Content Driven User Profiling for Comment-Worthy Recommendations of News and Blog Articles

Trapit Bansal
trapitbansal@gmail.com

Mrinal Das
mrinal@csa.iisc.ernet.in

Chiranjib Bhattacharyya
chiru@csa.iisc.ernet.in

Department of Computer Science and Automation
Indian Institute of Science, Bangalore

ABSTRACT

We consider the problem of recommending *comment-worthy* articles such as news and blog-posts. An article is defined to be comment-worthy for a particular user if that user is interested to leave a comment on it. We note that recommending comment-worthy articles calls for elicitation of *commenting-interests* of the user from the *content* of both the articles and the past comments made by users. We thus propose to develop *content-driven* user profiles to elicit these latent interests of users in commenting and use them to recommend articles for future commenting. The difficulty of modeling comment content and the varied nature of users' commenting interests make the problem technically challenging.

The problem of recommending comment-worthy articles is resolved by leveraging article and comment content through topic modeling and the co-commenting pattern of users through collaborative filtering, combined within a novel hierarchical Bayesian modeling approach. Our solution, Collaborative Correspondence Topic Models (CCTM), generates user profiles which are leveraged to provide a personalized ranking of *comment-worthy* articles for each user. Through these *content*-driven user profiles, CCTM effectively handle the ubiquitous problem of *cold-start* without relying on additional meta-data. The inference problem for the model is intractable with no off-the-shelf solution and we develop an efficient Monte Carlo EM algorithm. CCTM is evaluated on three real world data-sets, crawled from two *blogs*, ArsTechnica (AT) Gadgets (102,087 comments) and AT-Science (71,640 comments), and a *news* site, DailyMail (33,500 comments). We show average improvement of **14%** (warm-start) and **18%** (cold-start) in AUC, and **80%** (warm-start) and **250%** (cold-start) in Hit-Rank@5, over state of the art [1, 2].

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Probabilistic algorithms; H.4.m [**Information Systems Applications**]: Miscellaneous

Keywords

Comments; User Profiling; News; Blogs; Topic Modeling; Collaborative Filtering; Hybrid Recommendation Systems

RecSys'15, September 16–20, 2015, Vienna, Austria.

DOI: http://dx.doi.org/10.1145/2792838.2800186.

1. INTRODUCTION

Online news and blog portals have emerged as convenient tools to gather information and exchange thoughts. Facility of commenting on such news and blog sites have raised the experience of users and, through active user-engagement, play a pivotal role in the hosting site's popularity [1]. Comments have thus garnered much research interest, ranging from detecting spam [3] to summarization [4], ranking [5] and retrieval [6]. Instead of studying comments, in this paper we explore the key question of what stimulates a user to make a comment on such online media. We demonstrate that *content* can reveal a significant amount of knowledge about users' commenting interests which can be leveraged to elicit further commenting through specific and personalized recommendations of *comment-worthy* articles. This is a form of recommendation aimed at enhancing user engagement through commenting.

Consider a blog post reviewing *Samsung Galaxy S5*, dated Feb 25, 2014[1], with over 160 comments, shown in Fig. 1. The review discusses various features in different *segments* like the *cover material* (purple colored), the new features of *fingerprint scanner* (colored red), *heartbeat reader* (colored green), the *MicroUSB* support (colored blue), etc. Notice that some users commented almost entirety on the new feature of heartbeat scanner, like Carl (*..An inconspicuous way to check your pulse..*, colored green); while some commented on the *MicroUSB* aspect, like Bob (*..I do hope USB 3.0 C comes out soon..*, colored blue). In the same article, some users commented on the story solely for the purpose of replying to an existing comment, like Mark who replied to users Alice and Carl (*..it will only measure your pulse rate..*, colored green). This demonstrates the varied nature of *specific* interests of users in commenting.

Investigating the user-comments of such articles, we observe that a user can comment on an article mainly for four reasons, if he is interested in (1) the content of the article, (2) content of other users' comments on the article, (3) content of a specific part of the article or (4) if users with similar interests have commented on the article. While user-interest in existing comments and users is a well-known phenomenon which is realized in the form of explicit *reply-to* feature on most sites, the concept of commenting-interest in *specific* parts of articles is a recent finding [7]. Indeed, realizing the importance of this very behavior, recent blogging-platforms like Medium[2] and WordPress[2] have started to offer per-paragraph commenting facility, to wide-spread popularity.

[1] ArsTechnica Gadgets: http://tinyurl.com/ktlyv6x
[2] medium.com; wordpress.org/plugins/inline-comments

Explicitly modeling users' commenting interests is the key component in making comment-worthy recommendations. Realize that without modeling all the above interests explicitly, the user-article indicator matrix of comments is not fully indicative of commenting interest. Thus, traditional collaborative filtering (CF) methods [8, 9] used to recommend articles for viewing [10] are not very effective in recommending articles to users for commenting. Moreover, such CF methods [8, 9] are known to suffer from the problem of item *cold-start*, new content on which no user has commented. On the other hand, state of the art hybrid models like collaborative topic regression [2], do not model *comment-content* and the varied user interests, giving unsatisfactory performance. Recently, an attempt was made to recommend news articles to users *for commenting* [1]. However, due to the inability to leverage article and comment content, the approach is unable to distinguish users' specific interests in commenting, leading to sub-optimal results. Moreover, the reliance of the approach on less informative meta-data, like tags, causes unsatisfactory performance specially in cold-start scenario which is ubiquitous in the realm of online media.

Contributions.

We explore the role of content in recommending *comment-worthy* articles and propose content-driven user profiling aimed at elicitation of users' commenting interests. We identify that user profiles should depend on content of articles on which user previously commented, content of user's previous comments and the co-commenting pattern of users. The resulting problem (Section 2) turns out to be an instance of a correspondence problem between the article and comment content, and a collaborative filtering problem of finding co-commenting patterns. To this end, we propose (Section 3) a novel hierarchical Bayesian model, namely *Collaborative Correspondence Topic Models* (CCTM), which solve the problem by bringing together topic modeling [11], collaborative filtering [8] and Bayesian personalized ranking [9].

