Efficient Structured Learning for Personalized Diversification

Shangsong Liang, Fei Cai, Zhaochun Ren, and Maarten de Rijke

Abstract—This paper is concerned with the problem of personalized diversification of search results, with the goal of enhancing the performance of both plain diversification and plain personalization algorithms. In previous work, the problem has mainly been tackled by means of unsupervised learning. To further enhance the performance, we propose a supervised learning strategy. Specifically, we set up a structured learning framework for conducting supervised personalized diversification, in which we add features extracted directly from tokens of documents and those utilized by unsupervised personalized diversification algorithms, and, importantly, those generated from our proposed user-interest latent Dirichlet topic model. We also define two constraints in our structured learning framework to ensure that search results are both diversified and consistent with a user's interest. To further boost the efficiency of training, we propose a fast training framework for our proposed method by adding additional multiple highly violated but also diversified constraints at every training iteration of the cutting-plane algorithm. We conduct experiments on an open dataset and find that our supervised learning strategy outperforms unsupervised personalized diversification methods as well as other plain personalization and plain diversification methods. Our fast training framework significantly saves training time while it maintains almost the same performance.

Index Terms—Personalization, diversity, structured SVMs, ad hoc retrieval

1 Introduction

Search result diversification has gained attention as a method to tackle query ambiguity. In search result diversification one typically considers the relevance of a document in light of the other retrieved documents. The goal is to identify the probable "aspects" of the ambiguous query, retrieve documents for each of these aspects and make the search results more diverse [2]. By doing so, in the absence of any knowledge of users' context or preferences, the chance that users who issue an ambiguous query will find at least one of these results to be relevant to their underlying information need is maximized [3].

In both search result diversification and personalized web search, an issued query is often viewed as an incomplete expression of a user's underlying need [4]. Unlike search result diversification, where the system accepts and adapts its behavior to a situation of uncertainty, personalized web search strives to address this situation by enhancing the system's knowledge about users' information needs. Rather than aiming to satisfy as many users as possible, personalization aims to build a sense of who the user is, and maximize the satisfaction of a specific user [5].

Although different, diversification and personalization are not incompatible and do not have mutually exclusive goals [6]. Search results generated by diversification techniques should be more diverse when a user's preferences are unrelated to the query. Likewise, personalization can improve the effectiveness of aspect weighting in diversification, by favoring query interpretations that are predicted to be more related to each specific user [5].

We study the problem of *personalized diversification of search results*, with the goal of enhancing both diversification and personalization performances. As an example, consider a case where a query has three aspects, I, II and III, and a user is only interested in aspects I and III. Diversification algorithms may return a top-k of documents that covers all aspects, including aspect II that the user is not interested in. On the other hand, personalized algorithms may retrieve a top-k of documents such that the first m ($m < k$) documents only covering aspect I but not III that the user is also interested in, while the remaining $k - m$ documents cover other aspects that are not covered by the first m documents. In contrast, personalized diversification algorithms try to retrieve a top-k of documents that covers topics I and III, one by one. (See Section 7.7 for a real example). The problem has previously been investigated by Radlinski and Dumais [7] and Vallet and Castells [5]. They have present a number of effective *unsupervised learning* approaches that combine both personalization and diversification components to tackle the problem. To further improve the performance we propose a *supervised learning* approach.

Accordingly, we formulate the task of personalized search result diversification as a problem of predicting a diverse set of documents given a specific user and query. We formulate a discriminant based on maximizing search

- *S. Liang is with the Department of Computer Science, University College London, London WC1E 6BT, United Kingdom. E-mail: shangsong.liang@ucl.ac.uk.*
- *F. Cai is with the Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, China, and the Informatics Institute, University of Amsterdam, Amsterdam 1012, WX, The Netherlands. E-mail: f.cai@uva.nl.*
- *Z. Ren and M. de Rijke are with Informatics Institute, University of Amsterdam, Amsterdam 1012, WX, The Netherlands. E-mail: {z.ren, derijke}@uva.nl.*

Manuscript received 5 Apr. 2015; revised 24 Mar. 2016; accepted 11 July 2016. Date of publication 26 July 2016; date of current version 3 Oct. 2016. Recommended for acceptance by H. Zha.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2016.2594064

result diversification, and perform training using the well-known structured support vector machines (SSVMs) framework [8]. The main idea is to use a user-interest LDA-style [9, Latent Dirichlet Allocation] topic model, from which we can infer a per-document multinomial distribution over topics and determine whether a document can cater for a specific user.

Like most previous learning algorithms based on SSVMs, our framework for personalized search result diversification is formulated as a constrained quadratic program with many constraints, which can be solved by applying the cutting-plane approach [8]. However, a large number of parameters needs to be tuned during the training, at every iteration of the cutting-plane procedure inference must be performed on the entire training dataset, and a large quadratic program must be solved, as a result of which training our proposed model is time-consuming. To speed up convergence in our training procedure, we propose to utilize multiple constraints that are not only informative with regards to the current approximation, i.e., they are highly violated, but also have marginal relevance with regards to the constraints added at the current iteration, i.e., they are diversified.

Then, during training we use features extracted directly from the tokens' statistical information in the documents and those utilized by unsupervised personalized diversification algorithms, and, more importantly, those generated from our proposed topic model. Two types of constraint in SSVMs are explicitly defined to enforce the search results to be both diverse and relevant to a user's personal interest.

We evaluate our approach on a publicly available personalized diversification dataset and compare it (1) to unsupervised approaches that focus on either personalization or diversification alone, (2) to combined approaches like those in [5], [7] and (3) to two standard structured learning approaches [10], [11]. We also evaluate our proposed method in terms of training time and number of iterations. The four main contributions of our work are:

(i) We tackle the problem of personalized diversification of search results in a new way, using a supervised learning method.
(ii) We propose a user-interest latent topic model to capture a user's interest and infer per-document multinomial distributions over topics.
(iii) We explicitly enforce diversity and personalization through two types of constraint in structured learning for personalized diversification.
(iv) We boost the efficiency of the training stage of our approach by adding not only highly violated but also diversified constraints during training iterations.

2 Related Work

2.1 Personalized Search Result Diversification

Two main components, viz., personalized web search and search result diversification, play important roles in tackling the problem of personalized search result diversification. The task of *personalized* web search aims at identifying the most relevant search results for an individual by leveraging their information. Many personalized web search methods have been proposed, such as the one based on social tagging profiles [12], ranking model adaption for personalized search [13], search personalization by modeling the impact of users' behavior [14], and personalized search using interaction behaviors in search sessions [15]. Interestingly, after analyzing large-scale query logs, Dou et al. [16], [17] reveal that personalized search yields significant improvements over common web search on some queries but has little effect on others; they propose click entropy to measure whether users have adverse information needs by issuing a query and features to automatically predict whether the results should be personalized.

In contrast, *diversification* aims to make the search results diversified given an ambiguous query so that users can find at least one of these results to be relevant to their underlying information need [18]. Well-known diversification methods include the maximal marginal relevance model [19], probabilistic model [20], subtopic retrieval model [21], xQuAD [22], RxQuAD [23], IA-select [18], PM-2 [24], and more recently, DSPApprox [2], text-based measures [25], term-level [2], and fusion-based [26], [27]. All of the above methods focus on either personalization or diversification only.

Only Radlinski and Dumais [7] and Vallet and Castells [5] have studied the problem of combining both personalization and diversification. Radlinski and Dumais [7] analyze a large sample of individual users' query logs from a web search engine such that individual users' query reformulations can be obtained. Then they personalize web search by reranking some top results using query reformulations to introduce diversity into those results. Their evaluation suggests that using diversification is a promising method to improve personalized reranking of search results. Vallet and Castells [5] present approaches that combine personalization and diversification components. They investigate the introduction of the user as an explicit variable in state-of-the-art diversification models. Their algorithms achieve competitive performance and improve over plain personalization and diversification baselines.

All of the previous personalized diversification models are unsupervised. However, we argue that to enhance the performance, it is better to employ a supervised learning approach, and our experiments show that supervised learning can indeed improve the performance of unsupervised approaches. To the best of our knowledge, this is the first attempt to tackle the problem of personalized diversification using supervised learning methods.

2.2 Structured Learning

Structured learning has provided principled techniques for learning structured-output models, with the structured support vector machines being one of the most important ones [8]. In structured learning, a set of training pairs, $\{(\mathbf{x}, \mathbf{y}) \,|\, \mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}\}$, is assumed to be available to the learning algorithm, and the goal is to learn a mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$ from the input space $\mathcal{X}$ to the output space $\mathcal{Y}$, such that a regularized task-dependent loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+$ can be minimized, where $\Delta(\mathbf{y}, \bar{\mathbf{y}})$ denotes the cost of predicting output $\bar{\mathbf{y}}$ when the correct prediction is $\mathbf{y}$. During the past few years, Structured SVMs have been studied and applied in many areas, such as summarization [28], speech recognition [29], optimizing average precision of a ranking [11], and diversification [10]. For us, the most interesting prior application of SSVMs is the one for

predicting diverse subsets [10]. However, our personalized search result diversification method differs from that proposed in [10]: we work on personalized diversification where we propose a user-interest LDA-style model to capture a user's interest distribution over topics, whereas the authors of [10] directly apply existing SSVMs algorithm to tackle the problem of search result diversification but not personalized diversification; our model makes results diverse and consistent with the user's interest by enforcing both diversity and interest constraints, whereas the model in [10] only directly diversifies the results by adopting standard SSVMs. The work in [28] is of interesting to us too, as the authors also propose to utilize additional constraints in SSVMs for their purpose: enhancing diversity, coverage and balance for summarization. Compared to their model, again, ours targets a different task, works with different types of constraints, and moreover, utilizes a user topic model to infer users' personal interests for personalization diversification and try to boost the training efficiency by generating additional constraints. To the best of our knowledge, ours is the first attempt to enforce diversity and personalization through additional constraints in SSVMs.

Additionally, most previous SSVMs based algorithms, including those aforementioned, utilize the well-known cutting-plane technique for training, which involves many constraints (in total $|\mathcal{Y}| \times (|\mathcal{Y}| - 1)$ constraints). Unfortunately, those algorithms also contain many parameters that need to be tuned, require a large number of iterations due to the large number of constraints, and at each iteration the cutting-plane that they apply needs to solve a quadratic program performing on the entire training dataset. Because of this, training SSVMs based algorithms is time-consuming and applying such algorithms to some real-time applications, e.g., those that need to rapidly update the model in data streams, becomes unpractical. Guzman-Rivera et al. [30] and Branson et al. [31] have proposed methods to boost the efficiency of SSVMs cutting-plane training. Their formulation for finding additional solutions in [30] cannot be used directly in our setting, as their formulation aims at maximizing a discrete Markov random field based function, whereas we need to find diversified solutions, all of which should maximize both relevance and diversification of the returned documents in our formulation. We cannot directly apply the method in [31] either, as, again, the formulation for finding additional solutions is not directly applicable for our personalized diversification task. Accordingly, we propose a formulation to generate additional constraints that are not only informative with regards to the current approximation, i.e., should be highly violated, but also diversified, i.e., the potential solutions involved in the set of constraints should be different from each other to some extent. As far as we are aware, this is the first proposal to boost the efficiency of training SSVM based personalized diversification.

2.3 Topic Modeling

Topic modeling provides a suite of algorithms to discover hidden thematic structure in a collection of documents. A topic model takes a collection of documents as input, and discovers a set of "latent topics"—recurring themes that are discussed in the collection—and the degree to which each document exhibits those topics [9]. *Latent dirichlet allocation* (LDA) [9] is one of the simplest topic models, and it decomposes a collection of documents into topics—biased probability distributions over terms—and represents each document with a subset of these topics. Many LDA-style models have been proposed, such as the syntactic topic model [32], multilingual topic model [33], topic over time model [34], and more recently, the max-margin model [35], hierarchical sentiment-LDA model [36], fusion-based model [26], [27], [37], user clustering topic model [38] and dynamic clustering topic model [39]. We propose a user-interest LDA-style model to capture a multinomial distribution of topics specific to a user. From our model, we infer a per-document multinomial distribution over topics so that we can easily identify whether a document caters to a user's interest. Accordingly, we use the output of our user-interest LDA-style model, i.e., multinomial distributions over topics, as one of the three types of features (see Section 5.2) to tune a weight vector $\mathbf{w}$ in training. The advantages of preprocessing the users' distributions rather than integrating the topic model into the SSVMs are that the system can update users' interest distributions offline, while making its predictions online, so that the response time given a user and a query can be significantly shortened. A unified model that integrates user-interest LDA-style model and SSVMs would be possible; we leave it as future work to explore this alternative. Our experimental results show that the model can help to enhance the performance of personalized search result diversification. To the best of our knowledge, this is the first time that a topic model is utilized to enhance the performance of personalized diversification.

