A Two-Stage Active Learning Method for Learning to Rank

Rodrigo M. Silva, Marcos A. Gonçalves, and Adriano Veloso
Department of Computer Science, Federal University of Minas Gerais, Av. Antonio Carlos 6627, CEP 31270-901, Belo Horizonte—MG, Brazil. E-mail: {rmsilva, mgoncalv, adrianov}@dcc.ufmg.br

Learning to rank (L2R) algorithms use a labeled training set to generate a ranking model that can later be used to rank new query results. These training sets are costly and laborious to produce, requiring human annotators to assess the relevance or order of the documents in relation to a query. Active learning algorithms are able to reduce the labeling effort by selectively sampling an unlabeled set and choosing data instances that maximize a learning function's effectiveness. In this article, we propose a novel two-stage active learning method for L2R that combines and exploits interesting properties of its constituent parts, thus being effective and practical. In the first stage, an association rule active sampling algorithm is used to select a very small but effective initial training set. In the second stage, a query-by-committee strategy trained with the first-stage set is used to iteratively select more examples until a preset labeling budget is met or a target effectiveness is achieved. We test our method with various LETOR benchmarking data sets and compare it with several baselines to show that it achieves good results using only a small portion of the original training sets.

Introduction

Ranking is an essential feature of many applications: From web search to product recommendation systems and online advertising, results have to be ordered based on their estimated relevance with respect to a query or based on a user profile or personal preferences. Learning to rank (L2R) algorithms, which deliver superior performance when compared with more traditional approaches such as BM25 (Robertson, Walker, & Hancock-Beaulieu, 1995), rely on labeled or ordered training sets to build ranking models that are used to rank results at query time. To create these training sets, human annotators must evaluate a portion or all documents returned by a set of queries. After selecting a group of representative queries, a classic information retrieval (IR) method (such as BM25) is used to retrieve and rank the documents associated with each query and an expert evaluates the first *n* documents, labeling each according to its relevance to the given query. Depending on the type of rank learning algorithm used, it may be necessary to provide a binary relevance judgment (i.e., relevant, not relevant) or a relevance level (e.g., somewhat relevant, very relevant, extremely relevant), a pairwise ordering (e.g., document d_i is more relevant than document d_j or $d_i \succ d_j$), or a complete or partial ordering of the documents returned by a query (e.g., $d_i \succ d_j \succ [\ldots] \succ d_k$). These different relevance judgment types correlate to the three main approaches used by L2R methods: pointwise, pairwise, and listwise (Liu, 2009). Independently of how the training set is constructed, it is costly and laborious to produce any amount of it.

Active learning techniques have been proposed to help deal with the labeling effort problem in L2R (Cai, Gao, Zhou, & Wong, 2011; Donmez & Carbonell, 2009, 2008; Long et al., 2010; Radlinski & Joachims, 2007; Silva, Gonçalves, & Veloso, 2011; Yu, 2005). The motivation behind active learning is that it may be possible to achieve highly effective learned functions by carefully selecting and labeling instances that are "informative" to the learning algorithm. Using active learning, we can reduce the cost of producing training sets for rank learning algorithms and improve the effectiveness of the learned functions by avoiding adding "noisy" or redundant instances to the training sets. Furthermore, human annotators can spend more time analyzing the relevance of each selected instance, which produces better training sets (Geng, Qin, Liu, Cheng, & Li, 2011). Active learning can dramatically reduce the size of the training sets created without affecting the quality of the resulting learned models by carefully selecting the documents to be labeled.

In a typical active learning scenario, data instances are selected from an unlabeled set one at a time and labeled by a human expert. Every time a new sample is selected and

Received October 30, 2012; revised February 14, 2013; accepted February 15, 2013

© 2013 ASIS&T • Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/asi.22958

labeled, a new learning model is produced and the active learning algorithm again chooses a new instance from the unlabeled set. This process is repeated as long as necessary or until the labeling budget is exhausted. Several studies propose active learning methods for classification tasks (see, e.g., Donmez, Carbonell, & Bennett, 2007; Mccallum, 1998; Nguyen & Smeulders, 2004; Schohn & Cohn, 2000; Tong & Koller, 2002). Whereas classification functions output a distinct class for each data item, ranking functions must produce partial orders of items either through some scoring function, pairwise ordering, or listwise ordering of the items. Most active sampling methods for classification try to directly minimize the classification error, but it is not straightforward to extend this approach to the ranking problem because, as noted by Liu (2009), position-based measures such as the mean average precision (MAP) and the normalized discounted cumulative gain (NDCG) are usually noncontinuous and nondifferentiable. In addition, in most supervised classification learning settings, samples can be treated as independent of each other, which is not the case for L2R where each sample represents a document *relative to a query*. In classification, two instances that have very similar feature-values usually will be assigned the same class. In L2R, most features characterize the document given the query: Two documents returned by different queries can have similar feature-values and yet appear at diverse points of each query's rankings (or be considered relevant in one case and not relevant in the other). Thus, in L2R, instances are conditionally independent given a query (see Long et al., 2010).

Despite the differences between classification and L2R, active learning methods proposed in both fields have common general outlines or strategies. In some methods, the most ambiguous, or those instances for which the learner is most *uncertain*, are selected (Lewis & Gale, 1994; Tian & Lease, 2011). In a related strategy, a query-by-committee (QBC) strategy is used where competing learners vote on the label of the candidate samples and the one selected for labeling is the one about which the members of the committee most disagree in classifying (Cai et al., 2011; Seung, Opper, & Sompolinsky, 1992). Some algorithms select samples that would cause the greatest change to the current learned function (Donmez & Carbonell, 2008; Settles, Craven, & Ray, 2008). Other methods select instances that would lead to the minimal expected future error or, similarly, optimize some other metric such as precision or recall (Donmez & Carbonell, 2009; Settles, 2009).

In this article, we propose a novel two-stage active learning technique for L2R. The first stage uses an association rule-based strategy that selects nonredundant, informative samples from a completely unlabeled set so that "noisy" samples are avoided. The resulting data set, although very small (yet effective), may be limited in its representativeness. Moreover, as the first stage has a clear stopping criteria (it stops selecting new instances for labeling when it judges that no other candidate has useful information to be incorporated into the training set), it does not provide a simple way to select more instances (and possibly improve the ranking quality), even if there is a labeling budget available. Thus, in the second stage, we use a QBC procedure to expand the selected set using a completely different selection criterion that prioritizes a better coverage of the sample space. The result is still a very small, yet highly effective training set.

To the best of our knowledge, all previously proposed methods concerning active learning for L2R have assumed that an initial labeled seed set is available to be used as a base for further sample selection or use simple single-feature and/or semirandom procedures to select initial sets. Although some labeled samples may be available in certain scenarios, we believe that in many other cases it is desirable to create a new L2R training set from scratch. Unlabeled samples are easily obtained from existing collections and web crawling efforts. Thus, instead of using random sampling or a classic retrieval method to obtain a small initial set of documents, labeling them, and then using this seed set to bootstrap the actual active learning process, our method actively selects documents to be labeled from the start. This characteristic allows the proposed method to obtain competitive results right from the start, making it ideal for situations in which no previously labeled sets are available. Another advantage of using the association rule-based first stage to create the initial sets is the fact that it naturally converges; thus, it is not necessary to arbitrarily choose a number of documents per query that need to be labeled to produce the seed sets (as is the case with the Donmez baseline, as discussed in the Baselines section). These are important characteristics of an active learning method because in a real-world scenario, there is no simple way to evaluate the quality of the selected training sets; only after reasonably sized sets are actually selected and labeled (i.e., after a reasonable part of the labeling budget is spent) can we use cross-validation to evaluate their quality. We compare our method against several baselines: random sampling (Random), a QBC strategy using randomly selected initial sets (Random-QBC), a combination of the first-stage method and random second-stage selection (ARLR-Random), the active learning method proposed by Donmez and Carbonell (2008) (Donmez), the supervised (i.e., using the complete training sets) SVMRank results published by the LETOR producers (SVM Full), and the ARLR-RLR active learning method presented by Silva et al. (2011). Experiments were run on the six LETOR (LEarning To Rank) 3.0 web data sets, iteratively selecting up to 15% of each original training set for comparative purposes. As discussed later in the Experimental Evaluation section, our method obtained excellent results, selecting as little as 6% of the unlabeled sets. Moreover, its results surpass, in most cases (on average, all cases), state-of-the-art supervised algorithms that use the complete training sets, producing some of the best results ever reported for these data sets (e.g., considering all the reported LETOR 3.0 benchmark baselines). We also extend our method to perform both query- and document-level selections (see Adding Query-

Level Selection section) and test this modified version on the LETOR 4.0 collection. This modification allows the method to be used on data sets that have many queries. We believe this work is an original and important contribution as the method is both practical and effective, advancing the state of the art in active learning mechanisms for L2R.

In summary, the main contributions of this article are as follows:

- A practical and effective active learning method that can be used to produce training sets for L2R algorithms
- A method that does not rely on an initial labeled set: It can be applied directly to select samples from unlabeled sets
- A method that is general and obtains state-of-the-art results for data sets with very diverse characteristics, such as those based on informational or navigational queries

The remainder of this article is organized as follows: The Related Work section discusses related work. The Two-Stage Active Learning for L2R section presents the two stages of our proposed method and describes how the two stages work together. The Experimental Evaluation section describes our experimental evaluation along with several analyses. Finally, the Conclusions section wraps up the article.

Related Work

Some researchers have recently proposed active learning schemes for L2R based on the optimization of approximations of position-based measures. Long et al. (2010), for example, propose a general active learning framework based on expected loss optimization. Their framework uses function ensembles to select training examples that minimize a chosen loss function. The authors approach the unique challenges of active learning in L2R by separating their selection algorithm into query- and document-level parts. To approximate their chosen metric, namely, discounted cumulative gain (DCG), they use ensembles of learners to produce relevance scores and estimate predictive distributions for the documents in the active learning set. To produce the ensemble, they use a bootstrap technique that relies on an initial labeled set. Thus, their technique requires an initial labeled set to build the ensemble of learners, which needs to be large enough for the learners in the ensemble to be minimally effective. They evaluate their method using a large commercial web search data set (500K documents) and using initial (labeled) sets of 2K, 4K, and 8K documents.

An SVM-specific strategy is presented by Donmez and Carbonell (2008). The method starts with a per-query labeled seed set. It progresses in rounds, selecting a preset number of new documents per query that are then labeled and added to the training set. The basic idea is to estimate, for each query, what is the capacity of an unlabeled example to update the current model if it is labeled and added to the training set. The top five samples for each query that have the highest estimated impact in the current model are selected and labeled. The idea is that those instances that change the current learned model the most will accelerate the model's convergence to the true hypothesis. The authors present results using this sampling technique adapted to SVMRank and RankBoost. Their method achieves competitive results with around 10% of the original training sets selected. We have chosen to use this method as a baseline because it is elegant, simple, and relatively easy to implement. It also produces results on par with those presented in the authors' more recent work, which is described later. More details about this method are provided in the Baselines section.

Donmez and Carbonell (2009) rely on the relationship between the area under the receiver operating characteristic curve and the hinge rank loss proposed by Steck (2007) to develop a loss minimization framework for active learning in ranking. Instead of testing each and every unlabeled sample to determine the one that has the smallest expected future error, the authors suggest selecting the examples that have the largest contribution to the estimated current error. These are the ones that, when labeled, will potentially bring more benefit for the functions that will be trained in the next rounds of the method. The proposed selection criterion is based on the hinge rank loss calculated on a per-query basis and depends on the determination of a rank threshold that estimates the rank position that separates the lowest ranked relevant element from the highest ranked nonrelevant example. The algorithm starts with a small, semirandom labeled per query set and proceeds selecting unlabeled samples that have the highest uncertainty (as defined by the rank threshold). These samples are then labeled and added to the per-query labeled sets, and the process is repeated as many times as necessary.

Some other studies apply active learning strategies in association with other techniques such as transfer learning or relevance feedback (RF). Cai et al. (2011), for instance, propose a method that integrates domain adaptation and active learning as a way to reduce labeling costs. They first use a QBC scheme built on a mixture of source domain and target domain data to select the most "informative" queries from the target domain. Then these new labeled sets are used to adjust the importance weights of source-domain queries for boosting the transfer of prior ranking knowledge. Their QBC strategy is based on the query-by-bagging concept (Abe & Mamitsuka, 1998) where from the currently labeled set a number of partitions is created by sampling uniformly with replacement, and the same learning algorithm is trained using these partitions to obtain different models to be used as committee. Their method performs only query-level selection, meaning that all the documents for the selected queries have to be labeled. Although their method is not directly comparable with ours (because it uses rank adaptation on top of active learning), it must be noted that by using query-level selection only, the amount of labeled documents grows very fast for data sets such as those in LETOR 3.0 (which is used to evaluate their method). In contrast, our

DOI: 10.1002/asi

work uses an ensemble of learners approach to QBC and performs document-level selection (although, as we discuss later, it can be easily extended to use query-, document-, or query-document–level selection), achieving better results using 15% (usually much less) of the unlabeled sets than those presented in their work using 100% of the training sets.

