Continuous Experience-aware Language Model

Subhabrata Mukherjee[†]   Stephan Günnemann[‡]   Gerhard Weikum[†]

[†]Max Planck Institute for Informatics, [‡]Technical University of Munich

smukherjee@mpi-inf.mpg.de, guennemann@in.tum.de, weikum@mpi-inf.mpg.de

Abstract

Online review communities are dynamic as users join and leave, adopt new vocabulary, and adapt to evolving trends. Recent work has shown that recommender systems benefit from explicit consideration of user experience. However, prior work assumes a fixed number of discrete experience levels, whereas in reality users gain experience and mature continuously over time.

This paper presents a new model that captures the *continuous evolution* of user experience, and the resulting *language model* in reviews and other posts. Our model is unsupervised and combines principles of Geometric Brownian Motion, Brownian Motion, and Latent Dirichlet Allocation to trace a smooth temporal progression of user experience and language model respectively. We develop practical algorithms for estimating the model parameters from data and for inference with our model (e.g., to recommend items). Extensive experiments with five real-world datasets show that our model not only fits data better than discrete-model baselines, but also outperforms state-of-the-art methods for predicting item ratings.

CCS Concepts

•**Information systems** → **Recommender systems;** •**Mathematics of computing** → **Dimensionality reduction;** •**Human-centered computing** → *Collaborative filtering; Social recommendation;*

Keywords

Review Community; User Experience; Language Evolution; Recommendation; Topic Modeling; Brownian Motion

1. INTRODUCTION

Motivation: Review communities about items like movies, cameras, restaurants, beer, newspapers and more are a key asset for recommender systems. State-of-the-art methods harness different signals for predictions: user-user and item-item similarities in addition to user-item ratings. These are typically cast into latent factor models [12] that exploit user-user interactions, user bias, bursty posting behavior and other community-level features (e.g., [11, 14, 7]). None of the above methods, however, consider the role of *user experience* and the *evolution* of how users mature over time. These dimensions have been recognized, investigated and incorporated into recommender models only recently by [15, 18].

Example: Experienced users often appreciate certain facets of an item differently from novices and amateurs. As users gain experience they mature, and may later appreciate these more intricate facets. Consider the following reviews about Christopher Nolan movies. The facet of interest is the (non-linear) narrative style.

- *User 1 on Memento (2001): "A story does not become interesting if told backwards."*
- *User 2 on The Dark Knight (2008): "Memento was very complicated. The Dark Knight was flawless. Heath Ledger rocks!"*
- *User 3 on Inception (2010): "Inception is a triumph of style over substance. It is complex only in a structural way, not in terms of plot. It doesn't unravel in the way Memento does."*

The first user does not appreciate complex narratives. The second user prefers simpler blockbusters. The third user seems to appreciate the non-linear narration style of Inception and, more so of Memento. In terms of maturity, we would consider User 3 to be more experienced in the underlying facet, and use this assessment when generating future recommendations to her or similar users.

State-of-the-Art and its Limitations: The evolution of user experience and how it affects ratings has first been studied in [15, 18]. However, these works make the simplifying assumption that user experience is categorical with discrete levels (e.g. $[1, 2, 3, \ldots, E]$), and that users progress from one level to the next in a discrete manner. As an artifact of this assumption, the experience level of a user changes abruptly by one transition. Also, an undesirable consequence of the discrete model is that all users at the same level of experience are treated similarly, although their maturity could still be far apart (if we had a continuous scale of measuring experience). Therefore, the assumption of *exchangeability* of reviews, for the latent factor model in [18], for users at the same level of experience may not hold as the language model changes.

The prior work [15] assumes user *activity* (e.g., number of reviews) to play a major role in experience evolution, which biases the model towards highly active users (as opposed to an experienced person who posts only once in a while). In contrast, our own prior work [18] captures *interpretable* evidence for a user's experience level using her vocabulary, cast into a language model with latent facets. However, this approach also exhibits the drawbacks of discrete levels of experience, as discussed above.

The current paper overcomes these limitations by modeling the evolution of user experience, and the corresponding language model, as a *continuous-time* stochastic process. We model time *explicitly* in this work, in contrast to the prior works.

Other prior work on item recommendation that considered review texts (e.g., [16], [23], [17]) did this with the sole perspective of learning topical similarities in a static snapshot-oriented manner, without considering time at all.

Approach and Technical Challenges: This paper is the first work to develop a continuous-time model of user experience and language evolution. Unlike prior work, we do not rely on explicit features like ratings or number of reviews. Instead, we capture a user's experience by a latent language model learned from the user-specific vocabulary in her review texts. We present a generative model where the user's experience and language model evolve according to a Geometric Brownian Motion (GBM) and Brownian Motion process, respectively. Analysis of the GBM trajectory of users offer interesting insights; for instance, users who reach a high level of experience progress faster than those who do not, and also exhibit a comparatively higher variance. Also, the number of reviews written by a user does not have a strong influence, unless they are written over a long period of time.

The facets in our model (e.g., narrative style, actor performance, etc. for movies) are generated using Latent Dirichlet Allocation. User experience and item facets are latent variables, whereas the observables are *words* at explicit *timepoints* in user reviews.

The parameter estimation and inference for our model are challenging since we combine discrete multinomial distributions (generating words per review) with a continuous Brownian Motion process for the language models' evolution, and a continuous Geometric Brownian Motion (GBM) process for the user experience.

Contributions: To solve this technical challenge, we present an inference method consisting of three steps: a) estimation of user experience from a user-specific GBM using the Metropolis Hastings algorithm, b) estimation of the language model evolution by Kalman Filter, and c) estimation of latent facets using Gibbs sampling. Our experiments, with real-life data from five different communities on movies, food, beer and news media, show that the three components *coherently* work together and yield a better fit of the data (in terms of log-likelihood) than the previously best models with discrete experience levels. We also achieve an improvement of ca. 11% to 36% for the mean squared error for predicting user-specific ratings of items compared to the baseline of [15, 18]. Finally, we present a use-case study with a news-media community, where experience-aware models can be used to identify experienced *citizen journalists* — where our method performs well in capturing user maturity.

In summary, our contributions are:

- Model: We devise a probabilistic model for tracing *continuous* evolution of *user experience*, combined with a language model for facets that explicitly captures smooth evolution over time.

- Algorithm: We introduce an effective learning algorithm, that infers each users' experience progression, time-sensitive language models, and latent facets of each word.

- Experiments: We perform extensive experiments with five real-word datasets, together comprising of 12.7 million ratings from 0.9 million users on 0.5 million items, and demonstrate substantial improvements of our method over state-of-the-art baselines.

The rest of the paper is organized as follows. Section 2 introduces the fundamental components for modeling the continuous evolution of user experience and language. Section 3 presents the generative model for the joint evolution of facets, experience, and language, and also our inference methods based on Markov Chain Monte Carlo (MCMC) sampling. Section 4 shows our experimental results, comparing our model against a variety of baselines. Section 5 discusses the use-case study, followed by related prior work.

2. MODEL COMPONENTS

2.1 Importance of Time

Previous works on experience evolution [15, 18] model time only implicitly by assuming the (discrete) latent experience to progress from one review to the next. In contrast, this current paper models time explicitly, and allows experience to *continuously* evolve over time — so that we are able to trace the joint evolution of experience, and vocabulary. This is challenging as the discrete Multinomial distribution based language model (to generate words) needs to be combined with a continuous stochastic process for experience evolution.

