Question Recommendation for Collaborative Question Answering Systems with RankSLDA

Jose San Pedro
Telefonica Research
Barcelona, Spain
jspw@tid.es

Alexandros Karatzoglou
Telefonica Research
Barcelona, Spain
alexk@tid.es

ABSTRACT

Collaborative question answering (CQA) communities rely on user participation for their success. This paper presents a supervised Bayesian approach to model expertise in online CQA communities with application to question recommendation, aimed at reducing waiting times for responses and avoiding question starvation. We propose a novel algorithm called *RankSLDA* which extends the supervised Latent Dirichlet Allocation model by considering a learning-to-rank paradigm. This allows us to exploit the inherent collaborative effects that are present in CQA communities where users tend to answer questions in their topics of expertise. Users can thus be modeled on the basis of the topics in which they demonstrate expertise. In the supervised stage of the method we model the pairwise order of expertise of users on a given question. We compare *RankSLDA* against several alternative methods on data from the *Cross Validate* community, part of the Stack Exchange network. *RankSLDA* outperforms all alternative methods by a significant margin.

Categories and Subject Descriptors

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

Keywords

Learning to Rank; Supervised Latent Dirichlet Allocation; Latent Topics Model; Question Recommendation; Community Question Answering; Expert Modeling

1. INTRODUCTION

Many users resort to the Internet for finding answers to their questions and solving their information needs [22]. Community Question Answering (CQA) websites, such as Yahoo! Answers or Stack Overflow, allow users to formulate questions to leverage the expertise of other members participating in the community. In the past years, CQA websites have rapidly grown in size and popularity. For instance, Yahoo! Answers includes over 300 million questions posted since it launched in 2005; only during 2012, it received an average of $7,000$ questions and $21,000$ answers per hour[1].

CQA websites are heavily dependent on community participation. While most provide passive question discovery tools, such as search or unanswered question lists, they still require significant effort from expert users to find the questions to which they can provide responses. This additional load on the community can potentially reduce participation and degrade the overall quality of the content.

A solution to this problem studied in previous work considers the task of automatically finding potential responders to questions [24, 10, 13, 18, 19, 11, 6, 20, 8]. These *question recommendation* systems aim at increasing participation by proactively warning users about the presence of questions suitable to their interests and expertise. The task of matching experts to questions given their previous answers to similar questions can be posed as a Recommendation problem. The expected result of implementing such a system is that less questions are left unanswered and the elapsed time for answers is significantly decreased. Note that such a system can be effectively used along with other participation encouragement tools, such as rewards, karma points, etc.

One common feature of most CQA systems is the presence of community feedback tools, which serve as a crowd-sourced and distributed curation mechanism. Users can easily vote, positively or negatively, for questions (if they consider them interesting for future reference) or answers (if they correctly solve the problem stated in the associated question). All votes casted for a post, and hence for the user that posted it, get aggregated into a single score, which serves as a proxy for question/answer quality. This rich source of information can be used for modeling user expertise.

State-of-the-art Recommender Systems are often based on factor models e.g. [9, 17, 12, 14]. Factor-based collaborative filtering methods represent both "items" and "users" with a vector of latent features. In our problem setting, the presence of text (questions and answers) allows us to use a more interpretable basis for modeling the experts and the questions, namely topic modeling [3]. The main idea behind the model we introduce is that users with expertise in similar topics are likely to answer similar questions, we use this fact both to recommend questions to experts but also to reinforce the learning of the corpus topics.

In this paper, we present *RankSLDA*, Rank Supervised Latent Dirichlet Allocation, a supervised probabilistic topic model with direct application to question recommendation. *RankSLDA* is based on a Bayesian inference framework that

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

RecSys'14, October 6–10, 2014, San Jose or vicinity, CA, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2668-1/14/10 ...$15.00.
http://dx.doi.org/10.1145/2645710.2645736 .

[1] Source: http://searchengineland.com

extends the LDA model to account for the authorship of questions and answers as well as for community feedback. The proposed model combines the semantic content modeling benefits of LDA with supervised ranking learning to model the observed community scores based on the latent topics assigned to each question.

The contribution of this paper is two-fold. First, we propose a novel learning-to-rank extension to supervised LDA, and provide the derivation of a Gibbs sampler to perform inference. Second, we apply this model to the question recommendation task and provide experimental results of its performance compared to several state-of-the art methods.

The paper is organized as follows. In Section 2 we review previous literature on related topics. In Section 3 we provide an in-depth description of the generative model proposed, and the procedure to train and tune its parameters. In Section 4 we describe the setup used to study the proposed method, and in Section 5 we present the results obtained. We conclude with Section 6.

2. RELATED WORK

A body of literature exists around expertise modeling in the CQA context. Two main approaches have been used to this end: network and content-based. Network-based methods are specifically targeted to CQA communities with a narrow topic focus. Networks built from the interaction of users in question threads are analyzed to infer their relative expertise order. Network centrality has been proven to be a good indicator of expertise on them [24]. Different network creation approaches have been proposed for comparing expertise between users to establish a global rank. For instance, competition-based expertise networks are formalized by establishing directed links between the best answerer and all the other contributors [10, 2].

Content-based methods consider an Information Retrieval centric vision, where users are profiled according to their contributions in the website, and are ranked with respect to an expertise query (e.g. a new question). Methods based on TF-IDF [13], pLSA [18], and probabilistic topic models [19] have been proposed using this general framework. A related line of work considers the use of a classification-based approach, where the user-question relationships are represented in a common feature space to find their reciprocal relevance [6]. Tensor factorization approaches, capturing known relationships between askers, question and answerers, have also been proposed to predict best answerers [20].

Several bayesian generative models have been used in this setting. Guo *et al.* introduced the User-Question-Answer model, which considers user profiles as topic mixtures, and uses question categories to improve the recommendation performance [8]. Ni *et al.* proposed the Topic-based User Interest model, which leverages community selected best answers to promote users that contribute high quality posts [11].