To tackle the challenge of modeling comment-content, we use the recently introduced concept of *multiple topic vectors* [7], to discover user interest in *specific* article segments and create topic profiles of users from previous *comment-content*. By associating each user and each *segment* of article with *latent offsets*, we show how the topic profiles can be leveraged to model users commenting interests through a Bayesian personalized ranking approach [9]. Through these content-driven profiles, CCTM naturally handles cold-start problem in recommendation without relying on meta-data.

The resulting inference problem becomes non-standard due to dependency among several variables introduced through the three different modeling components and there are no off-the-shelf solutions. We develop (Section 4) an efficient stochastic Monte Carlo expectation maximization (MCEM) inference algorithm for CCTM. CCTM is used to generate article recommendations for future commenting and is rigorously evaluated (Section 5) on *three real datasets*, consisting of crawled copies of 1 popular News sites and 2 popular Blogs. We find that, on average, CCTM achieves **16%** improvement in AUC and **165%** improvement in Hit-Rank@5 over state-of-the-art recommendation systems [1, 2].

Notation.

K is the number of topics and V is number of words in vocabulary. β_k is a V-dimension vector such that $\sum_{j=1}^{V} \beta_{kj} = 1$, popularly called as a "*topic*". $x^T y$ is dot product of vectors x and y. *Dir* denotes Dirichlet distribution, *Ber* denotes Bernoulli distribution, and $\mathcal{N}$ denotes K dimension multivariate normal distribution. $[n] = \{1, 2, \ldots, n\}$ and $|R|$: cardinality of set R. $I[.]$ is the indicator function, $\sim$ means "distributed as" and 1_n is a n-dimension vector of ones.

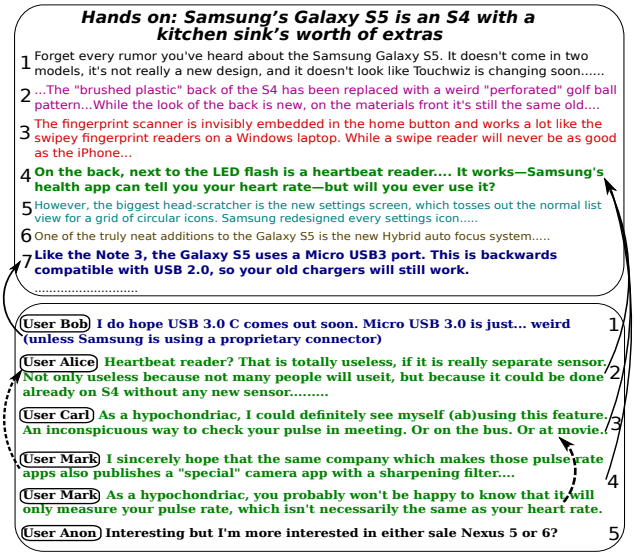

Figure 1: Examples of various commenting interests of users. Article segments and comments are color coded so that they correspond to same topic. Dashed arrow indicates a reply-to comment.

2. THE PROBLEM OF RECOMMENDING COMMENT-WORTHY ARTICLES

We begin by introducing relevant notation to describe the data. An article A_d, indexed by d or $j \in [D]$, is composed of S_d segments (sentences or paragraphs), $A_d = \{s_{da} | a \in [S_d]\}$, with each segment being a bag-of-words $s_{da} = \{w_{dan} | n \in [N_{da}]\}$. Each article has a set of C_d comments denoted by $E_d = \{c_{de} \mid e \in [C_d]\}$, with comment $c_{de} = \{w'_{dem} | m \in [n_{de}]\}$ where w'_{dem} is the mth word of comment e on article d. Furthermore $w_{dan}, w'_{dem} \in [V]$, where V is vocabulary size. The unique set of U users is indexed by user-id $i \in [U]$. For each user i, $A_i^+ \subseteq [D]$ denotes the set of articles i commented on and $A_i^- \subseteq [D]$ denotes the set he did not comment on.

Commenting interests of users.

Analyzing commenting interests of users is a crucial step in achieving our goal. Commenting interest of user i can be on three aspects; (1) general content of article A_d, (2) a *specific* segment s_{da} ($a \in [S_d]$) or (3) a *subset* of existing comments ($\subseteq [C_d]$) and their corresponding users. The set $R_{di} \subseteq [S_d] \cup [C_d]$, for each user i, will be denoted as the set of his *commenting interests* in article d. If user i makes a comment being interested on any component of R_{di}, we say that user i has made a comment on article A_d.

Problem formulation.

Given article-comment pairs $\{(A_d, E_d)\}_{d=1}^{D}$, classify for every user $i \in [U]$, whether i will comment on A_d, by finding the set R_{di} of commenting interests for user i in article d.

As a glimpse of things to come, the model we propose will associate with each potential element of R_{di} a real-valued score for each user, which is then combined to give an overall *commenting-interest score* of that article for the user.

Key challenges.

The task of recommending comment-worthy articles is tightly bound with finding commenting interests. We describe some key challenges below.

(1) Inapplicability of supervised approaches. Explicit supervision about R_{di} is generally unavailable for the user's prior commenting history and creating such labeled data is costly, enforcing an unsupervised approach.

(2) Low correlation between user's interest and article's main topic. A user may be interested in a specific part of an article (see Fig. 1) denoted by R_{di}. The proportion over topics for R_{di} can be very different from that of the entire article and size of R_{di} can be as small as a sentence. That makes correlation between user's interest in the article and main content of the article very low, enhancing the difficulty for statistical models.

(3) Low correlation between user interest and majority of the comments. Most of the articles in a popular site will receive comments of varied topics. Although many of the comments will focus on main topic of the article, but a significant number of comments diverge from that [7], and a user may be interested in comments which are on different topic than most of the comments.