We use two levels of temporal granularity. Since experience is naturally continuous, it is beneficial to model its evolution at a very fine resolution (say, minutes or hours). On the other hand, the language model has a much coarser granularity (say, days, weeks or months). We show in Section 3 how to smoothly merge the two granularities using continuous-time models. Our model for language evolution is motivated by the seminal work of Wang and Blei et al. [21], with major differences and extensions. In the following subsections, we formally introduce the two components affected by time: the experience evolution and the language model evolution.

2.2 Continuous Experience Evolution

Prior works [15, 18] model experience as a discrete random variable . At each timepoint, a user is allowed to stay at level l, or move to level $l + 1$. As a result the transition is abrupt when the user switches levels. Also, the model does not distinguish between users at the same level of experience, (or even for the same user at beginning or end of a level) even though their experience can be quite far apart (if measured in a continuous scale). For instance, in Figure 1b the language model uses the same set of parameters as long as the user stays at level 1, although the language model changes. In order to address these issues, our goal is to develop a continuous experience evolution model with the following requirements:

- The experience value is always positive.

- Markovian assumption for the continuous-time process: The experience value at any time t depends only on the value at the *most recent observed time prior to t*.

- Drift: It has an overall *trend* to increase over time.

- Volatility: The evolution may not be smooth with occasional volatility. For instance, an experienced user may write a series of expert reviews, followed by a sloppy one.

To capture all of these aspects, we model each user's experience as a *Geometric Brownian Motion* (GBM) process (also known as Exponential Brownian Motion).

GBM is a natural continuous state alternative to the discrete-state space based Hidden Markov Model (HMM) used in our previous work [18]. Figure 1 shows a real-world example of the evolution of an experienced and amateur user in the BeerAdvocate community, as traced by our proposed model — along with that of its discrete counterpart from our previous work. The GBM is a stochastic process used to model population growth, financial processes like stock price behavior (e.g., Black-Scholes model) with random noise. It is a continuous time stochastic process, where the logarithm of the random variable (say, X_t) follows Brownian Motion with a *volatility* and *drift*. Formally, a stochastic process X_t, with an arbitrary initial value X_0, for $t \in [0, \infty)$ is said to follow Geometric Brownian Motion, if it satisfies the following Stochastic Differential Equation (SDE) [9]:

$$dX_t = \mu X_t dt + \sigma X_t dW_t \tag{1}$$

where, W_t is a Wiener process (Standard Brownian Motion); $\mu \in \mathbb{R}$ and $\sigma \in (0, \infty)$ are constants called the *percentage trend* and
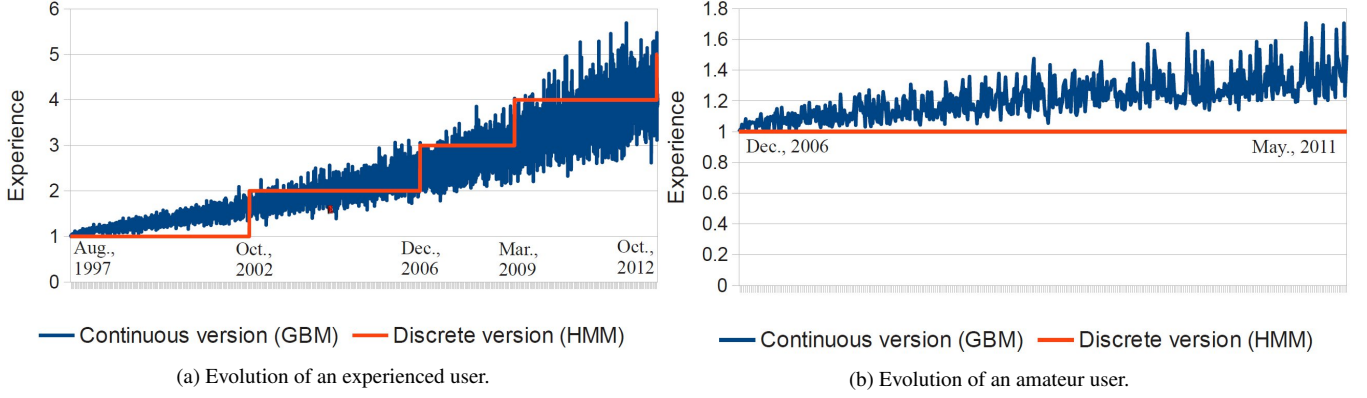
Figure 1: Discrete state and continuous state experience evolution of some typical users from the BeerAdvocate community.

percentage volatility respectively. The former captures deterministic trends, whereas the latter captures unpredictable events occurring during the motion.

In a Brownian Motion trajectory, $\mu X_t dt$ and $\sigma X_t dW_t$ capture the "trend" and "volatility", as is required for experience evolution. However, in real life communities each user might show a different experience evolution; therefore our model considers a *multivariate* version of this GBM – we model one trajectory *per-user*. Correspondingly, during the inference process we learn μ_u and σ_u for each user u.

Properties: A straightforward application of *Itô's* formula yields the following analytic solution to the above SDE (Equation 1):

$$X_t = X_0 \, exp\big((\mu - \frac{\sigma^2}{2})t + \sigma W_t\big) \qquad (2)$$

Since $log(X_t)$ follows a Normal distribution, X_t is Log-Normally distributed with mean $\big(log(X_0) + (\mu - \frac{\sigma^2}{2})t\big)$ and variance $\sigma\sqrt{t}$. The probability density function $f_t(x)$, for $x \in (0, \infty)$, is given by:

$$f_t(x) = \frac{1}{\sqrt{2\pi t}\sigma x} exp\left(-\frac{\big(log(x) - log(x_0) - (\mu - \frac{\sigma^2}{2})t\big)^2}{2\sigma^2 t}\right) \qquad (3)$$

It is easy to show that GBM has the Markov property. Consider $U_t = (\mu - \frac{\sigma^2}{2})t + \sigma W_t$.

$$\begin{aligned} X_{t+h} &= X_0 exp(U_{t+h}) \\ &= X_0 exp(U_t + U_{t+h} - U_t) \\ &= X_0 exp(U_t)exp(U_{t+h} - U_t) \\ &= X_t exp(U_{t+h} - U_t) \end{aligned} \qquad (4)$$

Therefore, future states depend only on the future increment of the Brownian Motion, which satisfies our requirement for experience evolution. Also, for $X_0 > 0$, the GBM process is always positive. *Note* that the start time of the GBM of *each* user is relative to her first review in the community.

2.3 Experience-aware Language Evolution

Once the experience values for each user are generated from a Log-Normal distribution (more precisely: the experience of the user at the times when she wrote each review), we develop the language model whose parameters evolve according to the Markov property for experience evolution.

As users get more experienced, they use more sophisticated words to express a concept. For instance, experienced cineastes refer to a movie's "protagonist" whereas amateur movie lovers talk about the "hero". Similarly, in a Beer review community (e.g., BeerAdvocate, RateBeer) experts use more *fruity* words to describe a beer like "caramel finish, coffee roasted vanilla", "and citrus hops". Facet preferences of users also evolve with experience. For example, users at a high level of experience prefer "hoppiest" beers which are considered too "bitter" by amateurs [15]. Encoding explicit time in our model allows us to trace the evolution of vocabulary and trends *jointly* on the temporal and experience dimension.

Latent Dirichlet Allocation (LDA): In the traditional LDA process [2], a document is assumed to have a distribution over Z facets (a.k.a. topics) $\beta_{1:Z}$, and each of the facets has a distribution over words from a fixed vocabulary collection. The per-facet word (a.k.a topic-word) distribution β_z is drawn from a Dirichlet distribution, and words w are generated from a Multinomial(β_z).

The process assumes that documents are drawn *exchangeably* from the same set of facets. However, this process neither takes experience nor the evolution of the facets over *time* into account.