Beside *best answers*, which can be biased and unreliable [5], none of these works make explicit use of the rich community feedback available from most CQA communities. One previous work that uses aggregated voting scores is [21]. In it, observed votes are drawn as part of the generative process from a Gaussian Mixture Model. In contrast, we consider a generative model with a supervised stage, where observed votes are used for building optimized user profiles based on the latent topics of the questions and answers contributed. Also, the fLDA algorigthm [1] predicts ratings of users for documents, expressed as topic mixtures, by using matrix factorization to model the affinity between users and topics. While it can be applied to question recommendation, the model is optimized for rating prediction instead of ranking.

3. METHODOLOGY

3.1 Problem Statement

In this section we provide the formalization of the question recommendation task that we use in the rest of the paper. The main element of this formalization is the question thread, which comprises the following components: one unique *question* (which originates the thread), one or more *answers* (that attempt to solve the problem stated in the question), *community feedback* (aggregated scores computed from the community votes to questions and answers), and *user information* (user identifier of each question and answer). For the sake of brevity, we will refer to the individual textual components of the thread, both questions and answers, as *posts*. In this scenario, each post has been authored by a single user and has an associated quality score given by the community. We can obtain an absolute ranking of experts for each question by sorting answers in decreasing order of aggregated voting scores. The question recommendation task is to predict this rank for new questions.

3.2 RankSLDA Model Description

RankSLDA builds on supervised latent Dirichlet allocation (sLDA [3]), an approach that combines LDA topic modeling, where document topic mixtures are drawn from a Dirichlet distribution, with a response variable associated to individual documents. The goal is to find the latent topics that best explain the observed responses. The key innovation of this paper is the extension of sLDA, originally restricted to a single response value per document, to a more flexible multitask scenario able to model the multiple pairwise preferential relationships observed for each document.

As noted, we pose question recommendation as a pairwise ranking problem. We denote as $s(d, u)$ the aggregated score for user u received for his/her contribution to question d. To optimize for the latent topics that best explain the observed preferential relationship of user pairs we consider r_d as the ordering for a given question thread d, so that user pairs $(u_i, u_j) \in r_d$ when $s(d, u_i) > s(d, u_j)$. The model observed responses are then defined by:

$$y_d^{(i,j)} = \begin{cases} 1, & (u_i, u_j) \in r_d \\ 0, & \text{otherwise} \end{cases}$$

Given $\Phi(d, u)$, a feature representation of the matching between a question thread d and a user u, the learning procedure finds the model $\boldsymbol{\gamma}$ that maximizes the number of pairs $(u_i, u_j) \in r_d$ where

$$\langle \boldsymbol{\gamma}, \Phi(d, u_i) \rangle > \langle \boldsymbol{\gamma}, \Phi(d, u_j) \rangle$$

A binary classification approach on pairwise differences of vectors $(\Phi(d, u_i) - \Phi(d, u_j))$ can be used to model this problem, which requires inverting some of the duplets (u_j, u_i) to balance the training dataset. The plate diagram of the *RankSLDA* model is shown in Figure 1.

3.3 Generative Model

Our model, depicted in Fig 1, can be expressed as a generative process that generates question threads and assigns

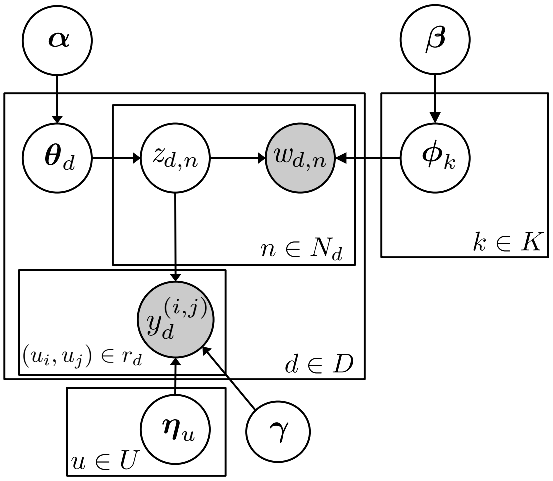

Figure 1: Graphic model representing *RankSLDA*. Shaded nodes represent observed variables and edges probabilistic dependencies.

pairwise preference scores, $y_d^{(i,j)}$. The observed scores, $y_d^{(i,j)}$, which encode the relative rank order between users, are assumed to come from a Bernouilli distribution parametrized by: 1) the ranking model (γ) and 2) the question-user features ($\Phi(d, u_i)$ and $\Phi(d, u_j)$):

$$y_d^{(i,j)} \sim \text{Bern}(p_d^{(i,j)} = \pi(\boldsymbol{\gamma}, \Phi(d, u_i), \Phi(d, u_j)))$$
$$\text{with } \pi(\boldsymbol{\gamma}, \boldsymbol{x}, \boldsymbol{y}) = \text{sigmoid}(\langle \boldsymbol{\gamma}, (\boldsymbol{x} - \boldsymbol{y}) \rangle)$$

Given the topic mixture of the current question, $\boldsymbol{\theta}_d$, and its matching to user profiles, $\Phi(d, u)$, the ranking model determines pairwise preference scores, where their sign determines the predicted ranking order. The sigmoid function maps these scores in the $[0, 1]$ interval, which serves as the parameter $p_d^{(i,j)}$ of the Bernouilli distribution from which the observed variable is drawn. Depending on the discrepancies of predicted and observed values, the model adapts both the ranking coefficients and the topic mixtures to maximize the likelihood of the observed data.

