Active Learning for Ranking through Expected Loss Optimization

Bo Long
Yahoo! Labs
Sunnyvale, CA
bolong@yahoo-inc.com

Olivier Chapelle
Yahoo! Labs
Sunnyvale, CA
chap@yahoo-inc.com

Ya Zhang
Shanghai Jiao Tong University
Shanghai, China
yazhang@sjtu.edu.cn

Yi Chang
Yahoo! Labs
Sunnyvale, CA
yichang@yahoo-inc.com

Zhaohui Zheng
Yahoo! Labs
Sunnyvale, CA
zhaohui@yahoo-inc.com

Belle Tseng
Yahoo! Labs
Sunnyvale, CA
belle@yahoo-inc.com

ABSTRACT

Learning to rank arises in many information retrieval applications, ranging from Web search engine, online advertising to recommendation system. In learning to rank, the performance of a ranking model is strongly affected by the number of labeled examples in the training set; on the other hand, obtaining labeled examples for training data is very expensive and time-consuming. This presents a great need for the active learning approaches to select most informative examples for ranking learning; however, in the literature there is still very limited work to address active learning for ranking. In this paper, we propose a general active learning framework, Expected Loss Optimization (ELO), for ranking. The ELO framework is applicable to a wide range of ranking functions. Under this framework, we derive a novel algorithm, Expected DCG Loss Optimization (ELO-DCG), to select most informative examples. Furthermore, we investigate both query and document level active learning for raking and propose a two-stage ELO-DCG algorithm which incorporate both query and document selection into active learning. Extensive experiments on real-world Web search data sets have demonstrated great potential and effectiveness of the proposed framework and algorithms.

Categories and Subject Descriptors

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

General Terms

Algorithms, Design, Theory, Experimentation

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.
Copyright 2010 ACM 978-1-60558-896-4/10/07 ...$10.00.

Keywords

Active Learning, Ranking, Expected Loss Optimization

1. INTRODUCTION

Ranking is the core component of many important information retrieval problems, such as web search, recommendation, computational advertising. Learning to rank represents an important class of supervised machine learning tasks with the goal of automatically constructing ranking functions from training data. As many other supervised machine learning problems, the quality of a ranking function is highly correlated with the amount of labeled data used to train the function. Due to the complexity of many ranking problems, a large amount of labeled training examples is usually required to learn a high quality ranking function. However, in most applications, while it is easy to collect unlabeled samples, it is very expensive and time-consuming to label the samples.

Active learning comes as a paradigm to reduce the labeling effort in supervised learning. It has been mostly studied in the context of classification tasks [19]. Existing algorithms for learning to rank may be categorized into three groups: pointwise approach [8], pairwise approach [25], and listwise approach [21]. Compared to active learning for classification, active learning for ranking faces some unique challenges. First, there is no notion of classification margin in ranking. Hence, many of the margin-based active learning algorithms proposed for classification tasks are not readily applicable to ranking. Further more, even some straightforward active learning approach, such as query-by-committee, has not been justified for the ranking tasks under regression framework. Second, in most supervised learning setting, each data sample can be treated completely independent of each other. In learning to rank, data examples are not independent, though they are conditionally independent given a query. We need to consider this data dependence in selecting data and tailor active learning algorithms according to the underlying learning to rank schemes. There is a great need for an active learning framework for ranking.

In this paper, we attempt to address those two important and challenging aspects of active learning for ranking. We first propose a general active learning framework, Expected

Loss Optimization (ELO), and apply it to ranking. The key idea of the proposed framework is that given a loss function, the samples minimizing the expected loss are the most informative ones. Under this framework, we derive a novel active learning algorithm for ranking, which uses function ensemble to select most informative examples that minimizes a chosen loss. For the rest of the paper, we use web search ranking problem as an example to illustrate the ideas and perform evaluation. But the proposed method is generic and may be applicable to all ranking applications. In the case of web search ranking, we minimize the expected DCG loss, one of the most commonly used loss for web search ranking. This algorithm may be easily adapted to other ranking loss such as NDCG or Average Precision. To address the data dependency issue, the proposed algorithm is further extended to a two-stage active learning schema to seamlessly integrate query level and document level data selection.

The main contributions of the paper are summarized as follows.

- We propose a general active learning framework based on expected loss optimization. This framework is applicable to various ranking scenarios with a wide spectrum of learners and loss functions. We also provides a theoretically justified measure of the informativeness .

- Under the ELO framework, we derive novel algorithms to select most informative examples by optimizing the expected DCG loss. Those selected examples represent the ones that the current ranking model is most uncertain about and they may lead to a large DCG loss if predicted incorrectly.

- We propose a two stage active learning algorithm for ranking, which addresses the sample dependence issue by first performing query level selection and then document level selection.

