U}(S)$ from Equation (3) **(with approximate Q^* sampling) aim to estimate.**

ements that get shifted beyond the last position get reintroduced at the first. S_{REV-m} reverses the order of the top m elements in S_{OPT}. The collection of systems was $\mathcal{S} = \{S_{OPT}, S_{REV-75}, S_{REV-150}, S_{SHIFT-5}, S_{SHIFT-7}\}$. As evaluation measure, we use DCG@2000.

The TREC dataset contains binary relevance judgments and mimics a typical information retrieval scenario. To generate the dataset, we start with the TREC-8 ad hoc track [25], which comprises 149 systems that submitted rankings $\boldsymbol{y}$ of size 1000 as response to 50 queries $\boldsymbol{X}$. To eliminate unwanted biases due to unjudged documents which could confound our empirical evaluation, we only consider a random subset of 20 systems for which we ensured that all the top 100 documents of each of the 50 queries were fully judged. Correspondingly, we truncated all rankings to length 100 and evaluate in terms of DCG@100.

Approximate Utilities. If not noted otherwise, we use approximate utilities $\tilde{u}_S(\boldsymbol{x}, y)$ when designing the sampling distribution in all of our experiments. We use the following simple heuristic to define $\tilde{u}_S(\boldsymbol{x}, y)$ and we will evaluate empirically to what degree this can be improved. For SYNTH and any given system S, we use $\tilde{u}_S(\boldsymbol{x}, y) = 4 \cdot (1 - \frac{\text{rank}(S(\boldsymbol{x}),y)}{2000})$ – reflecting the fact that relevances ranged between 0 and 4. We have an analogously rank-decreasing function $\tilde{u}_S(\boldsymbol{x}, y) = \frac{16}{\text{rank}(S(\boldsymbol{x}),y)+34}$ for TREC. To define approximate utilities for a set of systems $\mathcal{S} = \{S_1, \ldots, S_k\}$, we simply average the $\tilde{u}_{S_j}(\boldsymbol{x}, y)$.

Comparing systems. When performing comparative evaluations in Sections 7.3 to 7.5, we focus on the most difficult comparisons in the following way. We rank all systems by their true performance $U(S)$ on our ground truth set, and then compare adjacent systems in Section 7.3 and a sliding window of five systems in Sections 7.4 and 7.5. In Section 7.4, the middle system is used as the baseline S'. Each experiment is replicated 100 times.

7.2 Empirical Verification of Design Principles

We first evaluate how the design of the sampling distribution $Q(x, y)$ affects the efficiency of the $\hat{U}(\boldsymbol{x}, y)$ estimator in Equation (3). To study the effect in isolation and in comparison to conventional pooling methods, we first focus on the single-system evaluation problem.

Comparison to Deterministic Pooling. To ground our experiments with respect to conventional practice, we start by comparing the sampling approach to deterministic pooling [24] historically used in TREC. The depth of the pool b, i.e., the number of top-b documents being judged, vs the number of queries $|\boldsymbol{X}|$ is the key design choice when pooling under a judgment budget. We use two pooling methods that choose this trade-off differently. Given a budget of n assessments, the *ShallowPool* methods gets judgments for the top $b' = \frac{n}{|\boldsymbol{X}|}$ items of each query and computes the average performance based on these judgments. The *DeepPool* method randomly selects $l' = \frac{n}{b}$ queries and judges all pooled documents for those queries.

	Q^*	Q^P	Q^{unif}
OPT	1.22	1.98	3.05
REV-75	1.07	1.45	2.64
REV-150	0.97	1.33	2.20
SHIFT-5	1.10	1.67	2.63
SHIFT-7	1.12	1.45	2.45
mds08a3	0.76	0.84	0.97
nttd8ale	0.87	0.98	1.18
weaver2	0.83	0.92	1.05

Table 3: Standard deviation of $\hat{U}(S)$ for DCG@k estimates under different sampling methods across 100 **trials for all systems in SYNTH (top) and three randomly chosen systems in TREC (bottom).**

Table 2 compares estimated DCG@k for the deterministic baselines ShallowPool and DeepPool with the sampling estimates $\hat{U}(S)$ from Eq. (4) with a total budget of $n = 5 \cdot |\boldsymbol{X}|$ assessments. We repeated each experiment 100 times, and average the estimates across all 100 runs. The error intervals correspond to the empirical standard deviations of the estimates. Note that ShallowPool has standard deviation of zero, since it is deterministic given $\boldsymbol{X}$.