The influence of question topics and users' expertise on the observation is included in the feature vector $\Phi(d, u)$. As later described, we infer user's expertise during the supervised step using multitask regression over the actual observed scores, $s(d, u)$. The regression coefficients, $\boldsymbol{\eta}_u$, can be interpreted as the relevance of each topic for a given user u, effectively encoding their topical knowledge. With these user profiles, we can obtain the final feature vector as a function $\Phi(d, u) = f(\boldsymbol{\theta}_d, \boldsymbol{\eta}_u)$ (point-wise vector multiplication in our case). The generative model proceeds as follows:

1. Draw topic distributions $\boldsymbol{\phi}_k \sim Dir(\boldsymbol{\beta})$ with $i = 1 \ldots K$.
2. For each question thread $d = 1 \ldots D$:
   (a) Draw distribution over topics $\boldsymbol{\theta}_d \sim Dir(\boldsymbol{\alpha})$.
   (b) For each word in the document $n = 1 \ldots N_d$, draw topic assignment $z_{d,n} \sim Mult(\boldsymbol{\theta}_d)$.
   (c) For each user pair $(u_i, u_j) \in r_d$, choose a response variable $y_d^{(i,j)} \sim \text{Bern}(\pi(\boldsymbol{\gamma}, f(\bar{\boldsymbol{z}}_d, \boldsymbol{\eta}_{u_i}), f(\bar{\boldsymbol{z}}_d, \boldsymbol{\eta}_{u_j})))$, with $\bar{\boldsymbol{z}}_d \equiv \frac{1}{N_d} \sum_{n=1}^{N_d} z_{d,n} = \hat{\boldsymbol{\theta}}_d$.

Note that we use the empirical topic distribution to define the documents' mixtures $\bar{\boldsymbol{z}}_d$. Statistical inference is then used to find the distribution over latent variables and the parameter values that best explain the observed data. We follow the approach described in [7] using stochastic EM, where the E-step is performed using collapsed Gibbs sampling to infer topic assignments for the terms in the documents. The process is started by randomly initializing the topics, and then alternates between sampling topics of words $z_{d,n}$ and optimizing the regression parameters $\boldsymbol{\eta}_u$ and ranking model $\boldsymbol{\gamma}$ for the given topic assignments and observations. Because there is a dependency between topic assignments and observed responses, the inferred topic distribution favors topic assignments that minimize the difference between the predicted and the observed responses.

Table 1: Notation used in the paper

Symbol	Description
d,u,k,n	index for documents, users, topics and words
U	Number of users in the community
D	Number of documents (i.e. question threads)
N_d	Number of words in document d
$z_{d,n}$,$w_{d,n}$	n-th topic assignment and term of document d
$y_d^{(i,j)}$	Response for user pair (u_i, u_j) and document d
r_d	Binary ordering between users for document d
$\boldsymbol{z}_d$	Topic mixture proportion of document d
$\boldsymbol{z}_d^{\neg n}$	Same as $\boldsymbol{z}_d$ excluding the n-th word of d
$\boldsymbol{\theta}_d$	Multinomial topic distribution of document d
$\boldsymbol{\phi}_k$	Multinomial word distribution of topic k
$\boldsymbol{\eta}_u$	User topical expertise model
$\boldsymbol{\gamma}$	Pair-wise ranking model

In the remaining of the paper we used the notation summarized in Table 1 to refer to the variables and parameters of the model.

3.4 Gibbs Sampler

We use a collapsed Gibbs sampler to collect samples from the posterior distribution of the model. In this section, we derive the equations of the Gibbs sampler used, which presents interesting differences to LDA and sLDA due to its multitask nature. The collapsed Gibbs sampler needs to compute the probability distribution of $P(z_{d,n} = k)$ conditioned on the rest of the variables:

$$P(z_{d,n} | \boldsymbol{z}^{\neg n}, \boldsymbol{w}, \boldsymbol{y}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\gamma}) = \frac{P(z_{d,n}, \boldsymbol{z}^{\neg n}, \boldsymbol{w}, \boldsymbol{y}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\gamma})}{P(\boldsymbol{z}^{\neg n}, \boldsymbol{w}, \boldsymbol{y}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\gamma})}$$

For Gibbs sampling purposes, we can compute a proportional expression and then normalize:

$$\begin{aligned} &P(z_{d,n} = k | \boldsymbol{z}^{\neg n}, \boldsymbol{w}, \boldsymbol{y}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\gamma}) \\ &\propto P(z_{d,n} = k, \boldsymbol{z}^{\neg n}, \boldsymbol{w}, \boldsymbol{y}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\gamma}) = P(\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{y}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\gamma}) \\ &= \int P(\boldsymbol{\theta} | \boldsymbol{\alpha}) P(\boldsymbol{z} | \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} \cdot \int P(\boldsymbol{w} | \boldsymbol{z}, \boldsymbol{\phi}) P(\boldsymbol{\phi} | \boldsymbol{\beta}) \, \mathrm{d}\boldsymbol{\phi} \cdot P(\boldsymbol{y} | \boldsymbol{z}, \boldsymbol{\eta}, \boldsymbol{\gamma}) \end{aligned}$$

where the last equality is the joint probability distribution of the model. The first two terms of this expression are the standard factors of a collapsed LDA Gibbs sampler (expanded in equation 3). The term $P(\boldsymbol{y} | \boldsymbol{z}, \boldsymbol{\eta}, \boldsymbol{\gamma})$ of this conditional probability distribution includes information on the likelihood that new topic assignments $\boldsymbol{z}_d$ explain the observations, shifting the distribution towards topics that generate responses in consonance to their observed value $y_d^{(i,j)}$.

