Dynamic Collaborative Filtering With Compound Poisson Factorization

Ghassen Jerfel
Princeton University

Mehmet E. Basbug
Princeton University

Barbara E. Engelhardt
Princeton University

Abstract

Model-based collaborative filtering (CF) analyzes user–item interactions to infer latent factors that represent user preferences and item characteristics in order to predict future interactions. Most CF approaches assume that these latent factors are static; however, in most CF data, user preferences and item perceptions drift over time. Here, we propose a new conjugate and numerically stable dynamic matrix factorization (DCPF) based on hierarchical Poisson factorization that models the smoothly drifting latent factors using gamma-Markov chains. We propose a conjugate gamma chain construction that is numerically stable within our compound-Poisson framework. We then derive a stochastic variational inference approach to estimate the parameters of our model. We apply our model to time-stamped ratings data sets from Netflix, Yelp, and Last.fm. We empirically demonstrate that DCPF achieves a higher predictive accuracy than state-of-the-art static and dynamic factorization algorithms.

1 Introduction

Collaborative filtering (CF) (Koren et al., 2009) is a popular framework for recommender systems to personalize item recommendations for each customer based on their previous item interactions; these approaches are used by large online platforms where user decision-making is influenced by item recommendations, including Amazon (Linden et al., 2003), Spotify, and Netflix (Koren, 2009). Latent factor models have recently become a research focus within CF.

These models represent user preferences and item attributes in terms of latent factors that exploit the low-dimensional latent structure to impute missing user–item interactions (Koren et al., 2009). In this context, Poisson factorization models (PF) (Cemgil, 2009; Gopalan et al., 2013; Charlin et al., 2015; Basbug and Engelhardt, 2016; Gopalan et al., 2014) have demonstrated better accuracy in predicting held-out user–item ratings than other CF models such as Probabilistic Matrix Factorization (PMF) (Salakhutdinov and Mnih, 2008), a Gaussian factorization model.

Poisson factorization models for CF generally assume that user preferences and item attributes are static over time (Cemgil, 2009). However, in real-world scenarios, user interests and item attributes drift in time (Charlin et al., 2015; Chua et al., 2013). Items' popularity and reception continuously evolve as new items and categories emerge and global environments change. Customers' preferences and needs change over time, leading them to interact with items in a time-varying way. However, static PF models may mistakenly recommend sports documentaries to a current fan of horror movies, or suggest rock songs to a new fan of electronic music. Similarly, they might not recognize that a 50's statistical physics paper (Robbins and Monro, 1951) is again relevant but to a new machine learning audience. Thus, the static assumption of traditional PF models is not well-aligned with the true generative model of ratings in real-world scenarios.

Temporal dynamics have been explored in the context of other collaborative filtering frameworks. Early approaches underweighted older user–item interactions (Yu et al., 2003). TimeSVD++ proposed an extension to singular value decomposition (SVD) by augmenting matrix factorization with a user–item–time bias term and user-evolving factors, but ignored the drifting nature of the item attributes (Koren, 2010). Nevertheless, TimeSVD++ was one of the winning algorithms of the Netflix Prize in 2009, motivating the need for further work on temporal dynamics in CF models. Bayesian Poisson tensor factorization proposes an independent latent factor for time (or con-

text) then factorizes a user–item–time tensor (Xiong et al., 2010). This approach models the user and item factors at each time index independently from previous ones, which makes evaluating trends of specific user and item factors impossible.

Several dynamic extensions to PMF have also been proposed by leveraging a Gaussian state space model to represent the evolving user and item latent factors (Gultekin and Paisley, 2014; Sun et al., 2012; Chua et al., 2013). However, the conjugate gamma-Poisson structure of Poisson factorization models ensures computationally tractable approaches for high-dimensional data using variational inference (Cemgil, 2009). Additionally, long tailed gamma priors powerfully capture the overdispersed user and item behavior in sparse matrices, which leads to better predictive results—in terms of precision and recall—than PMF (Salakhutdinov and Mnih, 2008) and latent Dirichlet allocation (Blei and Lafferty, 2006), among other CF approaches as shown by Gopalan et al. (2013).