Unsurprisingly, Table 2 shows that ShallowPool is heavily biased by the missing judgments and systematically underestimates the true $U(S)$ given in the first column. The bias of ShallowPool is so strong that even its average estimates do not reflect the correct ordering of the ranking functions on SYNTH.

Table 2 shows that our $\hat{U}(S)$ estimator is indeed unbiased. DeepPool is also unbiased, since it has a non-zero probability of eliciting judments for any returned document. However, $\hat{U}(S)$ has substantially lower standard deviation especially on TREC. This is not surprising, since DeepPool suffers from between-query variability when sampling only a few queries from $\boldsymbol{X}$. In fact, the standard deviations of DeepPool are much larger than the estimated differences on TREC, indicating that DeepPool cannot reliably distinguish the ranking performance of the systems.

Impact of Prior Knowledge. We now study the impact of the sampling distribution on sampling efficiency. Table 3 compares three sampling approaches: Q^* with approximate utilities as in the previous experiment, Q^P with $\tilde{u}(x, y) = 1$, and uniform sampling Q^{unif}.

We see in Table 3 that the standard deviation increases as we transition from Q^* to Q^P, and it increases even further if we sample uniformly in Q^{unif}. Note that an estimator with half the standard deviation requires only roughly half the sample size to reach the same level of estimation accuracy.

	$\hat{U}(S)$	ConfInt	$\hat{P}_{\text{cov}}$
OPT	284.54 ± 1.22	± 2.21	0.92
REV-75	277.58 ± 1.07	± 2.10	0.96
REV-150	271.18 ± 0.97	± 2.05	0.95
SHIFT-5	274.73 ± 1.10	± 2.18	0.93
SHIFT-7	269.98 ± 1.12	± 2.18	0.94
mds08a3	6.86 ± 0.76	± 1.50	0.95
nttd8ale	8.97 ± 0.87	± 1.65	0.94
weaver2	7.53 ± 0.83	± 1.58	0.95

Table 4: Average confidence intervals and coverage probabilities for the $\hat{U}(S)$ estimator using Q^*.

We conclude that the choice of Q is of great practical significance. This is in line with importance sampling theory that indicates that good estimates of $\tilde{u}$ can substantially improve estimation quality. Given these findings, all following experiments will be based on the Q^* sampler with approximate utilities.

Confidence Intervals. Another feature of the sampling-based approach is that one can quantify its accuracy using confidence intervals. But are the approximate confidence intervals proposed in Section 3.7 accurate in practice? For each of the 100 runs from Table 2, we computed 95% confidence intervals according to Equation (6). We also computed the coverage probability $\hat{P}_{\text{cov}}$ of the confidence intervals, defined as the number of times the true value was within the confidence interval. As we can see from Table 4, the coverage probabilities of the approximate intervals are usually very close to 95% as desired. We can also verify that the empirical standard deviation of $\hat{U}(S)$ is roughly 1/1.96 of the average confidence intervals.

7.3 Is pairwise comparative evaluation more efficient than individual evaluation?

We now evaluate how far comparative estimation can improve upon individual system evaluation. Prior work [28] uses $Q(\boldsymbol{x}, y) \propto \tilde{u}(\boldsymbol{x}, y)(\mathrm{P}(\boldsymbol{x}, y|S) + \mathrm{P}(\boldsymbol{x}, y|S'))/2$ which we compare to the pairwise $Q^*(\boldsymbol{x}, y)$ with approximate utilites from Equation (9).