2. RELATED WORK

The main motivation for active learning is that it usually requires time and/or money for the human expert to label examples and those resources should not be wasted to label non-informative samples, but be spent on interesting ones.

Optimal Experimental Design [12] is closely related to active learning as it attempts to find a set of points such that the variance of the estimate is minimized. In contrast to this "batch" formulation, the term *active learning* often refers to an incremental strategy [7].

There has been various types of strategies for active learning that we now review. A comprehensive survey can be found in [20]. The simplest and maybe most common strategy is *uncertainty sampling* [18], where the active learning algorithm queries points for which the label uncertainty is the highest. The drawback of this type of approach is that it often mixes two types of uncertainties, the one stemming from the *noise* and the *variance*. The noise is something intrinsic to the learning problem which does not depend on the size of the training set. An active learner should not spend too much effort in querying points in noisy regions of the input space. On the other hand, the variance is the uncertainty in the model parameters resulting from the finiteness of the training set. Active learning should thus try to minimize this variance and this was first proposed in [7].

In Bayesian terms, the variance is computed by integrating over the posterior distribution of the model parameters. But in practice, it may be difficult or impossible to compute this posterior distribution. Instead, one can randomly sample models from the posterior distribution [9]. An heuristic way of doing so is to use a bagging type of algorithm [1]. This type of analysis can be seen as an extension of the *Query-By-Committee* (QBC) algorithm [14] which has been derived in a noise free classification setting. In that case, the posterior distribution is uniform over the *version space* – the space of consistent hypothesis with the labeled data – and the QBC algorithm selects points on which random functions in the version space have the highest disagreement.

Another fairly common heuristic for active learning is to select points that once added in the training set are expected to result in a large model change [20] or a large increase in the objective function value that is being optimized [4].

Compared with traditional active learning, there is still limited work on the active learning for ranking. Donmez and Carbonell studied the problem of document selection in ranking [11]. Their algorithm selects the documents which, once added to the training set, are the most likely to result in a large change in the model parameters of the ranking function. They apply their algorithm to RankSVM [17] and RankBoost [13]. Also in the context of RankSVM, [24] suggests to add the most ambiguous pairs of documents to the training set, that is documents whose predicted relevance scores are very close under the current model. Other works based on pairwise ranking include [6, 10]. In case of binary relevance, [5] proposed a greedy algorithm which selects document that are the most likely to differentiate two ranking systems in terms of average precision. Finally, an empirical comparison of document selection strategies for learning to rank can be found in [2].

There are some related works about query sampling. [23] empirically shows that having more queries but less number of documents per query is better than having more documents and less queries. Yang et. al. propose a greedy query selection algorithm that tries to maximize a linear combination of query difficulty, query density and query diversity [22].

3. EXPECTED LOSS OPTIMIZATION FOR ACTIVE LEARNING

As explained in the previous section, a natural strategy for active learning is based on variance minimization. The variance, in the context of regression, stems from the uncertainty in the prediction due to the finiteness of the training set. Cohn *et. al* [7] proposes to select the next instance to be labeled as the one with the highest variance. However, this approach applies only to regression and we aim at generalizing it through the *Bayesian expected loss* [3].

In the rest of the section, we first review Bayesian decision theory in section 3.1 and then introduce the *Expected Loss Optimization* (ELO) principle for active learning. In section 3.2 we show that in the cases of classification and regression, applying ELO turns out to be equivalent to standard active learning method. Finally, we present ELO for ranking in section 3.3.

3.1 Bayesian Decision Theory

We consider a classical Bayesian framework to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ parametrized by a vector θ. The training data D is made of n examples, $(x_1, y_1), \ldots, (x_n, y_n)$. Bayesian learning consists in:

1. Specifying a prior $P(\theta)$ on the parameters and a likelihood function $P(y|x, \theta)$.
2. Computing the likelihood of the training data, $P(D|\theta) = \prod_{i=1}^{n} P(y_i|x_i, \theta)$.
3. Applying Bayes rule to get the *posterior* distribution of the model parameters, $P(\theta|D) = P(D|\theta)P(\theta)/P(D)$.
4. For a test point x, computing the *predictive distribution* $P(y|x, D) = \int_\theta P(y|x, \theta)P(\theta|D)d\theta$.

Note that in such a Bayesian formalism, the prediction is a distribution instead of an element of the output space $\mathcal{Y}$. In order to know which action to perform (or which element to predict), Bayesian decision theory needs a *loss* function. Let $\ell(a, y)$ be the loss incurred by performing action a when the true output is y. Then the Bayesian expected loss is defined as the expected loss under the predictive distribution:

$$\rho(a) := \int_y \ell(a, y)P(y|x, D)dy. \tag{1}$$

The best action according to Bayesian decision theory is the one that minimizes that loss: $a^* := \arg\min_a \rho(a)$. Central to our analysis is the expected loss (EL) of that action, $\rho(a^*)$ or

$$\mathrm{EL}(x) := \min_a \int_y \ell(a, y)P(y|x, D)dy. \tag{2}$$

This quantity should be understood as follows: given that we have taken the best action a^* for the input x, and that the true output is in fact given by $P(y|x, D)$, what is, in expectation, the loss to be incurred once the true output is revealed?