Note that the current topic assignment can only affect the predicted response for the current document d, hence:

$$P(\boldsymbol{y} | \boldsymbol{z}, \boldsymbol{\eta}, \boldsymbol{\gamma}) \propto P(\boldsymbol{y}_d | \boldsymbol{z}_d, \boldsymbol{\eta}, \boldsymbol{\gamma})$$

where proportionality is kept with respect to $z_{d,n}$. Given a set of topic assignments to document d, the probability of observing the response value for a given user pair (u_i, u_j) is

$$P(y_d^{(i,j)}|\boldsymbol{z}_d; \boldsymbol{\eta}_{u_i}, \boldsymbol{\eta}_{u_j}, \boldsymbol{\gamma}) \sim \text{Bern}(\pi(\boldsymbol{\gamma}, f(\boldsymbol{z}_d, \boldsymbol{\eta}_{u_i}), f(\boldsymbol{z}_d, \boldsymbol{\eta}_{u_j})))$$

Assuming independence, we can express the joint probability of observed user responses for a given document as

$$P(\boldsymbol{y}_d|\boldsymbol{z}_d; \boldsymbol{\eta}, \sigma) = \prod_{(u_i,u_j)\in r_d} P(y_d^{(i,j)}|\boldsymbol{z}_d; \boldsymbol{\eta}_{u_i}, \boldsymbol{\eta}_{u_j}, \boldsymbol{\gamma}) \qquad (1)$$

Note that the expression considers only user pairs $(u_i, u_j) \in r_d$. The probability of topics assignments is therefore shifted towards those that maximize the correct number of ranking predictions from $\boldsymbol{\gamma}$. We can now completely specify the conditional distribution for our Gibbs sampler:

$$P(z_{d,n} = k|\boldsymbol{z}^{\neg n}, \boldsymbol{w}, \boldsymbol{y}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \sigma) \propto \qquad (2)$$

$$(n_{d,k}^{\neg n} + \alpha_k) \frac{n_{w_{d,n},k} + \beta_{w_{d,n}}}{\sum_{j=1}^{W} n_{j,k} + \beta_j} \qquad (3)$$

$$\prod_{(u_i,u_j)\in r_d} \frac{1}{1 + exp(\langle \boldsymbol{\gamma}, f(\boldsymbol{z}_d(\boldsymbol{\eta}_{\boldsymbol{u_i}} - \boldsymbol{\eta}_{\boldsymbol{u_j}}))\rangle)} \qquad (4)$$

where $n_{w,k}$ refers to the number of times word w has been assigned to topic k, and $n_{d,k}$ refers to the frequency of topic k in document d. We use the standard notation $\neg n$ to refer to the counts where the term being sampled is excluded.

Computing pairwise preference scores of users on new questions is equivalent to marginalizing over $\boldsymbol{y}_d$ and sampling topics using eq. 3, as shown by [7].

3.5 Supervised Stage

After sampling topics for each document, we find new regression parameters for all users, $\boldsymbol{\eta}_u$, as well as the ranking model, $\boldsymbol{\gamma}$, that maximize the likelihood of the response variables conditioned on the current state of topic assignments.

The main issue for learning parameters $\boldsymbol{\eta}$ is that most users have relatively few documents (questions) so that finding an accurate representation might be difficult. To alleviate this issue we resort to multitask learning which allows us to jointly learn all the η_u vectors. Multitask learning [4] can be seen as a form of transfer learning in which similar learning "tasks" share information contained in their training signals. This is particularly well suited for our setting as each user can represent a task and we can easily assume that groups of users will be experts in a set of common topics thus allowing for transfer of information among them. Moreover, we use group lasso [23] to select the relevant topics for all the users. We thus optimize the following multitask lasso objective using an l_1/l_2 regularization norm:

$$\|\boldsymbol{s} - \boldsymbol{z}\boldsymbol{\eta}\|_F^2 + \lambda \sum_u \sqrt{\sum_k \eta_{u,k}^2}$$

where $\boldsymbol{s}$ the matrix of scores of all documents d and user u combinations, $\boldsymbol{z}$ is the matrix of document d topic k representations, $\boldsymbol{\eta}$ the matrix of all user u, topic k representations and λ the regularization parameter.

Regarding the ranking model $\boldsymbol{\gamma}$, the proposed framework is flexible enough to accomodate different binary classification methods. A metric notion of the decision function is required to properly assess the variation experienced by the predicted responses for different topic assignments, which gets transferred into the the Gibbs samping distribution (eq. 4). We experimented with two linear models, Logistic Regression and Linear SVM, because the linear formulation allows for several optimizations in the Gibbs sampling implementation. While both performed similarly in terms of accuracy, we chose logistic regression with l_2 regularization in our experimental setting as it empirically showed to be more stable w.r.t. the choice of hyperparameters.

3.6 Hyperparameter Tuning

The model counts with a number of hyperparameters that need to be properly tuned, in particular the Dirichlet priors α and β, the regression regularizer λ, and the classification regularizer C. These are treated as constants to be estimated instead of random variables of the model.

To describe our estimation strategy let us first define how the stochastic EM procedure is carried out. Firstly, we allow for a number burn-in iterations of the Gibbs sampler, which we set empirically to $M_b = 40$. During these iterations, we remove from the sampling distribution the factor corresponding to $P(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{\eta}, \boldsymbol{\gamma})$ as we have not yet enough information to initialize both $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$. From this point onwards, the procedure follows these steps:

1. Smooth topic assignments: we compute a smoothed version of the topic mixtures of documents, $\bar{\boldsymbol{z}}_d$, by averaging the mixtures obtained on the previous M_s iterations. We fixed this value $M_s = 20$ empirically.
2. Build regression and ranking models: We find new values for $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$ using the current topic assignments $\bar{\boldsymbol{z}}_d$ and the observed training responses $\boldsymbol{y_d}$ following the procedure described in Section 3.5.
3. Draw topic assignments using CGS: We perform M_g iterations of the full Gibbs sampler to draw topics for all documents. We set $M_g = 40$ empirically.

This process is repeated until the total number of Gibbs sampling iterations reaches $M = 500$.