Table 5 shows the estimator variance $\Sigma(Q)$ as defined in Equation (8) for both sampling methods (up to scaling by the sample size n). On both datasets, the pairwise Q^* substantially improves estimator accuracy over Naive sampling. Note that reducing estimator variance by a quarter corresponds to halving the sample size needed to get a particular level of estimation accuracy. The table also contains the variance $\Sigma(Q)$ when using the true utilities $u^*(\boldsymbol{x}, y)$ for the design of the sampling distribution instead of the approximate utilities $\tilde{u}(\boldsymbol{x}, y)$. Comparing the variance gains between Naive and Q^* to the gains we could get by moving from $\tilde{u}(\boldsymbol{x}, y)$ to $u^*(\boldsymbol{x}, y)$, we see that the pairwise Q^* has improved sample efficiency far more than we could still hope to improve by finding a better $\tilde{u}(\boldsymbol{x}, y)$ for these datasets. Note that all numbers in Table 5 were computed analytically, so there are no errorbars.

While Table 5 measures estimator accuracy in terms of mean-squared-error, we are often merely interested in the binary decision of which system is better, i.e. $\operatorname{sign}(\Delta(S, S'))$. Figure 1 compares the accuracy with which the sign of our estimate $\operatorname{sign}(\hat{\Delta}(S, S'))$ predicts the true order of the systems on our query sample $\boldsymbol{X}$ correctly, $Acc = \mathbb{I}[\Delta(S, S') \cdot \hat{\Delta}(S, S') > 0]$. On both datasets, the pairwise Q^* sampling outperforms Naive. The magnitude of the gain is largest on the SYNTH dataset, where the rankings produced by different systems are more similar to each other than for the TREC systems.

	SYNTH		TREC	
	$\tilde{u}(x, y)$	$u^*(x, y)$	$\tilde{u}(x, y)$	$u^*(x, y)$
Naive	2.15	1.13	6.60	4.65
pair Q^*	0.24	0.21	1.45	0.77

Table 5: Variance $\Sigma(Q) \cdot n$ defined in Equation (8) for the pairwise comparison problem. The table compares naive averaging with our optimal Q^* from Equation (9) under perfect and approximate knowledge of utilities.

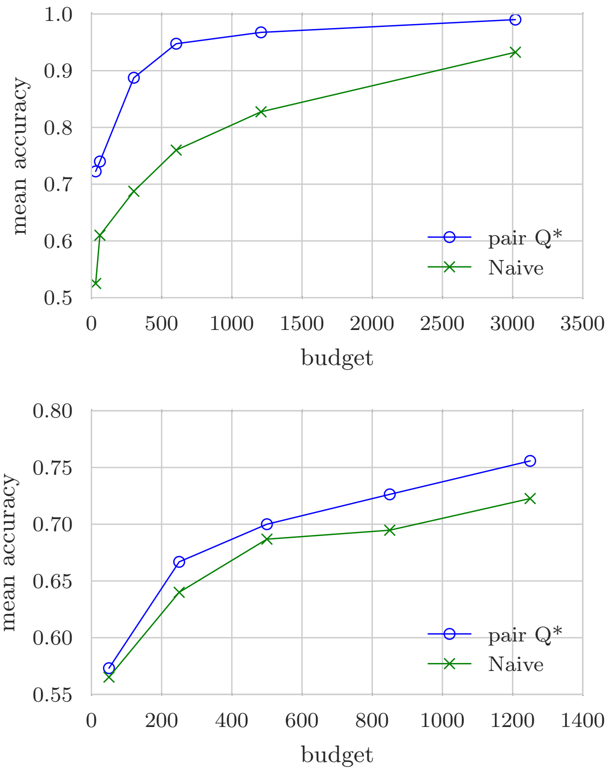

Figure 1: Mean pairwise accuracy on the SYNTH dataset (top) and the TREC dataset (bottom).