3 The Learning Problem

Let $\mathbf{u} = \{d_1, \ldots, d_{|\mathbf{u}|}\} \in \mathcal{U}$ be a set of documents of size $|\mathbf{u}|$ that a user u is interested in. For each query q, we assume that we are given $\mathbf{u}$ and a set of candidate documents $\mathbf{x} = \{x_1, \ldots, x_{|\mathbf{x}|}\} \in \mathcal{X}$, where $\mathcal{X}$ denotes the set of all possible document sets. Our task is to select a subset $\mathbf{y} \in \mathcal{Y}$ of K documents from $\mathbf{x}$ that maximizes the performance of personalized search result diversification given q and $\mathbf{u}$, where we let $\mathcal{Y}$ denote the space of predicted subsets $\mathbf{y}$. Following the standard machine learning setup, we formulate our task as learning a hypothesis function $h : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{Y}$ to predict a $\mathbf{y}$ given $\mathbf{x}$ and $\mathbf{u}$. To this end, we assume that a set of labeled training data is available

$$\{(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}^{(i)}) \in \mathcal{X} \times \mathcal{U} \times \mathcal{Y} : i = 1, \ldots, N\},$$

where $\mathbf{y}^{(i)}$ is the ground-truth subset of K documents from $\mathbf{x}^{(i)}$, and $\mathbf{u}^{(i)}$ is the set of documents that user u_i is interested in, and N is the size of the training data. We aim to find a function h such that the empirical risk

$$R_S^{\Delta}(h) = \frac{1}{N} \sum_{i=1}^{N} \Delta\Big(\mathbf{y}^{(i)}, h(\mathbf{x}^{(i)}, \mathbf{u}^{(i)})\Big),$$

can be minimized, where we quantify the quality of a prediction by considering a loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+$ that measures the penalty of choosing $\bar{\mathbf{y}} = h(\mathbf{x}^{(i)}, \mathbf{u}^{(i)})$. Given the ground-truth $\mathbf{y}$, viz., the ground truth ranking of relevant documents, and the prediction $\bar{\mathbf{y}}$, viz., the ranking of

predicted documents, we define the loss function by

$$\Delta(\mathbf{y}, \bar{\mathbf{y}}) \equiv 1 - E(\mathbf{y}, \bar{\mathbf{y}}), \tag{1}$$

where $E(\mathbf{y}, \bar{\mathbf{y}})$ is an evaluation metric that computes the evaluation score for the predicted ranking $\bar{\mathbf{y}}$ given the ground-truth ranking $\mathbf{y}$. We focus on hypothesis functions that are parameterized by a weight vector $\mathbf{w}$, and thus wish to find $\mathbf{w}$ to minimize the risk, $R_S^{\Delta}(\mathbf{w}) \equiv R_S^{\Delta}(h(\cdot; \mathbf{w}))$. We let a discriminant $\mathcal{F} : \mathcal{X} \times \mathcal{U} \times \mathcal{Y} \rightarrow \mathbb{R}^+$ compute how well the prediction $\bar{\mathbf{y}}$ fits for $\mathbf{x}$ and $\mathbf{u}$. Then the hypothesis predicts the $\bar{\mathbf{y}}$ that maximizes $\mathcal{F}$

$$\bar{\mathbf{y}} = h(\mathbf{x}, \mathbf{u}; \mathbf{w}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} \mathcal{F}(\mathbf{x}, \mathbf{u}, \mathbf{y}). \tag{2}$$

We describe each $(\mathbf{x}, \mathbf{u}, \mathbf{y})$ through a feature vector $\Psi(\mathbf{x}, \mathbf{u}, \mathbf{y})$; the extraction will be discussed later. The discriminant function $\mathcal{F}(\mathbf{x}, \mathbf{u}, \mathbf{y})$ is assumed to be linear in the feature vector $\Psi(\mathbf{x}, \mathbf{u}, \mathbf{y})$ such that

$$\mathcal{F}(\mathbf{x}, \mathbf{u}, \mathbf{y}) = \mathbf{w}^T \Psi(\mathbf{x}, \mathbf{u}, \mathbf{y}), \tag{3}$$

where $\mathbf{w}$ is a weight vector to be learned from training data and with which $\mathbf{y}$ can be predicted, given $\mathbf{u}$ and $\mathbf{x}$.

4 Structured Learning for Personalized Diversification

In this section, we introduce the standard SSVMs learning problem, propose constraints for our personalized diversification model, describe our optimization problem, the way we make predictions, and how we boost the efficiency of training our proposed algorithm.

4.1 Standard Structured SVMs

Our personalized diversification model builds on a structured learning framework. In our setting, the structured learning framework can be described as: given a training set $\{(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}^{(i)}) \in \mathcal{X} \times \mathcal{U} \times \mathcal{Y} : i = 1, \ldots, N\}$, structured SVMs are employed to learn a weight vector $\mathbf{w}$ for the discriminant function $\mathcal{F}(\mathbf{x}, \mathbf{u}, \mathbf{y})$ through the following quadratic programming problem.

Optimization Problem 1. (Standard structured SVMs)

$$\min_{\mathbf{w}, \xi \geq 0} \frac{1}{2} ||\mathbf{w}||^2 + \frac{C}{N} \sum_{i=1}^{N} \xi_i, \tag{4}$$

subject to $\forall i, \forall \mathbf{y} \in \mathcal{Y} \backslash \mathbf{y}^{(i)}, \xi_i \geq 0$,

$$\mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}^{(i)}) \geq \mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}) + \Delta(\mathbf{y}^{(i)}, \mathbf{y}) - \xi_i.$$

In the objective function (4), the parameter C is a tradeoff between model complexity, $||\mathbf{w}||^2$, and a hinge loss relaxation of the training loss for each training example, $\sum \xi_i$. These standard constraints enforce the requirement that given $\mathbf{x}^{(i)}$ and $\mathbf{u}^{(i)}$ the ground-truth personalized diversity document set $\mathbf{y}^{(i)}$ should have a greater value $\mathcal{F}(\mathbf{x}, \mathbf{u}, \mathbf{y})$ than alternative $\mathbf{y} \in \mathcal{Y}$.

4.2 Additional Constraints

As discussed above, we aim at training a personalized diversification model that can enforce both diversity and consistency with the user's interest. This can be achieved by introducing additional constraints to the structured SVM optimization problem defined in (4).

To start, diversity requires a set of retrieved documents that should not discuss the same aspects of an ambiguous query. In other words, aspects of documents returned by a diversification model should have little overlap with one another. Formally, we enforce diversity with the following constraint.

Constraint for Diversity

$$\mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}^{(i)}) \geq \sum_{y \in \mathbf{y}^{(i)}} \mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, y) - \xi_i. \tag{5}$$

In (5), the sum of each document's score, $\sum \mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, y)$, should not be greater than the overall score when documents in $\mathbf{y}^{(i)}$ are considered as an ideal ranking of the document sets. According to this constraint, commonly shared features are associated with relatively low weights, and a document set with less redundancy will be predicted as output given inputs $\mathbf{x}$ and $\mathbf{u}$.

Additionally, personalization requires a set of returned documents to match the user's personal interest. Formally, we enforce personalization with the following constraint.

Constraint for Consistency with User's Interest

$$\begin{aligned} \mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}^{(i)}) \geq\ & \mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}) + \\ & (1 - \mathrm{sim}(\mathbf{y}, \mathbf{u}^{(i)})) - \mu - \xi_i, \end{aligned} \tag{6}$$

where $\mathrm{sim}(\mathbf{y}, \mathbf{u}^{(i)}) \in [0, 1]$ is a function (see (15)) that measures subtopic distribution similarity between a set of documents $\mathbf{y}$ and the documents user u_i is interested in, i.e., $\mathbf{u}^{(i)}$, μ is a slack variable that tends to give slightly better performance, which can be defined as $\mu = \frac{1}{N} \sum_{i=1}^{N} (1 - \mathrm{sim}\,(\mathbf{y}^{(i)}, \mathbf{u}^{(i)}))$.

In (6), $(1 - \mathrm{sim}(\mathbf{y}, \mathbf{u}^{(i)}))$ quantifies how well a set of documents matches a user's interest. If the topics discussed in a set of documents $\mathbf{y}$ are not consistent with a user's personal interest, $\mathbf{w}^T \Psi(\mathbf{x}, \mathbf{u}, \mathbf{y})$ will result in a low score. During prediction, documents consistent with a user's interest will be preferred.

4.3 Our Optimization Problem

A set of documents produced in response to an ambiguous query should be diverse and consistent with the user's personal interest. To this end we integrate the proposed additional constraints with standard structured SVMs. We propose to train a personalized diversification model by tackling the following optimization problem:

Optimization Problem 2. (Structured SVMs for personalized diversification)

$$\min_{\mathbf{w}, \xi \geq 0} \frac{1}{2} ||\mathbf{w}||^2 + \frac{C}{N} \sum_{i=1}^{N} \xi_i, \tag{7}$$

subject to $\forall i, \forall \mathbf{y} \in \mathcal{Y} \backslash \mathbf{y}^{(i)}, \xi_i \geq 0$,

1) $\mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}^{(i)}) \geq \mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}) + \Delta(\mathbf{y}^{(i)}, \mathbf{y}) - \xi_i$,
2) $\mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}^{(i)}) \geq \sum_{y \in \mathbf{y}^{(i)}} \mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, y) - \xi_i$,
3) $\mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}^{(i)}) \geq \mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y})$
$+ ((1 - \mathrm{sim}(\mathbf{y}, \mathbf{u}^{(i)})) - \mu) - \xi_i$.

4.4 The Learning Algorithm

We can solve the optimization problem defined in (7) by employing the cutting plane algorithm [8]. The learning algorithm is shown in Algorithm 1. The algorithm iteratively adds constraints until we have solved the original problem within a desired tolerance ϵ. It starts with empty working sets $\mathcal{W}_i$, $\mathcal{W}'_i$ and $\mathcal{W}''_i$, for $i = 1, \ldots, N$. Then it iteratively finds the most violated constraints $\bar{\mathbf{y}}$, $\mathbf{y}'$ and $\bar{\mathbf{y}}''$ for each $(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}^{(i)})$ in terms of the three constraints (i), (ii) and (iii) in (7). If they are violated by more than ϵ, we add them into the corresponding working sets. We iteratively update $\mathbf{w}$ by optimizing (7) over the updated working sets. The outer loop in Algorithm 1 can halt within a polynomial number of iterations for any desired precision ϵ [8].

Algorithm 1. Cutting Plane Algorithm

Input: $(\mathbf{x}^{(1)}, \mathbf{u}^{(1)}, \mathbf{y}^{(1)}), \ldots, (\mathbf{x}^{(N)}, \mathbf{u}^{(N)}, \mathbf{y}^{(N)}), C, \epsilon$
$\mathcal{W}_i \leftarrow \emptyset, \mathcal{W}'_i \leftarrow \emptyset, \mathcal{W}''_i \leftarrow \emptyset$ for all $i = 1, \ldots, N$
$\mu = \frac{1}{N}\sum_{i=1}^{N}(1 - \mathrm{sim}(\mathbf{y}^{(i)}, \mathbf{u}^{(i)}))$
repeat
  for $i = 1, \ldots, N$ **do**
    $H(\mathbf{y}; \mathbf{w}) \equiv \Delta(\mathbf{y}^{(i)}, \mathbf{y}) + \mathbf{w}^T\Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}) - \mathbf{w}^T\Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}^{(i)})$
    $H'(\mathbf{y}; \mathbf{w}) \equiv \sum_{y \in \mathbf{y}^{(i)}} \mathbf{w}^T\Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, y) - \mathbf{w}^T\Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}^{(i)})$
    $H''(\mathbf{y}; \mathbf{w}) \equiv \mathbf{w}^T\Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}) +$
    $((1 - \mathrm{sim}(\mathbf{y}, \mathbf{u}^{(i)})) - \mu) - \mathbf{w}^T\Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}^{(i)})$
    compute $\bar{\mathbf{y}} = \mathrm{argmax}_{\mathbf{y}} H(\mathbf{y}; \mathbf{w})$, $\bar{\mathbf{y}}' = \mathrm{argmax}_{\mathbf{y}} H'(\mathbf{y}; \mathbf{w})$ and $\bar{\mathbf{y}}'' = \mathrm{argmax}_{\mathbf{y}} H''(\mathbf{y}; \mathbf{w})$
    compute $\xi_i = \max\{0, \max_{\mathbf{y} \in \mathcal{W}_i} H(\mathbf{y}; \mathbf{w}), \max_{\mathbf{y} \in \mathcal{W}'_i} H'(\mathbf{y}; \mathbf{w}), \max_{\mathbf{y} \in \mathcal{W}''_i} H''(\mathbf{y}; \mathbf{w})\}$
    if $H(\bar{\mathbf{y}}; \mathbf{w}) > \xi_i + \epsilon$ **or** $H''(\bar{\mathbf{y}}'; \mathbf{w}) > \xi_i + \epsilon$ **or** $H''(\bar{\mathbf{y}}''; \mathbf{w}) > \xi_i + \epsilon$ **then**
      Add constraint to working set $\mathcal{W}_i \leftarrow \mathcal{W}_i \cup \{\bar{\mathbf{y}}\}$, $\mathcal{W}'_i \leftarrow \mathcal{W}'_i \cup \{\mathbf{y}'\}$, $\mathcal{W}''_i \leftarrow \mathcal{W}''_i \cup \{\mathbf{y}''\}$
      $\mathbf{w} \leftarrow$ optimize (7) over $\bigcup_i \{\mathcal{W}_i, \mathcal{W}'_i, \mathcal{W}''_i\}$
until *no* $\mathcal{W}_i$, $\mathcal{W}'_i$ *and* $\mathcal{W}''_i$ *have changed during iteration*

4.5 Prediction

After $\mathbf{w}$ has been learned, given an ambiguous query, a set of candidate documents $\mathbf{x}$, and a set of documents $\mathbf{u}$ the user u is interested in, we try to predict a set of documents $\bar{\mathbf{y}}$ by tackling the following prediction problem:

$$\bar{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}} \mathcal{F}(\mathbf{x}, \mathbf{u}, \mathbf{y}) = \mathbf{w}^T\Psi(\mathbf{x}, \mathbf{u}, \mathbf{y}). \tag{8}$$

This is a special case of the Budgeted Max Coverage problem, and can be efficiently solved by Algorithm 2. Recall that the Budgeted Max Coverage problem [40] is defined as follows. A collection of sets $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ with associated costs $\{c_i\}_{i=1}^m$ is defined over a domain of elements $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$, with associated weights $\{w_i\}_{i=1}^n$. The goal is to find a collection of sets $\mathcal{S}' \subseteq \mathcal{S}$, such that the total cost of elements in $\mathcal{S}'$ does not exceed a given budget L while the sum of the weights of elements covered by $\mathcal{S}'$ is maximized. Following the definition of the Budgeted Max Coverage problem, we can define our personalized diversification prediction problem as: A collection of documents $\mathbf{x} = \{d_1, d_2, \ldots, d_m\}$ with uniform associated costs $\{c_i = 1\}_{i=1}^m$ of being selected in response to a query and the user's interest in query aspects $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ are represented with aspect weights $\{w_i\}_{i=1}^n$. The goal is to find a collection of documents $\mathbf{y} \subseteq \mathbf{x}$, such that the total cost (number) of documents in $\mathbf{y}$ does not exceed a given number K, and the total weight of different aspects the user is interested in covered by $\mathbf{y}$ is maximized. Thus our problem is a special case of the Budgeted Max Coverage problem.