Discrete Experience-aware LDA: Our previous work [18] incorporates a layer for *experience* in the above process. The user experience is manifested in the set of facets that the user chooses to write on, and the vocabulary and writing style used in the reviews. The experience levels were drawn from a *Hidden Markov Model* (HMM). The reviews were assumed to be exchangeable for a user at the same level of experience – an assumption which generally may not hold; since the language model of a user at the same discrete experience level may be different at different points in time (refer to Figure 1b) (if we had a continuous scale for measuring experience). The process considers *time* only *implicitly* via the transition of the latent variable for experience.

Continuous Time LDA: The seminal work of Blei et. al in [1, 21] captures evolving content, for instance, in scholarly journals and news articles where the themes evolve over time, by considering time *explicitly* in the generative LDA process. Our language model evolution is motivated by their Continuous Time Dynamic Topic Model [1], with the major difference that the facets, in our case, evolve over both *time* and *experience*.

Continuous Experience-aware LDA (this work): Since the assumption of exchangeability of documents at the same level of experience of a user may not hold, we want the language model to explicitly evolve over experience and time. To incorporate the effect of changing experience levels, our goal is to condition the parameter evolution of β on the experience progression.

In more detail, for the language model evolution, we desire the following properties:

- It should *smoothly* evolve over time preserving the Markov property of experience evolution.

- Its variance should *linearly increase* with the *experience change* between successive timepoints. This entails that if the experience of a user does not change between successive timepoints, the language model remains almost the same.

To incorporate the temporal aspects of data, in our model, we use multiple distributions $\beta_{t,z}$ for each time t and facet z. Furthermore, to capture the smooth temporal evolution of the facet language model, we need to chain the different distributions to sequentially evolve over time t: the distribution $\beta_{t,z}$ should affect the distribution $\beta_{t+1,z}$.

Since the traditional parametrization of a Multinomial distribution via its mean parameters is not amenable to sequential modeling, and inconvenient to work with in gradient based optimization – since any gradient step requires the projection to the feasible set, the simplex — we follow a similar approach as [21]: instead of operating on the mean parameters, we consider the natural parameters of the Multinomial. The natural parameters are unconstrained and, thus, enable an easier sequential modeling.

From now on, we denote with $\beta_{t,z}$ the natural parameters of the Multinomial at time t for facet z. For *identifiability* one of the parameters $\beta_{t,z,w}$ needs to be fixed at zero. By applying the following mapping we can obtain back the mean parameters that are located on the simplex:

$$\pi(\beta_{t,z,w}) = \frac{exp(\beta_{t,z,w})}{1 + \sum_{w=1}^{V-1} exp(\beta_{t,z,w})} \quad (5)$$

Using the natural parameters, we can now define the facet-model evolution: The underlying idea is that strong changes in the users' experience can lead to strong changes in the language model, while low changes should lead to only few changes. To capture this effect, let $l_{t,w}$ denote the average experience of a word w at time t (e.g. the value of $l_{t,w}$ is high if many experienced users have used the word). That is, $l_{t,w}$ is given by the average experience of all the reviews D_t containing the word w at time t.

$$l_{t,w} = \frac{\sum_{d \in D_t : w \in d} e_d}{|D_t|} \quad (6)$$

where, e_d is the experience value of review d (i.e. the experience of user u_d at the time of writing the review).

The language model evolution is then modeled as:

$$\beta_{t,z,w} \sim Normal(\beta_{t-1,z,w}, \sigma \cdot |l_{t,w} - l_{t-1,w}|) \quad (7)$$

Here, we simply follow the idea of a standard dynamic system with Gaussian noise, where the mean is the value at the previous timepoint, and the variance increases linearly with increasing change in the experience. Thereby, the desired properties of the language model evolution are ensured.

3. JOINT MODEL FOR EXPERIENCE-LANGUAGE EVOLUTION

3.1 Generative Process

Consider a corpus $D = \{d_1, \ldots, d_D\}$ of review documents written by a set of users U at timestamps T. For each review $d \in D$, we denote u_d as its user, t'_d as the fine-grained timestamp of the

review (e.g. minutes or seconds; used for experience evolution) and with t_d the timestamp of coarser granularity (e.g. yearly or monthly; used for language model evolution). The reviews are assumed to be ordered by timestamps, i.e. $t'_{d_i} < t'_{d_j}$ for $i < j$. We denote with $D_t = \{d \in D \mid t_d = t\}$ all reviews written at timepoint t. Each review $d \in D$ consists of a sequence of N_d words denoted by $d = \{w_1, \ldots, w_{N_d}\}$, where each word is drawn from a vocabulary V having unique words indexed by $\{1 \ldots V\}$. The number of facets corresponds to Z.

Let $e_d \in (0, \infty)$ denote the experience value of review d. Since each review d is associated with a unique timestamp t'_d and unique user u_d, the experience value of a review refers to the experience of the user at the time of writing it. In our model, each user u follows her own Geometric Brownian Motion trajectory – starting time of which is relative to the first review of the user in the community – parametrized by the mean μ_u, variance σ_u, and her *starting experience* value $s_{0,u}$. As shown in Equation 3, the analytical form of a GBM translates to a Log-Normal distribution with the given mean and variance. We use this user-dependent distribution to generate an experience value e_d for the review d written by her at timestamp t'_d.

Following standard LDA, the facet proportion θ_d of the review is drawn from a Dirichlet distribution with concentration parameter α, and the facet $z_{d,w}$ of each word w in d is drawn from a Multinomial(θ_d).

Having generated the experience values, we can now generate the language model and individual words in the review. Here, the language model $\beta_{t,z,w}$ uses the state-transition Equation 7, and the actual word w is based on its facet $z_{d,w}$ and timepoint t_d according to a Multinomial($\pi(\beta_{t_d,z_{d,w}})$), where the transformation π is given by Equation 5.

Note that technically, the distribution β_t and word w have to be generated simultaneously: for β_t we require the terms $l_{t,w}$, which depend on the experience and the words. Thus, we have a joint distribution $P(\beta_t, w | \ldots)$. Since, however, words are *observed* during inference, this dependence is not crucial, i.e. $l_{t,w}$ can be computed once the experience values are known using Equation 6.

We use this observation to simplify the notations and illustrations of Algorithm 1, which outlines the generative process, and Figure 2, which depicts it visually in plate notation for graphical models.
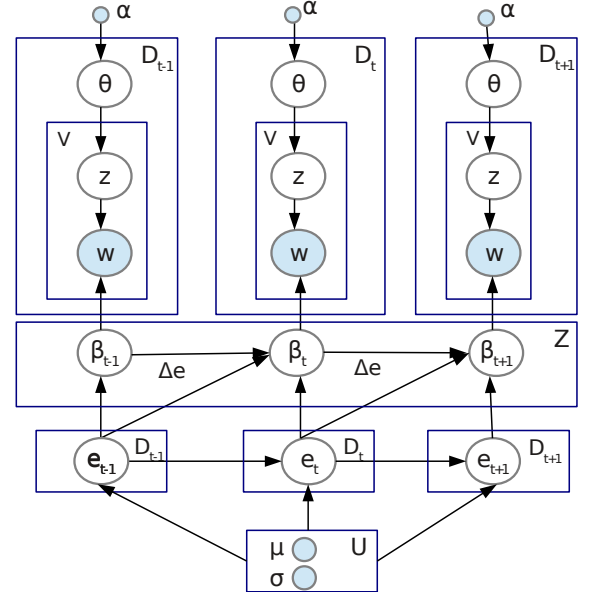

Figure 2: Continuous experience-aware language model. Words (shaded in blue), and timestamps (not shown for brevity) are observed.

Algorithm 1: Generative model for continuous experience-aware language model.