7.4 Is comparative evaluation against a baseline better than individual evaluation?

We now evaluate the statistical efficiency of our method for the problem of comparing multiple systems against a baseline. Table 2 shows the aggregate variance $\Sigma(Q)$ according to Eq. (10) computed analytically for both the SYNTH and the TREC datasets. The table shows that our Q^* for comparative evaluation as defined in Eq. (11) substantially outperforms the naive heuristic of sampling according to $Q(\boldsymbol{x}, y) \propto \tilde{u}(\boldsymbol{x}, y) \sum_j \mathrm{P}(\boldsymbol{x}, y|S_j)$ (which originates from Multiple Importance Sampling and Mixture Importance Sampling [17]). The gain is particularly large for the realistic

	SYNTH		TREC	
	$\tilde{u}(x,y)$	$u^*(x,y)$	$\tilde{u}(x,y)$	$u^*(x,y)$
Naive	1.31	0.70	15.08	1.77
comp Q^*	0.18	0.15	6.82	1.28

Table 6: Aggregate variance $\Sigma(Q) \cdot n$ defined in Equation (10) for the problem of comparing against a baseline. The table compares naive averaging with our optimal Q^* from Equation (11) under perfect and approximate knowledge of utilities.

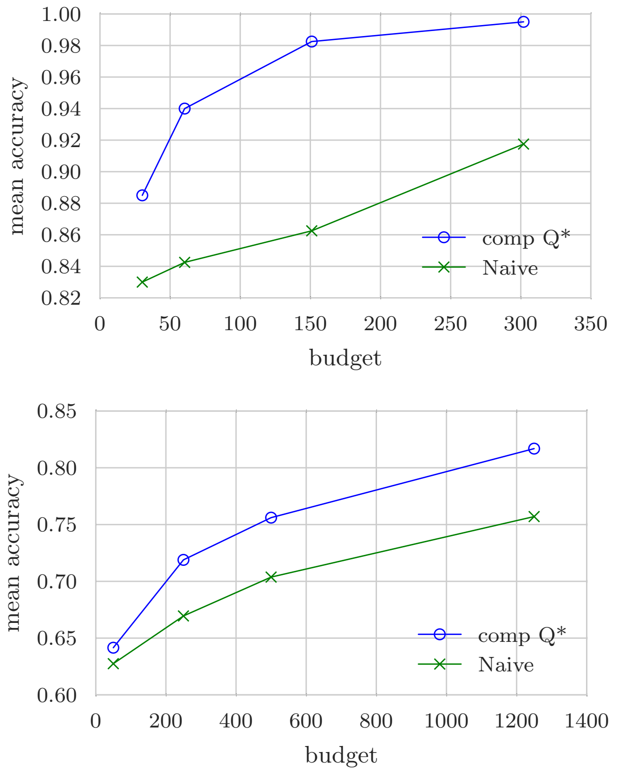

Figure 2: Average accuracy when comparing four new systems against a baseline system on the SYNTH dataset (top) and the TREC dataset (bottom).

case of approximate utilities.

Figure 2 evaluates the accuracy with which the estimated $\mathbb{I}[\hat{\Delta}(S_j, S') \geq 0]$ predicts the true $\mathbb{I}[\Delta(S_j, S') \geq 0]$. Mean accuracy is defined as $Acc = \mathbb{I}[\Delta(S_j, S') \cdot \hat{\Delta}(S_j, S') > 0]$ over all comparisons using the sliding window approach described in Section 7.1. Again, we see large gains in efficiency from using our Q^* over naive averaging – both samplers using approximate utilites – reducing the required sample size for a desired accuracy by a half or more on both datasets.

7.5 Is comparative ranking more efficient than individual evaluation?

Finally, we turn to the problem of estimating the relative performance of k systems. Following the pattern from the previous two subsections, we first compare our Q^* from Eq. (13) with the naive averaging sampler in terms of their aggregate variance $\Sigma(Q)$ as defined in Eq. (10). Sets of systems to compare were chosen using the sliding-window approach described in Section 7.1. Again, the gains in statistical efficiency are especially large for the practical case, where the utilities are approximated.

	SYNTH		TREC	
	$\tilde{u}(x,y)$	$u^*(x,y)$	$\tilde{u}(x,y)$	$u^*(x,y)$
Naive	0.86	0.46	38.64	1.79
rank Q^*	0.11	0.09	12.40	1.12

Table 7: Aggregate variance $\Sigma(Q) \cdot n$ defined in Equation (10) for the ranking problem. The table compares naive averaging with our optimal Q^* from Equation (13) under perfect and approximate knowledge of utilities.