In this paper, we address the problem of evolving user and item factors within the context of Poisson factorization. We propose a novel dynamic PF model: dynamic compound-Poisson factorization (DCPF). DCPF is a novel dynamic probabilistic model that represents the user and item latent factors as independent smoothly-evolving gamma-Markov chains. There has been a recent dynamic extension attempt for PF replacing the gamma priors with a Gaussian state space model (Charlin et al., 2015). However, this approach compromises the conjugacy of the model, leading to computational intractability or, alternatively, inaccurate numerical approximations for posterior inference. In fact, the lack of conjugacy prevents simple closed-form updates and consequently prevents convenient extensions to the model. Furthermore, Gaussian priors at each time step fail to capture the empirical response distribution and the long-tailed Gamma distributions, as demonstrated by Gopalan et al. (2013).

However, conjugacy cannot be attained with simple gamma chains in a Poisson factorization model. Therefore, we introduce auxiliary gamma chains to each chain of user and item factors to guarantee the conjugacy of the user and item chains within the model. We also introduce a compound Poisson generating distribution (Basbug and Engelhardt, 2016). The Poisson compounding allows us to rescale the non-negative multiplicative gamma chains, which would otherwise grow uncontrollably and thus lead to numerical instability during inference. It also provides more flexibility with ratings data types.

In Section 2, we propose a conjugate gamma chain construction with dampened, positively correlated states

(i.e., slowly and smoothly evolving in time). In Section 3, we describe the generative model for DCPF and the stochastic variational inference approach that allows the inference of DCPF to scale to large, sparse data sets. In Section 4, we present the experimental setup including the data sets, the static and dynamic baselines, the comparison metrics, our hyperparameter initialization choices, and discuss our results. We conclude with potential improvements.

2 Conjugate and Stable Gamma Markov Chains

In our approach to incorporating smooth temporal dynamics in Poisson factorization, we preserved the gamma-Poisson conjugate structure by placing gamma-Markov chains as priors on the latent factors associated with each user and item. Conjugacy guarantees closed form updates to infer the model in a scalable fashion, and combine it with other conjugate models in the future. Additionally, long tail gamma priors are able to capture diverse sets of users and items in terms of their activity and popularity (Gopalan et al., 2013). This allows us to better represent data where a few users might be proactive reviewers while the majority rarely contribute, for example.

We assume that the latent factors evolve smoothly in time and remain non-negative, which requires the gamma chain states to be positively correlated (i.e., each state matches the previous state's trajectory with a higher probability). Therefore, we constructed a Gamma chain by conditioning the state at every time step on the inverse of the state at the previous time step via the gamma scale parameters. This is different from the Acharya et al. (2015) approach of chaining on the shape parameter with no consideration to the temporal smoothness as they used the chains for the substantially different problem of factor analysis.

A simple and intuitive construction of this chain would draw the kth component of the user m's latent factor at time t as follows : $u_{m,k,t} \sim Ga(a, b \cdot u_{m,k,t-1})$.

Here, Ga is the gamma distribution with a fixed shape parameter a and a rate parameter $b \cdot u_{m,k,t-1}$.

However, this construction is non-conjugate because the full conditional distribution of the latent factor value, $p(u_{m,k,t}|u_{m,k,t-1}, u_{m,k,t+1})$, includes the terms u, $1/u$ and $log\ u$. A conjugate density should be in the same distribution family as $p(u_{m,k,t})$ (gamma). However, one can easily check that a gamma density for u cannot be expressed with the above terms.