Regarding the estimation of the concentration parameters, α and β, we estimate them directly from data. Wallach *et al.* provided empirical evidence of the performance gains obtained by using asymmetric priors [16], which are unfeasible to set using grid-search approaches. In particular, we used the Digamma Recurrence Relation proposed by Wallach [15] to optimize both concentration parameters. This process is performed every 5 iterations of the Gibbs sampler. Regarding the regularization parameters λ and C, we chose them using grid-search with a 3-fold cross validation strategy.

4. EXPERIMENTAL SETUP

We consider the task of question recommendation, which comprises the prediction of the best experts for previously unseen questions. To this end, we consider a set of questions that has a known set of experts, generate a rank of possible experts using our model, and compare both. We use community feedback as ground truth for the quality of responses, so the the ideal rank to predict is obtained by sorting users for each question in decreasing order of score (i.e. aggregated number of votes).

4.1 Dataset

The study presented in this paper considers data from the *Cross Validate*[2] community of the Stack Exchange net-

[2] http://stats.stackexchange.com

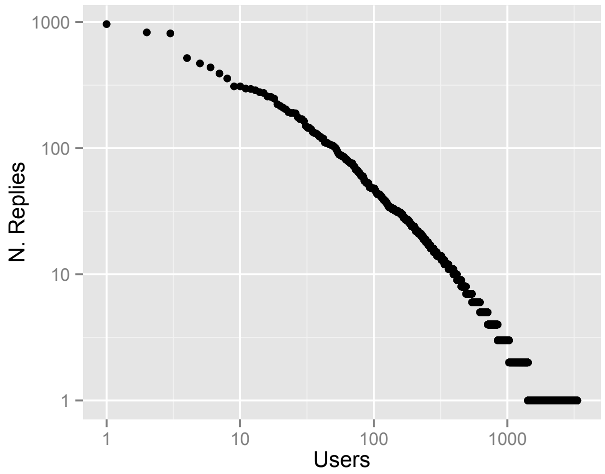

Figure 2: Number of replies for *Cross Validate* users, in decreasing order of participation (log-log scale)

work, focusing on Q&A about statistics and machine learning. Stack Exchange releases Creative Commons-licensed data dumps quarterly, which include the textual content of posts and other metadata (e.g. date, scores, authors).

Throughout the experiments of this paper, we used the Cross Validate data dump corresponding to June 2013, which comprises 3 years, 21,819 questions, 28,429 answers and 11,281 unique users. Only 15,894 of these questions have at least 1 answer, which represents 72.84% of the total. The median time to receive the first answer is 2.3 hours. Hence, there is room for improvement in terms of answer coverage and average waiting time.

Regarding participation, only 3,334 users ($< 30\%$) answered at least 1 question. Low participation levels are organic to CQA communities, where most of the answers are generated by a minority of contributors. In this case, 80% of the answers have been contributed by the top 430 most active users. The median number of questions replied per user is 1, with a mean of 8.527. In Figure 2 we show this long-tail distribution of answering participation in our set.

In terms of community feedback, users make active use of the rating mechanisms. Both questions and answers can be voted as positive or negative; votes get aggregated into a score value, which we used as ground truth for the supervised stage of our model. In total, 23,986 posts were voted by the community, representing almost 85% of all the posts.

4.2 Baseline Description

Section 5 evaluates the effectiveness of the presented model by comparing it to several other methods that we use as baselines. For every question in the test set we compare to:

- Popularity Ranking (**PR**): ranks users according to their answering frequency in decreasing order. This naive baseline approach does not entail profiling users from their history of previous contributions.

- TF-IDF (**BOW**): considers user profiles based on a *tf-idf* representation of the contributions of users to the system. In our setup, user profiles encompass all textual content (questions and answers) posted by the user and, additionally, the questions they have replied. Ranking is established in increasing order of cosine distance between the *tf-idf* representation of new questions with all user profiles.

- LSI (**LSI**): considers latent semantic indexing (LSI) for representing user profiles and questions. We consider question threads (including all contributed answers) as the documents for finding the latent topics of the corpus. User profiles are built by aggregating all their contributions as well as the questions they have provided answers to. These are then used as input documents to the learned LSI model to obtain users' topic mixtures. Ranking is established using the cosine distance between user profiles and questions.

- LDA+Ranking (**LDA-R**): considers a *RankSLDA* model where the supervised factor of the Gibbs sampler (eq. 4) is omitted. This corresponds to an unsupervised LDA model where topics are learned from the corpus and then used to train a pairwise ranking model using the observed responses $s(d, u)$ as ground truth. Similarly to *RankSLDA*, question threads act as the documents for finding the latent topics of the corpus $\boldsymbol{z}_d$, and user profiles $\boldsymbol{\eta}_u$ are learned from the observed responses via regression. These two pieces of information are used to build feature vectors $\Phi(d, u)$ as explained in section 3.2, which serve to train the pairwise ranking model. Comparing with this baseline enables us to observe the effect of including the observed scores as an integral part of the Bayesian inference process for the RandSLDA model.

4.3 Evaluation Metrics

We used ranking evaluation metrics to assess the precision of our method and the different baseline approaches considered. Note that our target rank to predict is obtained by sorting users for each question in decreasing order of community-provided score.

- **P@k:** Precision at cut-off k measures the number of users that provided a response in the top k positions of the predicted rank, normalized by the cut-off value. To binarize our ground truth we consider any user with an aggregated score over 0 as relevant. We evaluate precision at cut-off levels $K = \{1, 5, 10\}$.

- **nDCG@k:** Discounted Cumulative Gain (DCG) extends $P@k$ to allow for multiple relevance values. It is defined by the expression:

$$\text{DCG@k} = \sum_{i=1}^{k} \frac{2^{s_i} - 1}{log_2(i+1)}$$

where s_i stands for the relevance score of the i-th author. The normalized DCG (nDCG) normalizes this score to allow comparison across different queries. The normalizing factor is obtained by computing the DCG@k for the ideal rank. We evaluate nDCG at cut-off levels $K = \{1, 5, 10\}$.