Algorithm 2. Greedy Subset Selection for Prediction

Input: $\mathbf{w}$, $\mathbf{x}$, $\mathbf{u}$
$\bar{\mathbf{y}} \leftarrow \emptyset$
for $k = 1, \ldots, K$ **do**
  $\bar{x} = \arg\max_{x: x \in \mathbf{x}, x \notin \bar{\mathbf{y}}} \mathbf{w}^T\Psi(\mathbf{x}, \mathbf{u}, \bar{\mathbf{y}} \cup \{x\})$
  $\bar{\mathbf{y}} \leftarrow \bar{\mathbf{y}} \cup \{\bar{x}\}$
return $\bar{\mathbf{y}}$

4.6 Fast Training

Utilizing Algorithm 1 to train our personalized diversification model is time-consuming, as there is a large number of parameters that need to be tuned in $\mathbf{w}$ and many iterations of the training procedure in Algorithm 1 are required where at each iteration inference must be performed on the entire training dataset and a large quadratic program must be solved [8]. In cases where we need to apply structured learning to update the model online, e.g., for online structured prediction [41], such slow training and updating are bottlenecks.

To boost the efficiency of training our personalized diversification algorithm, we propose a fast training cutting plane algorithm, which is shown in Algorithm 3. Algorithm 3 is almost the same as the original cutting plane algorithm presented in Algorithm 1 except that in the current iteration it not only adds the most violated constraints $\bar{\mathbf{y}}$, $\mathbf{y}'$ and $\bar{\mathbf{y}}''$ for each $(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}^{(i)})$ but also constraints $\bar{\mathbf{y}}'''_1, \ldots, \bar{\mathbf{y}}'''_M$ that are potentially violated in later iterations (see lines 13-15 in Algorithm 3). In other words, Algorithm 1 is a special case of Algorithm 3 when no additional constraints are added excepted the standard ones. Here, M is the number of additional constraints. According to [31], [30], to speed up convergence in our fast cutting plane training procedure, these M additional constraints should not only be informative, i.e., highly violated (as required by the standard cutting-plane approach), but also diversified, i.e., these constraints should be different from each other. Therefore, for the ground truth $(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}^{(i)})$ in Algorithm 3, besides $\bar{\mathbf{y}}, \bar{\mathbf{y}}'$ and $\bar{\mathbf{y}}''$ we propose to find these additional constraints $\bar{\mathbf{y}}'''_1, \ldots, \bar{\mathbf{y}}'''_j, \ldots, \bar{\mathbf{y}}'''_M$ by

$$\bar{\mathbf{y}}'''_j = \arg \max_{\mathbf{y} \in \mathcal{Y} \setminus \mathcal{W}_i} \mathcal{A}(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}; \mathbf{y}), \tag{9}$$

subject to

$$\min_{\mathbf{y}' \in \mathcal{W}_i} \Delta(\mathbf{y}', \mathbf{y}) \geq \min_{\mathbf{y}'' \in \widetilde{\mathcal{W}}_i,\, \mathbf{y}''' \in \widetilde{\mathcal{W}}_i \setminus \{\mathbf{y}''\}} \Delta(\mathbf{y}'', \mathbf{y}''').$$

Here, $\mathcal{A}(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}; \mathbf{y})$ is an unsupervised personalized diversification algorithm that retrieves $\mathbf{y}$ given $\mathbf{x}^{(i)}$ and $\mathbf{u}^{(i)}$; $\widetilde{\mathcal{W}}_i$ is the working set that contains existing constraints, i.e., $\widetilde{\mathcal{W}}_i \equiv \mathcal{W}_i \cup \mathcal{W}'_i \cup \mathcal{W}''_i \cup \mathcal{W}'''_i \cup \{\bar{\mathbf{y}}'''_1, \ldots, \bar{\mathbf{y}}'''_{j-1}\}$; $\Delta(\mathbf{y}', \mathbf{y})$ is a dissimilarity function computed by (1) where all documents in $\mathbf{y}'$ are assumed to be relevant and other documents not in $\mathbf{y}'$

TABLE 1
Main Notation Used in User-Interest Topic Model

Notation	Gloss	Notation	Gloss
q	query	d	document
u	user	z	topic
T	number of topics	U	number of users
D	number of documents	V	number of tokens
N_d	number of tokens in d		
$\widetilde{\mathbf{u}}$	a set of users	$\widetilde{\mathbf{w}}$	a set of tokens
b_z	Beta distribution parameter for z		
α	the parameter of user Dirichlet prior		
β	the parameter of token Dirichlet prior		
θ_d	multinomial distribution of topics specific to d		
ϕ_z	multinomial distribution of tokens specific to z		
ϑ_u	multinomial distribution of topics specific to u		
z_{di}	topic associated with the ith token in d		
w_{di}	the ith token in d		
r_{di}	relevance of the ith token in d		

are assumed to be irrelevant (note that the assumption we make here is only to compute the dissimilarity between two rankings). The target function defined in (9) aims to find the most violated additional constraints, whereas the condition defined in (9) makes sure that the additional constraints are different to each other in the working constraint set. The maximum size of the working set in Algorithms 1 and 3 is provide by the following theorem:

Theorem 1 (Theorem 18 in [8]). *Let* $\Delta_i = \max_{\mathbf{y}}\{\Delta(\mathbf{y}^{(i)}, \mathbf{y}), \sum_{y\in\mathbf{y}^{(i)}} \mathbf{w}^T\Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, y) - \mathbf{w}^T\Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}), ((1-\mathrm{sim}(\mathbf{y}, u^{(i)})) - \mu)\}$, *and* $R_i = \max_{\mathbf{y}}\{||\Psi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - \Psi(\mathbf{x}^{(i)}, \mathbf{y})||\}$. *With* $\bar{R} = \max_i R_i$, $\bar{\Delta} = \max_i \Delta_i$ *and for a given* $\xi > 0$, *Algorithms 1 and 3 terminate after incrementally adding at most* $\max\{\frac{2N\bar{\Delta}}{\xi}, \frac{8C\bar{\Delta}\bar{R}^2}{\xi^2}\}$ *constraints to the working set.*

Algorithm 3. Fast Training for Cutting Plane Algorithm

Input: $(\mathbf{x}^{(1)}, \mathbf{u}^{(1)}, \mathbf{y}^{(1)}), \ldots, (\mathbf{x}^{(N)}, \mathbf{u}^{(N)}, \mathbf{y}^{(N)}), C, \epsilon, M$
$\mathcal{W}_i \leftarrow \emptyset, \mathcal{W}'_i \leftarrow \emptyset, \mathcal{W}''_i \leftarrow \emptyset, \mathcal{W}'''_i \leftarrow \emptyset$ for all $i = 1, \ldots, N$
$\mu = \frac{1}{N}\sum_{i=1}^{N}(1 - \mathrm{sim}(\mathbf{y}^{(i)}, \mathbf{u}^{(i)}))$
repeat
  for $i = 1, \ldots, N$ **do**
    $H(\mathbf{y}; \mathbf{w}) \equiv \Delta(\mathbf{y}^{(i)}, \mathbf{y}) + \mathbf{w}^T\Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}) - \mathbf{w}^T\Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}^{(i)})$
    $H'(\mathbf{y}; \mathbf{w}) \equiv \sum_{y\in\mathbf{y}^{(i)}} \mathbf{w}^T\Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, y) - \mathbf{w}^T\Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}^{(i)})$
    $H''(\mathbf{y}; \mathbf{w}) \equiv \mathbf{w}^T\Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}) + ((1 - \mathrm{sim}(\mathbf{y}, \mathbf{u}^{(i)})) - \mu) - \mathbf{w}^T\Psi(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{y}^{(i)})$
    compute $\bar{\mathbf{y}} = \mathrm{argmax}_{\mathbf{y}} H(\mathbf{y}; \mathbf{w})$, $\mathbf{y}' = \mathrm{argmax}_{\mathbf{y}} H'(\mathbf{y}; \mathbf{w})$ and $\mathbf{y}'' = \mathrm{argmax}_{\mathbf{y}} H''(\mathbf{y}; \mathbf{w})$
    compute $\xi_i = \max\{0, \max_{\mathbf{y}\in\mathcal{W}_i} H(\mathbf{y}; \mathbf{w}), \max_{\mathbf{y}\in\mathcal{W}'_i} H'(\mathbf{y}; \mathbf{w}), \max_{\mathbf{y}\in\mathcal{W}''_i} H''(\mathbf{y}; \mathbf{w})\}$
    if $H(\bar{\mathbf{y}}; \mathbf{w}) > \xi_i + \epsilon$ **or** $H'(\bar{\mathbf{y}}'; \mathbf{w}) > \xi_i + \epsilon$ **or** $H''(\bar{\mathbf{y}}''; \mathbf{w}) > \xi_i + \epsilon$ **then**
      Add constraint to working set $\mathcal{W}_i \leftarrow \mathcal{W}_i \cup \{\bar{\mathbf{y}}\}$, $\mathcal{W}'_i \leftarrow \mathcal{W}'_i \cup \{\mathbf{y}'\}$, $\mathcal{W}''_i \leftarrow \mathcal{W}''_i \cup \{\mathbf{y}''\}$
      for $j = 1, \ldots, M$ **do**
        $\bar{\mathbf{y}}'''_j \leftarrow$ optimize (9)
        $\mathcal{W}'''_i \leftarrow \mathcal{W}'''_i \cup \{\bar{\mathbf{y}}'''_j\}$
      $\mathbf{w} \leftarrow$ optimize (7) over $\bigcup_i \{\mathcal{W}_i, \mathcal{W}'_i, \mathcal{W}''_i, \mathcal{W}'''_i\}$
until *no* $\mathcal{W}_i, \mathcal{W}'_i, \mathcal{W}''_i$ *and* $\mathcal{W}'''_i$ *have changed during iteration*

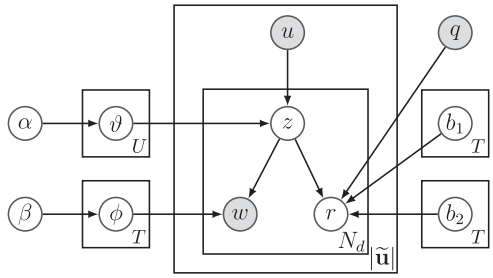

Fig. 1. Graphical representation of user-interest topic model.

5 User-Interest Topic Model and Feature Space

In this section, we first review the notation and terminology used in our user-interest topic model, and then describe the model and the features used in our structured learning framework.

We summarize the main notation used in our user-interest topic model (UIT) in Table 1. We distinguish between queries, aspects and topics. A *query* is a user's expression of an information need. An *aspect* (sometimes called *subtopic* at the TREC Web tracks [42]) is an interpretation of an information need. We use *topic* to refer to latent topics as identified by a topic modeling method (LDA).

5.1 User-Interest Topic Model

To capture per-user and per-document multinomial distributions over topics such that we can measure whether a document caters for the user's interest, we propose a user-interest latent topic model (UIT). Topic discovery in UIT is influenced not only by token co-occurrences, but also by the relevance scores of documents evaluated by users. In our UIT model, we use a Beta distribution over a (normalized) document relevance span covering all the data, and thus various skewed shapes of rising and falling topic prominence can be flexibly represented.