(4) Comments are short and diverse. A key observation we make here is that the information of a user's specific interests is hidden in his prior comment *content*, as can be seen in Fig. 1. However, simple textual overlap is unsuitable to discover this relation. Moreover, due to the short and noisy nature of comments, modeling comment-content is challenging and approaches often rely on external text enrichment [4, 12], which is impractical for ever-increasing datasets.

(5) Cold-Start conditions. New online articles are generated at a very rapid pace, leading to the problem referred to as *cold-start*. That is, finding user interest in fresh articles with no existing comments ($|C_d| = 0$). This prohibits use of vanilla matrix factorization [8, 9].

2.1 Related work

Our approach lies in the literature on correspondence topic models and recommendation systems. CorrLDA [11] was the first topic model to model correspondence between images and their annotations. Within user modeling, [13] apply the LDA model on user's view logs to predict which stories a user will view. [14] developed topic models to profile experts in community Q&A. Outside topic modeling, [6] developed language models incorporating comment content for retrieving related news stories, but ignores user-information and personalization. None of these [6, 13, 14] model commenting behavior or recommend articles for commenting.

Collaborative filtering (CF) is an active area of research with rich literature [15, 16]. The most common example is probabilistic matrix factorization (PMF) [8], which analyzes interdependencies between items and users. However, CF approaches are prone to the problem of *cold-start* which has led to research in *hybrid* CF methods. [5] developed a hybrid regression based latent factor model to rank comments on news articles, by leveraging meta-data. Recent research in hybrid CF methods, like collaborative topic regression (CTR)[2], has combined LDA and PMF for generating article recommendations, by leveraging item content. Quite recently, [17] showed the advantage of using articles in users' libraries to generate relevant recommendations of scientific articles. These methods [2, 17] cannot model comment content and are unsuitable for the given task.

More germane to our work is the recent study of [1], who look at recommending news articles for commenting. They use a CF approach relying crucially on article *tags* as meta-data to handle cold-start. This has the following problems: *a)* Extensive editorial effort is required to ensure fine-grained tags with every new article. Indeed, the datasets that we crawled (5.1), only have generic *categories* like *Tennis*, making it impossible for the model to distinguish user's commenting interest on any two new articles on Tennis. *b)* It ignores both article and comment content, making it impossible to analyze commenting interests of users. As opposed to this, our model does not rely on meta-data by modeling article and comment content.

3. COLLABORATIVE CORRESPONDENCE TOPIC MODELS

In order to solve the problem of recommending comment-worthy articles, we propose in this section a novel hierarchical Bayesian modeling approach, namely *collaborative correspondence topic models* (CCTM). Full generative process is given in Fig. 2. We describe the modeling details below.

3.1 Modeling principle

Our objective in this paper is to analyze specific reasons behind commenting activity of users and apply that suitably to recommend articles for further commenting. Our approach is based on four key steps; (1) create *topic profiles*, (2) combine topic profile with *latent offsets*, (3) quantify commenting interests and (4) finally rank the preferences. CCTM thus brings in specific correspondence modeling (step 1), collaborative filtering (step 2 and 3) and Bayesian personalized ranking (step 4) together. We describe the steps below.

3.2 Modeling specific correspondence to capture topic profiles

As evinced by the example of Fig. 1, articles cover multiple topics in different segments and comment content can be related to such very specific segments, which in addition may not be contiguous. Due to this fact, proportion over topics should vary across the segments within an article and CorrLDA [11] fails to capture this aspect. We resort to the concept of multiple topic distributions (topic vectors) [7].

3.2.1 Multiple topic vectors (MTV) to vary proportion over topics

For every article, there are J_d topic vectors $\{\theta_{dt}\}$, representing its varied themes. For each word of a segment, a topic vector θ_{dt} is sampled from a multinomial ρ, and the topic assignment of the word is then sampled from θ_{dt}. Contrast this to CorrLDA, where a *single* θ_d is fixed for the article and any random segment has same distribution as the entire article (in expectation), whereas MTV allows it to be significantly different. Following [7], stick-breaking process (SBP) [18] is used as prior for ρ. MTV thus allows us to model low correlation in topic of a comment with an article but high correlation with topic of a *specific* article segment.