---

1. Set granularity t for language model evolution (e.g., years, months, days)
2. Set granularity for experience evolution, timestamp t' (e.g., minutes, seconds)

for *each coarse timepoint t* **do**
    for *each review $d \in D_t$* **do**
        // retrieve user $u = u_d$ and fine-grained timepoint $t' = t'_d$
        3. Draw
        $e_d \sim$ Log-Normal$((\mu_u - \frac{\sigma_u^2}{2})t' + log(s_{0,u}), \sigma_u \sqrt{t'})$
        4. Draw $\theta_d \sim$ Dirichlet(α)
        for *each word w in d* **do**
            5. Draw $z_{d,w} \sim$ Multinomial(θ_d)
        end
    end
    6. Draw $\beta_{t,z,w} \sim$ Normal$(\beta_{t-1,z,w}, \sigma \cdot |l_{t,w} - l_{t-1,w}|)$
    for *each review $d \in D_t$* **do**
        for *each word w in d* **do**
            7. Draw $w \sim$ Multinomial$(\pi(\beta_{t_d, z_{d,w}}))$
        end
    end
end

3.2 Inference

Let E, L, Z, T and W be the set of experience values of all reviews, experience values of words, facets, timestamps and words in the corpus, respectively. In the following, d denotes a review and j indexes a word in it. θ denotes the per-review facet distribution, and β the language model respectively.

The joint probability distribution is given by:

$$P(E, L, Z, W, \theta, \beta | U, T; \alpha, \langle \mu \rangle, \langle \sigma \rangle) \propto \prod_{t \in T} \prod_{d \in D_t} P(e_d; s_{0,u_d}, \mu_{u_d}, \sigma_{u_d})$$

$$\cdot \left(P(\theta_d; \alpha) \cdot \prod_{j=1}^{N_d} P(z_{d,j} | \theta_d) \cdot P(w_{d,j} | \pi(\beta_{z_{d,j},t})) \right)$$

$$\cdot \left(\prod_{z \in Z} \prod_{w \in W} P(l_{t,w}; e_d) \cdot P(\beta_{t,z,w}; \beta_{t-1,z,w}, \sigma \cdot |l_{t,w} - l_{t-1,w}|) \right) \tag{8}$$

The exact computation of the above distribution is intractable, and we have to resort to approximate inference. Exploiting conjugacy of the Multinomial and Dirichlet distributions, we can integrate out θ from the above distribution. Assuming θ has been integrated out, we can decompose the joint distribution as:

$$P(Z, \beta, E, L | W, T) \propto P(Z, \beta | W, T) \cdot P(E | Z, \beta, W, T) \cdot P(L | E, W, T) \tag{9}$$

The above decomposition makes certain conditional independence assumptions in line with our generative process.

3.2.1 Estimating Facets Z

We use Collapsed Gibbs Sampling [5], as in standard LDA, to estimate the conditional distribution for each of the latent facets $z_{d,j}$, which is computed over the current assignment for all other hidden variables, after integrating out θ. Let $n(d, z)$ denote the count of the topic z appearing in review d. In the following equation, $n(d, .)$ indicates the summation of the above counts over all possible $z \in Z$.

The subscript $-j$ denotes the value of a variable excluding the data at the jth position.

The posterior distribution $P(Z | \beta, W, T; \alpha)$ of the latent variable Z is given by:

$$P(z_{d,j} = k | z_{d,-j}, \beta, w_{d,j}, t, d; \alpha)$$

$$\propto \frac{n(d, k) + \alpha}{n(d, .) + Z \cdot \alpha} \cdot P(w_n = w_{d,j} | \beta, t, z_n = k, z_{-n}, w_{-n})$$

$$= \frac{n(d, k) + \alpha}{n(d, .) + Z \cdot \alpha} \cdot \pi(\beta_{t,k,w_n}) \tag{10}$$

where, the transformation π is given by Equation 5.

3.2.2 Estimating Language Model β

In contrast to θ, the variable β cannot be integrated out by the same process, as Normal and Multinomial distributions are not conjugate. Therefore, we refer to another approximation technique to estimate β.

In this work, we use *Kalman Filter* [8] to model the sequential language model evolution. It is widely used to model linear dynamic systems from a series of observed measurements over time, containing statistical noise, that produces robust estimates of unknown variables over a single measurement. It is a continuous analog to the Hidden Markov Model (HMM), where the state space of the latent variables is continuous (as opposed to the discrete state-space HMM); and the observed and latent variables evolve with Gaussian noise.

We want to estimate the following state-space transition model:

$$\beta_{t,z,w} | \beta_{t-1,z,w} \sim N(\beta_{t-1,z,w}, \sigma \cdot |l_{t,w} - l_{t-1,w}|)$$

$$w_{d,j} | \beta_{t,z,w} \sim Mult(\pi(\beta_{t,z,w})) \text{ where, } z = z_{d,j}, t = t_d. \tag{11}$$

However, unlike standard Kalman Filter, we do not have any *observed* measurement of the variables — due to the presence of *latent* facets Z. Therefore, we resort to *inferred* measurement from the Gibbs sampling process.

Let $n(t, z, w)$ denote the number of times a given word w is assigned to a facet z at time t in the corpus. Therefore,

$$\beta_{t,z,w}^{inf} = \pi^{-1} \left(\frac{n(t, z, w) + \gamma}{n(t, z, .) + V \cdot \gamma} \right) \tag{12}$$

where, we use the inverse transformation of π given by Equation 5, and γ is used for smoothing.

Update Equations for Kalman Filter: Let p_t and g_t denote the *prediction error*, and *Kalman Gain* at time t respectively. The variance of the process noise and measurement is given by the difference of the experience value of the word observed at two successive timepoints. Following standard Kalman Filter calculations [8], the predict equations are given by:

$$\widehat{\beta}_{t,z,w} \sim N(\beta_{t-1,z,w}, \sigma \cdot |l_{t,w} - l_{t-1,w}|)$$

$$\widehat{p}_t = p_{t-1} + \sigma \cdot |l_{t-1,w} - l_{t-2,w}| \tag{13}$$

and the update becomes:

$$g_t = \frac{\widehat{p}_t}{\widehat{p}_t + \sigma \cdot |l_{t,w} - l_{t-1,w}|}$$

$$\beta_{t,z,w} = \widehat{\beta}_{t,z,w} + g_t \cdot (\beta_{t,z,w}^{inf} - \widehat{\beta}_{t,z,w}) \tag{14}$$

$$p_t = (1 - g_t) \cdot \widehat{p}_t$$

Thus, the new value for $\beta_{t,z,w}$ is given by Eq. 14.

If the experience does not change much between two successive timepoints, i.e. the variance is close to zero, the Kalman Filter just emits the counts as estimated by Gibbs sampling (assuming, $P_0 = 1$). This is then similar to the Dynamic Topic Model [1]. Intuitively, the Kalman Filter is smoothing the estimate of Gibbs sampling taking the experience evolution into account.

3.2.3 Estimating Experience E

The experience value of a review depends on the user and the language model β. Although we have the state-transition model of β, the previous process of estimation using Kalman Filter cannot be applied in this case, as there is no observed or inferred value of E. Therefore, we resort to Metropolis Hastings sampling. Instead of sampling the E's from the complex true distribution, we use a proposal distribution for sampling the random variables — followed by an acceptance or rejection of the newly sampled value. That is, at each iteration, the algorithm samples a value of a random variable — where the current estimate depends only on the previous estimate, thereby, forming a Markov chain.