The latent topic model used in UIT is a generative model of relevance and tokens in the documents. The generative process used in Gibbs sampling [43] for parameter estimation, is:

i) Draw T multinomials ϕ_z from a Dirichlet prior β, one for each topic z;
ii) For each user u, draw a multinomial ϑ_u from a Dirichlet prior α; then for each token w_{di} in document $d \in \mathbf{u}$:
  a) Draw a topic z_{di} from multinomial ϑ_u;
  b) Draw a token w_{di} from multinomial $\phi_{z_{di}}$;
  c) Draw a relevance score r_{di} for w_{di} from Beta $(b_{z_{di}1}, b_{z_{di}2})$.

Fig. 1 shows a graphical representation of our model. In the generative process, the relevance scores of tokens observed in the same document are the same and evaluated by a user, although a relevance score is generated for each token from the Beta distribution. In our experiments, there is a fixed number of latent topics, T, although a non-parametric Bayes version of UIT that automatically integrates over the number of topics is possible. The posterior distribution of topics depends on information from two modalities: tokens and document relevance scores.

Inference is intractable in this model. Following [32], [33], [34], [43], [44], we employ Gibbs sampling to perform approximate inference. We adopt a conjugate prior (Dirichlet) for the multinomial distributions, and thus we can easily integrate out ϑ and ϕ, analytically capturing the uncertainty associated with them. In this way we facilitate the sampling, i.e., we need not sample ϑ and ϕ at all. Because we use the continuous Beta distribution rather than discretizing document relevance scores, sparsity is not a big concern in fitting the model. For simplicity and speed we estimate these Beta distributions (b_{z1}, b_{z2}) by the method of moments, once per iteration of Gibbs sampling. We find that the sensitivity of the hyper-parameters α and β is not very strong. Thus, for simplicity, we use fixed symmetric Dirichlet distributions ($\alpha = 50/T$ and $\beta = 0.1$) in all our experiments.

In the Gibbs sampling procedure above, we need to calculate the conditional distribution $P(z_{di}|\widetilde{\mathbf{w}}, \mathbf{r}, \mathbf{z}_{-di}, \widetilde{\mathbf{u}}, \alpha, \beta, \mathbf{b}, q)$, where $\mathbf{z}_{-di}$ represents the topic assignments for all tokens except w_{di}. We begin with the joint probability of a dataset, and using the chain rule, we can obtain the conditional probability as

$$P(z_{di}|\widetilde{\mathbf{w}}, \mathbf{r}, \mathbf{z}_{-di}, \widetilde{\mathbf{u}}, \alpha, \beta, \mathbf{b}, q) \propto \frac{n_{z_{di}w_{di}} + \beta_{w_{di}} - 1}{\sum_{v=1}^{V}(n_{z_{di}v} + \beta_v) - 1} \times \frac{n_{u_{di}z_{di}} + \alpha_{z_{di}} - 1}{\sum_{z=1}^{T}(n_{u_{di}z} + \alpha_z) - 1} \times \frac{(1 - r_{di})^{b_{z_{di}1}-1} r_{di}^{b_{z_{di}2}-1}}{B(b_{z_{di}1}, b_{z_{di}2})},$$

where n_{zv} is the total number of tokens v that are assigned to topic z, n_{uz} represents the number of topics z that are assigned to user u. See Appendix A for details.

After the Gibbs sampling procedure, we can easily infer a user's interest, i.e., multinomial distributions over topics for user u as

$$\vartheta_{uz} = p(z|u) = \frac{n_{uz} + \alpha_z}{\sum_{z=1}^{T}(n_{uz} + \alpha_z)}, \tag{10}$$

and easily infer multinomial distributions over tokens for topic z

$$\phi_{zv} = p(v|z) = \frac{n_{zv} + \beta_v}{\sum_{v=1}^{V}(n_{zv} + \beta_v)}, \tag{11}$$

where n_{zv} is the number of tokens of word v that are assigned to topic z. To obtain the multinomial distribution over topics for document d, i.e., θ_{dz}, we first apply the Bayes' rule

$$\theta_{dz} = p(z|d) = \frac{p(d|z)p(z)}{p(d)}, \tag{12}$$

where $p(d|z)$ is the probability of d belonging to topic z, and $p(z)$ is the probability of topic z. According to (11), $p(d|z)$ can be obtained as $p(d|z) = \prod_{v \in d} p(v|z) = \prod_{v \in d} \phi_{zv}$. According to (10), $p(z)$ can be obtained as $p(z) = \sum_{u=1}^{U} p(z|u)p(u)$, where U is the total number of users. Therefore, 12 can be represented as

$$\theta_{dz} = \frac{\prod_{v \in d} \phi_{zv} \sum_{u=1}^{U} p(z|u)p(u)}{p(d)}. \tag{13}$$

As any d has the same chance to be considered to be returned in response to q, we can assume that $p(d)$ is a constant, and likewise we also assume that $p(u)$ is a constant, such that (13) becomes

$$\theta_{dz} = \frac{1}{E} \prod_{v \in d} \phi_{zv} \sum_{u=1}^{U} \vartheta_{uz}, \tag{14}$$

where $E = \sum_{z=1}^{T} \prod_{v \in d} \phi_{zv} \sum_{u=1}^{U} \vartheta_{uz}$ is a normalization constant. Then, the topic distribution similarity $\mathrm{sim}(\mathbf{y}, \mathbf{u})$ between a set of documents $\mathbf{y}$ and the documents $\mathbf{u}$ a user u is interested in can be measured as

$$\mathrm{sim}(\mathbf{y}, \mathbf{u}) = \frac{1}{|\mathbf{y}|} \sum_{d \in \mathbf{y}} \cos(\theta_d, \vartheta_u), \tag{15}$$

where vectors $\theta_d = (\theta_{d1}, \ldots, \theta_{dT})$ and $\vartheta_u = (\vartheta_{d1}, \ldots, \vartheta_{dT})$ are the multinomial distribution of topics specific to document d and user u, respectively. We use the $\cos$ function in (15); other distance functions, e.g., based on euclidean distance, can be employed but we found no significantly different results.

5.2 Feature Space

The feature representation Ψ must enable meaningful discrimination between high quality and low quality predictions [10]. To predict a set of documents in the personalized diversification task, we propose to consider three main types of feature.

Tokens. Following [10], we define L token sets $V_1(\mathbf{y}), \ldots, V_L(\mathbf{y})$. Each token set $V_l(\mathbf{y})$ contains the set of tokens that appear at least l times in some document in $\mathbf{y}$. Then we use thresholds on the ratio $|D_l(v)|/|\mathbf{u}|$ (or $|D_l(v)|/|\mathbf{x}|$) to define feature values of $\psi_l(v, \mathbf{u})$ (or $\psi_l(v, \mathbf{x})$) that describe word v at l-th importance level. Here, $D_l(v)$ is the set of documents that have at least l copies of v in the whole set of documents $\mathbf{u}$ (or $\mathbf{x}$). We let $L = 20$ in our experiments, as quite a few tokens can appear more than 20 times in a document. Besides, we propose to directly utilize the tokens' statistics to capture similarity between a document $x \in \mathbf{y}$ and a set of documents $\mathbf{u}$ that a user u is interested in as features. We consider cosine, euclidean and Kullback-Leibler (KL) divergence similarity metrics. For each of these three metrics, we compute the minimal, maximal, and average similarity scores of the document $x \in \mathbf{y}$ and the standard deviations to a set of documents $\mathbf{u}$ based on the content of the documents and the standard LDA model [9]. In total, we have 49 features that fall in this feature category.

Interest. In addition, based on our UIT topic model, we also compute the cosine, euclidean and KL similarity between a document $x \in \mathbf{y}$ and a set of documents $\mathbf{u}$ based on a multinomial distribution over topics and the user's multinomial distribution over topics generated by UIT. Again, for each of these three similarity metrics, we compute the minimal, maximal, and average similarity scores and the standard deviation scores. In total, we have $S = 36$ features $\omega_s(x, \mathbf{u})$ that fall in this feature category.

Probability. The main probabilities used in state-of-the-art unsupervised personalized diversification methods are utilized in our learning model as features, i.e., $\gamma_m(x, \mathbf{x}, \mathbf{u})$. These probabilities include $p(d|q)$, the probability of d being relevant to q, $p(c|d)$, the probability of d belonging to category c, $p(c|q, u)$, the personalized query aspect distribution, $p(c|d, u)$, the personalized aspect distribution over d, and $p(d|c, u)$, the personalized aspect-dependent document

distribution, where c is a category that d belongs to in the Textwise Open Directory Project category service.[1] For $p(d|q)$, we obtain three versions, produced by BM25 [45], Jelinek-Mercer and Dirichlet language models [46]. To get the feature value of $p(c|d)$, we make use of the Textwise service, which returns up to three possible categories for d, ranked by a score in $[0, 1]$, and we use the normalized scores as features. We adopt five ways of computing $p(c|q, u)$ as feature values [5]; for details on how to compute $p(c|q, u)$, $p(c|d, u)$ and $p(d|c, u)$, see [5].

Then, we define $\Psi(\mathbf{x}, \mathbf{u}, \mathbf{y})$ as follows:

$$\Psi(\mathbf{x}, \mathbf{u}, \mathbf{y}) = \begin{bmatrix} \frac{1}{|\mathbf{y}|} \sum_{v \in V_1(\mathbf{y})} \psi_1(v, \mathbf{u}) \\ \frac{1}{|\mathbf{y}|} \sum_{v \in V_1(\mathbf{y})} \psi_1(v, \mathbf{x}) \\ \vdots \\ \frac{1}{|\mathbf{y}|} \sum_{v \in V_L(\mathbf{y})} \psi_L(v, \mathbf{u}) \\ \frac{1}{|\mathbf{y}|} \sum_{v \in V_L(\mathbf{y})} \psi_L(v, \mathbf{x}) \\ \frac{1}{|\mathbf{y}|} \sum_{x \in \mathbf{y}} \omega_1(x, \mathbf{u}) \\ \vdots \\ \frac{1}{|\mathbf{y}|} \sum_{x \in \mathbf{y}} \omega_S(x, \mathbf{u}) \\ \frac{1}{|\mathbf{y}|} \sum_{x \in \mathbf{y}} \gamma_1(x, \mathbf{x}, \mathbf{u}) \\ \vdots \\ \frac{1}{|\mathbf{y}|} \sum_{x \in \mathbf{y}} \gamma_M(x, \mathbf{x}, \mathbf{u}) \end{bmatrix}.$$

6 Experimental Setup

In this section, we describe our experimental setup. We begin by listing our research questions. We then describe our dataset, and our baselines and evaluation metrics, respectively. We conclude the section by detailing the settings of our experiments.

The research questions guiding the remainder of the paper are: (**RQ1**) Can supervised personalized diversification methods outperform state-of-the-art unsupervised methods? Can our method beat state-of-the-art supervised methods? (**RQ2**) What is the contribution of the user-interest topic model in our method? (**RQ3**) What is the effect of the constraints for diversity and consistence with user's interest in our method? (**RQ4**) Does our method outperform the best supervised baselines on each query? (**RQ5**) Can our method retrieve a competitive number of subtopics per query? (**RQ6**) What is the performance of our supervised methods when the C parameter is varied? (**RQ7**) Do the rankings generated by our supervised method and the baselines differ? (**RQ8**) Can additional highly violated and diversified constraints help to boost the efficiency of training our personalized diversification algorithm?

6.1 Dataset

In order to answer our research questions we use a publicly available personalized diversification dataset.[2,3] It contains private evaluation information from 35 users on 180 search queries. The queries are quite ambiguous, as the length of each query is no more than two keywords. In total, there are 751 subtopics for the queries, with most of the queries having more than two subtopics. Over 3,800 relevance judgements are available, for at least the top five results for each query. Each relevance judgement includes three main assessments: assessment-I—a two-grade assessment whether a specific subtopic is related to the evaluated query (resulting in subjective subtopics related to the search query); assessment-II—a four-grade scale assessment on how relevant the result is to the user's interests (resulting in the *user relevance* ground truth and a set of users' interesting documents); and assessment-III—a four-grade scale assessment on how relevant the result is to the evaluated query (resulting in the *topic relevance* ground truth being created). The format of the user and topic relevance ground truth is the same as that in the diversification task in the Web 2009-2014 Tracks [42] at TREC, viz. ⟨QueryID subtopicID DocumentID relevanceLevel⟩. Here, the judgements for subtopicID are generated from assessment-I for both types of ground truth, and the relevance level judgements for a document given a query are generated from assessment-II and assessment-III for user and topic relevance ground truth, respectively. See [5]. We apply Porter stemming, tokenization, and stopword removal (using the INQUERY list) to the documents using the Lemur toolkit.[4]

6.2 Baselines

Let PSVM_{div} denote our personalized diversification via structured learning method. We compare PSVM_{div} to 14 baselines: a traditional web search algorithm, BM25 [45]; three well-known plain (in the sense of "not personalized") search result diversification approaches, IA-Select [18], xQuAD [22] and PM-2 [24]; two plain (in the sense of "not diversified") personalized search approaches, Pers_{SL}, which combines the effect of long and short terms for personalized search [14], and Pers_{BM25}, which is based on BM25 [12]; a two-stage diversification and personalization approach, xQuAD_{BM25}, as suggested by [7], which first applies the xQuAD algorithm and then Pers_{BM25}; five state-of-the-art unsupervised personalized diversification methods [5], PIA-Select, $\mathrm{PIA-Select}_{BM25}$, PxQuAD, PxQuAD_{BM25} and PPM-2, which is the personalization version of PM-2 using the framework in [5]. As PSVM_{div} builds on a standard structured learning framework, we also consider two structured learning algorithms: SVM_{div} [10] that directly tries to retrieve relevant documents covering as many subtopics as possible, and a standard structured learning method, denoted as SVM_{rank} [11] that directly ranks documents by optimizing a relevance-biased evaluation metric.[5]