- For $k \in [K]$, sample topic $\beta_k \sim Dir(\eta\ 1_V)$
- For each user $i \in [U]$,
  - Sample $\vartheta_i \sim Dir(\alpha_u\ 1_K)$, $\epsilon_i \sim Beta(\lambda_1, \lambda_2)$
- For each article-comment pair, $d \in [D]$
  - For $t \in [J_d]$, draw topic vectors $\theta_{dt} \sim Dir(\alpha\ 1_K)$
  - For each article segment, $a \in [S_d]$
    - Sample $\rho_{da} \sim SBP(\tau, \iota)$
    - For each word $n \in [N_{da}]$,
      - Sample topic, $z_{dan} \sim \theta_{db_{dan}}$, $b_{dan} \sim \rho_{da}$
      - Sample word, $w_{dan} \sim \beta_{z_{dan}}$
    - Set topic profile $\tilde{s}_{da}$, $\tilde{s}_{dak} = \frac{|\{z_{dan}=k,\ n \in N_{da}\}}{N_{da}}|$
  - For each comment $e \in [C_d]$ by user $i \in [U]$
    - For each segment a, selector $\xi_{dea} \sim Ber(\pi_{de})$
    - Set $\varphi_{dek} = \frac{\#\{z_{dan}=k\ \forall (a,n) | \xi_{dea}=1\}}{\sum_{a=1}^{S_d} \xi_{dea} N_{da}}$
    - For each comment word $m \in [n_{de}]$,
      - Sample topic $y_{dem} \sim \epsilon_{de}\varphi_{de} + (1-\epsilon_{de})\vartheta_{de}$
      - Sample word, $w'_{dem} \sim \beta_{y_{dem}}$
    - Set topic profile $\tilde{c}_{de}$, $\tilde{c}_{dek} = \frac{|\{y_{dem}=k,\ m \in n_{de}\}|}{n_{de}}$
- For articles $d \in [D]$, sample $v_{d0} \sim \mathcal{N}(0, \lambda_v^{-1} I)$
  - For segment $a \in [S_d]$, sample $v_{da} \sim \mathcal{N}(0, \lambda_s^{-1} I)$,
- For users $i \in [U]$,
  - Set topic profile $\tilde{q}_i$, $\tilde{q}_{ik} = \frac{\sum_{(d,e) \in A_i^+} n_{de} c_{dek}}{\sum_{(d,e) \in A_i^+} n_{de}}$
  - Sample $u_i \sim \mathcal{N}(0, \lambda_q^{-1} I)$
- For each user-article pair, $(i,d) \in [U] \times [D]$,
  - Commenting interest, $r_{id} = r_{id}^{mf} + r_{id}^{art} + r_{id}^{cmnt}$,
    $r_{id}^{mf} = h_d + (\tilde{q}_i + u_i)^T v_{d0}$
    $r_{id}^{art} = \max_{1 \le a \le S_d} \{(\tilde{q}_i + u_i)^T (\tilde{s}_{da} + v_{da})\}$
    $r_{id}^{cmnt} = \sum_{e=1}^{C_d} (\tilde{q}_i + u_i)^T (\tilde{c}_{de} + u_{de}) p_{ide}$, where
    $p_{ide} = softmax\{(\tilde{q}_i + u_i)^T (\tilde{q}_{de} + u_{de})\}$

Figure 2: Generative Process of CCTM

3.2.2 Generating comment content

To relate comment content to specific segments of the article, topic assignment of comment words is sampled from subset of segments, rather than uniformly from entire article like CorrLDA. For every comment, the subset of article-segments is sampled through selector variable ξ. Then the topic assignment of each word in comment e is generated from a mixture distribution $\epsilon_{de}\varphi_{de} + (1-\epsilon_{de})\vartheta_{de}$; where φ_{de} is the uniform distribution over selected segments and ϵ_{de} is user's propensity to select from the article d or his own interests[3], ϑ_i. To capture commenting interests of users, each user is associated with topic interests defined by a distribution over topics, ϑ_i. ϑ_i relates user i's comment content across his commenting history and allows comment-topics to vary from the article distribution based on the user's other interests. Through this mechanism, we allow comments to exhibit content which is better modeled by the user's diverging interests from the current article. This is essential for modeling all the varied types of commenting activity and not ignore a substantial amount of comment content.

[3]we index user variables by comment-id for simplicity

3.2.3 Creating topic profiles

After modeling the correspondence between article and comments, we create topic profiles. For each user i, we construct an empirical topic distribution $\tilde{q}_i$ from the topic assignment of all his prior comments which can be considered a summary of user's topic interests. Similarly, each article d's segment a and comment e get empirical topic distributions $\tilde{s}_{da}$ and $\tilde{c}_{de}$, respectively. These *topic profiles* will now be refined *collaboratively* to model users' commenting interests.

3.3 Complementing topic profiles with latent offsets

To model users' commenting interests, we introduce *latent offsets* to the topic profiles, much in the vein of [2, 19].

3.3.1 Latent offsets to model commenting interests

We introduce latent offsets for each *user i (u_i), each segment $a \in [S_d]$ (v_{da}) of article d* and a global offset for each article d (v_{d0}), in order to model the varied commenting interests. Following PMF approach [8], the latent offsets are drawn from zero-mean K-dimensional Gaussian distribution. These offsets will allow the topic profiles to change according to observed commenting interests.

Intuition behind latent offsets.

Suppose that the user *Mark* (in Fig. 1) has till now commented on *mobile apps* which gets reflected in his topic-profile. However, now his comments on the related topic of *health* app (green article-segment), cannot be explained by the content-model alone, due to lack of his prior comment-content on this topic. The latent offset will model this behavior by increasing the value for the *health* topic in the user's offset and *mobile app* topic in the corresponding segment's offset, due to the commenting activity of similar users like *Alice* who were also interested in mobile apps but also commented on health topic in this article and other articles.

The latent offsets, thus, will serve the purpose of allowing the topic profiles to deviate according to observed commenting patterns. The offsets are learned from the overall commenting activity, so the more comments users make, the better idea the model gets of the value of the offsets.

3.3.2 Content-driven user profiles

Latent offsets and topic profiles will be combined to give an overall *commenting-interest score*, $\hat{r}_{id}$, for each user i on article d, by taking into account interest in each element of R_{di}. To this end, we create content-driven user profiles, by combining topic profiles of users (content information) with the latent offset (co-commenting information) to obtain user i's profile, $(\tilde{q}_i + u_i)$. Following Bayesian approach, by associating (zero mean) latent offsets with topic profiles, we ensure good back-off estimates to the topic profile in sparse commenting scenarios. Similar profiles are considered for article segments and comments in the following section.

3.4 Quantifying commenting interests

A user can comment on the main article, a specific segment of the article or on some existing comments. We model interest in these aspects using scores r^{mf}, r^{art} and r^{cmnt}, respectively. The idea behind computing these scores is that, the closer the user's profile is to the profiles of these three aspects, the higher his interest is in that specific aspect.

3.4.1 Interest in article popularity

Users like *Anon* in Fig. 1 are interested in the main topic or popularity of article. Similar to PMF[8], we capture this:

$$r_{id}^{mf} = h_d + (\tilde{q}_i + u_i)^T v_{d0} \quad (1)$$

where $h_d \sim \mathcal{N}(0, \lambda_h^{-1})$ is a popularity bias and v_{d0} is a global latent offset for article. Notice that unlike vanilla PMF (which would consider $u_i^T v_{d0}$), we use the complete user profile $(\tilde{q}_i + u_i)$ which allows the model to ensure that user's interest do not deviate much from his previous content interests. For a new article with no existing comments (i.e. cold-start), there is no contribution from this term.