- **MAP:** Mean average precision for a set of queries is

$$\text{MAP} = \frac{\sum_{q=1}^{Q} AP(q)}{Q}$$

where $AP(q)$ denotes the average precision for query q, computed as

$$AP(q) = \frac{\sum_{k=1}^{U} P@k(q) \cdot \mathbb{I}(s_k > 0)}{\sum_{k=1}^{U} \mathbb{I}(s_k > 0)}$$

where $\mathbb{I}$(cond) is an indicator function with value 1 if *cond* is true, and 0 otherwise.

- **MRR:** Mean Reciprocal Rank is the average of the Reciprocal Rank for a set of queries. The reciprocal rank for a query q is defined as

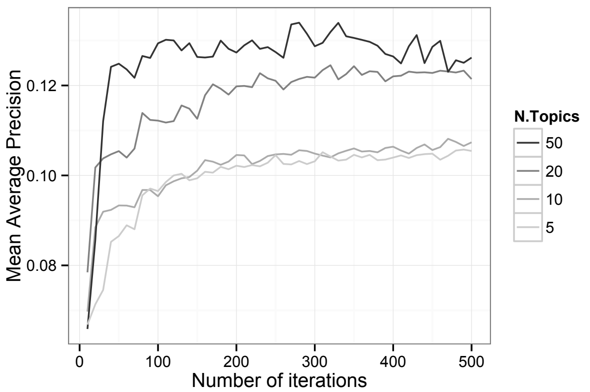

Figure 3: Evolution of MAP for *RankSLDA*

$$RR(q) = \frac{1}{r_q}$$

with r_q being the rank of the first relevant user for question q, using the same notion of relevance as for P@k.

5. RESULTS

In this section we present the results obtained from the application of the *RankSLDA* model, as well as the baseline methods, to the question recommendation scenario using the dataset introduced in Section 4. We split the original dataset into two partitions for training and test purposes following a strictly chronological criterion. Our training set comprises 75% of the question threads, from July 2010 to February 2013. The rest of the dataset, from February 2013 onwards, was used as the test collection.

The number of active users in the training set was 8, 805, from which 2, 930 contributed replies. We decided to set a minimum threshold of activity by discarding from the training set all users with 5 or less contributed replies, which led to a total of 967 users. Regarding the test set, we consider the same collection of 967 users as we do not have information to model new answerers. After applying this criterion for user selection, we removed the contributions of "inactive" users from the dataset. We then proceeded to eliminate question threads with less than two answers, leading to a total of 6, 108 question threads for training and 2, 092 for test. The dataset information is gathered in Table 2.

We show the overall results for all performance metrics in Table 3. We can observe that the *RankSLDA* method systematically outperforms the baselines for all metrics considered. The best performance is achieved by the *RankSLDA* method for $K = 50$ topics, as the higher number of topics enables the model to generate finer-grained models of user expertise, boosting the question-user matching accuracy.

Despite its simplicity, popularity ranking achieves comparable results to more sophisticated approaches (LSI with up to 500 topics) and not far from the *tf-idf* approach, especially if we factor in the high dimensionality of the latter, over 45, 000 features in our corpus. The supervised approaches, LDA-R and *RankSLDA*, provide significant improvements over more simple baselines. Even with a low dimensionality, $K = 5$ topics, *RankSLDA* achieves a MAP of over 0.10, meaning that in average at least 1 of the top-10 users ranked is relevant for the question. None of the unsupervised methods was able to obtain this score.

Table 2: Characteristics of the test dataset

Collection	Type	N.Question threads	N.Users
Training	Unfiltered	16, 893	2, 930
	Filtered	6, 108	967
Test	Unfiltered	4, 926	811
	Filtered	2, 092	967

The MAP evolution across iterations of the Gibbs sampler for different number of topics, K is depicted in Figure 3. After the initial burn-in period of the first 40 iterations, the full conditional sampling distribution starts being taken into account. This is followed by a steady increase in MAP, that is particularly noticeable for $K = \{5, 10, 20\}$.

For $K = 50$, we observe an initial growing trend that plateaus after the first 100 iterations. This evidences a tendency of the *RankSLDA* model to overfit the training data when increasing the number of topics. An early termination strategy could be used to avoid this artifact and get closer to the maximum test ranking performance, obtained in this case at iteration $i = 280$ with an MAP = 0.1339.

In Figure 4 we compare the evolution of MAP between *RankSLDA* and LDA-R across iterations of the Gibbs sampler for different number of topics. LDA-R shows a similar pattern in all cases, with a rapid increase in MAP after the burn-in period, and a plateau effect once the Gibbs sampler converges to the posterior distribution. *RankSLDA*, on the other hand, continuously improves MAP by alternating the EM steps described in Section 3. These plots show the impact of equation 4 in the assignment of document topics for the *RankSLDA* model, which effectively shifts topic assignments towards mixtures that better help explain the observed rankings during training.

Figures 5 and 6 show similar plots comparing *RankSLDA* and LDA-R for P@k and nDCG@k respectively. The results depict a similar scenario to the one described previously for Figure 4, with LDA-R plateauing after a small number of iterations, and *RankSLDA* improving in time thanks to the model capacity to adapt to observed scores when sampling topic assignments. We can observe the most notable difference between LDA-R and *RankSLDA* in the *P@1* case, which highlights the better ability of *RankSLDA* to choose a relevant user at the top of the rank.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed *RankSLDA*, a Bayesian framework that combines supervised ranking with topic modeling. It can be applied to question recommendation, where both community feedback and text content topics are jointly modeled for ranking users according to their relevance for new questions. Our experiments using data from the Cross Validate community show empirical evidence of the ability of the model to influence topic assignments during training to better explain the observed community scores. CQA communities could benefit from question recommendation for decreasing the rate of unanswered questions and reducing the average waiting time for answers to new questions.