Assume all reviews $\{\cdots d_{i-1}, d_i, d_{i+1} \cdots\}$ from all users are sorted according to their timestamps. As discussed in Section 2.1, for computational feasibility, we use a coarse granularity for the language model β. For the inference of E, however, we need to operate at the fine temporal resolution of the reviews' timestamps (say, in minutes or seconds). Note that the process defined in Eq. (7) represents the aggregated language model over multiple fine-grained timestamps. Accordingly, its corresponding fine-grained counterpart is $\beta_{t'_{d_i},z,w} \sim Normal(\beta_{t'_{d_{i-1}},z,w}, \sigma \cdot |e_{d_i} - e_{d_{i-1}}|)$ — now operating on t' and the review's individual experience values. Since the language model is given (i.e. previously estimated) during the inference of E, we can now easily refer to this fine-grained definition for the Metropolis Hastings sampling.

As the proposal distribution for the experience of review d_i at time t'_{d_i}, we select the corresponding user's GBM ($u = u_d$) and sample a new experience value $\widehat{e_{d_i}}$ for the review:

$$\widehat{e}_{d_i} \sim \text{Log-Normal}((\mu_u - \frac{\sigma_u^2}{2})t'_{d_i} + log(s_{0,u}), \sigma_u \sqrt{t'_{d_i}})$$

The language model $\beta_{t'_{d_i}}$ at time t'_{d_i} depends on the language model $\beta_{t'_{d_{i-1}}}$ at time $t'_{d_{i-1}}$, and experience value difference $|e_{d_i} - e_{d_{i-1}}|$ between the two timepoints. Therefore, a change in the experience value at any timepoint affects the language model at the *current* and next timepoint, i.e. $\beta_{t'_{d_{i+1}}}$ is affected by $\beta_{t'_{d_i}}$, too.

Thus, the acceptance ratio of the Metropolis Hastings sampling becomes:

$$Q = \prod_{w,z} \left[\frac{N(\beta_{t'_b,z,w}; \beta_{t'_a,z,w}, \sigma \cdot |\widehat{e_b} - e_a|)}{N(\beta_{t'_b,z,w}; \beta_{t'_a,z,w}, \sigma \cdot |e_b - e_a|)} \right.$$
$$\left. \cdot \frac{N(\beta_{t'_c,z,w}; \beta_{t'_b,z,w}, \sigma \cdot |e_c - \widehat{e_b}|)}{N(\beta_{t'_c,z,w}; \beta_{t'_b,z,w}, \sigma \cdot |e_c - e_b|)} \right] \quad (15)$$

where $a = d_{i-1}$, $b = d_i$ and $c = d_{i+1}$. The numerator accounts for the modified distributions affected by the updated experience value, and the denominator discounts the old ones. Note that since the GBM has been used as the proposal distribution, its factor cancels out in the term Q.

Overall, the Metropolis Hastings algorithm iterates over the following steps:

1. Randomly pick a review d at time $t' = t'_d$ by user $u = u_d$ with experience e_d

2. Sample $\widehat{e}_d \sim \text{Log-Normal}\left((\mu_u - \frac{\sigma_u^2}{2})t' + log(s_{0,u}), \sigma_u \sqrt{t'}\right)$

3. Accept $\widehat{e}_d$ as the new experience with probability $P = min(1, Q)$

3.2.4 Estimating Parameters for the Geometric Brownian Motion

For each user u, the mean μ_u and variance σ_u of her GBM trajectory are estimated from the sample mean and variance.

Consider the set of all reviews $\langle d_t \rangle$ written by u, and $\langle e_t \rangle$ be the corresponding experience values of the reviews.

Let $\widehat{m}_u = \frac{\sum_{d_t} log(e_t)}{|d_t|}$, and $\widehat{s}_u^2 = \frac{\sum_{d_t}(log(e_t) - \widehat{m}_u)^2}{|d_t - 1|}$.

Furthermore, let Δ be the average length of the time intervals for the reviews of user u.

Now, $log(e_t) \sim N((\mu_u - \frac{\sigma_u^2}{2})\Delta + log(s_{0,u}), \sigma_u \sqrt{\Delta})$.

From the above equations we can obtain the following estimates using Maximum Likelihood Estimation (MLE):

$$\widehat{\sigma}_u = \frac{\widehat{s}_u}{\sqrt{\Delta}}$$
$$\widehat{\mu}_u = \frac{\widehat{m}_u - log(s_{0,u})}{\Delta} + \frac{\widehat{\sigma}_u^2}{2} \quad (16)$$
$$= \frac{\widehat{m}_u - log(s_{0,u})}{\Delta} + \frac{\widehat{s}_u^2}{2\Delta}$$

3.2.5 Overall Processing Scheme

Exploiting the results from the above discussions, the overall inference is an iterative process consisting of the following steps:

1. Estimate facets Z using Equation 10.

2. Estimate β using Equations 13 and 14.

3. Sort all reviews by timestamps, and estimate E using Equation 15 and the Metropolis Hastings algorithm, for a random subset of the reviews.

4. Once the experience values of all reviews have been determined, estimate L using Equation 6.

4. EXPERIMENTS

We perform experiments with data from five communities in different domains:

- BeerAdvocate (`beeradvocate.com`) and RateBeer (`ratebeer.com`) for beer reviews
- Amazon (`amazon.com`) for movie reviews
- Yelp (`yelp.com`) for food and restaurant reviews
- NewsTrust (`newstrust.net`) for reviews of news media

Table 1 gives the dataset statistics[1]. We have a total of 12.7 million reviews from 0.9 million users over 16 years from all of the five communities combined. The first four communities are used for product reviews, from where we extract the following quintuple for our model $< userId, itemId, timestamp, rating, review >$. NewsTrust is a special community, which we discuss in Section 5.

4.1 Data Likelihood, Smoothness and Convergence

Inference of our model is quite involved with different Markov Chain Monte Carlo methods. It is imperative to show that the resultant model is not only stable, but also improves the log-likelihood of the data. Although there are several measures to evaluate the quality

---

Dataset	#Users	#Items	#Ratings	#Years
Beer (BeerAdvocate)	33,387	66,051	1,586,259	16
Beer (RateBeer)	40,213	110,419	2,924,127	13
Movies (Amazon)	759,899	267,320	7,911,684	16
Food (Yelp)	45,981	11,537	229,907	11
Media (NewsTrust)	6,180	62,108	89,167	9
TOTAL	885,660	517,435	12,741,144	-

Table 1: Dataset statistics.

of facet models, we report the following from [20]:
$LL = \sum_d \sum_{j=1}^{N_d} log\, P(w_{d,j}|\beta; \alpha)$. A higher likelihood indicates a better model.

Figure 3 contrasts the log-likelihood of the data from the continuous experience model and its discrete counterpart [18]. We find that the continuous model is stable and has a *smooth* increase in the data log-likelihood *per iteration*. This can be attributed to how smoothly the language model evolves over time, preserving the Markov property of experience evolution. Empirically our model also shows a fast convergence, as indicated by the number of iterations.

On the other hand, the discrete model not only has a worse fit, but is also less smooth. It exhibits abrupt state transitions in the Hidden Markov Model, when the experience level changes (refer to Figure 1). This leads to abrupt changes in the language model, as it is coupled to experience evolution.

4.2 Experience-aware Item Rating Prediction

In the first task, we show the effectiveness of our model for item rating prediction. Given a user u, an item i, time t, and review d with words $\langle w \rangle$ — the objective is to predict the rating the user would assign to the item based on her *experience*.

For prediction, we use the following features: The experience value e of the user is taken as the last experience attained by the user during training. Based on the learned language model β, we construct the language feature vector $\langle F_w = log(max_z(\beta_{t,z,w})) \rangle$ of dimension V (size of the vocabulary). That is, for each word w in the review, we consider the value of β corresponding to the best facet z that can be assigned to the word at the time t. We take the log-transformation of β which empirically gives better results.