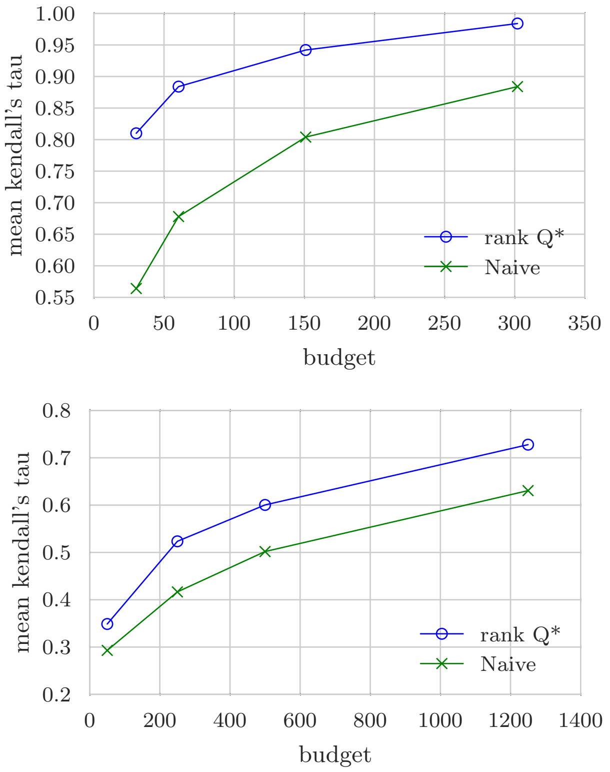

Figure 3: Kendall's tau for one set of 5 systems on SYNTH (top) and 16 sets of 5 systems on TREC (bottom).

Figure 3 evaluates to what extent this gain in mean squared estimation error translates into an improved accuracy for system ranking. In particular, we measure Kendall's tau when ranking by $\hat{\Delta}(S_j, \tau)$, which is identical to ranking by $\hat{U}(S_j)$. Analogous to the previous experiments, we see that our Q^* sampler for ranking is again substantially more accurate than naive sampling.

8. CONCLUSIONS AND FUTURE WORK

We developed a general and practical framework for evaluating ranking systems, making explicit the tight connections to Monte-Carlo estimation. This formal framework brings improved clarity and generality to the problem of sampling-based evaluation, including the design of estimators for new performance measures, conditions for unbiasedness, the reuse of data, and the design of sampling distributions. In particular, we focused on the question of how to design estimators and sampling distributions for comparative system evaluations, deriving variance-optimizing strategies for pairwise evaluation, comparing k-systems against a baseline, and ranking k systems. Empirical results show that these evaluation strategies lead to substantial improvements over previously used heuristics.

There are many directions for future work. First, much of the methodology should be applicable in other complex evaluation tasks as well, e.g., in natural language processing. Second, it would be interesting to see how to better approximate $u(x, y)$ by learning from previously collected judgments. This could potentially also be done in an adaptive fashion – similar to adaptive importance sampling. Third, there are various other evaluation questions involving k systems, e.g., (adaptive) sampling so as to maximize the probability of finding the best out of k systems.

9. ACKNOWLEDGMENTS

This work was supported in part through NSF Awards IIS-1247637, IIS-1217686, IIS-1513692, and a gift from Bloomberg.

10. REFERENCES

[1] P. Ahlgren and L. Grönqvist. Retrieval evaluation with incomplete relevance data: A comparative study of three measures. In *CIKM*, pages 872–873, 2006.

[2] J. A. Aslam, V. Pavlu, and E. Yilmaz. A statistical method for system evaluation using incomplete judgments. In *SIGIR*, pages 541–548, 2006.

[3] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR*, pages 25–32, 2004.

[4] S. Büttcher, C. L. A. Clarke, P. C. K. Yeung, and I. Soboroff. Reliable information retrieval evaluation with incomplete and biased judgements. In *SIGIR*, pages 63–70, 2007.