Another work leverages active learning using RF. Tian and Lease (2011) propose a method that iteractively improves the rankings shown to the user based on feedback on which link the user clicks at each round. The method uses an SVM algorithm to classify the documents into relevant and not relevant. Their active learning scheme uses two approaches: In the Simple Margin version, those documents lying closest to the decision hyperplane are considered the ones the SVM is most *uncertain* about. They also propose a variation, called Local Structure, that takes into account the proximity of the unlabeled instances to those instances already labeled. This variation chooses instances close to the decision surface but also far from already labeled data and close to more unlabeled samples in an attempt to maximize the diversity of the selected set and improve the algorithm's learning curve. They evaluate their method using Robust04[1] and LETOR 3.0, comparing it with RF baselines. Their work is not directly comparable with ours because we provide a method for reducing the training set creation effort in which the human annotators evaluate each document selected by the system in turn until the (small) labeling budget is met.

Two-Stage Active Learning for L2R

Stage 1: Active Sampling Using Association Rules

To be able to explain how the active sampling technique that constitutes the first stage (Silva et al., 2011) of our two-stage approach works, we first briefly describe the supervised algorithm it is based on and detailed by Veloso, Almeida, Gonçalves, and Meira, Jr. (2008).

RLR. The rule-based learning to rank (RLR) algorithm is a supervised pointwise method that uses association rules (Agrawal, Imieliński, & Swami, 1993) to rank documents. To use it, we need a labeled training set $\mathcal{D}$ composed of records of the form <*q*, *d*, *r*>, where *q* is a query, *d* is a document returned by the query and is represented as a list of *m* feature-values or $\{f_1, f_2, \ldots, f_m\}$, and *r* is the *relevance* of *d* to *q*. Features include BM25, PageRank, and many other document and query-document properties that are discretized to reduce the feature space and allow for the enumeration of association rules. The relevance can be either binary (i.e., 0: not relevant, 1: relevant) or a set of discrete and ordered possibilities $\{r_0, r_1, \ldots, r_k\}$ (e.g., 0: not relevant, 1: somewhat relevant, 2: relevant, 3: very relevant). The *test set* $\mathcal{T}$ consists of records <*q*, *d*, ?> for which only the query *q* and the document *d* are known, whereas the relevance of *d* to *q* is unknown. From the training set $\mathcal{D}$, we can derive a rule-set $\mathcal{R}$ composed of rules of the form $\left\{f_j \wedge \ldots \wedge f_l \xrightarrow{\theta} r_i\right\}$. These rules can contain any mixture of the available features in the antecedent and a relevance level in the consequent. The strength of the association between antecedent and consequent is measured by a statistic, θ, which is known as *confidence* (Agrawal et al., 1993) and is simply the conditional probability of the consequent given the antecedent. The algorithm works by delaying rule extraction until query time: When a set of documents is retrieved for a given query from $\mathcal{T}$, each individual document *d* in $\mathcal{T}_q$ (i.e., the subset of $\mathcal{T}$ containing the documents returned by query *q*) is used as a filter to remove irrelevant features and examples from $\mathcal{D}$. This process produces a projected training set, $\mathcal{D}_d$, which is obtained after removing all feature-values not present in *d*. In other words, this projected training set contains only those instances that have values in common with the document *d* that is being ranked at this point. This process ensures that we can only derive association rules from the projected training set that are useful in ranking document *d*. Thus, a specific rule-set, $\mathcal{R}_d$ extracted from $\mathcal{D}_d$, is produced for each document *d* in $\mathcal{T}_q$.

It is necessary to combine all rules in $\mathcal{R}_d$ to estimate the relevance of a document *d*. $\mathcal{R}_d$ is viewed as poll, in which each rule $\left\{\mathcal{X} \xrightarrow{\theta} r_i\right\} \in \mathcal{R}_d$ is a vote given by a set of features $\mathcal{X}$ for relevance level r_i. Votes have different weights, depending on the strength of the association they represent (i.e., θ). The weighted votes for relevance level r_i are summed and then averaged (by the total number of rules in $\mathcal{R}_d$ that predict relevance level r_i), forming the score associated with relevance r_i for document *d*, as shown in Equation 1, where $\theta(\mathcal{X} \rightarrow r_i)$ is the value θ assumes for rule $\{\mathcal{X} \rightarrow r_i\}$, and $\mathcal{R}_d^{r_i}$ is the set of rules derived for document *d* that predict relevance level r_i:

$$s(d, r_i) = \frac{\sum \theta(\mathcal{X} \rightarrow r_i)}{\left|\mathcal{R}_d^{r_i}\right|}, \text{ where } \mathcal{X} \subseteq d. \qquad (1)$$

Therefore, for a document *d*, the score associated with relevance r_i is given by the average θ values of the rules in $\mathcal{R}_d$ predicting r_i. The likelihood of *d* having a relevance level r_i is obtained by normalizing the scores, as expressed by $\hat{p}(r_i|d)$, shown in Equation 2:

$$\hat{p}(r_i|d) = \frac{s(d, r_i)}{\sum_{j=0}^{k} s(d, r_j)}. \qquad (2)$$

Finally, the relevance or score of document *d* is estimated by a linear combination of the likelihoods associated with each relevance level, as expressed by the ranking function *rank*(*d*), which is shown in Equation 3:

$$rank(d) = \sum_{i=0}^{k} \left(r_i \times \hat{p}(r_i|d)\right). \qquad (3)$$

[1] http://trec.nist.gov/data/robust/04.guidelines.html

The value of *rank*(d) is an estimate of the true relevance of document d using $\hat{p}(r_i|d)$. This estimate ranges from r_0 to r_k, where r_0 is the lowest relevance and r_k is the highest one. After *rank*(d) is calculated for all documents returned for the query, we can sort the documents by this score in descending order and obtain the final ranked list.

ARLR. The active learning variation of the rule-based method briefly discussed earlier was introduced by Silva et al. (2011). We call it ARLR (Active RLR) throughout this article. The idea behind the method is that the number of rules generated by the documents found in the unlabeled set is an indication of how much information each document shares with the current selected (and labeled) set.

More formally, from an unlabeled set $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$, we want to select highly informative documents to compose a new labeled training set $\mathcal{D}$ such that $|\mathcal{D}| \ll |\mathcal{U}|$. Initially, $\mathcal{D}$ is empty and the algorithm cannot extract any rules from it, so it selects from $\mathcal{U}$ the document that shares *the most* feature-values with the other unlabeled documents. This document is labeled and put into $\mathcal{D}$ (but also remains in $\mathcal{U}$). Then, at each round, the algorithm selects the document that demands the fewest rules (i.e., the document in $\mathcal{U}$ for which there are less matching rules) because it shares fewer feature-values with the documents already selected. If only a few rules are extracted for a document u_i, then this is evidence that $\mathcal{D}$ does not contain documents that are similar to u_i; thus, the information provided by document u_i is not redundant, and u_i is a highly informative document given the documents already in $\mathcal{D}$. If $u_i \in \mathcal{U}$ is inserted into $\mathcal{D}$, then the number of rules for documents in $\mathcal{U}$ that share feature-values with u_i will increase. However, the number of rules for those documents in $\mathcal{U}$ that do not share any feature-values with u_i will remain unchanged. Therefore, the number of rules extracted for each document in $\mathcal{U}$ can be used as an approximation of the amount of redundant information between documents already in $\mathcal{D}$ and documents in $\mathcal{U}$. The result is a very small training set based on a *diversity* criterion: The more diverse documents we have in the training set, the more we cover the feature space with the smallest possible amount of documents.

The sampling function used by ARLR exploits this key idea by selecting documents that contribute primarily with nonredundant information: These informative documents are those likely to demand the fewer number of rules from $\mathcal{D}$. More specifically, the sampling function $\gamma(\mathcal{U})$ returns a document in $\mathcal{U}$ according to Equation 4:

$$\gamma(\mathcal{U}) = \left\{u_i \text{ such that } \forall u_j : |\mathcal{R}_{u_i}| < |\mathcal{R}_{u_j}|\right\}. \tag{4}$$

Each document returned by the sampling function is labeled and inserted into $\mathcal{D}$, but it also remains in $\mathcal{U}$. In the next round, the sampling function is executed again, but the number of rules extracted from $\mathcal{D}$ for each document in $\mathcal{U}$ is likely to change due to the document recently inserted into $\mathcal{D}$. After selecting the first document and at each of the posterior rounds, ARLR executes the sampling function and a new example is inserted into $\mathcal{D}$. At iteration i, the selected document is denoted as $\gamma_i(\mathcal{U})$, and it is likely to be as dissimilar as possible from the documents already in $\mathcal{D} = \{\gamma_{i-1}(\mathcal{U}), \gamma_{i-2}(\mathcal{U}), \ldots \gamma_i(\mathcal{U})\}$. All steps of ARLR are shown in Algorithm 1: The loop (steps 1–12) runs until a document already inserted into $\mathcal{D}$ is selected again (step 12). In each loop, for every document in $\mathcal{U}$ (2), a projection of $\mathcal{D}$ is created (3) and rules useful for the document under consideration are derived (4). Initially, $\mathcal{D}$ is empty, so we select the document that shares the most feature-values with the other unlabeled documents (7). Once $\mathcal{D}$ is nonempty, we select the document that generates the smallest amount of rules (9).

Algorithm 1 ARLR.

```
Require: Unlabeled data $\mathcal{U}$, and $\sigma_{min}$ ($\approx 0$)
Ensure: The training data $\mathcal{D}$
 1: continue
 2:     for all document $u_i \in \mathcal{U}$ do
 3:         $\mathcal{D}_{u_i} \Leftarrow \mathcal{D}$ projected according to $u_i$
 4:         $\mathcal{R}_{u_i} \Leftarrow$ rules extracted from $\mathcal{D}_{u_i} | \sigma \ge \sigma_{\min}$
 5:     end for
 6:     if $\mathcal{D} = \emptyset$ then
 7:         $\gamma_i(\mathcal{U}) \Leftarrow u_i$ such that $\forall u_j : |\mathcal{U}_{u_i}| \ge |\mathcal{U}_{u_j}|\}$
 8:     else
 9:         $\gamma_i(\mathcal{U}) \Leftarrow u_i$ such that $\forall u_j : |\mathcal{R}_{u_i}| \le |\mathcal{R}_{u_j}|\}$
10:     end if
11:     if $\gamma_i(\mathcal{U}) \in \mathcal{D}$ then break
12:     else append $\gamma_i(\mathcal{U})$ to $\mathcal{D}$
```

Natural stop condition. The algorithm stops when all available documents in $\mathcal{U}$ are less informative than any document already inserted into $\mathcal{D}$. This occurs when ARLR selects a document that is already in $\mathcal{D}$. When this condition is reached, ARLR will keep selecting the same document over and over again, and there is no information gain with the inclusion of these documents.

As explained earlier, ARLR tries to maximize diversity by selecting documents to cover the feature space as best and as fast as possible using the number of rules demanded by each candidate document as a basis. This is in contrast with other initial set selection strategies (such as the one used by the Donmez method, as explained in more detail in the Baselines section), which use the value of only one feature and random sampling to build the initial sets. By using ARLR to select our initial sets, we are trying to provide the QBC stage with a representative and diverse set of examples that will allow a faster convergence of the learned model, thus obtaining better results right from the beginning of the active selection process.