We plan to explore how this model could be used to encourage participation in the community by promoting the rank of less active users in the long-tail of participation. We are also interested in exploring alternative ranking models in this framework, e.g. list-wise ranking approaches.

Table 3: Performance evaluation of the question recommendation task for the different methods considered. The best score for each evaluation metric is highlighted in boldface.

#Topics	Method	MRR	MAP	P@1	P@5	P@10	nDCG@1	nDCG@5	nDCG@10
-	PR	0.0679	0.0597	0.0320	0.0269	0.0196	0.0304	0.0612	0.0757
-	BOW	0.0946	0.0908	0.0502	0.0316	0.0265	0.0433	0.0775	0.1030
100	LSI	0.0358	0.0350	0.0124	0.0123	0.0102	0.0106	0.0265	0.0359
500		0.0610	0.0598	0.0292	0.0202	0.0163	0.0292	0.0486	0.0629
1000		0.0713	0.0694	0.0382	0.0238	0.0197	0.0313	0.0572	0.0766
5	LDA-R	0.1060	0.0946	0.0654	0.0326	0.0249	0.0611	0.0912	0.1123
	RankSLDA	0.1176	0.1054	0.0831	0.0338	0.0237	0.0767	0.1016	0.1180
10	LDA-R	0.1085	0.0979	0.0530	0.0385	0.0283	0.0485	0.0976	0.1208
	RankSLDA	0.1205	0.1073	0.0697	0.0391	0.0300	0.0652	0.1063	0.1325
20	LDA-R	0.1249	0.1100	0.0745	0.0406	0.0294	0.0696	0.1102	0.1328
	RankSLDA	0.1375	0.1214	0.0874	0.0437	0.0325	0.0816	0.1222	0.1483
50	LDA-R	0.1359	0.1198	0.0869	0.0464	0.0316	0.0799	0.1251	0.1453
	RankSLDA	**0.1403**	**0.1261**	**0.0917**	**0.0481**	**0.0340**	**0.0832**	**0.1312**	**0.1559**

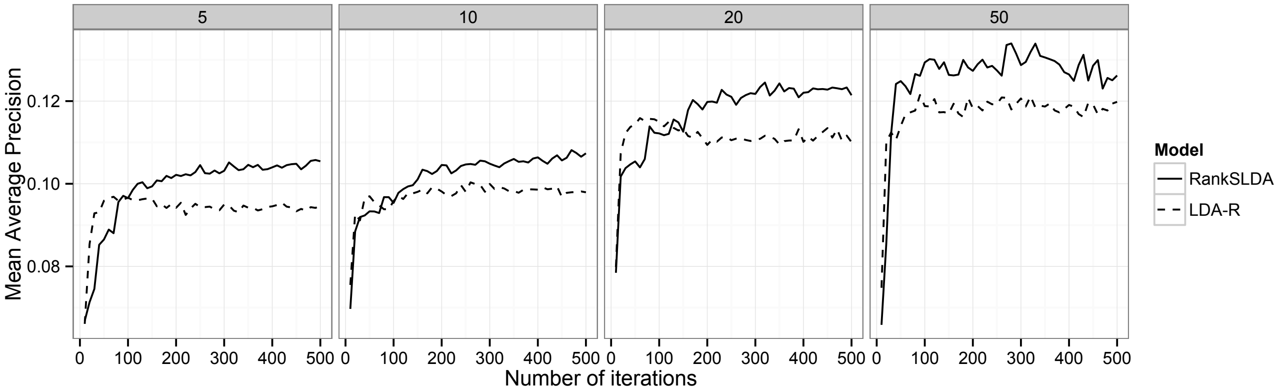

Figure 4: Evolution of MAP for *RankSLDA* vs LDA-R for different n. topics $K = \{5, 10, 20, 50\}$

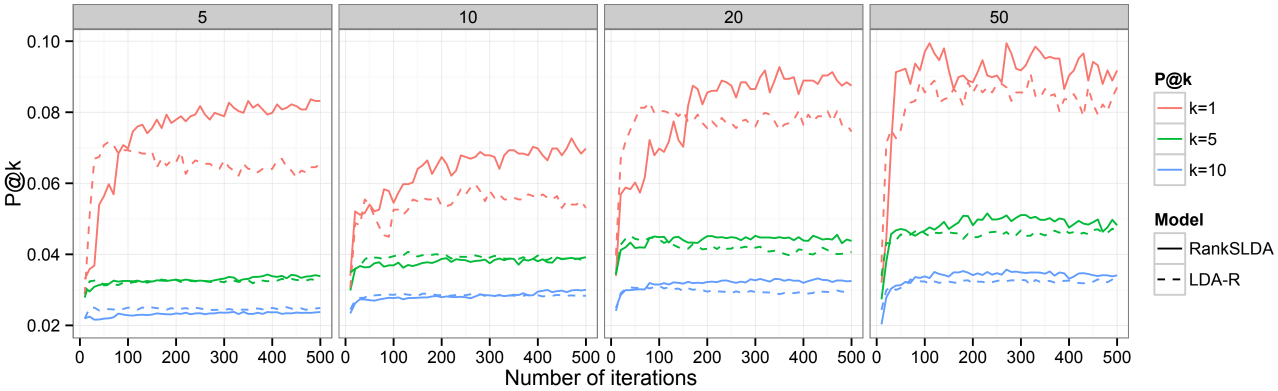

Figure 5: Evolution of P@k for *RankSLDA* vs LDA-R at levels for different n. topics $K = \{5, 10, 20, 50\}$