The overall generalization error (i.e. the expected error on unseen examples) is the average of the expected loss over the input distribution: $\int_x \mathrm{EL}(x)P(x)dx$. Thus, in order to minimize this generalization error, our active learning strategy consists in selecting the input instance x to maximize the expected loss:

$$\arg\max_x \mathrm{EL}(x).$$

3.2 ELO for Regression and Classification

In this section, we show that the ELO principle for active learning is equivalent to well known active learning strategies for classification and regression. In the cases of regression and classification, the "action" a discussed above is simply the prediction of an element in the output space $\mathcal{Y}$.

Regression.
The output space is $\mathcal{Y} = \mathbb{R}$ and the loss function is the squared loss $\ell(a, y) = (a - y)^2$. It is well known that the prediction minimizing this square loss is the mean of the distribution and that the expected loss is the variance:

$$\arg\min_a \int_y (a - y)^2 P(y|x, D)dy = \mu$$
$$\text{and} \quad \min_a \int_y (a - y)^2 P(y|x, D)dy = \sigma^2,$$

where μ and σ^2 are the mean and variance of the predictive distribution. So in the regression case, ELO will choose the point with the highest predictive variance which is exactly one of the classical strategy for active learning [7].

Classification.
The output space for binary classification is $\mathcal{Y} = \{-1, 1\}$ and the loss is the 0/1 loss: $\ell(a, y) = 0$ if $a = y$, 1 otherwise. The optimal prediction is given according to $\arg\max_{a \in \mathcal{Y}} P(y = a|x, D)$ and the expected loss turns out to be:

$$\min(P(y = 1|x, D), P(y = -1|x, D)),$$

which is maximum when $P(y = 1|x, D) = P(y = -1|x, D) = 0.5$, that is when we are completely uncertain about the class label. This uncertainty based active learning is the most popular one for classification which was first proposed in [18].

3.3 ELO for Ranking

In the case of ranking, the input instance is a query and a set of documents associated with it, while the output is a vector of relevance scores. If the query q has n documents, let us denote by $X_q := (x_1, \ldots, x_n)$ the feature vectors describing these (query,document) pairs and by $Y := (y_1, \ldots, y_n)$ their labels. As before we have a predictive distribution $P(Y|X_q, D)$. Unlike active learning for classification and regression, active learning for ranking can select examples at different levels. One is query level, which selects a query with all associated documents X_q; the other one is document level, which selects documents x_i individually .

Query level. In the case of ranking, the "action" in ELO framework is slightly different than before because we are not directly interested in predicting the scores, but instead we want to produce a ranking. So the set of actions is the set of permutations of length n and for a given permutation π, the rank of the i-th document $\pi(i)$. The expected loss for a given π can thus be written as:

$$\int_Y \ell(\pi, Y)P(Y|X_q, D)dY, \tag{3}$$

where $\ell(\pi, Y)$ quantifies the loss in ranking according to π if the true labels are given by Y. The next section will detail the computation of the expected loss where ℓ is the DCG loss.

As before, the ELO principle for active learning tells us to select the queries with the highest expected losses:

$$\mathrm{EL}(q) := \min_\pi \int_Y \ell(\pi, Y)P(Y|X_q, D)dY. \tag{4}$$

As an aside, note that the ranking minimizing the loss (3) is not necessarily the one obtained by sorting the documents according to their mean predicted scores. This has already been noted for instance in [26, section 3.1].

Document level. Selecting the most informative document is a bit more complex because the loss function in ranking is defined at the query level and not at the document level. We can still use the expected loss (4), but only consider the predictive distribution for the document of interest and consider the scores for the other documents fixed. Then we take an expectation over the scores of the other documents.

This leads to:

$$\mathrm{EL}(q,i) = \int_{Y^i} \min_{\pi} \int_{y_i} \ell(\pi, Y) P(Y|X_q, D) dy_i dY^i, \qquad (5)$$

where $\mathrm{EL}(q,i)$ is the expected loss for query q associated with the i-th document and Y^i is the vector Y after removing y_i.

4. ALGORITHM DERIVATION

We now provide practical implementation details of the ELO principle for active learning and in particular specifie how to compute equations (4) and (5) in case of the DCG loss.