Stage 2: Expanding the Selection

QBC. Stage 1 of our method selects a very small initial set that can be used as a training set for our query-by-committee (QBC) iteractive active selection stage. The concept of using a committee of learners to identify "interesting" data instances is well known (Baum, 1991; Schapire, 1989;

Seung et al., 1992). The basic idea is to use an ensemble of learners or models to classify or rank an unlabeled set, and those data instances that the various models most disagree about are deemed the most "informative" and are selected for labeling. In Seung et al. (1992), the authors call this process "incremental learning," where a training algorithm is used to produce a committee of $2k$ learners and a query algorithm selects a sample that is classified positive by half of the committee and negative by the other half. They show that, for the Gibbs training algorithm, in the $k \to \infty$ limit, each query bisects the version space, maximizing the information gain. More general methods have been proposed and successfully used such as query-by-bagging and query-by-boosting (Abe & Mamitsuka, 1998; Schapire, 1989). In the bagging approach, different models are produced using the same learning algorithm trained on partitions created by uniformly sampling the current labeled set. Boosting uses a more sophisticated, round-based resampling method where the sampling distributions (or instances' weights) are varied at each round to focus the learned models on the parts of the training data on which previous learners performed poorly. The bagging concept is immediately applicable to active learning, where the instances selected at each round are those that the various models most disagree about. In an L2R scenario, this measure of disagreement could be, for instance, the Kendall's τ rank correlation coefficient or, as proposed by Cai et al. (2011), a vote entropy (Dagan & Engelson, 1995) variation adapted for pairwise ranking. These metrics would allow the measurement of the disagreement of the learners in ranking *each query*. This means we would need to select whole queries (i.e., query-level selection) instead of sampling documents from the unlabeled set. This may be fine for data sets that contain few documents per query, but doing query-level selection on the LETOR 3.0 data sets would imply selecting at least 1,000 documents per round, jeopardizing our goal of minimizing the size of the training sets that need to be labeled.

Accordingly, our method uses an ensemble approach to QBC where different algorithms are used to produce diverse rankings at each round. Thus, we train three distinct algorithms using the same training data (the labeled data available at each round) and rank the remainer of the unlabeled set using these three learners. Then, for each document of each query, we calculate a simple metric to determine which documents of that query the three learners most disagree in ranking. At each round, we select the first m documents from each query that have the highest value for the disagreement metric described later. To rank the unlabeled sets, we use three algorithms as our committee: SVMRank (Joachims, 2002), RankBoost (Freund, Iyer, Schapire, & Singer, 2003), and RLR (Veloso et al., 2008). These algorithms are trained using the labeled set gathered so far and then used to rank the remaining instances in the unlabeled set.

We chose to sample $m = 5$ documents per query per round simply to be able to compare our results with our main baseline (proposed by Donmez & Carbonell, 2008). As we discuss in the Results section, this works fine for the LETOR 3.0 data sets, which have few queries and many documents per query. However, our method can be easily adapted to either (a) perform a query-level selection for data sets with many queries and few documents per query (using Kendall's τ as a measure of rank disagreement, for instance) or (b) perform first a query-level selection and then a document-level selection in data sets with many queries and many documents per query. The latter option is probably the most reasonable because selecting all documents of a query, even in data sets with few documents per query, would likely introduce noisy instances into the selected set, possibly hurting the results. Furthermore, the second option provides greater flexibility because we could tune the number of queries selected per round, as well as the number of documents selected per query per round, adapting the number of selected samples per round to a number compatible with the characteristics of the data set at hand and the labeling budget. We have implemented both variations (a and b) and discuss the implementation and results from experiments run on the LETOR 4.0 collection in the Adding Query-Level Selection section.

Measuring rank disagreement. At each round, the unlabeled documents of each query are ranked by the members of the committee using models trained on the labeled sets accrued so far. If we were performing query-level selection, as in Cai et al. (2011), we would need to measure the disagreement of each ranking on the query level. Instead, our method works at the document level; that is, we want to measure the disagreement among the learners about the ranking of each document. We select those documents that vary the most in their positions in each ranking. One simple metric would be to calculate the variance or standard deviation of each document's positions. For example, if we have three learners and document d_i is ranked in positions 10, 15, and 20 of each ranking, this document would have a ranking variance of 25 (or a standard deviation, σ, of 5). The problem of this approach is that it gives the same importance to documents whether they appear at the top or at the bottom of the rankings. For instance, document d_j appearing in positions 110, 115, and 120 would have the same variance of 25. In L2R, we are usually more concerned with the very top of the ranking (Granka, Joachims, & Gay, 2004); thus, it would seem preposterous to assign the same value of disagreement to documents d_i and d_j in the earlier example. A simple solution is to use the coefficient of variation (CV), which is a normalized measure of dispersion. The CV is defined as σ/μ (standard deviation over the mean). Now, documents far down the rank will need a much greater variation in their rankings to be selected for labeling. For example, a document d_k with rankings 200, 300, and 400 will have the same CV as document d_i with rankings 10, 15, and 20. Thus, this metric will prioritize the choice of documents with disagreeing rankings that are closer to the top in all three rankings but will still sometimes select documents that are farther from the top if they have a much bigger divergence in the rankings produced by the three models.

DOI: 10.1002/asi

The Two Stages Together

Figure 1 shows how the two stages are combined to produce a small yet effective training set. In the first stage (left), ARLR is used to produce a training set from scratch. It does this by inserting an initial document into the empty training set $\mathcal{D}$ (1) (it chooses the document that shares the most feature-values with all the other samples in the unlabeled set, $\mathcal{U}$). Then it loops through each and every of the unlabeled documents (2), creating a projection of the training set accrued so far (2a) and deriving association rules from these projected sets (2b). The document that produces the smallest amount of rules is selected (3), labeled, and inserted into the training set $\mathcal{D}$, but not removed from $\mathcal{U}$ (5). This process (2, 3, 5) is repeated until a document already selected is chosen again (4). Once ARLR converges and stops, the selected training set $\mathcal{D}$ is passed on to the QBC stage. At this point, the selected documents are *removed* from the unlabeled set $\mathcal{U}$, from which the QBC stage will select more samples.

In the second stage (QBC, to the right), the training set produced by ARLR is initially used to train the committee members (I). After ranking the unlabeled set (II), the *CV* is calculated for all documents in $\mathcal{U}$ (III) and used to select five documents from each query in the unlabeled set (IV). After being labeled, these documents are inserted into the training set $\mathcal{D}$ and removed from $\mathcal{U}$. This ends one round of the QBC stage. If more rounds are required, the process is repeated using the augmented training set (transition from V to I).

In our experiments, we ran the loop to the right (QBC) for 25 rounds for evaluation purposes. At each new round, the training set composed of the ARLR-selected documents plus the instances selected in all previous rounds is used to produce the committee ranking models, rank the unlabeled set, and select new documents based on the calculated *CV* values.

Experimental Evaluation

Data Sets

To evaluate the effectiveness of our method, we performed extensive experimentation on the L2R (LETOR) benchmark data sets version 3.0. LETOR 3.0 is composed of six web data sets plus the OHSUMED corpus. The web data sets contain labeled instances selected from web pages obtained from a 2002 crawl of the .gov top level domain (TLD). These collections are separated in three search

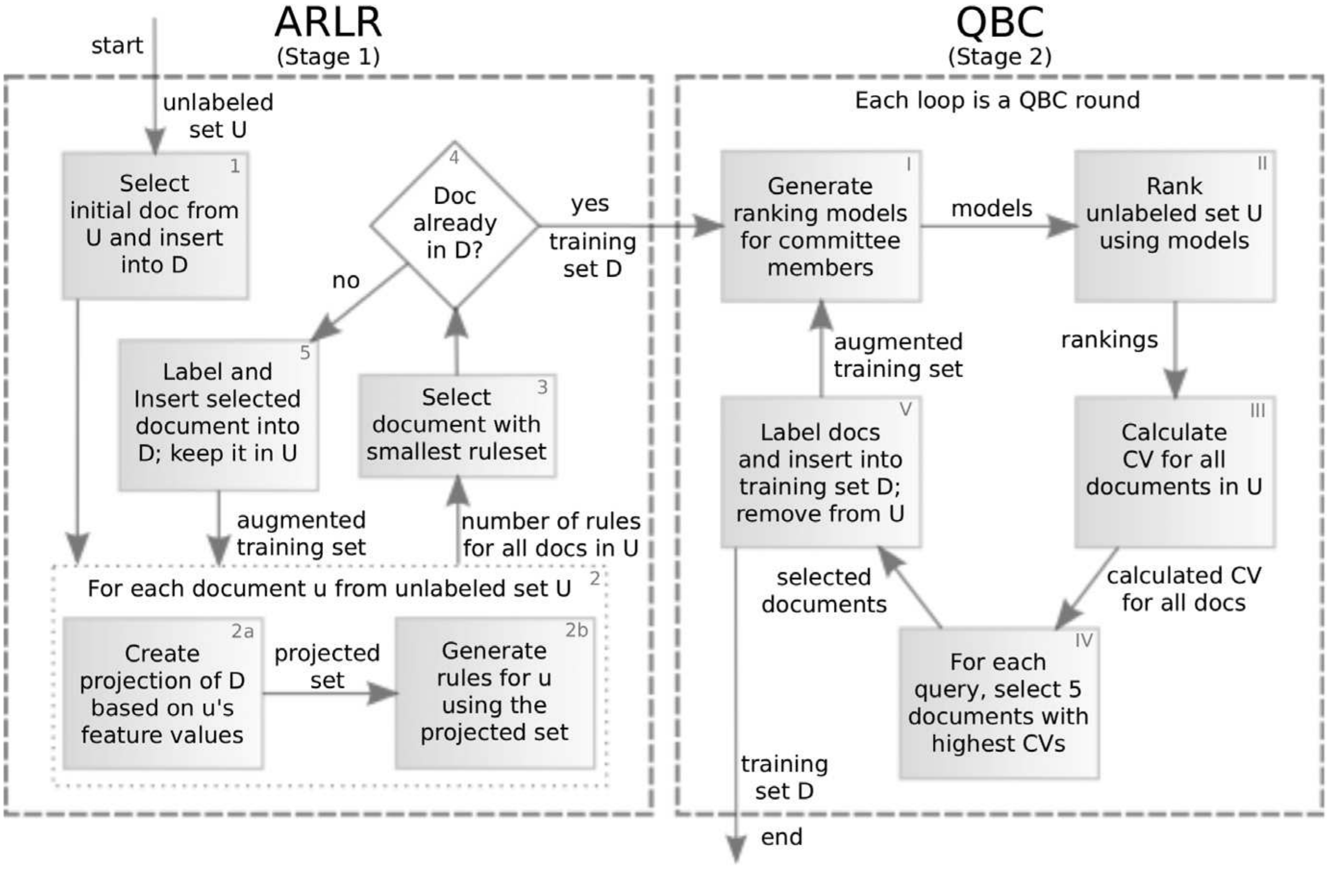
ARLR
(Stage 1)
start
unlabeled set U
1 Select initial doc from U and insert into D
4 Doc already in D?
yes
training set D
no
5 Label and Insert selected document into D; keep it in U
3 Select document with smallest ruleset
augmented training set
number of rules for all docs in U
2 For each document u from unlabeled set U
2a Create projection of D based on u's feature values
projected set
2b Generate rules for u using the projected set
QBC
(Stage 2)
Each loop is a QBC round
I Generate ranking models for committee members
models
II Rank unlabeled set U using models
augmented training set
rankings
V Label docs and insert into training set D; remove from U
III Calculate CV for all documents in U
selected documents
calculated CV for all docs
IV For each query, select 5 documents with highest CVs
training set D
end

FIG. 1. Diagram depicting the two-stage active learning method.

DOI: 10.1002/asi

tasks—topic distillation (TD), home-page finding (HP), and named page finding (NP)—and contain two sets each (namely, Text REtrieval Conference [TREC] 2003 and 2004). The TD data sets are basically composed of "informational" queries, that is, queries whose targets are documents containing information about a specific topic, theme, or entity. The HP and NP data sets are focused on "navigational" queries whose target is usually a single, specific web page.

These collections contain instances represented by 64 features for the top 1,000 documents returned for a specific set of queries using the BM25 model (Qin, Liu, Xu, & Li, 2010). These data sets use a binary relevance judgment indicating whether a document is relevant to a given query. We evaluate our method on all the largest, most diverse LETOR 3.0 web data sets. In all data sets, we have used the query-based normalized versions as suggested by the producers of the LETOR benchmarking data sets. We also use five-fold cross validation for all results reported and the evaluation script provided in the LETOR package to generate the final MAP and NDCG metrics (see Baeza-Yates & Ribeiro-Neto [2011], for definitions of these metrics).

Experimental Setup

In our active learning scenario, we consider the original training sets of each data set as unlabeled sets from which we select instances to be labeled. The label for these instances is not used until they are selected, and we assume that a human annotator evaluates and labels each selected document. The active learning process is done offline: After the documents are selected and labeled, the selected sets are used as training sets by an L2R algorithm to create a ranking model that can then be used to rank the test sets (or new user queries). Our active method, ARLR-QBC, is run on each fold of each data set in the following manner.