Furthermore, as also done in the baseline works [15, 18], we consider: γ_g, the average rating in the community; γ_u, the offset of the average rating given by user u from the global average; and γ_i, the rating bias for item i.

Thus, combining all of the above, we construct the feature vector $\langle \langle F_w \rangle, e, \gamma_g, \gamma_u, \gamma_i \rangle$ for each review with the user-assigned ground rating for training. We use Support Vector Regression [4], with the same set of default parameters as used in our discrete model [18], for rating prediction.

4.2.1 Baselines

We consider baselines [b – e] from [15], and use their code[2] for experiments. Baseline (f) is our prior discrete experience model[18].

a) *LFM*: A standard latent factor recommendation model [10].

b) *Community at uniform rate*: Users and products in a community evolve using a single "global clock" [11, 26, 25], where the different stages of the community evolution appear at uniform time intervals.

c) *Community at learned rate*: This extends b) by learning the rate at which the community evolves with time, eliminating the uniform rate assumption.

[2]Code available from http://cseweb.ucsd.edu/ jmcauley/code/

d) *User at uniform rate*: This extends b) to consider individual users, by modeling the different stages of a user's progression based on preferences and experience levels evolving over time. The model assumes a uniform rate for experience progression.

e) *User at learned rate*: This extends d) by allowing the *experience* of each user to evolve on a "personal clock", where the time to reach certain (*discrete*) experience levels depends on the user [15]. This is reportedly the best version of their experience evolution models.

f) *Discrete experience model*: This is our prior work [18] on the discrete version of the experience-aware language model, where the experience of a user depends on the evolution of the user's maturing rate, facet preferences, and writing style.

4.2.2 Quantitative Results

Table 2 compares the *mean squared error (MSE)* for rating predictions in this task, generated by our model versus the six baselines. Our model outperforms all baselines — except in the NewsTrust community, performing slightly worse than our prior work [18] (discussed in Section 5) — reducing the MSE by ca. 11% to 36%. Our improvements over the baselines are statistically significant at 99% level of confidence determined by *paired sample t-test*.

For all models, we used the three most recent reviews of each user as withheld test data. All experience-based models consider the *last* experience value reached by each user during training, and the corresponding learned parameters for rating prediction. Similar to the setting in [15], we consider users with a minimum of 50 reviews. Users with less than 50 reviews are grouped into a background model, and treated as a single user. We set $Z = 5$ for BeerAdvocate, RateBeer and Yelp facets; and $Z = 20$ for Amazon movies and $Z = 100$ for NewsTrust which have richer latent dimensions. All *discrete* experience models consider $E = 5$ experience levels. In the continuous model, the experience value $e \in (0, \infty)$. We initialize the parameters for our joint model as: $s_{0,u} = 1, \alpha = 50/Z, \gamma = 0.01$. Our performance improvement is strong for the *BeerAdvocate* community due to large number of reviews per-user for a long period of time, and low for NewsTrust for the converse.

4.3 Qualitative Results

User experience progression: Figure 4 shows the variation of the users' *most recent* experience (as learned by our model), along with the number of reviews posted, and the number of years spent in the community. As we would expect, a user's experience increases with the amount of *time* spent in the community. On the contrary, number of reviews posted does not have a strong influence on experience progression. Thus, if a user writes a large number of reviews in a short span of time, her experience does not increase much; in contrast to if the reviews are written over a long period of time.

Figure 5 shows the variation of the users' *most recent* experience, along with the mean μ_u and variance σ_u of her Geometric Brownian Motion (GBM) trajectory — all learned during inference. We observe that users who reach a high level of experience progress faster (i.e. a higher value of μ_u) than those who do not. Experienced users also exhibit comparatively higher variance than amateur ones. This result also follows from using the GBM process, where the mean and variance tend to increase with time.

Language model evolution: Figure 6 shows the variation of the frequency of a word — used in the community in "2011" — with the *learned* experience value $l_{t,w}$ associated to each word. The plots depict a bell curve. Intuitively, the experience value of a word does not increase with general usage; but increases if it has been used by experienced users. Highlighted words in the plot give some interesting insights. For instance, the words "beer, head, place, food,
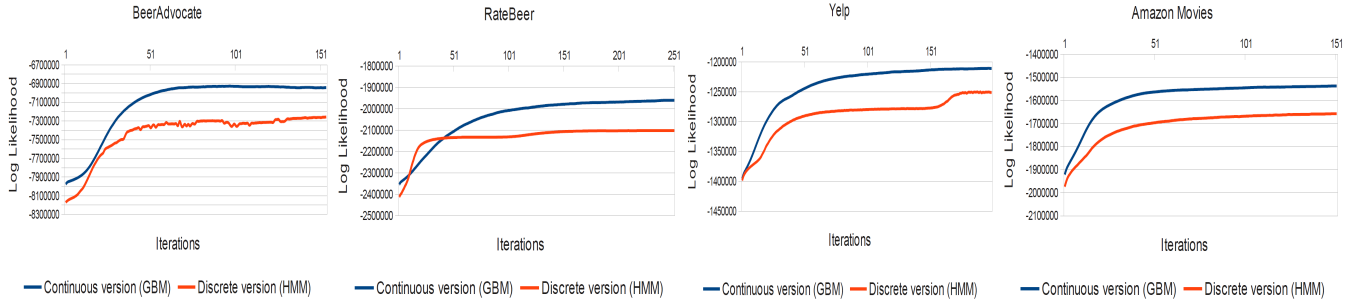
Figure 3: Log-likelihood per iteration of discrete [18] vs. continuous experience model (this work).
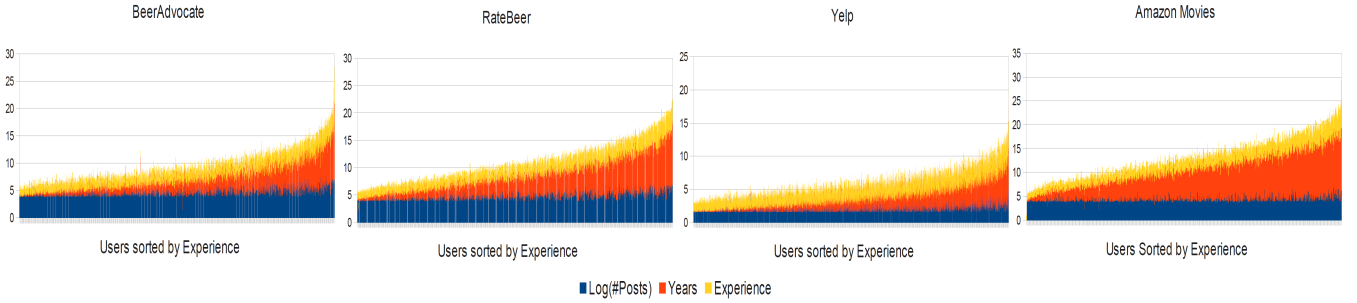