The difficulty of implementing the formulations of the previous section lies in the fact that the computation of the posterior distributions $P(y_i|x_i, D)$ and the integrals is in general intractable. For this reason, we instead use an ensemble of learners and replace the integrals by sums over the ensemble. As in [1], we propose to use bootstrap to construct the ensemble. More precisely, the labeled set is subsampled several times and for each subsample, a relevance function is learned. The predictive distribution for a document is then given by the predicted relevance scores by various functions in the ensemble. The use of bootstrap to estimate predictive distributions is not new and there has been some work investigating whether the two procedures are equivalent [16].

Finally note that in our framework we need to estimate the relevance scores. This is why we concentrate in this paper on pointwise approaches for learning to rank since pairwise and listwise approaches would not produce such relevance estimates.

4.1 Query Level Active Learning

If the metric of interest is DCG, the associated loss is the difference between the DCG for that ranking and the ranking with largest DCG:

$$\ell(\pi, Y) = \max_{\pi'} \mathrm{DCG}(\pi', Y) - \mathrm{DCG}(\pi, Y), \qquad (6)$$

where $\mathrm{DCG}(\pi, Y) = \sum_i \frac{2^{y_i}-1}{\log_2(1+\pi(i))}$.

Combining equations (4) and (6), the expected loss for a given q is expressed as follows:

$$\mathrm{EL}(q) = \int_Y \max_{\pi} \mathrm{DCG}(\pi, Y) P(Y|X_q, D) dY - \max_{\pi} \int_Y \mathrm{DCG}(\pi, Y) P(Y|X_q, D) dY. \qquad (7)$$

The maximum in the first component of the expected loss can easily be found by sorting the documents according to Y. We rewrite the integral in the second component as:

$$\int_Y \mathrm{DCG}(\pi, Y) P(Y|X_q, D) dY = \sum_i \frac{1}{\log_2(1+\pi(i))} \underbrace{\int_{y_i} (2^{y_i} - 1) P(y_i|x_i, D) dy_i}_{:=t_i}, \qquad (8)$$

with which the maximum can now be found by sorting the t_i.

The pseudo-code for selecting queries based on equation (7) is presented in algorithm 1. The notations and definitions are as follows:

- G is the *gain* function defined as $G(s) = 2^s - 1$.
- The notation $\langle \cdot \rangle$ means average. For instance, $\langle d_i \rangle = \frac{1}{N} \sum_{i=1}^{N} d_i$.
- BDCG is a function which takes as input a set of gain values and returns the corresponding best DCG:

  $$BDCG(\{g_j\}) = \sum_j \frac{g_j}{\log_2(1 + \pi^*(j))},$$

  where π^* is the permutation sorting the g_j in decreasing order.

Algorithm 1 Query Level ELO-DCG Algorithm

```
Require: Labeled set $\mathcal{L}$, unlabeled set $\mathcal{U}$
  for i=1,...,N do                    $N$=size of the ensemble
    Subsample $L$ and learn a relevance function
    $s_j^i \leftarrow$ score predicted by that function on the $j$-th document in $U$.
  end for
  for q=1,...,Q do                    $Q$ = number of queries in $U$
    $\mathcal{I} \leftarrow$ documents associated to $q$
    for i=1,...,N do
      $d_i \leftarrow \mathrm{BDCG}(\{G(s_j^i)\}_{j\in\mathcal{I}})$
    end for
    $t_j \leftarrow \langle G(s_j^i) \rangle$
    $d \leftarrow \mathrm{BDCG}(\{t_j\}_{j\in\mathcal{I}})$
    $\mathrm{EL}(q) \leftarrow \langle d_i \rangle - d$
  end for
  Select the queries $q$ which have the highest values $\mathrm{EL}(q)$.
```

4.2 Document Level Active Learning

Combining equations (5) and (6), the expected loss of the i-th document is expressed as:

$$\mathrm{EL}(q,i) = \int_{Y^i} \left[\int_{y_i} \max_{\pi} \mathrm{DCG}(\pi, Y) P(Y|X_q, D) dy_i - \max_{\pi} \int_{y_i} \mathrm{DCG}(\pi, Y) P(Y|X_q, D) dy_i \right] dY^i, \qquad (9)$$

which is similar to equation (7) except that the uncertainty is on y_i instead of the entire vector Y and that there is an outer expectation on the relevance values for the other documents. The corresponding pseudo-code is provided in algorithm 2.