Thus, we add auxiliary variables $z_{m,k,t}$ to our chains while preserving the positive correlation between

$u_{m,k,t}$ and $u_{m,k,t-1}$ as first suggested by Cemgil and Dikmen (2007). The chain (Fig.1) for the kth component of user m's latent factor is constructed as follows:

$$
\begin{aligned}
z_{m,k,1} &\sim Ga(\epsilon, b_m^z) \\
u_{m,k,t}|z_{m,k,t} &\sim Ga(\iota, \omega \cdot z_{m,k,t}) \\
z_{m,k,t+1}|u_{m,k,t} &\sim Ga(\epsilon, \Omega \cdot u_{m,k,t}).
\end{aligned}
$$

This construction differs from prior work in the parametrization of the gamma chains. We separate the parameter b_m^z (used for the initialization of each chain when $t = 1$) from the transition or chaining parameters Ω and ω in contrast to Cemgil and Dikmen (2007). Decoupling these parameters allows us to capture the changes in user preferences $u_{m,k}$ over time independently from the initial state values. This allows us to model both the static interests (the initial state for each chain) and the temporal corrections.
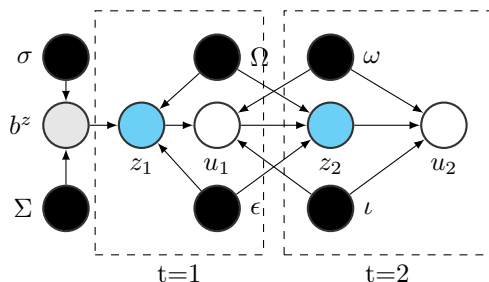

Figure 1: Gamma Chain for User Latent Factors u_t (white) Interleaved by Auxiliary Variables z_t (blue). Hyperparameters are in black.

However, our contribution is not limited to the change in the gamma construction but also how we incorporate these chains in our model to ensure their stability, which was not addressed in previous works despite the obvious problem it poses to the inference task.

First note that $E[u_t] = \frac{\iota}{\omega \cdot E[z_t]}$ and $E[z_t] = \frac{\epsilon}{\Omega \cdot E[u_{t-1}]}$, which compounds until $t = 1$.

In fact, the correlation is thus always positive (depends on absolute values of ϵ, ι, ϵ and Ω) whereas the skewness depends on the ratio $\frac{\iota \cdot \omega}{\Omega \cdot \epsilon}$. If the ratio < 1 the probability mass is shifted towards the region $u_t < ut - 1$ and the chain exhibit a systematic positive drift (the opposite is also true for a ratio> 1).

Thus, as t increases, $E[u_t]$ and $E[z_t]$ will increase (or decrease) monotonically ($z > 0$ and $u > 0$) depending on the ratio raised to the power of the time index. With a ratio significantly larger or smaller than 1 (due to posterior updates) the chains can grow too large or too small which we experienced when first attempting to use a simple Poisson factorization.

To ensure the numerical stability of the model during

the inference routine, we leverage the update of one of the compound Poisson hyperparameters Λ_t (explained in the next section). Consequently, our model rescales of the contribution of the latent factors towards the generating distributions to ensure the chains do not grow uncontrollably in time.

3 Dynamic Compound Poisson Factorization (DCPF) Model

3.1 The DCPF Generative Model

At each time step, we represent each user m as a vector u of K latent preferences and each item n as a vector v of K latent attributes. Let z and w be the auxiliary gamma variables for the chains of u and v, respectively. Let the variables b_m^z and b_n^w parametrize the initial states of the user m and item n chains.

One of our contributions is the compound-Poisson generating distribution (Basbug and Engelhardt, 2016). Instead of introducing a new parameter solely to rescale the gamma chains, independently from the generating distribution, we opted for compound-Poisson which includes a parameter that we can leverage for that purpose in addition to providing more flexibility with the type of observation data it can handle.