For the supervised methods, PSVM_{div}, SVM_{div} and SVM_{rank}, we use a 130/40/10 split for our training, validation and test sets, respectively. We train PSVM_{div}, SVM_{div} and SVM_{rank} using values of C (see (7)) that vary from 1e-4 to 1.0 and varying metric in (1). The best C value and metric

1. http://textwise.com
2. http://ir.ii.uam.es/~david/persdivers/
3. Two well-known corpora, ClueWeb09 and ClueWeb12 have been proposed to be used in the diversification tasks at TREC Web tracks [42]. However, they do not contain any user information or relevance judgments provided by specific users, and thus do not fit our research questions.
4. http://www.lemurproject.org
5. The source code for SVM_{rank} [11] and SVM_{div} [10] is available at http://www.cs.cornell.edu/People/tj/

in the loss function (1) are then chosen based on the validation set, and evaluated on the test queries. Similarly, we train the efficient version of PSVM_{div} by varying C, the metric and the unsupervised personalized diversification algorithm $\mathcal{A}(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}; \mathbf{y})$ integrated in it. The training/validation/test splits are permuted until all 180 queries were chosen once for the test set. We repeat the experiments 10 times and report the average evaluation results. Other baselines, such as xQuAD, PxQuAD, PM-2 and PPM-2, attempt to obtain a set of diversified documents $\mathbf{y}$ that maximizes the function $f(\mathbf{y}|q) = (1-\lambda) f_{\text{rel}}(\mathbf{y}|q) + \lambda f_{div}(\mathbf{y})$, where $f_{\text{rel}}(\mathbf{y}|q)$ estimates the relevance of the set of documents $\mathbf{y}$ to q and $f_{div}(\mathbf{y})$ estimates the dissimilarities between the documents in $\mathbf{y}$. For these baselines, we vary the parameter λ from 0 to 1.0.

6.3 Evaluation

We use the following diversity metrics for evaluation, most of which are official evaluation metrics in the TREC Web tracks [42]: α-nDCG@k [3], S-Recall@k [21], ERR-IA@k, Prec-IA@k [18], MAP-IA@k [18] and D#-nDCG@k [47], which utilizes multi-grade ratings.

For evaluating accuracy, we use nDCG [3], ERR, Prec@k and MAP. Since users mainly evaluated the top five returned results [5], we compute the scores at depth five for all metrics. For evaluating efficiency, we use time in seconds and the amount by which the most violated constraint is violated at the current iteration to see how fast the proposed algorithms can halt iterations.

Statistical significance of observed performance differences is tested using a two-tailed paired t-test and is denoted using $\blacktriangle$ (or $\blacktriangledown$) for significant differences for $\alpha = .01$, or $\triangle$ (and $\triangledown$) for $\alpha = .05$.

6.4 Experiments

We report on eight main experiments aimed at answering the research questions listed above. Our first experiment aims to understand whether supervised personalized diversification methods outperform unsupervised ones and whether PSVM_{div} beats the supervised algorithms that apply structured learning technique directly. We compare PSVM_{div} to two supervised baselines, SVM_{div} and SVM_{rank}, and the nine unsupervised baselines with topic relevance and user relevance ground truths, respectively.

To understand the contribution of the user-interest topic model, we conduct two experiments; in one we perform comparisons between PSVM_{div} using all features ("token," "interest" and "probability," see Section 5.2) including those extracted from the topic model and PSVM_{div} using basic features ("token" and "probability" only, see Section 5.2); in the other we also consider features extracted from LDA or author topic model [48] for comparison. In our third experiment, aimed at understanding the effect of our new constraints in PSVM_{div}, we employ different sets of constraints while training.

In order to understand how PSVM_{div} compares to the best baseline, our fourth and fifth experiment provide a query- and subtopic-level analysis, respectively. To understand the influence of the key parameter in our structured learning framework, C, and the metric defined in (1), we train PSVM_{div}, SVM_{div} and SVM_{rank} by varying the metric used in (1) between those listed in Section 6.3, and we vary C from 1e-4 to 1.0 and report the performance. We provide a case study to get an intuitive understanding of the algorithms. To determine whether adding highly violated and diversified constraints during iterations can boost the efficiency of training PSVM_{div}, we add $M = \{1, 2, \ldots, 8\}$ constraints, respectively, to PSVM_{div} and compare training times.

TABLE 2
Performance of Unsupervised Methods on Diversification Metrics

	α-nDCG	S-Recall	ERR-IA	Prec-IA	MAP-IA	D#-nDCG
User relevance						
BM25	.6443	.4557	.2267	.1659	.1245	.4973
IA-Select	.6099	.4282	.2241	.1624	.1177	.4642
Pers_{BM25}	.6427	.4541	.2318	.1639	.1206	.4951
Pers_{SL}	.6487	.4673	.2342	.1650	.1267	.5075
xQuAD	.6421	.4635	.2299	.1675	.1267	.4993
xQuAD_{BM25}	.6270	.4558	.2249	.1646	.1123	.4937
PM-2	.6501	.4674	.2347	.1655	.1278	.5061
PIA-Select	.5766	.4407	.2006	.1480	.1085	.4623
PIA-Select_{BM25}	.6457	.4752	.2364	.1581	.1180	.5074
PxQuAD	.6409	.4588	.2313	.1629	.1296	.4983
PxQuAD_{BM25}	.6497	.4713	**.2367**	**.1676**	.1296	.5113
PPM-2	**.6512**	$\blacktriangle$**.4775**	.2352	.1675	**.1342**	**.5154**
Topic relevance						
BM25	.7599	.4456	.2315	.1717	.1241	.5423
IA-Select	.7685	.4425	.2365	.1767	.1212	.5326
Pers_{BM25}	.7746	.4555	.2330	.1794	.1219	.5413
Pers_{SL}	.7752	.4587	.2375	.1771	.1223	.5512
xQuAD	.7711	.4600	.2348	.1747	.1245	.5489
xQuAD_{BM25}	.7763	.4741	.2336	.1773	.1225	.5534
PM-2	.7791	.4633	.2382	.1783	.1275	.5546
PIA-Select	.7410	.4641	.2227	.1650	.1206	.5343
PIA-Select_{BM25}	**.7854**	**.4798**	$\triangle$**.2415**	.1740	.1300	.5587
PxQuAD	.7744	.4543	.2350	.1747	.1278	.5493
PxQuAD_{BM25}	.7827	.4718	.2396	**.1797**	.1245	.5643
PPM-2	.7834	.4752	.2390	.1783	$\blacktriangle$**.1313**	$\blacktriangle$**.5664**

*The best performance per metric is in boldface. The best plain retrieval method (*BM25, IA-Select, Pers_{BM25}, Pers_{SL}, xQuAD, xQuAD_{BM25}, *and* PM-2*) is underlined. Statistically significant differences between the best performance per metric and the best plain retrieval method are marked in the upper left hand corner of the best performance score.*

7 Results and Analysis

7.1 Supervised versus Unsupervised

Table 2 lists the diversity scores of the unsupervised baseline methods. For all metrics, whether based on user relevance or topic relevance, we see that none of the plain methods, viz., BM25, IA-Select, Pers_{BM25}, Pers_{SL}, xQuAD, xQuAD_{BM25} and PM-2, beats the best unsupervised personalized diversification methods, viz., $\text{PIA} - \text{Select}_{BM25}$, PxQuAD_{BM25} or PPM-2. Moreover, in some cases the performance differences between the best plain method and the best unsupervised personalized diversification method are significant. This indicates that diversity and personalization are complementary and can enhance each other. The same observation can be found in Table 6 where performance is evaluated by relevance-oriented metrics.

Table 3 shows the diversity-oriented evaluation results of three supervised methods using basic features ("token",

TABLE 3
Performance of Supervised Methods Utilizing the Basic Features on Diversification Metrics

	User relevance					
	α-nDCG	S-Recall	ERR-IA	Prec-IA	MAP-IA	D#-nDCG
SVM_{rank}	▲.6667	△.4837	.2396	.1683	▲.1856	△.5273
SVM_{div}	▲.6750	△.4887	.2412	△.1698	▲.1974	▲.5325
PSVM_{div}	**▲.7234▲**	**▲.5756▲**	**▲.2514▲**	**▲.1702△**	**▲.2037△**	**▲.6134▲**
	Topic relevance					
SVM_{rank}	.7889	.4805	.2437	△.1812	▲.1848	▲.6254
SVM_{div}	▲.8003	△.4893	△.2479	△.1833	▲.2045	▲.6327
PSVM_{div}	**▲.8533▲**	**▲.5834▲**	**▲.2649▲**	**▲.1846△**	**▲.2113▲**	**▲.6475▲**

The best performance per metric is in boldface. Statistically significant differences between supervised and the best unsupervised method (in Table 2) per metric, between PSVM_{div} *and* SVM_{div}*, are marked in the upper left hand corner of the supervised method' score, in the right hand corner of the* PSVM_{div} *score, respectively.*

TABLE 5
Performance of Supervised Methods Utilizing All Features on Diversification Metrics

	User relevance					
	α-nDCG	S-Recall	ERR-IA	Prec-IA	MAP-IA	D#-nDCG
SVM_{rank}	△.6782	△.4973	.2416	△.1710	▲.2887	▲.5537
SVM_{div}	△.6867	△.4973	.2456	△.1729	▲.2911	▲.5745
PSVM_{div}	**▲.7513▲**	**▲.6140▲**	**△.2628▲**	**△.1742△**	**▲.2979△**	**▲.6424△**
	Topic relevance					
SVM_{rank}	▲.8422	▲.5068	▲.2554	△.1903	▲.3001	▲.6724
SVM_{div}	△.8569	▲.5068	▲.2628	▲.1907	▲.3036	▲.6937
PSVM_{div}	**▲.9549▲**	**▲.6730▲**	**▲.2849▲**	**△.1917△**	**▲.3096△**	**▲.7135**

The best performance per metric is in boldface. All the scores here are statistically significant compared to those in Table 2. Statistically significant differences between the method here and the method in Table 3, between PSVM_{div} *and* SVM_{div}*, are marked in the upper left hand corner of the corresponding score, in the right hand corner of the* PSVM_{div} *score, respectively.*

and "probability" features, see Section 5.2) in terms of both ground truths. In terms of diversity-oriented evaluation metrics all of the supervised methods significantly outperform the best unsupervised methods when making comparisons between the scores and the scores of unsupervised methods in Table 2 in most cases. We include further comparisons in Tables 6 and 7 in terms of relevance-oriented metrics, and find that supervised methods can statistically significantly outperform unsupervised ones. These two findings attest to the merits of taking supervised personalized diversification methods for the task of personalized search result diversification.

Next, we compare supervised strategies to each other. Tables 3 and 5 show the diversity-oriented evaluation resutls in terms of both grounth truths. It is clear from both tables that our supervised method PSVM_{div} statistically significantly beats plain supervised methods, SVM_{rank} and SVM_{div}. This is because PSVM_{div} considers both personalization and diversity factors, whereas the other two do not take both two factors into account. SVM_{rank} only tries to return more relevant documents, and SVM_{div} directly utilizes standard structured learning for diversification.

Table 7 shows that, in terms of relevance-oriented metrics, PSVM_{div} does not significantly outperform SVM_{rank} and SVM_{div}. PSVM_{div} returns the same number of relevant documents that do, however, cover more subtopics than the other supervised methods. Hence, PSVM_{div} mainly outperforms the other two in terms of diversity-oriented metrics. Sections 7.4 (query-level) and 7.5 (subtopic-level) have further analysis.

To understand the performance enhancement of our proposed algorithm compared to the baselines, we show the effect sizes of the performance differences between PSVM_{div} and *Other*, between PSVM_{div} and the best supervised learning algorithm SVM_{div}. Here *Other* is the best baseline except SVM_{rank} and SVM_{div} in Table 4. We use Cohen's d to compute the effect sizes. The effect sizes are quite large (> 0.5) in all cases, which again illustrates that PSVM_{div} does outperform the baselines.

7.2 Effect of the Proposed UIT Model

Next, to understand the contribution of our UIT topic model, we compare the performance of the supervised methods using basic features, i.e., all other features but not the features generated from the UIT model, with those using all the features. See Tables 3 and 5, which list the results of the supervised methods in terms of diversity-oriented metrics when using the basic features and all features, respectively. The use of all features outperforms only using the basic features.

We also compare the performance of our learning algorithm with the UIT topic model against the other topic models. In Table 8, we let $\text{PSVM}_{div}-\text{LDA}$ and $\text{PSVM}_{div}-\text{ATM}$ denote the algorithm using the basic features plus those generated by LDA, and the algorithm using the basic features plus those generated by the author topic model [48]. As can be seen from the table, $\text{PSVM}_{div}-\text{All}$ (which uses the basic features plus those generated by UIT; see Table 10) statistically significantly outperforms both $\text{PSVM}_{div}-\text{LDA}$ and $\text{PSVM}_{div}-\text{ATM}$. UIT integrates the relevance score in the topics' inference.