4.3 Two-stage Active Learning

Both query level and document level active learning have their own drawbacks. Since query level active learning selects all documents associated with a query, it is tend to include non-informative documents when there are a large number of documents associated with each query. For example, in Web search applications, there are large amount of Web documents associated for a query; most of them are non-informative, since the quality of a ranking function is mainly measured by its ranking output on a small number of top ranked Web documents. On the other hand, document level active learning selects documents individually. This selection process implies unrealistic assumption that documents are independent, which leads to some undesirable results. For example, an informative query could be

Algorithm 2 Document Level ELO-DCG Algorithm

```
Require: Labeled set $\mathcal{L}$, unlabeled doc $\mathcal{U}$ for a given query
  for i=1,...,N do                                  $N$=size of the ensemble
    Subsample $L$ and learn a relevance function
    $s_j^i \leftarrow$ score predicted by that function on the $j$-th document in $U$.
  end for
  for all $j \in U$ do
    EL($j$) $\leftarrow 0$                Expected loss for the $j$-th document
    for i=1,...,N do                              Outer integral in (5)
      $t_k \leftarrow s_k^i, \ \forall k \neq j$
      for p=1,...,N do
        $t_j \leftarrow s_j^p$
        $d_p \leftarrow$ BDCG($\{G(t_k)\}$)
      end for
      $g_k \leftarrow G(s_k^i), \ \forall k \neq j$
      $g_j \leftarrow \langle G(s_j^i) \rangle$
      EL($j$) $\leftarrow$ EL($j$) $+ \langle d_p \rangle -$ BDCG($\{g_k\}$)
    end for
  end for
  Select the documents (for the given query) which have the
  highest values of EL($j$).
```

missed if none of its documents is selected; or only one document is selected for a query, which is not a good example in ranking learning.

Therefore, it is natural to combine query level and document level into two-stage active learning. A realistic assumption for ranking data is that queries are independent and the documents are independent given on a query. Based on this assumption, we propose the following two-stage ELO-DCG algorithm: first, applying Algorithm 1 to select most informative queries; then, applying Algorithm 2 to select the most informative queries for each selected query.

5. EXPERIMENTAL EVALUATION

As a general active learning algorithm for ranking, ELO-DCG can be applied to a wide range of ranking applications. In this section, we apply different versions of ELO-DCG algorithms to Web search ranking to demonstrate the properties and effectiveness of our algorithm. We denote query level, document level, and two-stage ELO-DCG algorithms as ELO-DCG-Q, ELO-DCG-D, and ELO-DCG-QD, respectively.

5.1 Data Sets

We use Web search data from a commercial search engine. The data set consists of a random sample of about 10,000 queries with about half million Web documents. Those query-document pairs are labeled using a five-grade labeling scheme: {Bad, Fair, Good, Excellent, Perfect}.

For a query-document pair (q; d), a feature vector x is generated and the features generally fall into the following three categories. Query features comprise features dependent on the query q only and have constant values across all the documents, for example, the number of terms in the query, whether or not the query is a person name, etc. Document features comprise features dependent on the document d only and have constant values across all the queries, for example, the number of inbound links pointing to the document, the amount of anchor-texts in bytes for the document, and the language identity of the document, etc. Query-document features comprise features dependent on the relation of the query q with respect to the document d, for example, the number of times each term in the query q appears in the document d, the number of times each term in the query q appears in the anchor-texts of the document d, etc. We selected about five hundred features in total.

Data set	Number of examples
base set 2k	∼2,000
base set 4k	∼ 4,000
base set 8k	∼8,000
AL set	∼160,000
test set	∼180,000

Table 1: Sizes of the five data sets.

We randomly divide this data set into three subsets, base training set, active learning set, and test set. From the base training set, we randomly sample four small data sets to simulate small size labeled data sets $\mathcal{L}$. The active learning data set is used as a large size unlabeled data set $\mathcal{U}$ from which active learning algorithms will select the most informative examples. The true labels from active learning set are not revealed to the ranking learners unless the examples are selected for active learning. The test set is used to evaluate the ranking functions trained with the selected examples plus base set examples. We kept test set large to have rigorous evaluations on ranking functions. The sizes of those five data sets are summarized in Table 1.

5.2 Experimental Setting

For the learner, we use Gradient Boosting Decision Tree (GBDT) [15].

The input for ELO-DCG algorithm is a base data set $\mathcal{L}$ and the AL data set $\mathcal{U}$. The size of the function ensemble is set as 8 for all experiments. ELO-DCG algorithm selects top m informative examples; those m examples are then added to the base set to train a new ranking function; the performance of this new function is then evaluated on the test set. Each algorithm with each base set is tested on 14 different m, ranging from 500 to 80,000. For every experimental setting, 10 runs are repeated and in each run the base set is re-sampled to generate a new function ensemble.

For the performance measure for ranking models, we select to use DCG-k, since users of a search engine are only interested in the top-k results of a query rather than a sorted order of the entire document collection. In this study, we select k as 10. The average DCG of 10 runs is reported for each experiment setting.

5.3 Document Level Active Learning

We first investigate document level active learning, since documents correspond to basic elements to be selected in the traditional active learning framework. We compare document level ELO-DCG algorithm with random selection (denoted by Random-D) and a classical active learning approach based on Variance Reduction (VR) [7], which selects document examples with largest variance on the prediction scores.