Figure 4: Variation of *experience* (*e*) with *years* and *reviews* of each user. Each bar in the above stacked chart corresponds to a user with her most recent experience, number of years spent, and number of reviews posted in the community.
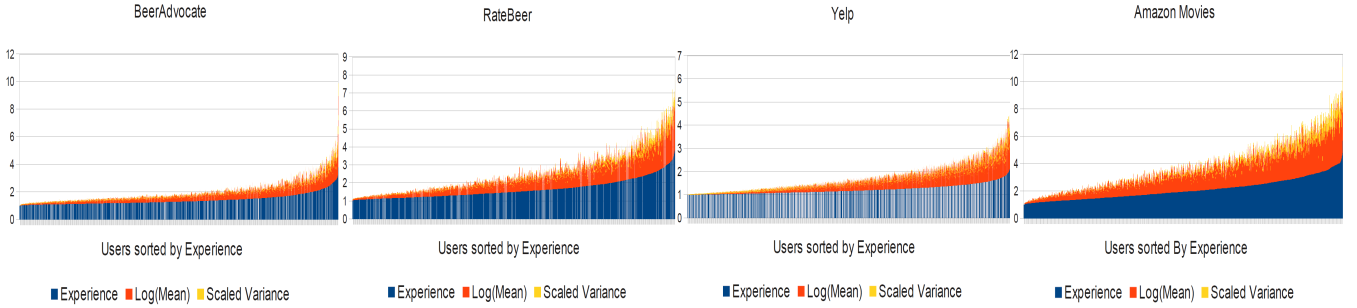

Figure 5: Variation of *experience* (*e*) with *mean* (μ_u) and *variance* (σ_u) of the GBM trajectory of each user (*u*). Each bar in the above stacked chart corresponds to a user with her most recent experience, mean and variance of her experience evolution.
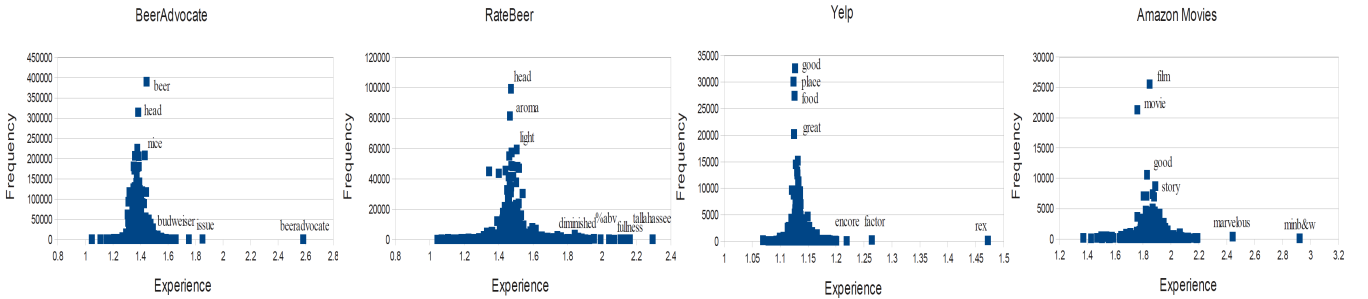

Figure 6: Variation of *word frequency* with *word experience*. Each point in the above scatter plot corresponds to a word (*w*) in "2011" with corresponding frequency and experience value ($l_{t=2011,w}$).
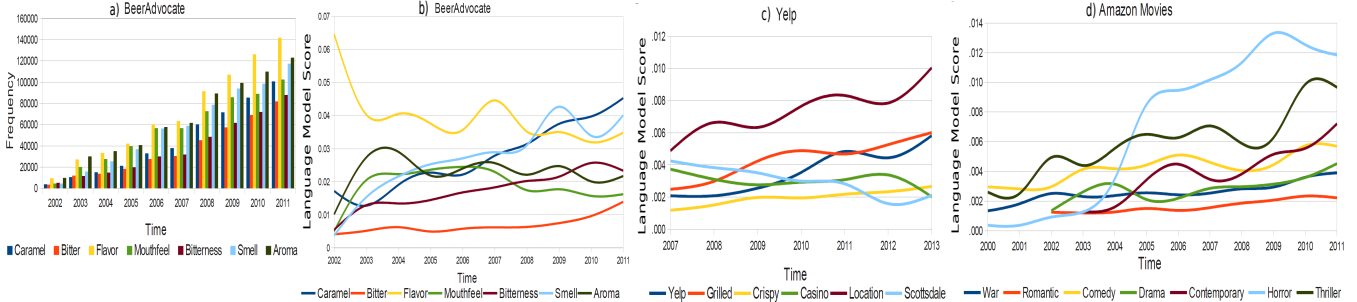

Figure 7: *Language model* score ($\beta_{t,z,w} \cdot l_{t,w}$) variation for sample words with *time*. Figure a) shows the count of some sample words over time in BeerAdvocate community, whose evolution is traced in Figure b). Figures c) and d) show the evolution in Yelp and Amazon Movies.

Models	BeerAdvocate	RateBeer	NewsTrust	Amazon	Yelp
Continuous experience model (this work)	**0.247**	**0.266**	0.494	**1.042**	**0.940**
Discrete experience model [18]	0.363	0.309	**0.464**	1.174	1.469
User at learned rate [15]	0.379	0.336	0.575	1.293	1.732
Community at learned rate [15]	0.383	0.334	0.656	1.203	1.534
Community at uniform rate [15]	0.391	0.347	0.767	1.203	1.526
User at uniform rate [15]	0.394	0.349	0.744	1.206	1.613
Latent factor model [12]	0.409	0.377	0.847	1.248	1.560

Table 2: Mean squared error (MSE) for rating prediction. Our model performs better than competing methods.

movie, story" etc. are used with high frequency in the beer, food or movie community, but have an average experience value. On the other hand specialized words like "beeradvocate, budweiser, %abv, fullness, encore, minb&w" etc. have high experience value.

Table 3 shows some top words used by *experienced* users and amateur ones in different communities, as learned by our model. Note that this is a ranked list of words with numeric values (not shown in the table). We see that experienced users are more interested about fine-grained facets like the mouthfeel, "fruity" flavors, and texture of food and drinks; narrative style of movies, as opposed to popular entertainment themes; discussing government policies and regulations in news reviews etc.

The word "rex" in Figure 6 in Yelp, appearing with low frequency and high experience, corresponds to a user "Rex M." with "Elite" status who writes humorous reviews with *self* reference.

Figure 7 shows the evolution of some sample words over *time* and experience (as given by our model) in different communities. The score in the *y-axis* combines the language model probability $\beta_{t,z,w}$ with the experience value $l_{t,w}$ associated to each word w at time t. Figure 7 a) illustrates the frequency of the words in BeerAdvocate, while their evolution is traced in Figure 7 b). It can be seen that the overall usage of each word increases over time; but the evolution path is different for each word. For instance, the "smell" convention started when "aroma" was dominant; but the latter was less used by *experienced* users over time, and slowly replaced by (increasing use of) "smell". This was also reported in [3] in a different context. Similarly "caramel" is likely to be used more by *experienced* users, than "flavor". Also, contrast the evolution of "bitterness", which is used more by experienced users, compared to "bitter".

In Yelp, we see certain food trends like "grilled" and "crispy" increasing over time; in contrast to a decreasing feature like "casino" for restaurants. For Amazon movies, we find certain genres like "horror, thriller" and "contemporary" completely dominating other genres in recent times.

5. USE-CASE STUDY

So far we have focused on traditional item recommendation for items like beers or movies. Now we switch to different kind of items - newspapers and news articles - by analyzing the NewsTrust online community (`newstrust.net`) (data available from [19]). It features news stories posted and reviewed by members, some of whom are professional journalists and content experts. Stories are reviewed based on their objectivity, rationality, and general quality of language to present an unbiased and balanced narrative of an event with focus on *quality journalism*. Unlike the other datasets, NewsTrust contains expertise of members that can be used as ground-truth for evaluating our model-generated *experience* values.

In our framework, each story is an item, which is rated and reviewed by a user. The facets are the underlying topic distribution of the reviews, with topics being *Healthcare, Obama Administration, NSA*, etc. The facet preferences can be mapped to the (political) polarity of users in this news community.