3.4.2 Interest in article segments

The *segment's* latent profile is considered as $(\tilde{s}_{da} + v_{da})$, where $\tilde{s}_{da}$ is the topic-distribution of the segment. This profile is expected to capture the topic-level popularity of this segment among users. We thus set $r_{ida}^{art} = (\tilde{q}_i + u_i)^T(\tilde{s}_{da} + v_{da})$, for $a \in [S_d]$. Now considering the maximum value we get the score for the segment with highest interest as below.

$$r_{id}^{art} = \max_{1 \le a \le S_d} \{(\tilde{q}_i + u_i)^T(\tilde{s}_{da} + v_{da})\} \quad (2)$$

3.4.3 Interest in existing comments and users

Consider a latent profile of existing comment e on A_d to be $(\tilde{c}_{de} + u_{de})$, where $\tilde{c}_{de}$ is the topic distribution of the comment and u_{de} is the latent offset associated with user of this comment. We thus consider the interest in comment e as $r_{ide}^c = (\tilde{q}_i + u_i)^T(\tilde{c}_{de} + u_{de})$. Note that $\tilde{c}_{de}$ is comment-specific but u_{de} is a global parameter for the user, avoiding explosion of the number of parameters to be learned (as quantity of comments can be high).

Main challenge in accounting for interest in $[C_d]$ is that users often don't respect the reply-to relations while commenting [12]. Moreover, such relations are unavailable for articles on which user has not yet commented. To model this, let γ_{ide} denote a Bernoulli random variable, with $\gamma_{ide} = 1$ if user i's comment on d is a *reply-to* comment to the comment e. Thus, the expected overall interest in existing comments:

$$r_{id}^{cmnt} = \mathbb{E}_{\gamma|u,\tilde{c}}\left[\sum_{e=1}^{C_d} r_{ide}^c I(\gamma_{ide} = 1)\right] = \sum_{e=1}^{C_d} r_{ide}^c p_{ide} \quad (3)$$

where $p_{ide} = P(\gamma_{ide} = 1|u_i, u_{de}, \tilde{c}_{de})$. The case of observed reply-to relations is trivial with $p_{ide} \propto 1$ for only the observed relations. Here we model the user as equally interested in all the users he replied to. For unobserved relations and uncommented articles, we use the intuition that a user i is interested in other comment if the user profiles are similar. Thus, for this case we consider $p_{ide} \sim softmax\{(\tilde{q}_i + u_i)^T(\tilde{q}_{de} + u_{de})\}$. Equation (3) and probability model p_{ide}, define the comment-interest part of the rating, r_{id}^{cmnt}. Apart from modeling *content*-relatedness, the cross-term of the latent offsets $(u_i^T u_{de})$ models user similarity. Also note that r_{id}^{cmnt} causes the user's interest in an article d to evolve with new comments received on d.

3.4.4 Overall commenting interests of users

We take into account all the potential user-interests to consider an overall interest score of user i in article d:

$$\hat{r}_{id} = r_{id}^{mf} + r_{id}^{art} + r_{id}^{cmnt} \quad (4)$$

where r_{id}^{mf} (1) is a popularity term for user interest in the topic popularity of the article (only for warm-start), r_{id}^{art} (2) accounts for the interest in specific segments of the article, and r_{id}^{cmnt} (3) accounts for the interest in the existing comment-content and co-commenters.

3.5 Ranking commenting preferences

With the interest score defined, we need to model users' personalized preferences for commenting. Note that the dataset consists of only positive item feedback, that is we only have access to information of which articles the user was interested in commenting on. This is a much harder problem of implicit feedback [9, 20]. We take recourse to Bayesian Personalized Ranking (BPR) [9, 21] which provides a Bayesian model for learning a personalized total ranking for each user. Given the predicted scores $\hat{r}_{id}$ and $\hat{r}_{ij}$, the probability of user i preferring article d over article j is $\sigma(\hat{r}_{id} - \hat{r}_{ij})$; where σ is the sigmoid function, $\sigma(x) = \frac{1}{1+e^{-x}}$.

4. INFERENCE AND PREDICTION

Our objective is to develop an efficient inference procedure to learn the model's latent offsets (denoted by Θ) and the latent content variables associated with articles, comments and users (denoted by Ω). The task is thus to maximize the log-posterior of the model parameters $p(\Theta, \Omega|R, W)$, where $R = \{(i,d)|i \in [U], d \in A_i^+\}$ is the observed commenting pattern and W are the words of the articles and comments. Inference is intractable and we resort to a stochastic MCEM algorithm [22], an inference method that alternates between Gibbs sampling for content variables Ω (keeping Θ fixed) and gradient ascent to estimate latent offsets Θ (Ω fixed).

4.1 Sampling content variables (E-step)

We develop an efficient *collapsed Gibbs sampling* inference for sampling the content-variables. The real-valued random variables $\beta, \vartheta, \epsilon, \theta, \rho, \pi$ (Fig. 2) are marginalized out, and only discrete variables b, z, y, and ξ are inferred, leading to accelerated convergence. Sampling of y requires introducing an auxiliary binary variable κ, with $\kappa = 0$ if comment-word topic is sampled from article-segments ϕ and $\kappa = 1$ if sampled from the user's topic vector ϑ. Note that unlike [2, 19], who add latent offsets to the document topic distributions θ, we add latent offset to *empirical* topic distributions $(\tilde{s}, \tilde{q})$, allowing us to collapse θ and ϑ, reducing the number of variables to be sampled. Derivation of conditional distributions is now conventional and we omit this due to space constraints. Refer to supplementary for more details.