Figure 1 compares the three document level active learning methods in terms of DCG-10 of the resulting ranking functions on the test set. Those ranking functions are trained with base data set and the selected examples. X-axis de-

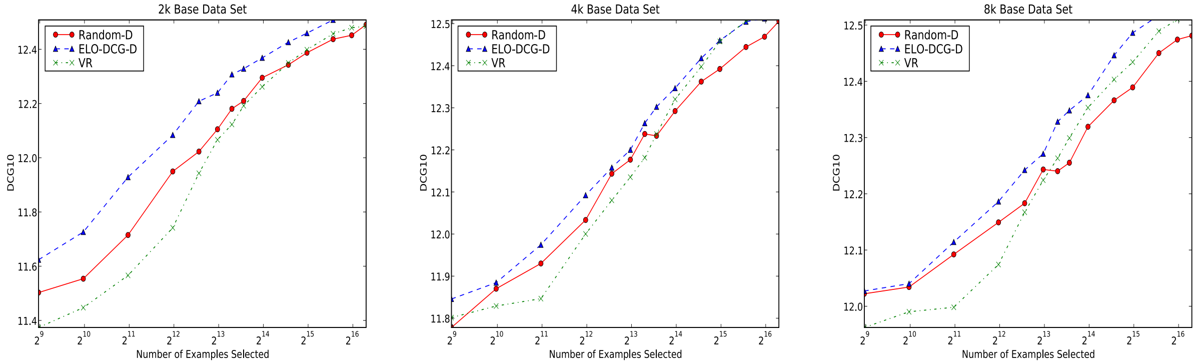

Figure 1: DCG comparison of document level ELO-DCG, variance reduction based document selection, and random document selection with base sets of sizes 2k,4k, and 8k shows that ELO-DCG algorithm outperforms the other two document selection methods at various sizes of selected examples.

notes number of examples selected by the active learning algorithm. For all three methods, the DCG increases with the number of added examples. This agrees with the intuition that the quality of a ranking function is positively correlated with the number of examples in the training set. ELO-DCG consistently outperforms the other two methods. An possible explanation is that ELO-DCG optimizes the expected DCG loss that is directly related to the objective function DCG-10 used to evaluate the ranking quality; on the other hand, VR reduces the score variance that is not directly related to the objective function. In fact, VR performs even worse than random document selection when the size of the selected example is small. An advantage of the ELO-DCG algorithm is its capability to optimize directly based on the ultimate loss function to measure ranking quality.

5.4 Query Level Active Learning

In this section, we show that query level ELO-DCG algorithm effectively selects informative queries to improve the learning to rank performance. Since traditional active learning approaches cannot directly applied to query selection in ranking, we compare it with random query selection (denoted by Random-Q) used in practice.

Figure 2 shows the DCG comparison results. we observe that for all three base sets, ELO-DCG performs better than random selection for all different sample sizes (from 500 to 80,000) that are added to base sets. Moreover, ELO-DCG converges much faster than random selection, i.e., ELO-DCG attains the best DCG that the whole AL data set can attain with much less examples added to the train data.

5.5 Two-stage Active Learning

In this section, we compare two-stage ELO-DCG algorithm with other two two-stage active learning algorithms. One is two-stage random selection, i.e. random query selection followed by random document selection for each query. The other one is a widely used approach in practice, which first randomly selects queries and then select top k relevant documents for each query based on current ranking functions (such as top k Web sites returned by the current search engine)[23]. In our experimental setting, this approach corresponds to randomly query selection followed by selecting k documents with highest mean relevance scores within each selected query. We denote this approach as top-K. In all three two-stage algorithms, we simply fix the number of documents per query at 15 based on the results from [23].

Figure 3 shows the DCG comparison results for two-stage active learning. We observe that among all three base sets, ELO-DCG performs the best and top-K performs the second. This result demonstrates that two-stage OLE-DCG effectively select most informative documents for most informative queries. A possible reason that top-K performs better than random selection is that top-k selects more perfect and excellent examples. Those examples contribute more to DCG than bad and fair examples.

We have observed that ELO-DCG algorithms perform best in all three active learning scenarios, query level, document level, and two stage active learning. Next, we compare three versions of ELO-DCG with each other.

Figure 4 shows DCG comparisons of two-stage ELO-DCG, query level ELO-DCG, and document level ELO-DCG. We observe that for all three based sets, two stage ELO-DCG performs best. The reason that two-stage algorithm performs best may root in its reasonable assumption for the ranking data: queries are independent; the documents are conditionally independent given a query. On the other hand, the document level algorithm makes the incorrect assumption about document independence and may miss informative information at the query level; the query level algorithm selects all documents associated with a query, which are not all informative.