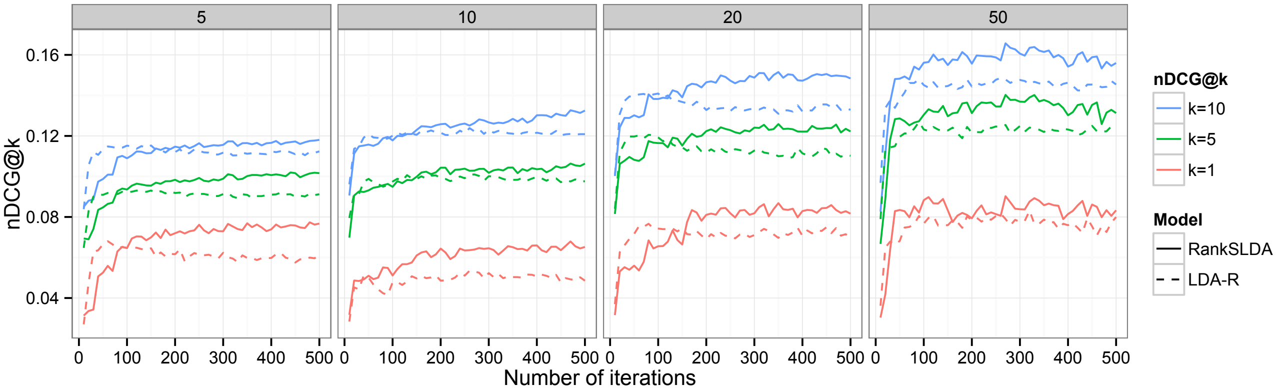

Figure 6: Evolution of nDCG@k for *RankSLDA* vs LDA-R at levels for different n. topics $K = \{5, 10, 20, 50\}$

7. ACKNOWLEDGEMENTS

The work leading to these results has received partial funding from the EU 7th Framework Programme (FP7/2007-2013) under grant agreement n^o 610594 (CrowdRec).

8. REFERENCES

[1] D. Agarwal and B.-C. Chen. flda: Matrix factorization through latent dirichlet allocation. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 91–100, 2010.

[2] c. Aslay, N. O'Hare, L. M. Aiello, and A. Jaimes. Competition-based Networks for Expert Finding. In *Proceedings of the 36th International ACM SIGIR Conference*, pages 1033–1036, 2013.

[3] D. M. Blei and J. D. McAuliffe. Supervised Topic Models, Mar. 2010.

[4] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, July 1997.

[5] B. C. Chen, A. Dasgupta, X. Wang, and J. Yang. Vote Calibration in Community Question-answering Systems. In *Proceedings of the 35th International ACM SIGIR Conference*, pages 781–790. ACM, 2012.

[6] G. Dror, Y. Koren, Y. Maarek, and I. Szpektor. I Want to Answer; Who Has a Question?: Yahoo! Answers Recommender System. In *Proceedings of the 17th ACM SIGKDD Conference*, pages 1109–1117. ACM, 2011.

[7] J. B. Graber and P. Resnik. Holistic sentiment analysis across languages: multilingual supervised latent Dirichlet allocation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 45–55, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[8] J. Guo, S. Xu, S. Bao, and Y. Yu. Tapping on the Potential of Q&A Community by Recommending Answer Providers. In *Proceedings of the 17th ACM CIKM Conference*, pages 921–930, 2008.

[9] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug. 2009.

[10] J. Liu, Y. I. Song, and C. Y. Lin. Competition-based user expertise score estimation. In *Proceedings of the 34th international ACM SIGIR conference*, SIGIR '11, pages 425–434, 2011.

[11] X. Ni, Y. Lu, X. Quan, L. Wenyin, and B. Hua. User interest modeling and its application for question recommendation in user-interactive question answering systems. *Information Processing & Management*, 48(2):218–233, Mar. 2012.

[12] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, 2009.

[13] F. Riahi, Z. Zolaktaf, M. Shafiei, and E. Milios. Finding expert users in community question answering. In *Proceedings of the 21st WWW conference*, pages 791–798. ACM, 2012.

[14] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic. Climf: Learning to maximize reciprocal rank with collaborative less-is-more filtering. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 139–146, 2012.

[15] H. M. Wallach. *Structured Topic Models for Language*. PhD thesis, University of Cambridge, 2008.

[16] H. M. Wallach, D. M. Mimno, and A. Mccallum. Rethinking LDA: Why Priors Matter. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1973–1981. 2009.

[17] M. Weimer, A. Karatzoglou, Q. V. Le, and A. Smola. Maximum margin matrix factorization for collaborative ranking. *Advances in Neural Information Processing Systems, NIPS*, 2007.

[18] H. Wu, Y. Wang, and X. Cheng. Incremental Probabilistic Latent Semantic Analysis for Automatic Question Recommendation. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 99–106, New York, NY, USA, 2008. ACM.

[19] F. Xu, Z. Ji, and B. Wang. Dual role model for question recommendation in community question answering. In *Proceedings of the 35th international ACM SIGIR conference*, pages 771–780. ACM, 2012.

[20] Z. Yan and J. Zhou. A New Approach to Answerer Recommendation in Community Question Answering Services. In *34th European Conference on Advances in Information Retrieval*, pages 121–132, 2012.

[21] L. Yang, M. Qiu, S. Gottipati, F. Zhu, J. Jiang, H. Sun, and Z. Chen. CQArank: Jointly Model Topics and Expertise in Community Question Answering. In *Proceedings of the 22nd ACM International CIKM Conference*, pages 99–108. ACM, 2013.

[22] T. Yeh, B. White, J. San Pedro, B. Katz, and L. S. Davis. A case for query by image and text content: Searching computer help using screenshots and keywords. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 775–784. ACM, 2011.

[23] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, Feb. 2006.

[24] J. Zhang, M. S. Ackerman, and L. Adamic. Expertise networks in online communities: structure and algorithms. In *16th International Conference on WWW*. ACM, 2007.