TABLE 4
Effect Size of the Differences between PSVM_{div} and *Other*, and between PSVM_{div} and SVM_{div} on the Two Ground Truth

	User relevance					
	α-nDCG	S-Recall	ERR-IA	Prec-IA	MAP-IA	D#-nDCG
PSVM_{div} versus *Other*	2.832	3.425	1.543	1.475	3.458	3.126
PSVM_{div} versus SVM_{div}	2.136	3.278	1.249	0.754	0.642	2.437
	Topic relevance					
PSVM_{div} versus *Other*	2.546	3.725	2.147	2.459	3.736	2.875
PSVM_{div} versus SVM_{div}	2.334	3.328	1.479	0.513	0.857	0.954

Here, Other *is the baseline (excluding* SVM_{rank} *and* SVM_{div}*) that performs the best on the corresponding metrics.*

TABLE 6
Performance of Unsupervised Methods on Relevance Metrics

	User relevance			
	nDCG	ERR	Prec	MAP
BM25	.5697	.9364	.7113	.2038
IA-Select	.5126	.9389	.6796	.1813
Pers_{BM25}	.5713	.9276	.7183	.2076
Pers_{SL}	.5724	.9314	.7213	.2142
xQuAD	.5526	.9352	.6858	.1915
xQuAD_{BM25}	.5540	.9133	.6921	.1841
PM-2	.5643	.9357	.7042	.2013
PIA-Select	.4783	.9034	.6417	.1774
PIA-Select_{BM25}	.5482	.9271	.6687	.1803
PxQuAD	.5631	.9246	.7050	.2073
PxQuAD_{BM25}	.5764	.9374	**.7258**	.2145
PPM-2	**.5780**	**.9378**	.7253	**.2152**
	Topic relevance			
BM25	.7775	.9440	.9146	.2239
IA-Select	.7340	.9452	.9250	.2299
Pers_{BM25}	.7741	.9374	.9298	.2316
Pers_{SL}	.7725	.9473	.9345	.2374
xQuAD	.7518	.9367	.9125	.2231
xQuAD_{BM25}	.7605	.9278	.9312	.2281
PM-2	.7623	.9367	.9350	.2293
PIA-Select	.6709	.9062	.8667	.2043
PIA-Select_{BM25}	.7264	.9418	.9042	.2223
PxQuAD	.7679	.9435	.9229	.2306
PxQuAD_{BM25}	**.7793**	.9466	**.9396**	.2355
PPM-2	.7763	**.9472**	.9396	**.2372**

Notational conventions are the same as in Table 2.

These results illustrate that our proposed UIT model can capture users' interest distributions and that this kind of information can be applied to improve performance. Due to space limitations, we do not report the results in terms of relevance-oriented metrics; the findings there are qualitatively similar.

7.3 Effect of the Proposed Constraints

Next, to understand the effect of the newly proposed constraints, we conduct experiments by employing different sets of constraints while training. The comparisons are again divided into those using all features and those using basic features. We write PSVM_{div}-C_i, PSVM_{div}-$\text{C}_{\text{i,ii}}$, PSVM_{div}-$\text{C}_{\text{i,iii}}$, and PSVM_{div}-All to denote the methods trained with the standard constraint (constraint i in (7)), standard and diversity-biased constraints (constraints i and ii in (7)), standard and interest-biased (constraints i and iii in (7)), and all constraints involved (constraints i, ii and iii in (7)), respectively. Again, we only report results on diversity-oriented metrics.

According to Tables 9 and 10, when employing one more constraint, either diversity-biased or interest-biased, the performance is significantly better than that of only employing the standard constraint. In terms of all metrics, the performance of PSVM_{div} employing all constraints statistically significantly outperforms the performance of using at most two constraints. Thus, combining diversification (the diversity-biased constraint) and personalization (the interest-biased constraint) boosts the performance.

TABLE 7
Performance of Supervised Methods Utilizing the Basic Features on Relevance Metrics

	User relevance			
	nDCG	ERR	Prec	MAP
SVM_{rank}	$^{\triangle}$.5805	$^{\triangle}$.9456	$^{\triangle}$.7345	$^{\triangle}$.2238
SVM_{div}	$^{\triangle}$.5813	$^{\triangle}$.9467	$^{\triangle}$.7396	$^{\triangle}$.2240
PSVM_{div}	$^{\triangle}$**.5833**	$^{\triangle}$**.9485**	$^{\triangle}$**.7412**	$^{\triangle}$**.2281**
	Topic relevance			
SVM_{rank}	$^{\triangle}$.7864	.9478	$^{\blacktriangle}$.9763	$^{\triangle}$.2446
SVM_{div}	$^{\triangle}$.7858	.9493	$^{\blacktriangle}$.9806	$^{\triangle}$.2482
PSVM_{div}	$^{\triangle}$**.7922**$^{\triangle}$	$^{\triangle}$**.9521**	$^{\blacktriangle}$**.9834**	$^{\triangle}$**.2496**

The best performance per metric is in boldface. Statistically significant differences between supervised and the best unsupervised method (in Table 6) per metric, between PSVM_{div} and SVM_{div}, are marked in the upper left hand corner of the supervised method' score, in the right hand corner of the PSVM_{div} score, respectively.

TABLE 8
Performance of PSVM_{div} Using Basic Features and Features Generated by Different Topic Models on Diversification Metrics with User Relevance Ground Truth

	User relevance					
	α-nDCG	S-Recall	ERR-IA	Prec-IA	MAP-IA	D#-nDCG
PSVM_{div}-LDA	.7223$^{\blacktriangledown}$	.5814$^{\blacktriangledown}$	.2524$^{\triangledown}$	.1712	.2254$^{\blacktriangledown}$	.6159$^{\blacktriangledown}$
PSVM_{div}-ATM	.7320$^{\blacktriangledown}$	.5927$^{\triangledown}$	.2572	.1715	.2314$^{\blacktriangledown}$	.6253$^{\blacktriangledown}$

Significant differences against PSVM_{div}-All in Table 10 are marked in the upper right hand corner of the corresponding scores.

TABLE 9
Performance of PSVM_{div} Involving Different Constraints Using Basic Features on Diversification Metrics with User Relevance Ground Truth

	User relevance					
	α-nDCG	S-Recall	ERR-IA	Prec-IA	MAP-IA	D#-nDCG
PSVM_{div}-C_i	.6713	.4842	.2403	.1673	.1969	.5325
PSVM_{div}-$\text{C}_{\text{i,ii}}$	.6973$^{\blacktriangle}$	.5262$^{\blacktriangle}$	.2437	.1681	.1977	.5437$^{\triangle}$
PSVM_{div}-$\text{C}_{\text{i,iii}}$	.6994$^{\blacktriangle}$	.5275$^{\blacktriangle}$	.2478$^{\triangle}$	.1687$^{\triangle}$	.1983	.5510$^{\blacktriangle}$
PSVM_{div}-All	**.7234**$^{\blacktriangle}$	**.5756**$^{\blacktriangle}$	**.2514**$^{\blacktriangle}$	**.1702**$^{\blacktriangle}$	**.2037**$^{\triangle}$	**.6134**$^{\blacktriangle}$

The best performance per metric is in boldface. Statistically significant differences against PSVM_{div}-C_i are marked in the upper right hand corner of the corresponding scores.

TABLE 10
Performance of PSVM_{div} Involving Different Constraints Using All Features on Diversification Metrics with User Relevance Ground Truth

	User relevance					
	α-nDCG	S-Recall	ERR-IA	Prec-IA	MAP-IA	D#-nDCG
PSVM_{div}-C_i	$^{\blacktriangle}$.6843	$^{\blacktriangle}$.4965	$^{\triangle}$.2434	$^{\triangle}$.1714	$^{\blacktriangle}$.2906	$^{\blacktriangle}$.5774
PSVM_{div}-$\text{C}_{\text{i,ii}}$	$^{\blacktriangle}$.7156$^{\blacktriangle}$	$^{\blacktriangle}$.5334$^{\blacktriangle}$	$^{\triangle}$.2494$^{\triangle}$	$^{\triangle}$.1720$^{\triangle}$	$^{\blacktriangle}$.2932	$^{\blacktriangle}$.5935$^{\blacktriangle}$
PSVM_{div}-$\text{C}_{\text{i,iii}}$	$^{\blacktriangle}$.7194$^{\blacktriangle}$	$^{\blacktriangle}$.5388$^{\blacktriangle}$	$^{\triangle}$.2501$^{\triangle}$	$^{\triangle}$.1723$^{\triangle}$	$^{\blacktriangle}$.2937$^{\triangle}$	$^{\blacktriangle}$.6048$^{\blacktriangle}$
PSVM_{div}-All	$^{\blacktriangle}$**.7513**$^{\blacktriangle}$	$^{\blacktriangle}$**.6140**$^{\blacktriangle}$	$^{\triangle}$**.2628**$^{\blacktriangle}$	$^{\triangle}$**.1742**$^{\blacktriangle}$	$^{\blacktriangle}$**.2979**$^{\triangle}$	$^{\blacktriangle}$.6424$^{\blacktriangle}$

Statistically significant differeneces between the score here and that in Table 9 are marked in the upper left hand corner of the scores. Other notational conventions are the same as in Table 9.

7.4 Query-Level Analysis

In order to figure out why PSVM_{div} enhances other supervised baselines, we take a closer look at per test query

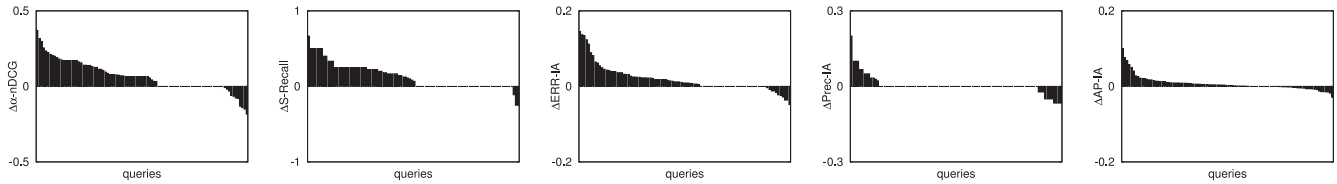

Fig. 2. Per query performance differences of PSVM_{div} against SVM_{div}. The figures shown are for α-nDCG, S-Recall, ERR-IA, Prec-IA, and MAP-IA, respectively. A bar extending above the center of a plot indicates that PSVM_{div} outperforms SVM_{div}, and vice versa for bars extending below the center.

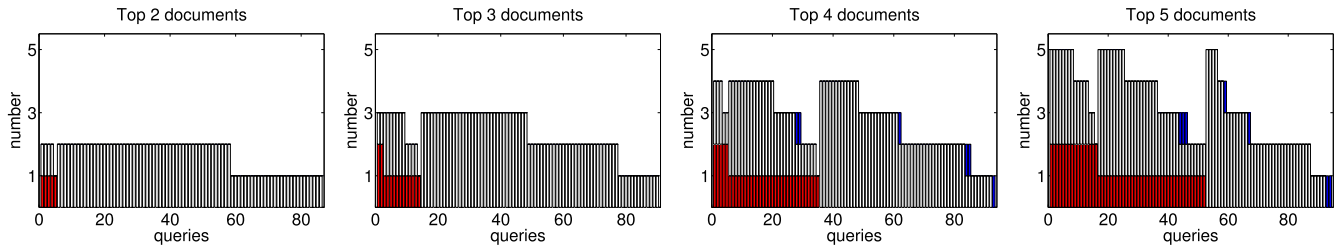

Fig. 3. How runs produced by PSVM_{div} and SVM_{div} differ. Red, white, and blue bars indicate the number of different subtopics that appear in PSVM_{div} but not in SVM_{div}, in both runs and not in PSVM_{div} but in SVM_{div}, respectively, at corresponding depth k (for k=2, 3, 4, 5).

improvements of PSVM_{div} over the best supervised baseline method, viz., SVM_{div}, which outperforms SVM_{rank} in most cases. Fig. 2 shows the per query performance differences in terms of the diversify-oriented metrics of PSVM_{div} against SVM_{div} when they use all the features. PSVM_{div} achieves performance improvements for many queries, especially in terms of α-nDCG, S-Recall, ERR-IA. In a small number of cases, SVM_{div} outperforms PSVM_{div}. This appears to be due to the fact that PSVM_{div} promotes some non-relevant documents when it tries to cover as many subtopics as possible for a given query.

7.5 Subtopic-Level Analysis

Next, we zoom in on the number of different subtopics that are returned by PSVM_{div} and SVM_{div}, respectively, to further analyze why PSVM_{div} beats SVM_{div}. Here, again, we use SVM_{div} as a representative. Specifically, we report changes in the number of subtopics for PSVM_{div} against SVM_{div} in Fig. 3 when they use all features. Red bars indicate the number of subtopics that appear in the run of PSVM_{div} but not in the run of SVM_{div}, white bars indicate the number of subtopics in both runs, whereas blue bars indicate the number of subtopics that are not in PSVM_{div} but in SVM_{div}; queries are ordered first by the size of the red bar, then the size of the white bar, and finally the size of the blue bar.