First stage (ARLR). In the first stage, the unlabeled and test sets are discretized using the Tree-based Unsupervised Bin Estimator (TUBE) proposed by Schmidberger and Frank (2005). TUBE is a greedy nonparametric density estimation algorithm that uses the log-likelihood measure and a top-down tree-building strategy to define varying-length bins for each attribute in the data set. All sets are discretized using 10 varying-length bins. The discretization is necessary because ARLR uses association rules with nominal values in the antecedent. Then the unlabeled set is partitioned into five vertical partitions as proposed by Silva et al. (2011). Each partition contains all unlabeled instances, but only a group of each instance's features. ARLR is run on each partition, selecting instances based on distinct feature sets. This process is done to increase the number of selected instances, because for each partition the algorithm will choose those that are most informative given the features in that partition. It also improves the diversity of the samples because they are selected based on distinct characteristics. The final product of this first stage of the method is a very small labeled set that we use as an initial training set for stage 2.

Second stage (QBC). The first step in stage 2 is to determine the best parameters for the algorithms used in the committee. To do that, we split the tiny labeled sets produced in stage 1 into training (50%) and validation (50%) sets. We then run the algorithms varying the parameters and choose the parameter(s) that gives us the best MAP. Stage 2 progresses in rounds where, at each round, five instances are selected per query. This number is used so that we can compare our method with Donmez, one of our baselines. At the first round, we use the labeled set produced in stage 1 to train our three supervised algorithms: SVMRank, RLR, and RankBoost. The models produced are used to rank the unlabeled sets (which, at this point, are the original training sets minus the instances selected in stage 1). Then, for each query, the rankings are evaluated and the metric described in the Stage 2: Expanding the Selection section (*CV*) is calculated for each document. The five documents with the highest *CV* value are then selected to be labeled. At the end of the round, five times the number of queries in the unlabeled set need to be labeled by the human annotators (in TD2004, for instance, $5 \times 45 = 225$ documents are selected per round; for all data sets, the number of documents selected per round represents 0.5% of the unlabeled sets). Once these documents are labeled, they are inserted into our labeled training set and removed from the unlabeled set. As another round starts, the augmented training sets are used to train the three committee members and the process described earlier is repeated. This procedure can be repeated as many times as desired or until the labeling budget is exhausted.

We performed extensive comparison of our method with the chosen baselines and provide the results of statistical significance tests where appropriate.

Baselines

Donmez. Our main baseline, which we call "Donmez," is an implementation we did of the method described by Donmez and Carbonell (2008). The Donmez method is based on the assumption that those instances that change the current model the most are more "interesting." The rationale behind this strategy is that the initial model is far from the ideal hypothesis and that by selecting the instances that are estimated to change the current model the most, it will approach the best hypothesis faster. Instead of retraining the learner on each and every unlabeled sample one by one, Donmez and Carbonell propose calculations to estimate the change in the current learner for SVMRank. The original method starts with an initial labeled set composed of 15 randomly picked documents plus exactly one relevant document per query. We have implemented this initial set selection (i.e., selecting an initial set containing 16 instances per query before starting the round-based selection proposed by Donmez) in the following manner: An "oracle" uses the labeled set to determine which feature gives the best mean

DOI: 10.1002/asi

reciprocal rank (MRR; see Baeza-Yates & Ribeiro-Neto [2011], for a definition of this metric) value by ranking all instances using the values of each feature in turn. Then we start selecting instances for each query in the order determined by this selected feature: If, before selecting 16 instances, one of them is found to be relevant, then the remaining instances (up to 16) are randomly sampled. If a relevant sample is not found in the first 16, then we keep selecting in the order determined by the oracle until either 50 samples are selected or a relevant one is found. Thus, the initial sets contain at least 1.6% of the original training sets but are usually a little larger, because for some queries, a relevant document is not found in the first 16 sampled documents. After gathering these initial sets, we proceed by using Donmez's method to select, at each round, five new documents per query (amounting to 0.5% of the original training sets selected per round). To make the comparison easier, our method also selects 5 documents per query per round, and we also run our experiments for 25 rounds. Because Donmez uses some randomly selected instances in the initial sets, results reported are averages for 10 runs for each fold.

Random-QBC. To understand how our initial selection method, ARLR, influences the second stage, we ran QBC with randomly chosen initial sets (16 samples per query). Results reported are averages for 10 runs on each fold.

ARLR-Random. Conversely, we also ran a random second stage to show that QBC is an effective selection strategy. In this baseline, we start with the ARLR-selected sets and then run a random selection process where five instances are selected per query per round. Results reported are averages for 10 runs on each fold.

Random. In this baseline, all instances are randomly selected. The amount of instances selected is the same as Donmez's original method: It starts with an initial random set of 16 instances per query (1.6% of each set) and selects 5 random instances per query per round (or 0.5% of each set). Results reported are averages for 10 runs on each fold.

SVM Full. These are the results provided by the LETOR publishers for the SVMRank algorithm. This is a natural baseline because SVMRank is used by Donmez and by us to generate the final ranking in each round, thus allowing a direct comparison regarding the active selection process. SVMRank is also a popular L2R method[2] and a strong baseline in several LETOR data sets. We include this baseline as a line in each plot as a reference to put our results and the other baselines in perspective. Notice that these results are for SVMRank using 100% of the training sets. In the Results section, we also compare our method with the other 11 supervised baselines published by the LETOR producers. We do not plot these baselines in Figures 2 and 3 so as not to clutter them.

ARLR-RLR. This line indicates the result obtained by using ARLR to select instances and RLR to rank the test set (these are the results that appear in Silva et al. [2011]). Notice that ARLR selects instances until it naturally converges and stops selecting new instances. Thus, it is not possible to run ARLR in a round-based fashion. We present this result as a line for readability (because this result should actually be a point in the extreme left of each plot).

Published LETOR 3.0 results for the considered collections. As mentioned earlier, we also compared the results of our method (ARLR-QBC) with all reported LETOR 3.0 results in the considered collections. As we shall see, when we consider the best results of all the 26 experimental rounds, as well as the best results in the first 8 rounds, ARLR-QBC is the best overall method for all data sets on average and the best performer in three of six data sets. These are some of the best results ever reported in the literature for these collections.

Results

Notice that our method (ARLR-QBC), Donmez's, Random-QBC, ARLR-Random, and Random methods are round-based active methods where an initial selection is performed as explained earlier (round 0 or stage 1 of our method); then at each new round, new instances are labeled and inserted into the (growing) training sets (rounds 1–25 or stage 2 of our method). For these three methods, at the end of each round we run SVMRank using the current training sets obtained by the active selection processes to rank the test sets and obtain the results presented. As a consequence, these three methods appear in the plots as curves containing 26 points each.

SVM Full and ARLR-RLR are not round based: The results for SVM Full were obtained by running SVMRank using the complete original training sets to rank the test sets, whereas ARLR-RLR results were obtained by running RLR with the sets selected by ARLR. Although SVM Full is a fully supervised baseline, whereas ARLR-RLR is an active learning baseline, neither is round based and both appear in the plots as flat lines only to improve readability. If correctly portrayed, ARLR-RLR would be a point in the extreme left of each plot, whereas SVM Full would not even appear in the plots because it uses 100% of the training sets and our plots show on the *x*-axis only up to 15%.

Informational data sets. Figure 2 shows the performance of our method against the chosen baselines on TD2003 and TD2004. These are the most relevant data sets because they are built using the more general TD queries (in contrast with the more specific navigational queries used to build the HP and NP data sets). The plots show on the *y*-axis the MAP

[2] Its code is freely available at http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html.

(left) and NDCG@10 (right) obtained at each round for each method. The *x*-axis indicates the percentage of the unlabeled set selected at each of the 26 rounds. The first point on the left of the plot for ARLR-QBC is the result obtained by running SVMRank with the ARLR-only selected set (stage 1). The other 25 points correspond to the 25 rounds of the stage 2 QBC method. ARLR, as explained in the Stage 1: Active Sampling Using Association Rules section, naturally converges; therefore, the amount of samples it selects for labeling varies from one data set to another. Thus, the ARLR-QBC lines start at 2.28% and 1.33% (*x*-axis) for TD2003 and TD2004, respectively.

TD2003 is exceptional in our results, as the ARLR-selected initial set performs very well in this data set using both SVMRank and RLR (i.e., ARLR-QBC and ARLR-RLR lines in the plots). Using only the initial data set, SVMRank obtains a MAP of 0.2881 and ARLR-RLR a MAP of 0.2728, surpassing not only SVMRank using the full training set (SVM Full) but also (in the case of ARLR-QBC) all supervised baselines published by the LETOR producers (see more details in the Analysis section). With such a good initial result, it is hard to expect much improvement. Although ARLR-QBC manages to obtain an even higher MAP at round 4 (0.2908) and a peak NDCG of 0.3710 at round 5, what we see is the MAP descending and stabilizing at around 0.265 and the NDCG staying mostly close to 0.365. Donmez starts at lower values for both metrics but eventually catches up (at round 4), obtaining similar results to ARLR-QBC at later rounds (from 10 up), specifically in terms of the MAP.

TD2004 shows a different picture, with our method also surpassing Donmez in the initial rounds for both metrics, but starting with a lower performance. We can see that ARLR does not select such a good initial set as in the case of TD2003, but the QBC selection performs very well, bringing the metrics on par with SVMRank using the complete training set (i.e., SVM Full) just after round 1 (1.84% selected). The MAP keeps growing until reaching a peak of 0.2388 in round 22, whereas the NDCG reaches 0.3335 in round 6 and then slowly declines to 0.32. These results are not as exceptional as those for TD2003; still, the MAP for all rounds (with the exception of rounds 0 and 2) beat 9 of the

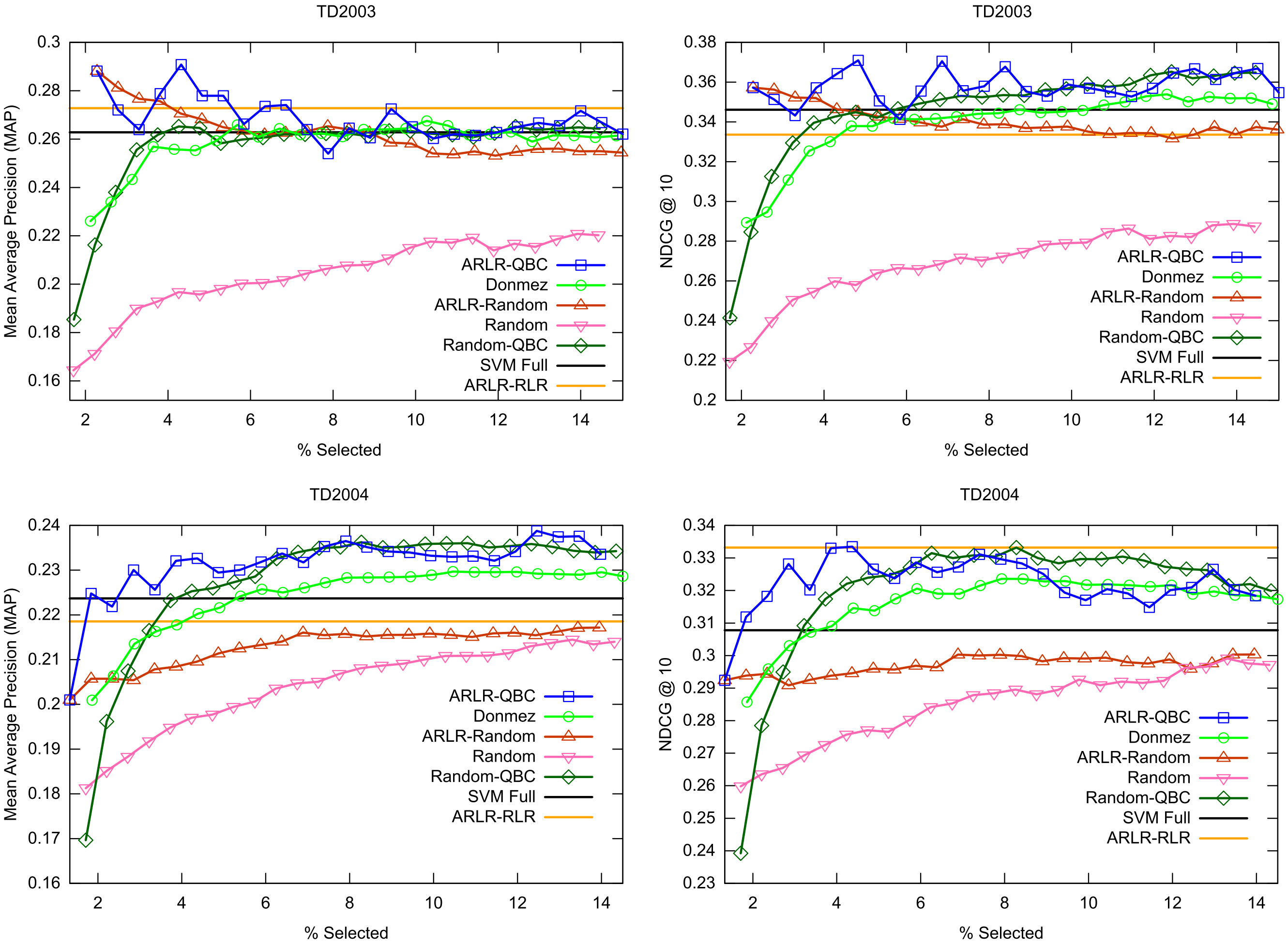

FIG. 2. TD2003 and TD2004: ARLR-QBC and baselines comparison. MAP (left) and NDCG@10 (right) versus % of samples selected from unlabeled set. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