5.6 Cost reduction of Two-stage ELO-DCG Algorithm

In this section, we show the reduction in labeling cost achieved by ELO-DCG compared with the widely used top-K approach in practice.

In Table 2, the saturated size means that when the examples of this size are selected and added back to the base set to train a ranking function, the performance of the learned ranking function is equivalent to the ranking function trained with all active learning data. From the first row of Table 2, we observe that for the base set of size 2k, 64k is the saturated size for two-stage ELO-DCG algorithm and 80k is the saturated size for top-K approach; hence, 64k selected examples from two-stage ELO-DCG algorithm is equivalent to 80k selected examples from top-K approach. This means that two-stage ELO-DCG algorithm can reduce the cost by

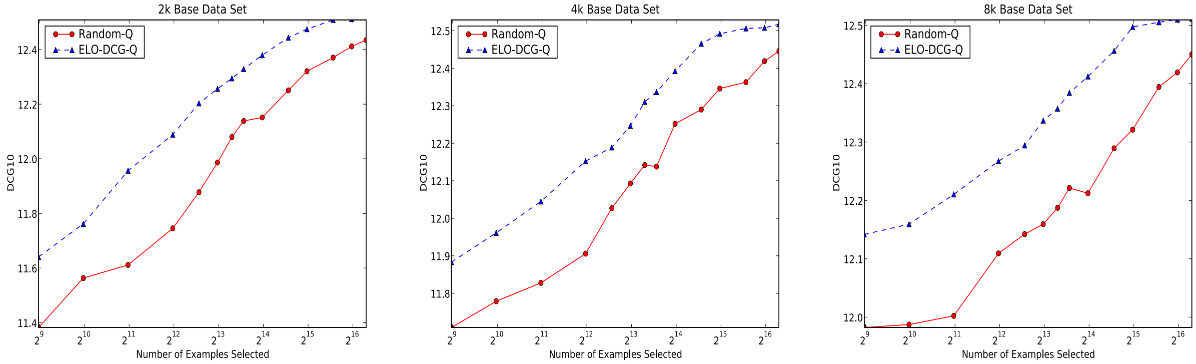

Figure 2: DCG comparisons of query level ELO-DCG and random query selection with base sets of sizes 2k,4k, and 8k shows that ELO-DCG algorithm outperforms random selection at various sizes of selected examples.

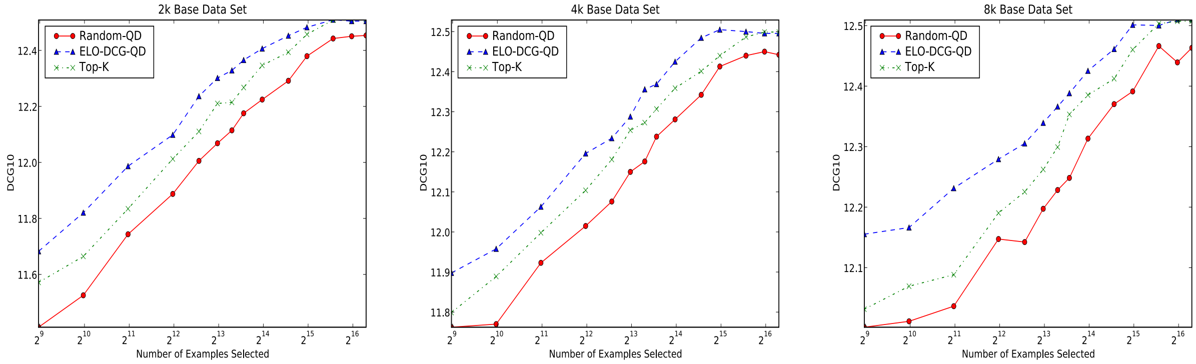

Figure 3: DCG comparisons of two-stage ELO-DCG, two-stage random selection, and top-K selection with base set 2k,4k, and 8k shows that ELO-DCG algorithm performs best.

20%. The largest percentage of cost reduced, 64%, is from base set 8k.

6. CONCLUSIONS

We propose a general expected loss optimization framework for ranking, which is applicable to active learning scenarios for various ranking learners. Under ELO framework, we derive novel algorithms, query level ELO-DCG and document level ELO-DCG, to select most informative examples to minimize the expected DCG loss. We propose a two stage active learning algorithm to select the most effective examples for the most effective queries. Extensive experiments on real-world Web search data sets have demonstrated great potential and effectiveness of the proposed framework and algorithms.

7. REFERENCES

[1] N. Abe and H. Mamitsuka. Query learning strategies using boosting and bagging. In *Proceedings of the Fifteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., 1998.

[2] J. A. Aslam, E. Kanoulas, V. Pavlu, S. Savev, and E. Yilmaz. Document selection methodologies for efficient and effective learning-to-rank. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 468–475. ACM, 2009.