Clearly, the differences between PSVM_{div} and SVM_{div} in the top two and three are more limited than the differences in the top four and five, but in all cases PSVM_{div} outperforms SVM_{div}. For example, in total there are 68 more subtopics in the top five of the run produced by PSVM_{div} than those in the SVM_{div} run (in terms of all the 180 test queries, 68 subtopics in PSVM_{div} but not in SVM_{div}, seven subtopics in SVM_{div} but not in PSVM_{div}).

7.6 Performance of Parameter Tuning

To understand the performance of the tradeoff parameter C used in (4) and (7), we show the performance of PSVM_{div} and the two supervised baselines using all features. To save space, we only report the performance on α-nDCG. Fig. 4 plots the results; it illustrates that PSVM_{div} performs best when C is small. This indicates the merit of our new constraints (as well as the standard constraint used in the baselines) focusing on weight modification rather than on low training loss.

7.7 A Case Study

To gain an intuitive understanding of why our personalized diversification algorithm works better than the baselines, we provide a case study with the query "Apple" from the dataset. This query has three aspects. They are "aspect I: Apple fruit," "aspect II: Apple product" and "aspect III: Apple company." The top six documents in the rankings in response to a user's query "Apple" generated by the plain personalization algorithm Per_{SL}, plain diversification algorithm PM-2, unsupervised personalized diversification algorithm PPM-2 (we take Per_{SL}, PM-2, PPM-2 as representatives), and our personalized diversification algorithm PSVM_{div} are shown in Table 11.

According to the ground truth, the user is interested in "Apple fruit" and "Apple product" but not "Apple company," as he submitted queries such as "Food" and "iPhone." The top three documents in the ranking of Per_{SL} are associated with the same Apple aspect, as the user just recently submitted some queries such as "Food" and Per_{SL} utilizes the short term history for search; the

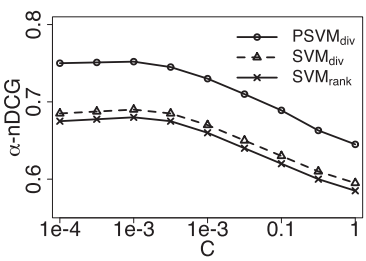

Fig. 4. Performance of the supervised methods using all features when varying the value of parameter C.

TABLE 11
Rankings Generated by Per_{SL}, PM-2, PPM-2, SVM_{div}, and PSVM_{div} in Response to the Query "Apple"

Per_{SL}		PM-2		PPM-2		PSVM_{div}	
Ranking	Aspect	Ranking	Aspect	Ranking	Aspect	Ranking	Aspect
Doc1	I	Doc1	I	Doc1	I	Doc1	I
Doc2	I	Doc4	II	Doc4	II	Doc4	II
Doc3	I	Doc7	III	Doc2	I	Doc2	I
Doc4	II	Doc2	I	Doc3	I	Doc5	II
Doc5	II	Doc5	II	Doc5	II	Doc3	I
Doc6	II	Doc8	III	Doc6	II	Doc6	II

Aspects "Apple fruit," "Apple product" and "Apple company" are denoted as "I," "II" and "III," respectively.

documents ranked 4-6 by Per_{SL} cover another aspect, viz. "Apple product," probably because the user submitted some queries such as "iPhone" before and Per_{SL} utilizes the long term history for search too. The plain diversification algorithm, PM-2, on the other hand, does not take the user's preference into account, diversifies the search results and covers all three aspects, one by one, including "Apple company" that the user is not interested in. In contrast, PPM-2 and PSVM_{div} try to retrieve and diversify the documents covering the two aspects amongst the top-k documents. Table 12 shows the performance against the user ground truth for the example query. PSVM_{div} outperforms the other baselines on all diversification metrics.

7.8 Fast Training Analysis

Finally, to understand the efficiency of the proposed fast training framework for our personalized diversification algorithm, we adapt the proposed Algorithm 3 to train PSVM_{div} with $M = \{1, 2, \ldots, 8\}$ additional highly violated but also diversified constraints being added to the original PSVM_{div} algorithm. We compare the experimental results with those of SVM_{div}, SVM_{rank}, and PSVM_{div} that is trained using the standard structured learning framework (no additional constraints are added, i.e., $M = 0$) as shown in Algorithm 1. Figs. 5 and 6 show the comparison results.

In Fig. 5, δ is the amount by which the most violated constraint is violated at that current iteration. The algorithm stops when $\delta \leq \epsilon = 0.01$. PSVM_{div} needs most iterations to converge, followed by the baselines SVM_{div} and SVM_{rank}. Even when only one additional constraint is added to PSVM_{div} (i.e., $M = 1$), PSVM_{div} needs considerably fewer

TABLE 12
Performance of PSVM_{div} in Response to the Query "Apple" Using the Basic Features and the Features Generated by Different Topic Models on Diversification Metrics with the User Relevance Ground Truth

	User relevance					
	α-nDCG	S-Recall	ERR-IA	Prec-IA	MAP-IA	D#-nDCG
Per_{SL}	.9414	1.000	.6112	.5000	.6277	1.000
PM-2	.9301	1.000	.6263	.4000	.4000	.9152
PPM-2	.9950	1.000	.6641	.5000	.6361	1.000
PSVM_{div}	1.000	1.000	.6687	.5000	.6917	1.000

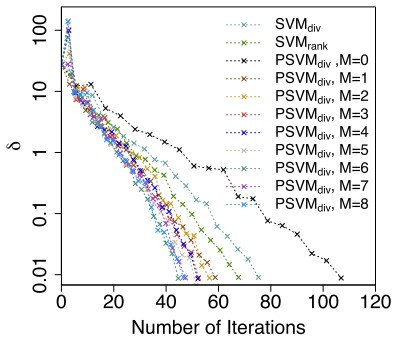

Fig. 5. Violation of the most violated constraint versus number of iterations to convergence for SVM_{div}, SVM_{rank}, PSVM_{div} ($M = 0$), and its efficient versions where $M = \{1, 2, \ldots, 8\}$ additional multiple constraints are added.

iterations compared to those of the original PSVM_{div} ($M = 0$), SVM_{div} and SVM_{rank}. When additional constraints are added ($M = \{1, 2, \ldots, 6\}$), fewer iterations are needed by PSVM_{div}. Including over six additional constraints (i.e., $M = 7, 8$) does not lead to decreases in the number of iterations for PSVM_{div}. This may be due to the fact that more constraints may become less informative for convergence. Clearly, the first additional constraint is the most informative one for PSVM_{div}, as the number of iterations drops most for $M = 1$.

In Fig. 6 we plot the training time in seconds versus the number of iterations for a range of values of M for PSVM_{div}, SVM_{div} and SVM_{rank}. Clearly, the baselines SVM_{div} and SVM_{rank} consume the smallest amount of training time although they are not the algorithms that need the smallest number of iterations for convergence. The original training framework for PSVM_{div} needs the most training time although its time versus iteration slope is just larger than those of SVM_{div} and SVM_{rank}. The more additional constraints are added during iterations, i.e,. the larger the value of M is, the larger the time versus iteration slopes are as solving the quadratic programing problem with more additional constraints needs more time. What is interesting for us in Fig. 6 is that for $M = 6$ PSVM_{div} needs almost the same amount of training time as for $M = \{1, 2, 3, 4\}$ while requiring fewer iterations for convergence. We do not show the diversification performance of PSVM_{div} when $M = \{1, 2, \ldots, 8\}$ as it is the same (no statistically significant) as for PSVM_{div} when $M = 0$ in most cases. In fact,

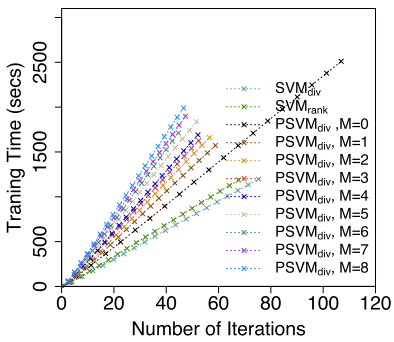

Fig. 6. Training time in seconds versus number of iterations to convergence for SVM_{div}, SVM_{rank}, PSVM_{div} ($M = 0$), and its efficient versions where $M = \{1, 2, \ldots, 8\}$ additional multiple constraints are added.

adding additional constraints does not impact or just negligibly impacts the final optimization.

Combining the findings in Figs. 5 and 6, we conclude that PSVM_{div} is able to maintain retrieval effectiveness, while adding highly violated and diversified constraints not only reduces the number of iterations but also the training time for PSVM_{div}.

8 Conclusion

Most previous work on personalized diversification of search results produces a ranking using unsupervised methods, either implicitly or explicitly. In this paper, we have adopted a different perspective on the problem, based on structured learning. We propose to boost the diversity and match to users' personal interests of search results by introducing two additional constraints into the standard structured learning framework. We also propose a user-interest topic model to capture users' multinomial distribution of interest over topics and infer per-document multinomial distributions over topics. Based on this, a number of user interest features are extracted and the similarity between a user and a document can be effectively measured for our learning method. To further boost the efficiency of training our proposed personalized diversification algorithm, we propose to add highly violated but also diversified constraints into our structured learning framework.

Our evaluation shows that supervised personalized diversification approaches outperform state-of-the-art unsupervised personalization diversification, plain personalization and plain diversification algorithms. The two proposed constraints are shown to play a significant role in the supervised method. We also find that the user-interest topic model helps to improve performance. Our proposed learning method is able to return more subtopics. Adding more informative constraints can help to make training faster, needs fewer iterations and still keep almost the same performance.

We aim to study other types of learning strategies for personalized diversification of search results. We use an unsupervised method to generate the additional highly violated and diversified constraints for our training; looking for other alternative ways to get more effective constraints is another follow-up research step. Finally, our experimental results were only evaluated on a single dataset. In future work we plan to pursue two alternatives: to use simulations based on click models [49] and to invite users to label the existing datasets, e.g., ClueWeb09, such that they can also be used for evaluating personalized diversification algorithms.

Acknowledgments

The authors thank the reviewers for their valuable comments. This research was supported by the China Scholarship Council, Amsterdam Data Science, the Dutch national program COMMIT, Elsevier, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 312827 (VOX-Pol), the ESF Research Network Program ELIAS, the Royal Dutch Academy of Sciences (KNAW) under the Elite Network Shifts project, the Microsoft Research Ph.D. program, the Netherlands eScience Center under project number 027.012.105, the Netherlands Institute for Sound and Vision, the Netherlands Organisation for Scientific Research (NWO) under project nos. 727.011.005, 612.001.116, HOR-11-10, 640.006.013, 612.066.930, CI-14-25, SH-322-15, 652.002.001, 612.001.551, the Yahoo Faculty Research and Engagement Program, and Yandex. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors. An earlier version of this article appeared in the *Proceedings of KDD* '14 [1]. In this substantially extended version, we propose an efficient structured learning algorithm for personalized diversification by adding additional violated and diversified constraints at every iteration of the standard cutting-plane algorithm. We detail the inference of our user interest topic model in Appendix A, examine its performance, add more baselines and metrics, study and answer more research questions, and provide a case study. We also expand the motivation of our work in the introduction, our discussion of related work and our analysis of experimental results.

Appendix A
Gibbs Sampling Derivation for UIT Model

We begin with the joint distribution $P(\widetilde{\mathbf{w}}, \mathbf{r}, \mathbf{z}, \widetilde{\mathbf{u}}|\alpha, \beta, \mathbf{b}, q)$. We can take advantage of conjugate priors to simplify the integrals. All symbols are defined in Sections 3, 4 and 5

$$
\begin{aligned}
&P(\widetilde{\mathbf{w}}, \mathbf{r}, \mathbf{z}, \widetilde{\mathbf{u}}|\alpha, \beta, \mathbf{b}, q) = P(\widetilde{\mathbf{w}}|\mathbf{z}, \beta)p(\mathbf{r}|\mathbf{b}, \mathbf{z}, q)P(\mathbf{z}|\widetilde{\mathbf{u}}, \alpha) \\
&= \int P(\widetilde{\mathbf{w}}|\Phi, \mathbf{z})p(\Phi|\beta)d\Phi \times p(\mathbf{r}|\mathbf{b}, \mathbf{z}, q) \int P(\mathbf{z}|\widetilde{\mathbf{u}}, \Theta)p(\Theta|\alpha)d\Theta \\
&= \int \prod_{d=1}^{D}\prod_{i=1}^{N_d} P(w_{di}|\phi_{z_{di}}) \prod_{z=1}^{T} p(\phi_z|\beta)d\Phi \\
&\quad \times \prod_{d=1}^{D}\prod_{i=1}^{N_d} p(r_{di}|b_{z_{di}1}, b_{z_{di}2}, q) \\
&\quad \times \int \prod_{d=1}^{D}\prod_{i=1}^{N_d} P(z_{di}|\vartheta_u) \prod_{u=1}^{U} p(\vartheta_u|\alpha)d\Theta \\
&= \int \prod_{z=1}^{T}\prod_{v=1}^{V} \phi_{z_v}^{n_{z_v}} \prod_{z=1}^{T} \left(\frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)} \prod_{v=1}^{V} \phi_{z_v}^{\beta_v - 1} \right) d\Phi \\
&\quad \times \prod_{d=1}^{D}\prod_{i=1}^{N_d} p(r_{di}|b_{z_{di}1}, b_{z_{di}2}, q) \\
&\quad \times \int \prod_{u=1}^{U}\prod_{z=1}^{T} \vartheta_{u_z}^{n_{u_z}} \prod_{u=1}^{U} \left(\frac{\Gamma(\sum_{z=1}^{T}\alpha_z)}{\prod_{z=1}^{T}\Gamma(\alpha_z)} \prod_{z=1}^{T} \vartheta_{u_z}^{\alpha_z - 1} \right) d\Theta \\
&= \left(\frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)} \right)^{T} \left(\frac{\Gamma(\sum_{z=1}^{T}\alpha_z)}{\prod_{z=1}^{T}\Gamma(\alpha_z)} \right)^{U} \\
&\quad \times \prod_{d=1}^{D}\prod_{i=1}^{N_d} p(r_{di}|b_{z_{di}1}, b_{z_{di}2}, q) \\
&\quad \times \prod_{z=1}^{T} \frac{\prod_{v=1}^{V}\Gamma(n_{z_v} + \beta_v)}{\Gamma(\sum_{v=1}^{V}(n_{z_v} + \beta_v))} \prod_{u=1}^{U} \frac{\prod_{z=1}^{T}\Gamma(n_{u_z} + \alpha_z)}{\Gamma(\sum_{z=1}^{T}(n_{u_z} + \alpha_z))}
\end{aligned}
$$