A compound Poisson distribution is defined as the sum of η i.i.d. random variables each with an element distribution $p_\psi(\theta, \kappa)$ (Basbug and Engelhardt, 2016). Here, the i.i.d. variables are drawn from exponential dispersion models (EDM) and η is a Poisson-distributed random variable of mean Λ. By virtue of the additivity property of EDM distributions, a compound Poisson variable has a distribution $p_\psi(\theta, \eta \cdot \kappa)$, which is also an EDM distribution with scale parameter $\eta \cdot \kappa$ and natural parameter θ (Basbug and Engelhardt, 2016). Examples of compound Poisson include zero-truncated Poisson and gamma-Poisson.

Accordingly, with a generating compound Poisson distribution of density $p_\psi(\theta, \eta \cdot \kappa)$, the generative process of our DCPF model is detailed in Algorithm 1.

The main computational problem for our model is the posterior inference of the parameters of the model given a matrix of observations indexed by user, item, and time. Once the posterior is estimated, we use the DCPF generative model to recommend items to users based on their latent factors at a given time point using the expectation of the score provided by $p_\psi(\theta, \eta \cdot \kappa)$.

3.2 Stochastic Variational Inference for DCPF

Given a matrix of user ratings for items across time, we wish to compute the posterior distribution of user

---

Algorithm 1 DCPF Generative Model

$\sigma,\Sigma,\delta,\Delta,\omega,\Omega$ were introduced earlier as the parameters of the user gamma-Markov chains. $\iota,\epsilon,\hat{\iota},\hat{\epsilon},\hat{\omega},\hat{\Omega}$ are the item chains counterparts.

- For each user m from 1 to M, sample the parameter of the initial state of the chain:
$$b^z_m \sim Ga(\delta, \delta/\Delta)$$

For each component k, sample the initial user latent factor or preference $(t=1)$:
$$z_{1,k} \sim Ga(\epsilon, b^z_m)$$
$$u_{1,k}|z_{1,k} \sim Ga(\iota, \omega z_{1,k})$$

Then, for time t from 2 to T, sample the states of the chain of user preferences:
$$z_{t,k}|u_{t-1,k} \sim Ga(\epsilon, \Omega u_{t-1,k})$$
$$u_{t,k}|z_{t,k} \sim Ga(\iota, \omega z_{t,k})$$

- For each item n from 1 to N, sample the parameter of the initial state of the chain:
$$b^w_n \sim Ga(\sigma, \sigma/\Sigma)$$

For each component k, sample the initial item latent factor $(t=1)$:
$$w_{1,k} \sim Ga(\hat{\epsilon}, b^w_n)$$
$$v_{1,k}|w_{1,k} \sim Ga(\hat{\iota}, \hat{\omega} w_{1,k})$$

Then, for time t from 2 to T, sample the states of the chain of item attributes:
$$w_{t,k}|v_{t-1,k} \sim Ga(\hat{\epsilon}, \hat{\Omega} v_{t-1,k})$$
$$v_{t,k}|w_{t,k} \sim Ga(\hat{\iota}, \hat{\omega} w_{t,k})$$

- For each user m and item n at time t:
  1. Sample counts
  $$\eta_{m,n,t} \sim Poisson(\Lambda_{m,n,t} = \sum_k u_{m,k,t} v_{k,n,t})$$
  2. Sample the compound Poisson response
  $$y_{m,n,t} \sim p_\psi(\theta, \eta_{m,n,t}\kappa)$$

---

preferences $u_{m,k,t}$ and item attributes $v_{n,k,t}$. As for many Bayesian models, the exact posterior for Poisson factorization is computationally intractable (Cemgil, 2009). Therefore, we will use variational inference, which proposes a family of distributions over the latent variables indexed by variational parameters (Jordan et al., 1999). Variational inference estimates the set of variational parameters that place the variational distributions of the posterior closest to the true posterior in Kullback-Leibler divergence (Kullback and Leibler, 1951). This inference problem thus becomes an optimization problem that can be scaled to large sample sizes using stochastic optimization, leading to stochastic variational inference (SVI) (Hoffman et al., 2013).