4.2 Estimating latent offsets (M-step)

We seek to optimize the log-posterior of latent offsets given observed user preferences R and the content variables, that is: $\ln p(\Theta|R, \Omega) = \sum_{i \in U} \sum_{d \in A_i^+} \sum_{j \in A_i^-} \ln \sigma(\hat{r}_{id} - \hat{r}_{ij}) + \ln p(\Theta|\Omega)$. This requires computing $U \times D^2$ terms for the gradient, which is computationally infeasible. BPR [9], thus

estimates the model parameters Θ by stochastic gradient ascent (SGA). In each step, a user i and a commented article d are sampled uniformly from R, an uncommented article j is sampled from A_i^- and a gradient step with respect to the associated terms in the log-posterior is performed.

However, a key challenge here is that $\hat{r}_{id}$ is *not differentiable* due to the use of the *max* function in the r_{id}^{art} (2), which explains users' interest in specific segments. We overcome this by a smooth approximation of the max function:

$$\underset{1\le k\le K}{diffmax}\{a_k\} = \frac{1}{\sum_k e^{\psi a_k}} \sum_k a_k e^{\psi a_k} \tag{5}$$

for some parameter $\psi \geq 0$. This is different from softmax but can be viewed as a weighted sum of the values a_k with weights given by the $softmax\{a_k\}$.
Note that $\lim_{\psi\to\infty} \underset{1\le k\le K}{diffmax}\{a_k\} \equiv \underset{1\le k\le K}{max}\{a_k\}$, and $\psi = 0$ corresponds to average of $\{a_k\}$.

Using *diffmax* allows us to carry out an efficient SGA algorithm. The computation of the gradients is now straightforward, details can be found in the supplementary[4].

Before concluding this section, we remark on our choice of BPR algorithm. Apart from its success for implicit feedback [21–23], we choose BPR as apposed to alternatives like weighted regularized matrix factorization (WRMF) [20], also used by CTR [2], due to practical considerations. The non-quadratic form of $\hat{r}$ (4), makes WRMF impractical, as an alternating least squares method like [2, 20] cannot be derived. Moreover, taking an SGA approach in WRMF is inferior to the SGA approach on BPR criterion which works with item-pair level as opposed to individual item level [9].

4.3 Predicting commenting interests

Given the learned latent offsets and topic profiles of articles, segments and comments, the predicted user interest in commenting on an article is given by equation (4). For an article with no comments, that is *cold-start*, this value is:

$$\mathbb{E}[r_{id}] = \mathbb{E}[r_{id}^{art}] \approx \underset{1\le a\le S_d}{max}\{(\tilde{q}_i + u_i)^T(\tilde{s}_{da})\} \tag{6}$$

5. EMPIRICAL EVALUATION

In this section, we evaluate the proposed model CCTM rigorously on three real datasets. We first show that CCTM is a good model of article-comment correspondence. Then we evaluate the main task of this paper, recommending articles to users *for commenting*.

5.1 Datasets

We crawled[4] live news and blog sites to collect article content, corresponding comments, user information of comments and reply-to relations (where available).

ATScience: We crawled 3 years of blog articles, from April 2011 to April 2014, from Science section of popular blog ArsTechnica[5]. This consists of 71,640 comments by 3,581 users on 2,500 articles.

ATGadgets: We crawled 102,087 comments by 4,872 users on 3,000 articles from Gadgets section of the site ArsTechnica[5], from June 2012 to April 2014.

DailyMail: We Crawled 33,468 comments by 2,534 users on 3,000 articles from Sports section of this popular news site[6], going chronologically backwards from 1 July 2013. This crawl of the dataset did not have reply-to relations.

5.2 Experimental setup

5.2.1 Baselines

We consider three baselines: *a)* TagCF [1]: This is the state-of-art for recommending articles for commenting. It associates *tags* with latent factors and uses BPR criterion. This completely ignores article and comment content. *b)* Collaborative Topic Regression (**CTR**) [2]: State-of-the-art *hybrid* model which models the content of the articles along with the ratings. This approach ignores the content of the comments, but promises to handle *cold-start* by leveraging content of articles and does not rely on meta-data like tags. *c)* Content-Only Method (**CoTM**): This is the *content-only* part of our model (section 3.2), equivalent to setting the latent offsets $\{u_i, v_{d0}, v_{da}\} = 0$. We will refer to this as CoTM (Commenter Topic Model).

Improvement of CoTM over CTR will demonstrate the importance of modeling *comment content* and its correspondence to article content. Improvement of CCTM over both CoTM and CTR will demonstrate importance of modeling commenting interests explicitly. Note that all baselines ignore modeling of commenting interests of users.

5.2.2 Implementation details

We remove standard stop words and restrict vocabulary to $15k$ words using term-frequency. Articles and users with less than 2 comments were removed since they cannot be evaluated. We used uninformative hyperparameter values for the content variables[7]: $\alpha, \alpha_u, \eta = 0.1$; $\tau = 1$, $\iota = 0.1$; $\lambda_1, \lambda_2 = 1$; and $J_d = 5$. For all models, $K = 150$. The precision parameters for latent-offsets $\lambda_v, \lambda_s, \lambda_q, \lambda_h$, similar parameters for baselines, were tuned on the first fold of DailyMail and the same value used for all other folds and datasets. The *diffmax* parameter ψ was tuned same way and found to be $\psi = 10$. We observed the estimation procedure for CTR to be sensitive to initialization, and initialized CTR with the output from LDA. We initialize CCTM randomly. CCTM is trained by 1000 EM-iterations, with 1k SGA updates per iteration. Due to unavailability of tags on articles, for TagCF we follow the advice of [1] to extract entities. We extract 10k entities by frequency (more entities did not improve performance) and do 1000k SGA updates.

5.2.3 Methodology

We test the performance of all the models in both warm-start and cold-start scenarios, following the approach of [2].