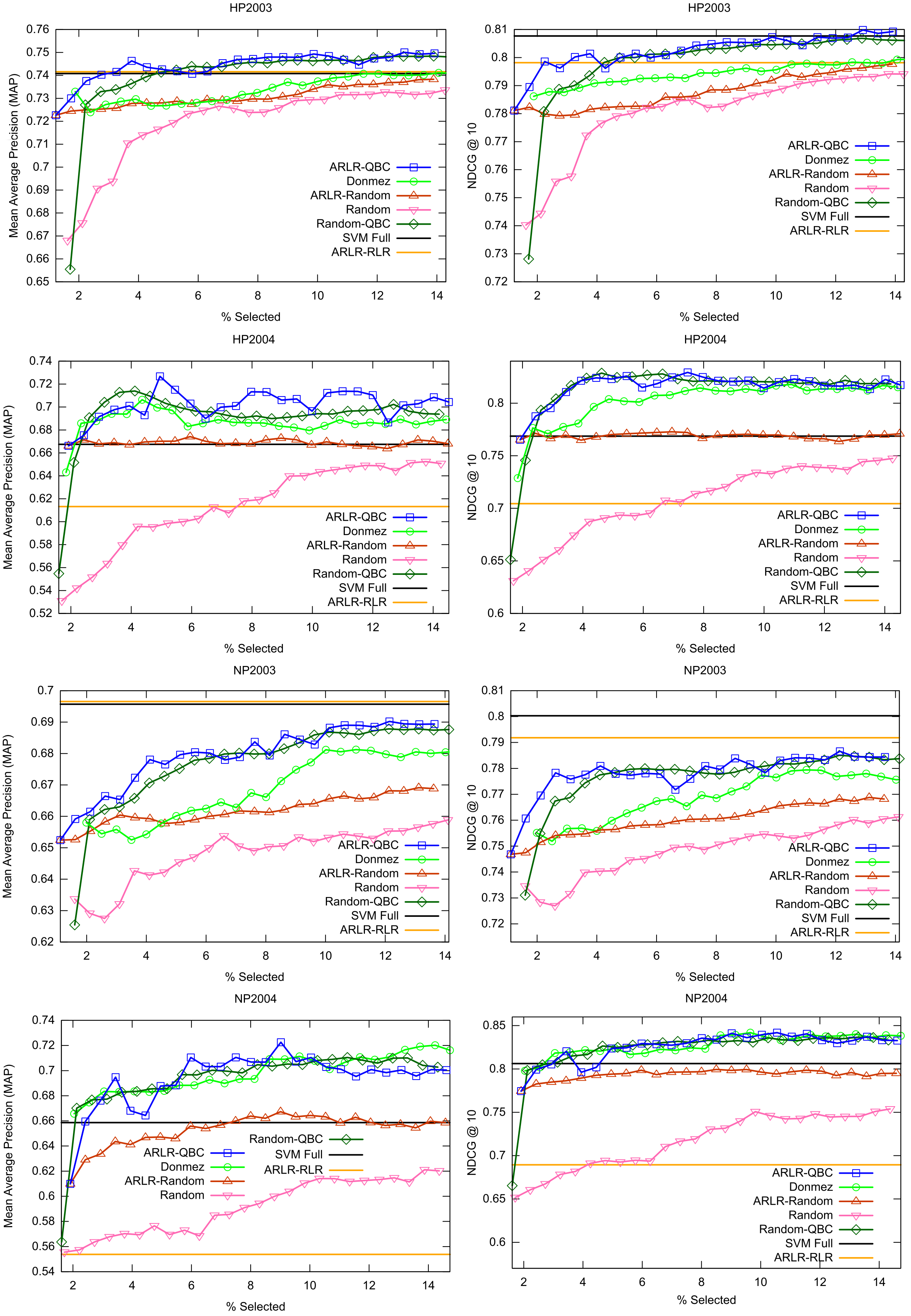

FIG. 3. HP2003, HP2004, NP2003, and NP2004: ARLR-QBC and baselines comparison. MAP (left) and NDCG@10 (right) versus % of samples selected from unlabeled set. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

12 supervised baselines published by the LETOR producers, and the NDCG (with the exception of rounds 0, 1, and 20) beat 8 of the 12 baselines. If we consider only the peak results, then ARLR-QBC beats 11 of the 12 supervised baselines (losing only to RankBoost, the best baseline in this collection).

Comparison with Random. Both ARLR-QBC and Donmez surpass the Random baseline with 99% confidence level on all rounds in both informational data sets.

ARLR-QBC versus Random-QBC. Figure 2 shows that using QBC with random initial sets severely affects the results in the initial rounds. With ARLR-selected initial sets, the QBC round-based method is able to select highly effective training sets in the first few rounds, which, we believe, is an important characteristic of a practical active learning method. A paired Student's *t* test shows that ARLR-QBC is better than Random-QBC with 99% confidence level in the first eight rounds (0–7) on both TD2003 and TD2004. If we consider all rounds, our method is significantly better with 95% confidence level on TD2003. It is also, on average, better than Random-QBC on TD2004 (considering all rounds), but not significantly so.

ARLR-QBC versus ARLR-Random. On TD2003, ARLR-Random starts with the high-quality set from ARLR and as random instances are added, the results steadily decline. A paired *t* test shows that, although ARLR-QBC is not significantly better in the first eight rounds, it is better with 99% confidence if we consider all rounds. TD2004 shows a different picture, with ARLR-QBC beating ARLR-Random with 99% confidence if we consider either the first eight rounds or all rounds. Together with Random-QBC, this baseline shows that the combination of ARLR and QBC is very effective.

ARLR-QBC versus Donmez. On TD2003, the ARLR-QBC results are significantly better than Donmez's with 95% confidence level up to the fourth round for the MAP (4.32% selected) and up to the fifth round for the NDCG (4.83% selected). On TD2004, our method is significantly better than Donmez's with 95% confidence level up until the 13th round (7.91% selected) for both metrics.

From these results we can see that ARLR-QBC selects highly effective sets for the informational data sets with only a few rounds (<6% selected). Depending on the metric used and the data set, peak results are obtained in the initial or later rounds, but in general, the results are quite good. Although we run the experiments for 26 rounds, selecting almost 15% of the unlabeled sets, Figure 2 shows that the sets selected in the initial rounds (0–7) are usually quite effective.

Navigational data sets. Figure 3 shows that our method performs very well on HP2003, NP2003, and NP2004 when compared with SVM Full. It surpasses this baseline on both metrics in round 1 on HP2004 and NP2004, and in round 4 on HP2003 (MAP). NP2003 is the exception, as ARLR-QBC never reaches the same level as both SVM Full and ARLR-RLR. To better understand these results, let us first take a look on how these results compare with the 12 supervised baselines published by the LETOR producers. In HP2003, ARLR-QBC obtains a MAP of 0.7463 by round 5 (selecting only 3.77% of the unlabeled set), surpassing five of these baselines (including SVMRank and RankBoost). The NDCG results are a little worse, beating only three of these baselines (Regression, SVMMAP, and FRank) by round 5 (with a value of 0.8013). The results are much better for HP2004, where by round 6 (4.95% selected) the MAP obtained (0.7268) beats all 12 published supervised baselines. The NDCG@10 for round 5 (0.8241) is better than that for all baselines with the exception of AdaRank-MAP. On NP2003, although ARLR-QBC never reaches the same performance level of SVM Full, by round 22 (12.14% selected) the MAP of 0.6902 is still better that of 9 of the 12 supervised baselines. The NDCG fares a little worse, surpassing only 4 of these baselines by round 3. Our method also obtains outstanding results on NP2004, beating all supervised baselines by round 3 with a MAP of 0.6948 (and reaching a peak of 0.7226 in round 14, which represents a gain of 5.24% over the best baseline, Regression-Reg). The NDCG@10 obtained at round 3, 0.8205, also surpasses all 12 baselines, achieving a peak value of 0.8419 in round 17 (a gain of 3.58% over the best baseline, ListNet).

Comparison with Random. ARLR-QBC and Donmez are significantly better in all 26 rounds with 99% confidence level on HP2004 and NP2004 (both metrics). In HP2003, ARLR-QBC is still significantly better with 95% confidence level from rounds 0 to 18. In contrast, for the same confidence level, Donmez is only better than Random up to the third round. In NP2003, our method is better than Random in all rounds with 95% confidence on both metrics, whereas Donmez is better only in rounds 0 to 2 and 11 to 25 if we consider the MAP.

ARLR-QBC versus Random-QBC. Differently from the informational data sets, here ARLR-QBC has a harder time beating Random-QBC. It is still significantly better with 95% confidence in the first eight rounds (0–7) in HP2003 and NP2003 (both metrics). It is also better with 95% confidence on all rounds in HP2004 and NP2003 (considering the MAP only).

ARLR-QBC versus ARLR-Random. In contrast with Random-QBC, ARLR-Random is easily beaten by ARLR-QBC on all data sets, as shown in Figure 3. In fact, the addition of randomly selected instances to the original ARLR sets almost does not improve the results on the 2004 data sets. On the 2003 data sets, there is a small improvement that mostly mirrors the curves for the Random baseline. On all data sets, a *t* test shows that ARLR-QBC is significantly better with 99% confidence if we consider the first eight rounds or all rounds.

ARLR-QBC versus Donmez. On a round-by-round basis, ARLR-QBC beats Donmez with 90% confidence in HP2003 from rounds 4 to 11 for the MAP. In HP2004, our method obtains a better MAP with 95% confidence on rounds 6 to 8 and 12 to 20, and a better NDCG on rounds 0 to 13. In NP2003, with a confidence of 95%, the MAP is better on rounds 5 to 10 and the NDCG on rounds 2 to 8 and 12 to 14. In NP2004, there is a statistical tie on all rounds.

Gains obtained over Donmez and SVM Full. Table 1 summarizes the gains obtained by our method compared with Donmez and SVM Full. We calculate the gains for each metric separately: MAP and NDCG@10. We also separate the calculations into two partitions: the average and maximum gains achieved in rounds 0 to 7 (columns AG7% and MG7%, respectively) and the overall (i.e., considering all rounds) average and maximum gains (AG% and MG%). The reason for doing this partitioning is that, although we run our method for 26 rounds, we believe that in most real-world scenarios, only a few rounds should be run. One of the advantages of an active learning method is exactly to introduce less “noise” into the selected training sets, thus allowing for the supervised L2R method to obtain better effectiveness. This insight is corroborated by the gains obtained in all data sets (except NP2003) by our method over supervised baselines that use the complete training sets (i.e., SVM Full and the published LETOR baselines) and by the stable or falling results that we see in many data sets in the final rounds. By providing the average and maximum gains obtained at the first eight rounds (0–7), we want to show that our method converges faster to good results as compared with Donmez and also that it obtains competitive results selecting less than 6% of the original training sets when compared with a strong supervised method using the complete sets (i.e., SVM Full).

Average and maximum gains over Donmez. From Table 1 we can see that ARLR-QBC obtains positive gains over Donmez on all data sets with the exception of NP2004. The improvement is more impressive on the informational data sets, where ARLR-QBC has average results on rounds 0 to 7 that are over 10% better than Donmez on TD2003 (both metrics) and more than 6% better on TD2004 (both metrics). The overall average gains are also good, reaching more than 5% for the NDCG on TD2003. The results for the navigational data sets are more modest, but still quite reasonable, with the average gain on rounds 0 to 7 reaching 3.4% on HP2004 (NDCG). Observe from the MG% column that the maximum gain is very often obtained in the initial rounds.

Average and maximum gains over SVM full. From the average gain obtained over SVM Full in the first eight rounds (Table 1, column AG7% to the right), we can see that ARLR-QBC surpasses this strong supervised baseline in four of six data sets for the MAP and half of the data sets for the NDCG@10. This means that our method is able to surpass this strong supervised baseline (which uses 100% of the training sets) while selecting and labeling less than 6% of the original training sets. Moreover, the overall average gain (AG% to the right) is positive in five of the six data sets for the MAP and four for the NDCG. The AG reaches more than 5% for the MAP on HP2004 and NP2004, more than 6% for the NDCG on HP2004, and more than 4% for TD2004.