[3] J. Berger. *Statistical decision theory and Bayesian analysis*. Springer, 1985.

[4] C. Campbell, N. Cristianini, and A. Smola. Query learning with large margin classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 111–118. Morgan Kaufmann, 2000.

[5] B. Carterette, J. Allan, and R. Sitaraman. Minimal test collections for retrieval evaluation. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2006.

[6] W. Chu and Z. Ghahramani. Extensions of gaussian processes for ranking: semi-supervised and active learning. In *Nips workshop on Learning to Rank*, 2005.

[7] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 705–712. The MIT Press, 1995.

[8] D. Cossock and T. Zhang. Subset ranking using regression. In *Proc. Conf. on Learning Theory*, 2006.

[9] I. Dagan and S. P. Engelson. Committee-based sampling for training probabilistic classifiers. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 150–157. Morgan Kaufmann, 1995.

[10] P. Donmez and J. Carbonell. Active sampling for rank

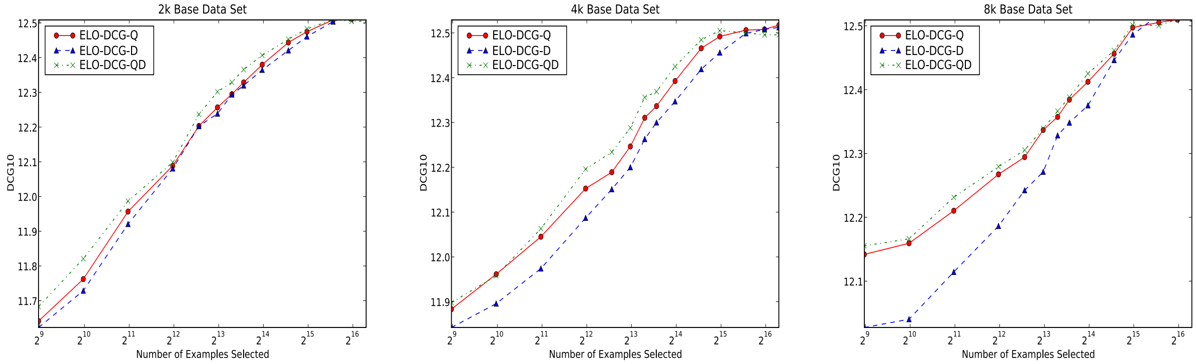

Figure 4: DCG comparisons of two-stage ELO-DCG, query level ELO-DCG, and document level ELO-DCG,with base sets of sizes 2k,4k, and 8k shows that two-stage ELO-DCG algorithm performs best.

Base set size	Saturated size for ELO-DCG	Saturated size for top-K	Percentage of cost reduced
2k	64k	80k	20%
4k	48k	80k	40%
8k	32k	80k	64%

Table 2: Cost reduction of two-stage algorithm compared with top-K approach.

learning via optimizing the area under the ROC curve. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, pages 78–89, 2009.

[11] P. Donmez and J. G. Carbonell. Optimizing estimated loss reduction for active sampling in rank learning. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 248–255, New York, NY, USA, 2008. ACM.

[12] V. Fedorov. *Theory of Optimal Experiments.* Academic Press, New York, 1972.

[13] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[14] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

[15] J. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[16] T. Fushiki. Bootstrap prediction and bayesian prediction under misspecified models. *Bernoulli*, 11(4):747–758, 2005.

[17] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In Smola, Bartlett, Schoelkopf, and Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA, 2000.

[18] D. Lewis and W. Gale. Training text classifiers by uncertainty sampling. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, 1994.

[19] A. McCallum and K. Nigam. Employing EM and pool-based active learning for text classification. In *Proceedings of the 5th international conference on Machine learning*, pages 359–367, 1998.

[20] B. Settles. Active learning literature survey. Technical Report Computer Sciences 1648, University of Wisconsin, Madison, 2009.

[21] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: theory and algorithm. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 1192–1199, New York, NY, USA, 2008. ACM.

[22] L. Yang, L. Wang, B. Geng, and X.-S. Hua. Query sampling for ranking learning in web search. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 754–755, New York, NY, USA, 2009. ACM.

[23] E. Yilmaz and S. Robertson. Deep versus shallow judgments in learning to rank. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 662–663, New York, NY, USA, 2009. ACM.

[24] H. Yu. SVM selective sampling for ranking with application to data retrieval. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 354–363. ACM, 2005.

[25] Z. Zheng, H. Zha, T. Zhang, O. Chapelle, K. Chen, and G. Sun. A general boosting method and its application to learning ranking functions for web search. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1697–1704, 2008.

[26] O. Zoeter, N. Craswell, M. Taylor, J. Guiver, and E. Snelson. A decision theoretic framework for implicit relevance feedback. In *NIPS Workshop on Machine learning for web search*, 2007.