Most Experience	Least Experience
BeerAdvocate	
chestnut_hued near_viscous rampant_perhaps faux_foreign cherry_wood sweet_burning bright_crystal faint_vanilla boned_dryness woody_herbal citrus_hops mouthfeel	originally flavor color didnt favorite dominated cheers tasted review doesnt drank version poured pleasant bad bitter sweet
Amazon	
aficionados minimalist underwritten theatrically unbridled seamless retrospect overdramatic diabolical recreated notwithstanding oblivious featurettes precocious	viewer entertainment battle actress tells emotional supporting evil nice strong sex style fine hero romantic direction superb living story
Yelp	
rex foie smoked marinated savory signature contemporary selections bacchanal delicate grits gourmet texture exotic balsamic	mexican chicken salad love better eat atmosphere sandwich local dont spot day friendly order sit
NewsTrust	
health actions cuts medicare oil climate major jobs house vote congressional spending unemployment citizens events	bad god religion iraq responsibility questions clear jon led meaningful lives california powerful

Table 3: Top words used by experienced and amateur users.

Recommending News Articles. Our first objective is to recommend news to readers catering to their viewpoints, and experience. We apply our model with the same setting as with earlier datasets. The mean squared error (MSE) results were reported in Section 4. Our model clearly outperforms most of the baselines; it performs only slightly worse regarding our prior work [18] in this task — possibly due to high rating sparsity in face of a large number of model parameters.

Identifying Experienced Users. Our second task is to find experienced members of this community, who have the potential of being *citizen journalists*. In order to evaluate the quality of the ranked list of experienced users generated by our model, we consider the following proxy measure for user experience. In NewsTrust, users have *Member Levels* determined by the NewsTrust staff based on community engagement, time in the community, other users' feedback on reviews, profile transparency, and manual validation. We use these member levels to categorize users as *experienced* or *inexperienced*. This is treated as the ground truth for assessing the ranking quality of our model against the baseline models [15, 18] — considering top 100 users from each model ranked by experience. Here we consider the top-performing baseline models from the previous task. We report the *Normalized Discounted Cumulative Gain (NDCG)* and the *Normalized Kendall Tau Distance* for the ranked lists of users generated by all the models. The better model should exhibit higher *NDCG*, and lower *Kendall Tau Distance*. As Table 4 shows, our model outperforms baseline models in capturing user maturity.

6. RELATED WORK

State-of-the-art recommender systems [10, 12] harness user-user and item-item similarities by means of latent factor models. Time-dependent phenomena such as bursts in item popularity, bias in ratings, and the temporal evolution of user community are investigated in [11, 26, 25]. There is also prior work on anomaly detection [6,

Models	NDCG	Kendall Tau Normalized Distance
Continuous experience model (this work)	**0.917**	**0.113**
Discrete experience model [18]	0.898	0.134
User at learned rate [15]	0.872	0.180

Table 4: Performance on identifying experienced users.

7], capturing changes of social links [14] and linguistic norms [3]. None of these prior works take into account the evolving experience and behavior of individual users.

Prior work that analyzed user review texts focused on sentiment analysis [13], learning latent aspects and their ratings [23, 17, 16], and user-user interactions [24]. However, all of these prior approaches operate in a static, snapshot-oriented manner, without considering time at all.

The work [15], one of our baselines, has modeled and studied the influence of evolving user experience on rating behavior and for targeted recommendations. However, it disregards the vocabulary in the users' reviews. Our own recent work [18] addressed this very limitation, by means of language models that are specific to the experience level of an individual user, and by modeling transitions between experience levels with a Hidden Markov Model. However, both of these works are limited to *discrete* experience levels leading to abrupt changes in both experience and language model. To address the above, and other related drawbacks, the current paper introduces continuous-time models for the smooth evolution of both user experience and language model.

Wang et al. [22] modeled topics over time. However, the topics themselves were constant, and time was only used to better discover them. Dynamic topic models have been introduced by Blei et al. in [1, 21]. This prior work developed generic models based on Brownian Motion, and applied them to news corpora. [21] argues that the continuous model avoids making choices for discretization and is also more tractable compared to fine-grained discretization. Our language model is motivated by the latter. We substantially extend it to capture evolving user behavior and experience in review communities using Geometric Brownian Motion.

7. CONCLUSION

In this paper, we have proposed an experience-aware language model that can trace the *continuous* evolution of user experience and language explicitly over *time*. We combine principles of Geometric Brownian Motion, Brownian Motion, and Latent Dirichlet Allocation to model a smooth temporal progression of user experience, and language model over time. This is the first work to develop a continuous and generalized version of user experience evolution.

Our experiments – with data from domains like beer, movies, food, and news – demonstrate that our model effectively exploits user experience for item recommendation that substantially reduces the mean squared error for predicted ratings, compared to the state-of-the-art baselines [15, 18].

We further demonstrate the utility of our model in a use-case study about identifying experienced members in the NewsTrust community, where these users would be top candidates for being citizen journalists. Another similar use-case for our model can be to detect experienced medical professionals in the health community.

8. REFERENCES

[1] David M. Blei and John D. Lafferty. Dynamic topic models. ICML '06, 2006.

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3, 2003.

[3] Cristian Danescu-Niculescu-Mizil, Robert West, Dan Jurafsky, Jure Leskovec, and Christopher Potts. No country for old members: User lifecycle and linguistic change in online communities. WWW, 2013.

[4] Harris Drucker, Chris J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. NIPS, 1997.

[5] Tom Griffiths. Gibbs sampling in the generative model of latent dirichlet allocation. Technical report, 2002.

[6] Nikou Günnemann, Stephan Günnemann, and Christos Faloutsos. Robust multivariate autoregression for anomaly detection in dynamic product ratings. In *WWW*, 2014.

[7] Stephan Günnemann, Nikou Günnemann, and Christos Faloutsos. Detecting anomalies in dynamic rating data: A robust probabilistic model for rating evolution. KDD, 2014.

[8] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME*, 1960.

[9] Ioannis Karatzas and Steven Eugene Shreve. *Brownian motion and stochastic calculus*. Graduate texts in mathematics. 1991.

[10] Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. KDD, 2008.

[11] Yehuda Koren. Collaborative filtering with temporal dynamics. *Commun. ACM*, 53(4), 2010.

[12] Yehuda Koren and Robert Bell. Advances in collaborative filtering. In *Recommender systems handbook*. 2011.

[13] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. CIKM, 2009.

[14] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender systems with social regularization. WSDM, 2011.

[15] J. J. McAuley and J. Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. WWW, 2013.

[16] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. RecSys, 2013.

[17] Subhabrata Mukherjee, Gaurab Basu, and Sachindra Joshi. Joint author sentiment topic model. SDM, 2014.

[18] Subhabrata Mukherjee, Hemank Lamba, and Gerhard Weikum. Experience-aware item recommendation in evolving review communities. ICDM, 2015.

[19] Subhabrata Mukherjee and Gerhard Weikum. Leveraging joint interactions for credibility analysis in news communities. CIKM, 2015.

[20] Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. ICML, 2009.

[21] Chong Wang, David M. Blei, and David Heckerman. Continuous time dynamic topic models. UAI, 2008.

[22] Xuerui Wang and Andrew McCallum. Topics over time: A non-markov continuous-time model of topical trends. KDD, 2006.

[23] Hongning Wang et al. Latent aspect rating analysis without aspect keyword supervision. KDD, 2011.

[24] Robert West, Hristo S. Paskov, Jure Leskovec, and Christopher Potts. Exploiting social network structure for person-to-person sentiment analysis. *TACL*, 2(2), 2014.

[25] Liang Xiang and et al. Temporal recommendation on graphs via long- and short-term preference fusion. KDD, 2010.

[26] L. Xiong, X. Chen, T. K. Huang, J. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. SDM, 2010.