TABLE 1. Gains obtained by ARLR-QBC over Donmez and SVM Full: MAP and NDCG@10.

	ARLR-QBC vs. Donmez				ARLR-QBC vs. SVM Full				Peak values*	
	AG7%	MG7%	AG%	MG%	AG7%	MG7%	AG%	MG%	AQ	LB
MAP										
TD2003	***10.54***	***26.32*** (0)	3.93	***26.32*** (0)	**5.39**	***10.65*** (4)	2.27	***10.65*** (4)	**.2907**	.2753
TD2004	**6.47**	***11.43*** (1)	3.65	***11.43*** (1)	0.44	3.99 (6)	3.38	**6.73** (22)	*.2388*	.2614
HP2003	1.69	2.31 (6)	1.55	2.31 (6)	*–0.37*	0.75 (5)	0.44	1.25 (23)	*.7500*	.7710
HP2004	0.99	3.71 (0)	2.20	4.30 (19)	4.24	**8.88** (6)	**5.16**	**8.88** (6)	.7267	.7219
NP2003	2.22	3.64 (6)	1.89	3.64 (6)	*–4.20*	*–2.53 (6)*	*–2.38*	*–0.79 (22)*	*.6902*	.7074
NP2004	*–1.80*	1.71 (3)	*–0.69*	3.22 (8)	1.49	**5.46** (3)	**5.24**	**9.68** (14)	**.7225**	.6866
NDCG@10										
TD2003	***10.39***	***22.92*** (0)	**5.68**	***22.92*** (0)	2.42	**7.19** (5)	3.47	**7.19** (5)	.3710	.3571
TD2004	**6.31**	**8.93** (1)	2.17	**8.93** (1)	4.12	**8.34** (6)	4.71	**8.34** (6)	*.3335*	.3504
HP2003	1.04	1.40 (4)	1.15	1.46 (23)	*–1.52*	*–0.78 (5)*	*–0.67*	0.26 (23)	*.8098*	.8384
HP2004	3.40	**5.15** (4)	1.66	**5.15** (4)	4.93	**7.43** (7)	**6.12**	**7.88** (11)	*.8293*	.8328
NP2003	2.73	3.51 (3)	1.48	3.51 (3)	*–3.67*	*–2.24 (6)*	*–2.79*	*–1.71 (22)*	*.7866*	.8068
NP2004	*–1.15*	0.93 (7)	*–0.28*	1.49 (12)	*–0.06*	2.26 (7)	2.40	4.43 (17)	.8419	.8128

Note. Average gain for rounds 0 to 7 (AG7%), maximum gain for rounds 0 to 7 (MG7%), average gain for all rounds (AG%), and maximum gain for all rounds (MG%). Numbers in parentheses indicate in which round the maximum gain was obtained. Values in italics indicate negative gains, values in bold indicate a gain greater than 5%, and values in italics and bold a gain greater than 10%.

*Peak value obtained by ARLR-QBC in all rounds (AQ) and the value obtained by the best supervised algorithm of the 12 results published by the LETOR producers for that data set (LB).

Peak values for ARLR-QBC and all LETOR 3.0 baselines. The last two columns of Table 1 show the best MAP and NDCG@10 obtained by ARLR-QBC in all rounds (column AQ) and the best supervised result published by the LETOR producers (column LB). The LETOR publishers encourage researchers to submit results obtained by new algorithms and methods, and have accrued so far results for 12 distinct L2R algorithms.[3] As listed in Table 1, ARLR-QBC's peak results beat the best LETOR baseline in three data sets for the MAP (obtaining a gain of more than 5% in TD2003 and NP2004, and a gain of less than 5% in HP2004) and in two for the NDCG@10 (namely, TD2003 and NP2004). The best LETOR baselines for each data set are as follows (MAP, NDCG@10): TD2003 ListNet, RankSVM-Primal; TD2004 RankBoost, RankBoost; HP2003 AdaRank-MAP, AdaRank-MAP; HP2004 AdaRank-MAP, AdaRank-MAP; NP2003 RankBoost, RankBoost; and NP2004 Regression-REG, ListNet. Observe that all of the 12 results for these baselines were obtained using the complete (i.e., 100%) training sets to rank the test sets. ARLR-QBC, being an active learning method, uses 15% or less of the training sets.

Paired Student's t test. We have also performed a paired difference *t* test using the MAP and NDCG differences for all rounds of our method in comparison with Donmez's results. The tests show that our method is significantly better with 95% confidence level in all data sets but NP2004 for the NDCG@10 and on four data sets for the MAP (the exceptions are NP2003 and NP2004).

Analysis

Comparison with other LETOR 3.0 baselines and ARLR-RLR. Figure 4 shows the comparison of the peak MAP obtained by ARLR-QBC in the first eight rounds (i.e., rounds 0–7), ARLR-RLR (as presented by Silva et al., 2011), and three published LETOR baselines (that use the complete training sets): SVMRank, RankBoost, and AdaRank-MAP. SVMRank (or SVM Full, as we have been calling it) is an obvious choice. We chose RankBoost and AdaRank-MAP because they are the only two algorithms (out of the 12 baselines published by the LETOR producers) that obtain the highest MAP scores in two data sets each: RankBoost in TD2004 and NP2003, and AdaRank-MAP in HP2003 and HP2004. The ARLR-QBC results shown are the peak values obtained in rounds 0 to 7 (i.e., using less than 6% of the original training sets).

As shown in Figure 4, ARLR-QBC obtains better results than ARLR-RLR in all data sets, with the exception of NP2003. ARLR-QBC obtains especially good results on the data sets where ARLR-RLR did worse: HP2004 and NP2004. On these data sets, ARLR-QBC obtains gains of 18% and 25% over the ARLR-RLR MAP results, respectively. On the informational data sets, the gain is also significant: 6.5% for both TD2003 and TD2004. On HP2003, the gain is marginal (less than 1%), and in NP2003, ARLR-QBC obtains a MAP that is 2.5% smaller than that achieved by ARLR-RLR. These results show that, although ARLR-RLR is able to select very small data sets with very good effectiveness, expanding the selection using the QBC second stage is very much worth the extra labeling cost. Using only ARLR, we select from 1.12% to 2.28% of the original sets; ARLR-QBC is able to obtain the gains listed earlier by expanding the selection to less than 6% of the original sets.

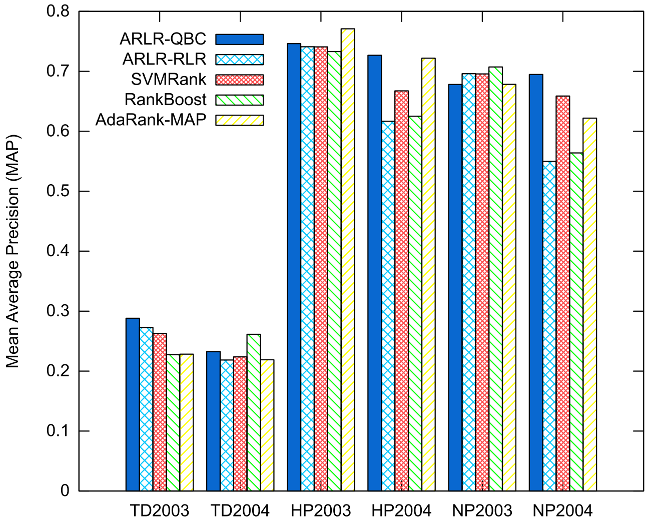

FIG. 4. Comparison of ARLR-QBC, ARLR-RLR, and three LETOR baselines: Best MAP was obtained in rounds 0 to 7. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

From Figure 4, we can also see that ARLR-QBC beats the chosen LETOR baselines in TD2003, HP2004, and NP2004.

Selection of relevant samples. As our final analysis, to better understand the differences between the compared methods and data sets, we look into how the selection of relevant instances may be affecting the results. Two simple metrics may help us gauge the influence of relevant instances in the results: the proportion of relevant instances in the selected sets and the fraction of all relevant instances present in the unlabeled sets that were selected.

In Figure 5, the plot on the left shows, for each method, how the proportion of relevant instances in the selected sets evolves as we run the rounds of the methods. On the *y*-axis, we have the percentage of relevant instances in the selected sets for each round. On the *x*-axis, as before, we have the percentage of the unlabeled sets selected at each round. The plot to the right shows the fraction the selected relevant samples represent of the *total* amount of relevant documents present in the unlabeled sets.

We first notice how the informational (first row of Figure 5) and navigational data sets (second and third rows) differ in these metrics. For the TD data sets, the proportion

[3] http://research.microsoft.com/en-us/um/beijing/projects/letor/letor3baseline.aspx

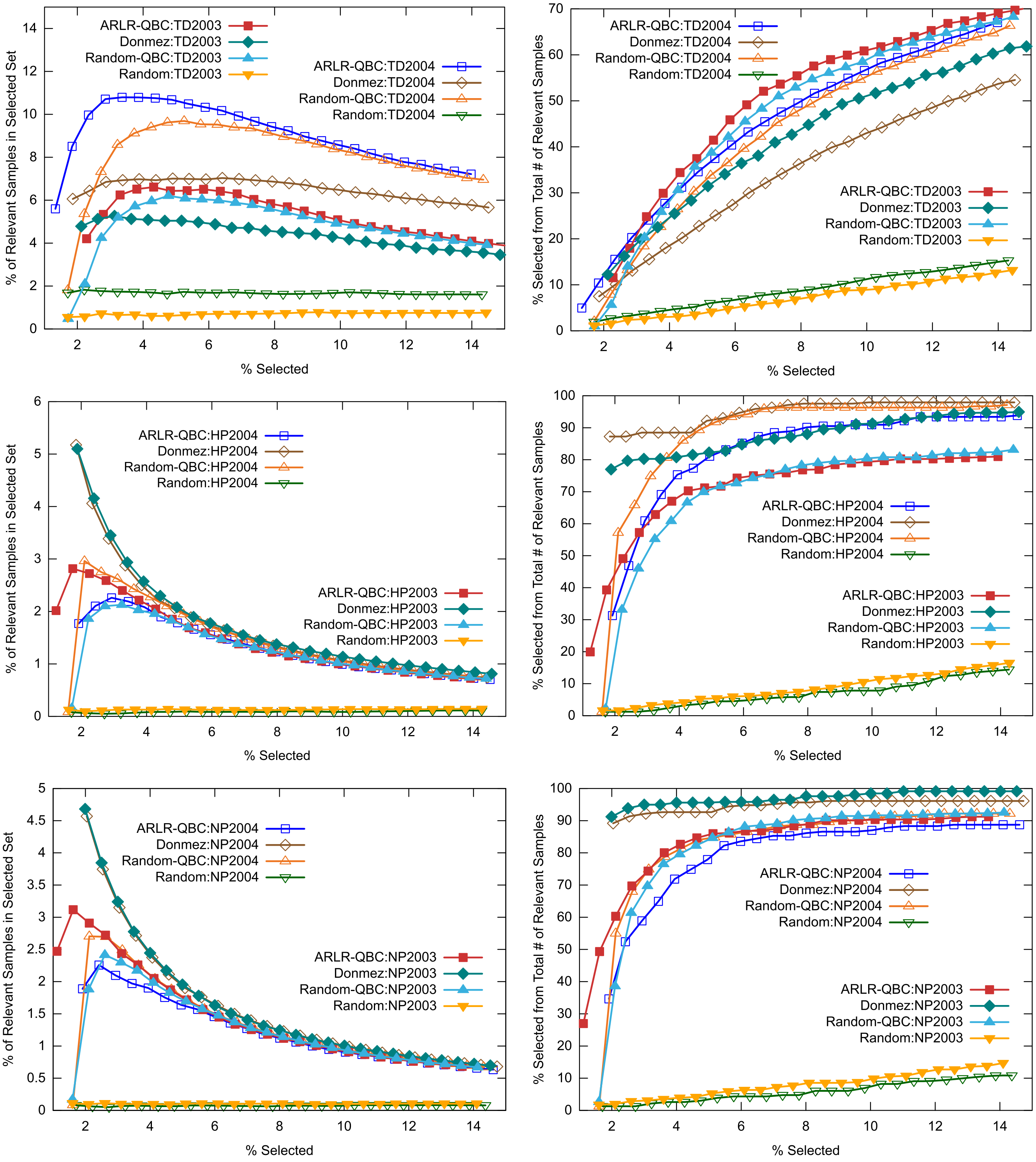

FIG. 5. All data sets: proportion of relevant instances in the selected sets (left) and the percentage they represent of the total number of relevant samples in the unlabeled sets (right). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

of relevant samples in the selected set grows in the first few rounds, indicating that both QBC and Donmez are selecting more relevant instances than nonrelevant. Then the proportion gently slopes down, although by the 26th round it is still higher (or the same) than in the initial round. Observe from the plot on the right that the initial relevant samples selected represent a minor proportion (1% to 12%) of all relevant samples in the sets, so both QBC and Donmez have room to find more relevant samples. On the HP and NP data sets, on the other hand, the Donmez initial selection method selects almost all relevant samples available in the initial round (plots on the right: from 76% to 91%), which means that as the rounds progress, the tendency is to select more nonrelevant samples, gradually reducing the proportion of relevant documents in the selected sets (plot to the left). This happens because the initial selection used by Donmez tries to start with at least one relevant sample per query, and the navigational data sets usually have only one relevant document per query (from 0.1% to 0.17% of the navigational sets are relevant documents). The informational data sets have a much higher proportion of relevant samples (2.92% in TD2004 and 1.52% in TD2003).