Warm-Start: In this case every test article had at least one comment in training data. For each user we do a stratified 5-fold split of articles (both 1's and 0's). For each fold, we fit the models on training data and test on within-fold articles of each user (users have different sets of within-fold articles).

Cold-Start: This is the task of predicting user interest in commenting on a new article with no existing comments. Articles are split into 5 folds. For each fold in turn, we remove all comments on the articles in that fold forming the test-set and keep the other folds as training-set. The models are fitted on the training set and tested on within-fold articles (same for each users).

[4]Resources: http://mllab.csa.iisc.ernet.in/recsys15
[5]arstechnica.com/science; arstechnica.com/gadgets
[6]dailymail.co.uk/sport

Table 1: AUC for recommending comment-worthy articles (higher is better). CCTM outperforms CTR & TagCF in both warm (14% better) and cold start (18% better). † means statistical significance over baselines, at 1% using paired t-test. CoTM is a content-only restriction of proposed CCTM (see section 5.2.1 for details).

Dataset	Warm-Start				Cold-Start			
	TagCF	CTR	CoTM	CCTM	TagCF	CTR	CoTM	CCTM
DailyMail	0.723	0.739	0.767	**0.860**†	0.601	0.693	0.725	**0.751**†
ATScience	0.658	0.643	0.628	**0.723**†	0.555	0.572	0.620	**0.646**†
ATGadgets	0.635	0.636	0.615	**0.721**†	0.514	0.571	0.602	**0.622**†

5.2.4 *Evaluation*

We perform our evaluation broadly on two aspects. First we validate CCTM's ability to model correspondence between articles and comments. Then, we focus on the main task of the paper, to recommend comment-worthy articles. For the second task, each model is allowed to present (hypothetically) each user with a ranked list of articles for commenting in the test set. The quality of the ranking thus presented is evaluated for each user through AUC (Area Under ROC Curve) metric [9], using the articles in the held-out set that the user actually commented on. AUC $\in (0, 1]$, measures the probability that a randomly chosen article on which user commented is ranked higher than one he didn't comment on. Eventually we averaged the AUC score for each user to get an overall performance score.

5.3 Evaluating content correspondence

Ability to effectively model correspondence relationship between the content of an article and its comments is crucial in detecting commenting interests of users. We first evaluate the ability of CCTM to model this aspect. For this purpose, we randomly select 20% articles in each dataset as test articles and remove all comments on the articles. The model is then fitted on the remaining set of article-comments, and the perplexity [11] of comment words in test-set is evaluated. We compare with CorrLDA [11], which also models correspondence but ignores specificity and user information. CCTM achieved perplexity measure (lower is better) of **459** (DailyMail), **998** (AT-Science), **843** (ATGadgets); while CorrLDA achieved 647 (DailyMail), 1324 (ATScience), 1100 (ATGadgets). This shows that CCTM is a better model of comment-content and article-comment correspondence.

5.4 Evaluating recommendations

We now evaluate CCTM on the main objective of recommending *comment-worthy* articles. Table 1 shows the performance for both warm-start and cold-start conditions. Under warm-start condition, note that a content-only approach (CoTM) gives almost similar performance to TagCF and state-of-art hybrid method CTR. This shows the importance of leveraging prior *comment-content* and article content through topic profiles. CCTM, by modeling *commenting interests* of users through content-driven profiles, gives significant improvements over all of these methods on all the datasets, showing that modeling commenting interest of users explicitly is crucial to generate relevant recommendations for commenting.

Cold-start recommendation is a much harder problem, as evinced by the relatively lower AUC values of all models. As explained earlier, TagCF particularly suffers in cold-start, with performance only slightly better than random guessing. Whereas the content-only approach (CoTM) itself gives significant gains over CTR and TagCF. CoTM gives almost similar performance for both warm-start and cold-start which is expected as it is a content-only approach. Performance is further improved through CCTM which uses a latent offset to model user profile. Note, the only difference in this scenario between CoTM and CCTM is the presence of a latent-offset in user topic profile.

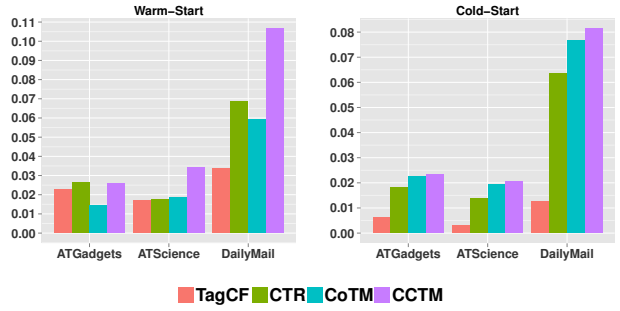

Figure 3: Average reciprocal Hit-Rank@5. CCTM is 165% better (average) than [1, 2]

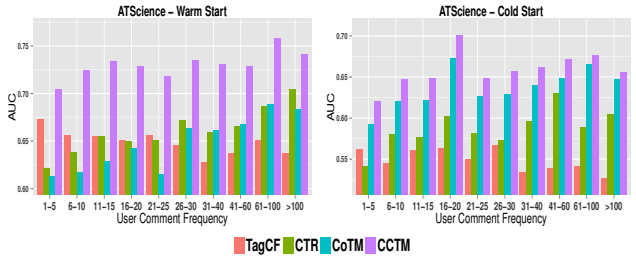

Figure 4: AUC by users' comment frequency.

5.5 Evaluating quality of ranking

To test which model ranks comment-worthy articles much higher in the ranked list of articles, we evaluate *average reciprocal hit-rank* (HR). Given a list of M ranked articles for user i with n_i test comments, let $c_1, c_2, \ldots, c_h$ denote the ranks of h articles in $[M]$ on which the user actually commented. HR is then defined as $\frac{1}{n_i}\sum_{t=1}^{h}\frac{1}{c_t}$ and tests whether top ranked articles are correct. Fig. 3 shows the results for both warm and cold start with $M = 5$, a realistic scenario where users can only be shown 5 articles. CCTM is significantly better than CTR and TagCF in both warm (**80%** better) and cold-start (**250%** better), establishing that relevant articles are ranked much higher. TagCF performs similar to CTR for warm-start but suffers severely in cold-start.