[5] B. Carterette, J. Allan, and R. Sitaraman. Minimal test collections for retrieval evaluation. In *SIGIR*, pages 268–275, 2006.

[6] B. Carterette, V. Pavlu, E. Kanoulas, J. A. Aslam, and J. Allan. If I had a million queries. In *ECIR*, pages 288–300, 2009.

[7] C. W. Cleverdon. The significance of the cranfield tests on index languages. In *SIGIR*, pages 3–12, 1991.

[8] G. V. Cormack and T. R. Lynam. Statistical precision of information retrieval evaluation. In *SIGIR*, pages 533–540, 2006.

[9] J. Guiver, S. Mizzaro, and S. Robertson. A few good topics: Experiments in topic set reduction for retrieval evaluation. *TOIS*, 27(4):21:1–21:26, 2009.

[10] M. Hosseini, I. J. Cox, N. Milic-Frayling, G. Kazai, and V. Vinay. On aggregating labels from multiple crowd workers to infer relevance of documents. In *ECIR*, pages 182–194, 2012.

[11] H. Kahn and A. W. Marshall. Methods of reducing sample size in monte carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278, 1953.

[12] L. Li, W. Chu, J. Langford, and X. Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *WSDM*, pages 297–306, 2011.

[13] A. Lipani, M. Lupu, and A. Hanbury. The curious incidence of bias corrections in the pool. In *ECIR*, pages 267–279, 2016.

[14] D. E. Losada, J. Parapar, and A. Barreiro. Feeling lucky? Multi-armed bandits for ordering judgements in pooling-based evaluation. In *SAC*, 2016.

[15] A. Moffat, W. Webber, and J. Zobel. Strategic system comparisons via targeted relevance judgments. In *SIGIR*, pages 375–382, 2007.

[16] A. Moffat and J. Zobel. Rank-biased precision for measurement of retrieval effectiveness. *TOIS*, 27(1):2:1–2:27, 2008.

[17] A. B. Owen. *Monte Carlo theory, methods and examples.* Draft, 2013.

[18] V. Pavlu and J. Aslam. A practical sampling strategy for efficient retrieval evaluation. Technical report, Northeastern University, 2007.

[19] D. D. Peng Ye. Combining preference and absolute judgements in a crowd-sourced setting. In *ICML Workshop: Machine Learning Meets Crowdsourcing*, 2013.

[20] S. E. Robertson and E. Kanoulas. On per-topic variance in IR evaluation. In *SIGIR*, pages 891–900, 2012.

[21] T. Sakai and N. Kando. On information retrieval metrics designed for evaluation with incomplete relevance assessments. *Information Retrieval*, 11(5):447–470, 2008.

[22] M. Sanderson. Test collection based evaluation of information retrieval systems. *Foundations and Trends in Information Retrieval*, 4(4):247–375, 2010.

[23] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, and T. Joachims. Recommendations as treatments: Debiasing learning and evaluation. *CoRR*, abs/1602.05352, 2016.

[24] K. Sparck-Jones and C. J. V. Rijsbergen. Report on the need for and provision of an "ideal" information retrieval test collection. Technical report, University of Cambridge, 1975.

[25] E. M. Voorhees and D. Harman. Overview of the ninth text retrieval conference (TREC-8). In *NIST Special Publication 500-246*, 1999.

[26] W. Webber and L. A. F. Park. Score adjustment for correction of pooling bias. In *SIGIR*, pages 444–451, 2009.

[27] E. Yilmaz and J. A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *CIKM*, pages 102–111, 2006.

[28] E. Yilmaz, E. Kanoulas, and J. A. Aslam. A simple and efficient sampling method for estimating AP and NDCG. In *SIGIR*, pages 603–610, 2008.

[29] C. Yuan and M. J. Druzdzel. How heavy should the tails be? In *FLAIRS*, pages 799–805, 2005.

[30] P. Zhao and T. Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *ICML*, pages 1–9, 2015.

[31] J. Zobel. How reliable are the results of large-scale information retrieval experiments? In *SIGIR*, pages 307–314, 1998.