On TD2004, ARLR-QBC starts with the highest proportion of relevant samples in the selected set (Figure 5, left plot, first row). Donmez also starts with a high proportion, but ARLR-QBC selects many more relevant instances in the first rounds, achieving a peak percentage of almost 11% of relevant samples in the selected sets by round 3, whereas Donmez never surpasses 7%. A similar trend is observable in TD2003: Although Donmez starts with a higher proportion of relevant samples, ARLR-QBC quickly surpasses it, achieving a peak of more than 6% by round 4 (with Donmez peaking at 5% on the third round). Another interesting observation is that, although Random-QBC starts with a very low proportion, the QBC method is able to select relevant samples in a very fast rate, bringing the fraction to the same level as ARLR-QBC by round 12 on both data sets. Remember that ARLR-QBC is significantly better that Random-QBC mainly in the initial rounds (0–7). It is tempting to attribute that advantage to the higher proportion of relevant samples selected by ARLR-QBC in the initial rounds, but notice that although Donmez starts with a higher proportion of relevant samples in round 0 on TD2003, ARLR-QBC obtains much higher MAP and NDCG@10 results in this round (26.32% higher MAP; see Table 1). Thus, on the informational data sets, a combination of relevant samples and informative (relevant or nonrelevant) samples seems to be the key to ARLR-QBC's better results in the initial rounds. On the navigational data sets, ARLR-QBC and Random-QBC start with much lower proportions of relevant samples as compared with Donmez, quickly achieving their peaks in the first two or three rounds and then gently sloping down (Figure 5, left graphs). The graphs to the right in Figure 5 show that QBC is very good at finding the relevant samples in these data sets, whereas Donmez starts with almost all relevant samples already selected and does not have much room to grow. On these data sets, the selection of more relevant documents in the initial rounds may affect the results more markedly, because SVMRank needs at least one relevant sample per query to be able to generate pairwise comparisons. Nevertheless, Donmez's initial selection contains almost one relevant sample per query, and that does not necessarily translate into better results in the initial rounds. Interestingly, the data sets where ARLR-QBC is not significantly better than Random-QBC in the first eight rounds (HP2004 and NP2004) are those where Random-QBC is able to select more relevant samples in those rounds. Although ARLR-QBC beats Donmez and Random-QBC (in the initial rounds) in two of the four navigational data sets and practically ties in the other two, it seems that devising some technique to increase the amount of relevant samples selected in the initial round (i.e., the first stage, ARLR) may be a simple way of improving our method's results in navigational data sets. This hypothesis will be tested in future work. Still, as we can clearly see in the graphs for the navigational data sets, the combination of the diversity and feature coverage provided by ARLR-QBC compensate for the smaller proportion of relevant samples, allowing our method to beat or at least tie with Donmez in the initial rounds.

Adding Query-Level Selection

The two-stage method described earlier performs only document-level selection: Both ARLR and QBC select documents from all queries. QBC selects a fixed amount of documents from each query per round, whereas ARLR selects documents without taking the queries into consideration. This is ideal for data sets with few queries and many documents per query (like the LETOR 3.0 data sets). If the unlabeled set contains many queries, it is necessary to perform some query selection before document selection to avoid selecting a huge amount of documents to be labeled in the QBC stage. Because ARLR naturally converges fast, selecting documents from many different queries, it is usually not necessary to add query selection to our method's first stage. Therefore, we need to devise a query selection mechanism only for the second stage, namely, QBC.

Query selection can be easily added to our method by defining a suitable committee disagreement metric that operates at query level. As stated earlier, a common query rank correlation coefficient is Kendall's τ, which is defined as

$$\tau(R_1, R_2) = 1 - \frac{2 \times \Delta(R_1, R_2)}{K(K-1)}, \qquad (5)$$

where R_1 and R_2 are two distinct rankings of K documents, and $\Delta(R_1, R_2)$ is a function that returns the number of discordant pairs in the two rankings (Baeza-Yates & Ribeiro-Neto, 2011). This value varies from −1 to 1, where 1 indicates two identical rankings (i.e., there are no discordant pairs) and −1 indicates one ranking is the reverse of the other (i.e., all pairs are discordant).

We modify the QBC stage so that at each round it first calculates Kendall's τ between the three rankings of the unlabeled set produced by the committee models [i.e., $\tau(R_1, R_2)$, $\tau(R_1, R_3)$, $\tau(R_2, R_3)$] and then averages these three values, choosing the q queries with the lowest average values (i.e., higher rank disagreement). Then it proceeds as before, calculating the CV for the documents of the chosen queries and selecting from each chosen query the m documents with the highest CV values.

Experiments using the LETOR 4.0 collection. To demonstrate how our method can be easily adapted to perform query-level selection, we decided to conduct some experiments using the LETOR 4.0 collection.[4]

There are two data sets in this collection: MQ2007 and MQ2008. LETOR 4.0 is very different from the 3.0 version. It is based on two informational query sets from the Million Query Track[5] of TREC 2007 and TREC 2008. Instead of many documents per query and few queries, these data sets contain many queries and few documents per query. MQ2007, contains a total of 1,692 queries and an average of 41 documents per query, for a total of 69,623 documents (if we take into consideration the cross-validation evaluation method, the training set has, on average, 41,774 documents for 1,015 queries, and the test and validation sets 13,924 documents related to 338 queries). MQ2008 is much smaller, with a total of 784 queries and 15,211 documents, and an average of only 19 documents per query (average training set size of 470 queries and 9,126 documents, and test and validation sets with 157 queries and 3,042 documents). These data sets also use three relevance levels instead of just two (i.e., 0: *not relevant*, 1: *relevant*, 2: *highly relevant*). One of the main differences between these data sets and those in LETOR 3.0 is the amount of relevant documents: On MQ2007, 20% of the documents have relevance 1, and 5.5% of the documents have relevance 2. For MQ2008, the numbers are 13% and 6%, respectively. In LETOR 3.0, the data set with the highest percentage of relevant documents is TD2004, with less than 3%. All navigational data sets have less than 0.2% relevant documents.

The LETOR 4.0 data sets were built using a very different document selection strategy. Whereas in LETOR 3.0 a fixed number of documents was selected per query using the BM25, in version 4.0 each query has 8, 16, 32, 64, or 128 documents selected according to two approaches: minimal test collections and statistical sampling. In the first method, documents are selected by a measure of how much they may change the MAP, given all judgments made up to that point, if they are considered relevant or nonrelevant (Carterette, Allan, & Sitaraman, 2006). In the second approach, a specific random sample of documents is drawn and judged to produce unbiased, low-variance estimates of average precision, R-precision, and precision at typical cutoffs. Additional documents may be included to improve the quality of the estimates, if necessary (see Aslam & Pavlu, 2007).

If we were to run ARLR-QBC unmodified on these data sets, each QBC round would select 5,075 documents from the unlabeled (i.e., training) set on MQ2007 and 2,350 samples for MQ2008. That is around 12% and 26% of the unlabeled sets per round, respectively. This would obviously defeat the purpose of our method (remember that in LETOR 3.0, only 0.5% of the unlabeled set was selected per QBC round). Thus, we need to introduce query-level selection: On each round, we first select some queries and then select documents from these queries.

We call the modified method ARLR-QBC-QueryDoc, to indicate that it performs query- and document-level selection. First, ARLR is run exactly as before on the full unlabeled sets. Once ARLR converges, the obtained labeled set is fed into the QBC stage as before. The three committee models are generated from the current training set and used to rank the unlabeled set. We then calculate average Kendall's τ for the three rankings and select the q queries that have the lowest average. q is chosen to keep the proportion of documents selected in each QBC round at around the same as for LETOR 3.0 (0.5%). After the queries are chosen, the same $m = 5$ documents are selected per query using their *CV* values as before. For MQ2007, the number q of chosen queries used was 42 and 35 for MQ2008.

To better understand the impact of using query- and document-level selection simultaneously, we also ran a variation of the method that relies only on query-level selection. In this variation, which we call ARLR-QBC-Query, once the queries are chosen, we select all documents from each query in turn (ordered in ascending order of the average Kendall's τ), until we reach a number of documents similar to the QueryDoc method. That is, we do not perform doc-level selection, ignoring the *CV* and selecting all documents from the first few ordered queries.

Baselines. The baselines were also adapted for this new scenario. ARLR-Random-QueryDoc corresponds to ARLR-Random: We start with the ARLR-selected sets and then randomly select q queries and m documents per query. In Random-QueryDoc, all instances are randomly selected, including the initial set. As before, we randomly select some queries and then some documents from each of the selected queries. Both random baselines were run 10 times on each Fold and the averages reported. Finally, we have SVM Full and ARLR-RLR. As before, SVM Full is the result reported by the LETOR producers.

Donmez. Donmez cannot be easily adapted to this new scenario. Running the regular Donmez on these data sets would produce initial data sets of more than 38% and 82% of the unlabeled sets for MQ2007 and MQ2008, respectively. Moreover, each QBC round would select an additional 12% and 26%. Adapting Donmez would require some query selection mechanism. We could randomly select q queries as with the random baselines described earlier; the problem is that Donmez estimates the impact that each document will have on the current model for each query separately. Thus, each query has a separate SVM model and the documents with the highest estimated impact on these models are selected. If we start with an initial set containing only a few queries, then we cannot proceed selecting documents from other queries, as there is no model for these other queries. Thus, we would only select documents from the same set of queries randomly chosen during the initial set selection. Reducing the number of documents selected per query would also not be very useful because we would still end up with very big initial sets and choose too many

[4] http://research.microsoft.com/en-us/um/beijing/projects/letor/letor4dataset.aspx

[5] http://ir.cis.udel.edu/million/index.html

DOI: 10.1002/asi

documents per round. Thus, we have not run Donmez on these data sets because it is not a competitive baseline in terms of the effectiveness versus cost trade-off.

Results for the LETOR 4.0 data sets. Figure 6 shows the results for ARLR-QBC-QueryDoc, ARLR-QBC-Query, and the baselines on MQ2007 (top) and MQ2008 (bottom). Notice that we show the NDCG@5 for MQ2008. The reason is that many queries in this data set have less than 10 documents, causing the NDCG@10 calculated by the evaluation script to be thrown off a bit. As shown in Figure 6, ARLR-QBC-QueryDoc does not beat SVM Full in the 26 rounds we ran on MQ2007, although both MAP and NDCG results steadily increase as more queries and documents are selected. On the other hand, in MQ2008, it is possible to achieve SVM Full-like performance on the eighth round (around 8% of the unlabeled set selected) and even beat it by the 19th round (14% selected). We can also see that ARLR-QBC-QueryDoc beats ARLR-QBC-Query in most rounds on both data sets. It seems that using both query- and document-level selection is a better strategy than choosing all documents from certain queries in these data sets, given their characteristics.

Paired t *test.* Paired Student's *t* tests show that on MQ2008, ARLR-QBC-QueryDoc is better than ARLR-QBC-Query, ARLR-Random-QueryDoc, and Random-QueryDoc with 99% confidence. It also beats ARLR-QBC-Query on MQ2007 with 99% confidence but is statistically tied to ARLR-Random-QueryDoc and Random-QueryDoc.