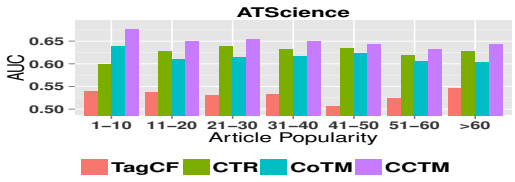

Figure 5: AUC by article's commenting popularity.

5.6 Evaluation in sparse commenting scenario

Ability of CCTM to model comment content and users latent interests should give CCTM advantage in extreme conditions such as when a user has made few comments or an article has received very few comments so far. We evaluate our approach on such cases here. In the following, we focus on ATScience, other results are similar (see supplementary).

Recommending to tail users.

In Fig. 4 we study performance by number of comments made by users in the training data. CCTM is superior in performance for all kinds of users and largest improvements are observed for tail-users, i.e. users with 1-5 comments in the training data. This shows that modeling commenting interests gives significant information about a user's commenting activity with as few as 5 comments made by the user.

Recommending tail articles.

We group articles by comment volume in training data (for warm-start) and evaluate AUC for each group. Average across these groups is the *stratified AUC* metric [1]. Fig. 5 shows the results. While CCTM is substantial superior throughout, largest improvements are observed for tail-articles, i.e. articles with 1-10 comments in the training data. This shows that modeling commenting interests gives significant information about users commenting activity with as few as 10 comments received on the article. CCTM is also substantially superior to the content-only model (CoTM) in this respect. CoTM, performs better than CTR for tail-items but worse for highly popular items. CTR's performance improves over CoTM for popular articles, showing that collaborative information dominates for high comment-volume, but CCTM is still significantly better than either.

6. CONCLUSIONS

We study a novel problem of eliciting *content-driven* user profiles to recommend articles which are *comment-worthy* of a particular user. We propose a novel hierarchical Bayesian model CCTM to solve this problem and demonstrate significant advancement in generating comment-worthy recommendations over state of art recommendation systems [1, 2], on three real life datasets using various metrics. There are many avenues for future work – incorporating comment sentiment, users' social information, modeling temporal nature of commenting preferences, and developing distributed and streaming inference [13] for web-scale deployment.

7. ACKNOWLEDGMENTS

We are thankful to all the reviewers for their valuable comments. The authors were partially supported by DST grant (DST/ECA/CB/1101).

References

[1] E. Shmueli, A. Kagian, Y. Koren, and R. Lempel. Care to comment?: Recommendations for commenting on news stories. In *WWW*, pages 429–438. ACM, 2012.

[2] C. Wang and D. Blei. Collaborative topic modeling for recommending scientific articles. In *SIGKDD*, pages 448–456. ACM, 2011.

[3] R. Kant, S. Sengamedu, and K. Kumar. Comment spam detection by sequence mining. In *WSDM*, pages 183–192. ACM, 2012.

[4] Z. Ma, A. Sun, Q. Yuan, and G. Cong. Topic-driven reader comments summarization. In *CIKM*, pages 265–274. ACM, 2012.

[5] D. Agarwal, B. Chen, and B. Pang. Personalized recommendation of user comments via factor models. In *EMNLP*, pages 571–582. ACL, 2011.

[6] Q. Li, J. Wang, Y. Chen, and Z. Lin. User comments for news recommendation in forum-based social media. *Information Sciences*, 180(24):4929–4939, 2010.

[7] M. Das, T. Bansal, and C. Bhattacharyya. Going beyond Corr-LDA for detecting specific comments on news & blogs. In *WSDM*, pages 483–492. ACM, 2014.

[8] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, 2007.

[9] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461. AUAI Press, 2009.

[10] J. Liu, P. Dolan, and E. Pedersen. Personalized news recommendation based on click behavior. In *IUI*, pages 31–40. ACM, 2010.

[11] D. Blei and M. Jordan. Modeling annotated data. In *SIGIR*, pages 127–134. ACM, 2003.

[12] J. Wang, C. Yu, P. Yu, B. Liu, and W. Meng. Diversionary comments under political blog posts. In *CIKM*, pages 1789–1793. ACM, 2012.

[13] A. Ahmed, Y. Low, M. Aly, V. Josifovski, and A. Smola. Scalable distributed inference of dynamic user interests for behavioral targeting. In *SIGKDD*, pages 114–122. ACM, 2011.

[14] T. Zhao, N. Bian, C. Li, and M. Li. Topic-level expert modeling in community question answering. In *SDM*, pages 776–784. SIAM, 2013.

[15] X. Su and T. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.

[16] F. Ricci, L. Rokach, and B. Shapira. *Introduction to recommender systems handbook*. Springer, 2011.

[17] L. Charlin, R. Zemel, and H. Larochelle. Leveraging user libraries to bootstrap collaborative filtering. In *SIGKDD*, pages 173–182. ACM, 2014.

[18] H. Ishwaran and L. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, 2001.

[19] W. Neiswanger, C. Wang, Q. Ho, and E. Xing. Modeling citation networks using latent random offsets. In *UAI*, 2014.

[20] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272. IEEE, 2008.

[21] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM*, pages 81–90. ACM, 2010.

[22] A. Ahmed, B. Kanagal, S. Pandey, V. Josifovski, L. Pueyo, and J. Yuan. Latent factor models with additive and hierarchically-smoothed user preferences. In *WSDM*, pages 385–394. ACM, 2013.

[23] L. Hong, A. Doumith, and B. Davison. Co-factorization machines: modeling user interests and predicting individual decisions in Twitter. In *WSDM*, pages 557–566. ACM, 2013.