Applying the chain rule, we can obtain the conditional probability

$$
\begin{aligned}
&P(z_{di}|\widetilde{\mathbf{w}}, \mathbf{r}, \mathbf{z}_{-di}, \widetilde{\mathbf{u}}, \alpha, \beta, \mathbf{b}, q) \\
&= \frac{P(z_{di}, w_{di}, r_{di}, u_{di}|\widetilde{\mathbf{w}}_{-di}, \mathbf{r}_{-di}, \mathbf{z}_{-di}, \widetilde{\mathbf{u}}_{-di}, \alpha, \beta, \mathbf{b}, q)}{P(w_{di}, r_{di}, u_{di}|\widetilde{\mathbf{w}}_{-di}, \mathbf{r}_{-di}, \mathbf{z}_{-di}, \widetilde{\mathbf{u}}_{-di}, \alpha, \beta, \mathbf{b}, q)} \\
&= \frac{P(\widetilde{\mathbf{w}}, \mathbf{r}, \mathbf{z}, \widetilde{\mathbf{u}}|\alpha, \beta, \mathbf{b}, q)}{P(\widetilde{\mathbf{w}}, \mathbf{r}, \mathbf{z}_{-di}, \widetilde{\mathbf{u}}|\alpha, \beta, \mathbf{b}, q)} \\
&\text{because } z_{di} \text{ depends only on } w_{di}, r_{di} \text{ and } u_{di} \\
&\propto \frac{P(\widetilde{\mathbf{w}}, \mathbf{r}, \mathbf{z}, \widetilde{\mathbf{u}}|\alpha, \beta, \mathbf{b}, q)}{P(\widetilde{\mathbf{w}}_{-di}, \mathbf{r}_{-di}, \mathbf{z}_{-di}, \widetilde{\mathbf{u}}_{-di}|\alpha, \beta, \mathbf{b}, q)} \\
&\propto \frac{n_{z_{di}w_{di}} + \beta_{w_{di}} - 1}{\sum_{v=1}^{V}(n_{z_{di}v} + \beta_v) - 1} \frac{n_{u_{di}z_{di}} + \alpha_{z_{di}} - 1}{\sum_{z=1}^{T}(n_{u_{di}z} + \alpha_z) - 1} \\
&\times \frac{(1 - r_{di})^{b_{z_{di}1}-1} r_{di}^{b_{z_{di}2}-1}}{B(b_{z_{di}1}, b_{z_{di}2})}.
\end{aligned}
$$

As relevance is drawn from continuous Beta distributions, sparsity is not a big problem for parameter estimation of $\mathbf{b}$. For simplicity, we update $\mathbf{b}$ after each Gibbs sample by the method of moments as

$$
\begin{aligned}
b_{z1} &= \bar{t}_z \left(\frac{\bar{t}_z(1 - \bar{t}_z)}{s_z^2} - 1 \right), \\
b_{z2} &= (1 - \bar{t}_z) \left(\frac{\bar{t}_z(1 - \bar{t}_z)}{s_z^2} - 1 \right),
\end{aligned}
$$

where $\bar{t}_z$ and s_z^2 are the sample mean and biased sample variance of the relevance belonging to topic z, respectively.

REFERENCES

[1] S. Liang, Z. Ren, and M. de Rijke, "Personalized search result diversification via structured learning," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 751–760.

[2] V. Dang and W. B. Croft, "Term level search result diversification," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2013, pp. 603–612.

[3] C. L. A. Clarke, et al., "Novelty and diversity in information retrieval evaluation," in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2008, pp. 659–666.

[4] X. Shen, B. Tan, and C. Zhai, "Implicit user modeling for personalized search," in *Proc. 14th ACM Int. Conf. Inf. Knowl. Manage.*, 2005, pp. 824–831.

[5] D. Vallet and P. Castells, "Personalized diversification of search results," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2012, pp. 841–850.

[6] Y. Shi, X. Zhao, J. Wang, M. Larson, and A. Hanjalic, "Adaptive diversification of recommendation results via latent factor portfolio," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2012, pp. 175–184.

[7] F. Radlinski and S. Dumais, "Improving personalized web search using result diversification," in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2006, pp. 691–692.

[8] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *J. Mach. Learn. Res.*, vol. 6, pp. 1453–1484, 2005.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[10] Y. Yue and T. Joachims, "Predicting diverse subsets using structural SVMS," in *Proc. 25th Int. Conf.*, 2008, pp. 1224–1231.

[11] Y. Yue, T. Finley, F. Radlinski, and T. Joachims, "A support vector method for optimizing average precision," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2007, pp. 271–278.

[12] D. Vallet, I. Cantador, and J. M. Jose, "Personalizing web search with folksonomy-based user and document profiles," in *Proc. 32nd Eur. Conf. Adv. Inf. Retrieval*, 2010, pp. 420–431.

[13] H. Wang, X. He, M.-W Chang, Y. Song, R. W. White, and W. Chu,, "Personalized ranking model adaptation for web search," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2013, pp. 323–332.

[14] P. N. Bennett, et al., "Modeling the impact of short- and long-term behavior on search personalization," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2012, pp. 185–194.

[15] C. Liu, N. J. Belkin, and M. J. Cole, "Personalization of search results using interaction behaviors in search sessions," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2012, pp. 205–214.

[16] Z. Dou, R. Song, and J.-R. Wen, "A large-scale evaluation and analysis of personalized search strategies," in *Proc. 16th Int. Conf. World Wide Web*, 2007, pp. 581–590.

[17] Z. Dou, R. Song, J.-R. Wen, and X. Yuan, "Evaluating the effectiveness of personalized web search," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 8, pp. 1178–1190, Aug. 2009.

[18] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong, "Diversifying search results," in *Proc. 2nd ACM Int. Conf. Web Search Data Mining*, 2009, pp. 5–14.

[19] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in *Proc. 21st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1998, pp. 335–336.

[20] H. Chen and D. R. Karger, "Less is more: Probabilistic models for retrieving fewer relevant documents," in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2006, pp. 429–436.

[21] C. Zhai, W. W. Cohen, and J. Lafferty, "Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2003, pp. 10–17.

[22] R. L. Santos, C. Macdonald, and I. Ounis, "Exploiting query reformulations for web search result diversification," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 881–890.

[23] S. Vargas, P. Castells, and D. Vallet, "Explicit relevance models in intent-oriented information retrieval diversification," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2012, pp. 75–84.

[24] V. Dang and W. B. Croft, "Diversity by proportionality: An election-based approach to search result diversification," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2012, pp. 65–74.

[25] K. Bache, D. Newman, and P. Smyth, "Text-based measures of document diversity," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 23–31.

[26] S. Liang, Z. Ren, and M. de Rijke, "Fusion helps diversification," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2014, pp. 303–312.

[27] S. Liang and M. de Rijke, "Burst-aware data fusion for microblog search," *Inf. Process. Manage.*, vol. 51, no. 2, pp. 89–113, 2015.

[28] L. Li, K. Zhou, G.-R. Xue, H. Zha, and Y. Yu,, "Enhancing diversity, coverage and balance for summarization through structure learning," in *Proc. 18th Int. Conf. World Wide Web*, 2009, pp. 71–80.

[29] S.-X. Zhang and M. Gales, "Structured SVMs for automatic speech recognition," *IEEE Trans. Audio Speech Language Process.*, vol. 21, no. 3, pp. 544–555, Mar. 2013.

[30] A. Guzman-Rivera, P. Kohli, and D. Batra, "DivMCuts: Faster training of structural SVMS with diverse M-best cutting-planes," in *Proc. AISTATS JMLR Workshop Conf.*, 2013, pp. 316–324.

[31] S. Branson, O. Beijbom, and S. Belongie, "Efficient large-scale structured learning," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2013, pp. 1806–1813.

[32] J. Boyd-Graber and D. M. Blei, "Syntactic topic models," in *Proc. 21st Adv. Neural Inf. Process. Syst.*, 2008, pp. 185–192.

[33] J. Boyd-Graber andD. M. Blei, "Multilingual topic models for unaligned text," in *Proc. 25th Conf. Uncertainty Artificial Intell.*, 2009, pp. 75–82.

[34] X. Wang and A. McCallum, "Topics over time: A non-Markov continuous-time model of topical trends," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 424–433.

[35] J. Zhu, X. Zheng, L. Zhou, and B. Zhang, "Scalable inference in max-margin topic models," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 964–972.

[36] Z. Ren and M. de Rijke, "Summarizing contrastive themes via hierarchical non-parametric processes," in *Proc. 38th Int. ACM SIGIR Conf. Research Develop. Inf. Retrieval*, 2015, pp. 93–102.

[37] S. Liang and M. de Rijke, "Formal language models for finding groups of experts," *Inf. Process. Manage.*, vol. 52, no. 4, pp. 529–549, 2016.

[38] Y. Zhao, S. Liang, Z. Ren, J. Ma, E. Yilmaz, and M. de Rijke, "Explainable user clustering in short text streams," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2016, pp. 155–164.

[39] S. Liang, E. Yilmaz, and E. Kanoulas, "Dynamic clustering of streaming short documents," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, to appear.
[40] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Inf. Process. Lett.*, vol. 70, no. 1, pp. 39–45, 1999.
[41] P. Shivaswamy and T. Joachims, "Online structured prediction via coactive learning," in *Proc. 29th Int. Conf. Mach. Learn.*, 2006, pp. 1431–1438.
[42] K. Collins-Thompson, C. Macdonald, P. Bennett, F. Diaz, and E. M. Voorhees, "TREC 2014 web track overview," in *Proc. 23rd Text REtrieval Conf.*, 2015, pp. 1–21.
[43] J. S. Liu, "The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem," *J. Amer. Statistical Assoc.*, vol. 89, no. 427, pp. 958–966, 1994.
[44] S. Jameel and W. Lam, "An unsupervised topic segmentation model incorporating word order," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2013, pp. 203–212.
[45] S. E. Robertson and D. A. Hull, "The TREC-9 filtering track final report," in *Proc. 9th Text REtrieval Conf.*, 2000, pp. 25–40.
[46] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to ad hoc information retrieval," in *Proc. 24th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2001, pp. 334–342.
[47] T. Sakai, et al., "Simple evaluation metrics for diversified search results," in *Proc. 3rd Int. Workshop Evaluation Inf. Access*, 2010, pp. 42–50.
[48] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *Proc. 20th Conf. Uncertainty Artificial Intell.*, 2004, pp. 487–494.
[49] A. Chuklin, I. Markov, and M. de Rijke, *Click Models for Web Search*. San Rafael, CA, USA: Morgan & Claypool Publishers, 2015.

Shangsong Liang received the BSc and MSc degrees in computer science from Northwest A&F University, Xi'an, China, in 2008 and 2011, respectively, and the PhD degree in computer science from the University of Amsterdam, in 2014. He is currently a postdoc at University College London, London, United Kingdom. His research interests include information retrieval and data mining. He has published at SIGIR, KDD, CIKM, ECIR, and in *Information Processing & Management*.

Fei Cai received the MSc degree in system engineering from the National University of Defense Technology, Changsha, China, in 2010. He is working toward the PhD degree at the University of Amsterdam under the supervision of Maarten de Rijke. His current research interests include information retrieval, learning to rank, and query understanding. He has published several papers at SIGIR and CIKM.

Zhaochun Ren received the BE and MSc degrees from Shandong University, in 2009 and 2012, respectively. He is working toward the PhD degree at the University of Amsterdam, supervised by Maarten de Rijke. He is interested in information retrieval, social media mining, and content analysis in e-discovery. Before joining UvA, he worked as a short-term visiting scholar in Max-Planck-Institut für Informatik, 2012. He has previously published at SIGIR, CIKM, and KDD.

Maarten de Rijke received the MSc degrees in philosophy and mathematics, and the PhD degree in theoretical computer science. He is a professor in computer science in the Informatics Institute, University of Amsterdam. He previously worked as a postdoc at CWI, before becoming a Warwick research fellow at the University of Warwick, United Kingdom. He is the editor-in-chief of the *ACM Transactions on Information Systems* and of the *Springer's Information Retrieval* book series.