What is interesting in these results is the high MAP and NDCG values obtained by the random baselines: Random selection is as good as ARLR-QBC-QueryDoc on MQ2007, although significantly worse on MQ2008. Although on TD2003 the average random MAP is more than 22% lower than the SVM Full MAP and on TD2004 it is around 9% lower, on MQ2007 it is only 4.7% lower and on MQ2008 only 1.6% lower. This seems to be a reflection of the very different natures of these informational data sets: In LETOR 3.0, there are 1,000 documents per query, with a low proportion of relevant documents (1.5% on TD2003 and almost 3% on TD2004). In LETOR 4.0, there are only a few documents per query with a fraction of more than 20% (25% for MQ2007) of relevant documents (considering two relevance levels, 1 and 2). We can also see that the MAP and NDCG

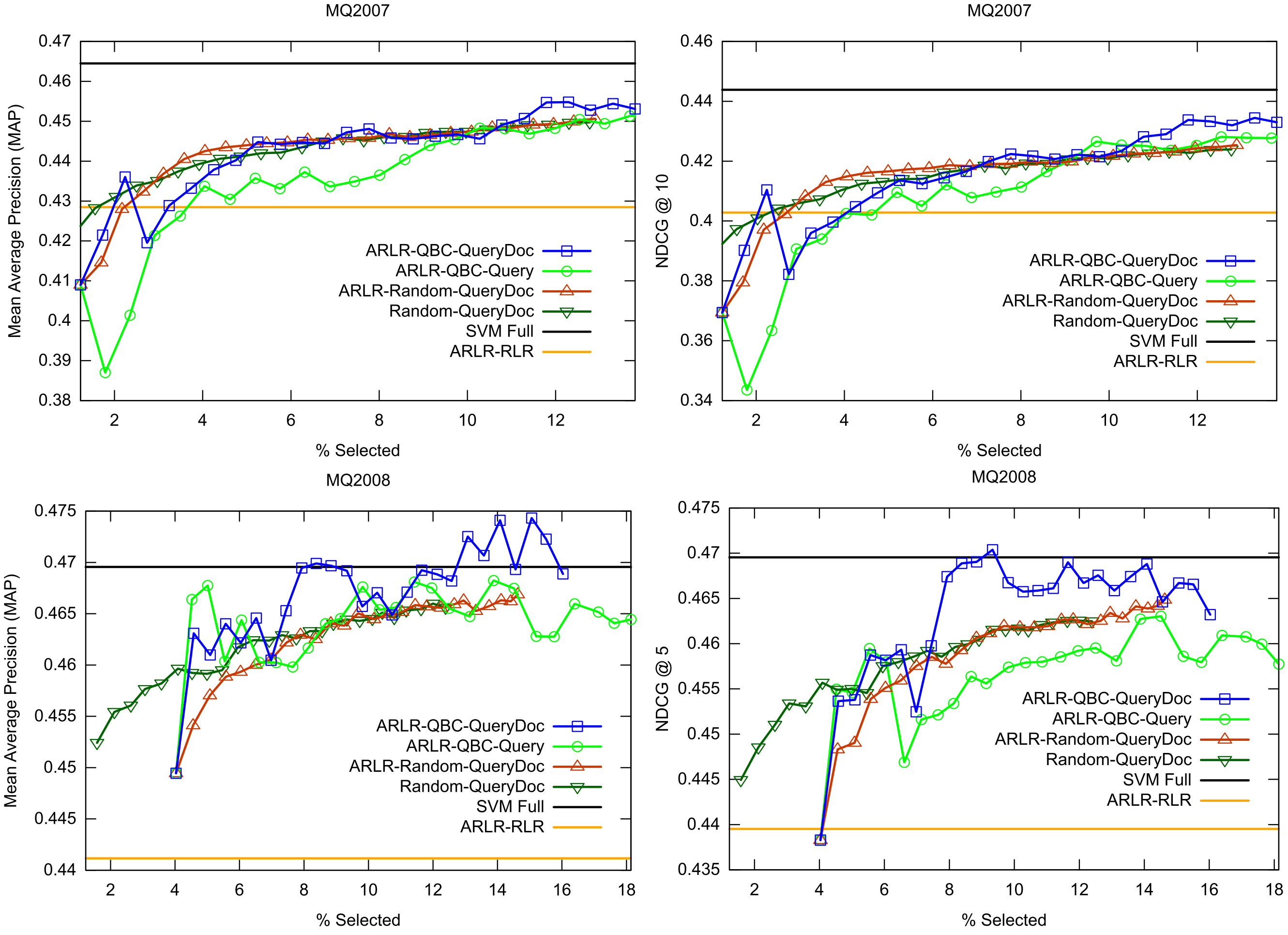

FIG. 6. MQ2007 and MQ2008: ARLR-QBC-QueryDoc and baselines comparison. MAP (left), NDCG@10 (MQ2007), and NDCG@5 (MQ2008) (right) versus percentage of samples selected from the unlabeled set. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

results for these data sets are almost double those obtained by the same (or similar) algorithms in the informational data sets of version 3.0 of the collection. Thus, the selection of only a few top documents per query has increased the overall quality of the documents in the 4.0 data sets, seemingly making it harder for an active method like ARLR-QBC to sieve out “noisy” instances, and thus achieve better results in the initial rounds (on MQ2008) and in general (on MQ2007). Nonetheless, ARLR-QBC-QueryDoc performs very well on MQ2008, showing that the combined query- and document-level QBC selection strategy can be very effective (top performance with a low percentage of the data set selected).

Conclusions

The proposed two-stage active learning technique provides an effective and practical method to reduce the cost of creating training sets for use with L2R techniques. The method is practical in the sense that it selects highly effective yet very small training sets from very diverse data sets and provides consistent quality results for small or bigger labeling budgets. Our method, selecting less than 6% of the unlabeled sets, achieves better MAP results than all 12 supervised baselines published by the LETOR 3.0 producers (using the full training sets) in three of the six data sets (TD2003 in round 0, HP2004 in round 6, and NP2004 in round 3). It still beats at least 9 of the 12 baselines in TD2004 and NP2003, and 5 in HP2003. Furthermore, it performs significantly better than a strong active learning baseline (Donmez) in five of the six data sets. We have also shown how a simple modification allows our method to be adapted for use with data sets with many queries (i.e., by adding query-level selection to the QBC stage). The results on the LETOR 4.0 data sets show that this variation can be very effective.

The strength of the proposed method relies on its ability to actively select documents without the need of an initial labeled set. As we have seen, the association rule active method (ARLR) is able to provide a very strong initial set, allowing the QBC second-stage process to expand the selection and obtain state-of-the-art results in just a few rounds on most data sets. We performed extensive experimentation that shows that the method is effective for very different data set types (i.e., data sets based on informational or navigational queries) and that it beats a very strong active learning method proposed in the literature.

In the future, we plan to measure the effect of the chosen committee algorithms by fine-tuning them or using other algorithms and to evaluate how the selected sets perform with other supervised methods such as RLR and RankBoost. Another interesting aspect to evaluate would be to change the measure of disagreement from the *CV* to the variance or standard deviation to determine how this metric affects the results.

References

Abe, N., & Mamitsuka, H. (1998). Query learning strategies using boosting and bagging. In Proceedings of the 15th International Conference on Machine Learning (pp. 1–9). Madison, WI: Morgan Kaufmann Publishers Inc.

Agrawal, R., Imieliński, T., & Swami, A. (1993, June). Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (pp. 207–216). Washington, DC: ACM Press.

Aslam, J.A., & Pavlu, V. (2007). A practical sampling strategy for efficient retrieval evaluation [working draft]. Retrieved from http://www.ccs.neu.edu/home/jaa/tmp/statAP.pdf

Baeza-Yates, R., & Ribeiro-Neto, B. (2011). Modern information retrieval: The concepts and technology behind search modern information retrieval: The concepts and technology behind search. Harlow, England: Addison Wesley.

Baum, E.B. (1991). Neural net algorithms that learn in polynomial time from examples and queries. IEEE Transactions on Neural Networks, 2(1), 5–19.

Cai, P., Gao, W., Zhou, A., & Wong, K. (2011). Relevant knowledge helps in choosing right teacher. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 115–124). Beijing, China: ACM Press.

Carterette, B., Allan, J., & Sitaraman, R. (2006). Minimal test collections for retrieval evaluation. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 268–275). Seattle, WA: ACM Press.

Dagan, I., & Engelson, S. (1995). Committee-based sampling for training probabilistic classifiers. In Proceedings of the 12th International Conference on Machine Learning (pp. 150–157). Tahoe City, CA: Morgan Kaufmann Publishers Inc.

Donmez, P., & Carbonell, J.G. (2008). Optimizing estimated loss reduction for active sampling in rank learning. In Proceedings of the 25th International Conference on Machine Learning (pp. 248–255). Helsinki, Finland: ACM Press.

Donmez, P., & Carbonell, J.G. (2009). Active sampling for rank learning via optimizing the area under the ROC curve. In Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval (pp. 78–89). Toulouse, France: Springer-Verlag.

Donmez, P., Carbonell, J.G., & Bennett, P.N. (2007). Dual strategy active learning. In Proceedings of the 18th European Conference on Machine Learning (pp. 116–127). Berlin, Heidelberg: Springer-Verlag.

Freund, Y., Iyer, R., Schapire, R.E., & Singer, Y. (2003, December). An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research, 4, 933–969.

Geng, X., Qin, T., Liu, T., Cheng, X., & Li, H. (2011, September). Selecting optimal training data for learning to rank. Information Processing and Management, 47(5), 730–741.

Granka, L.A., Joachims, T., & Gay, G. (2004). Eye-tracking analysis of user behavior in WWW search. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 478–479). New York: ACM Press.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 133–142). San Jose, CA: ACM Press.

Lewis, D.D., & Gale, W.A. (1994). A sequential algorithm for training text classifiers. In Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 3–12). Dublin, Ireland: Springer-Verlag.

Liu, T. (2009). Learning to rank for information retrieval. Foundations and Trends in Information Retrieval, 3(3), 225–331.

Long, B., Chapelle, O., Zhang, Y., Chang, Y., Zheng, Z., & Tseng, B. (2010). Active learning for ranking through expected loss optimization. In Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 267–274). Geneva, Switzerland: ACM Press.

Mccallum, A.K. (1998). Employing EM in pool-based active learning for text classification. In Proceedings of the 15th International Conference on Machine Learning (pp. 350–358). Madison, WI: Morgan Kaufmann Publishers Inc.

Nguyen, H.T., & Smeulders, A. (2004). Active learning using pre-clustering. In Proceedings of the 21st International Conference on Machine Learning (pp. 623–630). Banff, AB, Canada: ACM Press.

Qin, T., Liu, T., Xu, J., & Li, H. (2010, August). LETOR: A benchmark collection for research on learning to rank for information retrieval. Information Retrieval, 13, 346–374.

Radlinski, F., & Joachims, T. (2007). Active exploration for learning rankings from clickthrough data. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 570–579). San Jose, CA: ACM Press.

Robertson, S.E., Walker, S., & Hancock-Beaulieu, M.M. (1995, May). Large test collection experiments on an operational, interactive system: Okapi at TREC. Information Processing & Management, 31(3), 345–360.

Schapire, R.E. (1989). The strength of weak learnability. In Proceedings of the 30th Annual Symposium on Foundations of Computer Science (pp. 28–33). Washington, DC: IEEE Computer Society.

Schmidberger, G., & Frank, E. (2005). Unsupervised discretization using tree-based density estimation. In Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (pp. 240–251). Berlin, Heidelberg: Springer-Verlag.

Schohn, G., & Cohn, D. (2000). Less is more: Active learning with support vector machines. In Proceedings of the 17th International Conference on Machine Learning (pp. 839–846). San Francisco: Morgan Kaufmann Publishers Inc.

Settles, B. (2009). Active learning literature survey (Computer Sciences Technical Report 1648). Madison, WI: University of Wisconsin-Madison.

Settles, B., Craven, M., & Ray, S. (2008). Multiple-instance active learning. In Advances in neural information processing systems (Vol. 20, pp. 1289–1296). Cambridge, MA: MIT Press.

Seung, H.S., Opper, M., & Sompolinsky, H. (1992). Query by committee. In Proceedings of the 5th Annual Workshop on Computational Learning Theory (pp. 287–294). Pittsburgh, PA: ACM Press.

Silva, R., Gonçalves, M.A., & Veloso, A. (2011). Rule-based active sampling for learning to rank. In Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 240–255). Athens, Greece: Springer-Verlag.

Steck, H. (2007). Hinge rank loss and the area under the ROC curve. In Proceedings of the 18th European Conference on Machine Learning (pp. 347–358). Warsaw, Poland: Springer-Verlag.

Tian, A., & Lease, M. (2011). Active learning to maximize accuracy vs. effort in interactive information retrieval. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 145–154). Beijing, China: ACM Press.

Tong, S., & Koller, D. (2002, March). Support vector machine active learning with applications to text classification. Journal of Machine Learning Research, 2, 45–66.

Veloso, A.A., Almeida, H.M., Gonçalves, M.A., & Meira, W. Jr. (2008). Learning to rank at query-time using association rules. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 267–274). Singapore: ACM Press.

Yu, H. (2005). SVM selective sampling for ranking with application to data retrieval. In Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining (pp. 354–363). Chicago: ACM Press.

DOI: 10.1002/asi