The latent variables in our model are the user preferences u, item attributes v, their corresponding auxiliary chain variables z and w, and the rate parameters of the initial states of each chain b^z and b^w. Our mean-field variational distribution is thus given by:

$$q(u_{m,k,t} \mid \alpha^u_{m,k,t}, \beta^u_{m,k,t})q(z_{m,k,t} \mid \alpha^z_{m,k,t}, \beta^z_{m,k,t})$$
$$q(v_{n,k,t} \mid \alpha^v_{n,k,t}, \beta^v_{n,k,t})q(w_{n,k,t} \mid \alpha^w_{n,k,t}, \beta^w_{n,k,t})$$
$$q(b^z_m \mid \alpha^{bz}_m, \beta^{bz}_m)q(b^w_n \mid \alpha^{bw}_n, \beta^{bw}_n)q(s_{m,n,t} \mid \boldsymbol{\varphi_{m,n,t}})q(\eta_{m,n,t})$$

The variational distributions for latent factors and parameters u, z, v, w, and the parameters on the initial states of the chains, b^z and b^m, are all Gamma-distributed with independent variational shape and rate parameters α and β.

The variational distribution of s is a multinomial distribution with parameter φ. s and φ are auxiliary variables for each user–item–time tuple typically used in PF to facilitate the model inference (Cemgil, 2009).

We optimize the variational parameters with coordinate ascent which iteratively optimizes each variational parameter while holding the others fixed at their previous approximations (Hoffman et al., 2013). We assert convergence when there is $< 10^{-5}$ change in predictive likelihood on a validation set.

Given DCPF's conjugacy, we derived closed-form updates for each of our parameters (Algorithm 2). Note τ and ξ are the learning rate parameters for the stochastic optimization (Hoffman et al., 2013).

Computation: The update for each user or item parameter is independent of all the other users or items, respectively. Accordingly, this inference routine is easily parallelizeable. As for the runtime, the per-iteration complexity can be linear in terms of non-missing entries at each time step if we train on non-missing instead of the full matrix (difference discussed extensively in HCPF (Basbug and Engelhardt, 2016)).

Gamma chain stability: The variational parameters for u_t and z_t depend on Λ_t. The update for Λ_t simply equals the observation which can be large. As explained previously, such a change at time t can cause the chain to grow unstable. We address this issue with two deliberate model choices. First, through compounding, the contribution of the variational parameter for Λ_t towards the observations can be controlled to stay small which reflects on the updates for the variational parameters of u and v (see Basbug and Engelhardt (2016) for the interaction between Λ_t and the latent factors). Second, we carefully initialize the hyperparameters to control the skewness of the chain, at least for the first iteration of the posterior update.

Algorithm 2 SVI for DCPF

Initialize: Hyperparameters $\iota, \epsilon, \sigma, \Sigma, \omega, \Omega, \hat{\iota}, \hat{\epsilon}, \hat{\omega}, \hat{\Omega}$.
$N = \#items$, $K = \#factors$

$$\alpha_{m,t,k}^z = \iota + \epsilon \qquad\qquad t_u = \tau$$
$$\alpha_m^{bz} = \sigma + K \cdot \epsilon$$

repeat

    Sample a user m and item n from the data set

    repeat

        Compute local variational parameters for m, n, t

$$\Lambda_{m,n,t} = \sum_k \frac{\alpha_{m,k,t}^u \alpha_{n,k,t}^v}{\beta_{m,k,t}^u \beta_{n,k,t}^v}$$

$$q(\eta_{mn,t} = \eta) \propto \exp\left\{-\kappa\eta\Psi(\theta)\right\} h(y_{m,n,t}, \eta\kappa) \frac{\Lambda_{m,n,t}^\eta}{\eta!}$$

$$\varphi_{m,n,k,t} \propto \exp\left\{\Psi(\alpha_{m,k,t}^u) - \log\beta_{m,k,t}^u \right.$$
$$\left. + \Psi(\alpha_{n,k,t}^v) - \log\beta_{n,k,t}^v\right\}$$

        Update global variational parameters [only showing equations for user chains]

$$\alpha_{m,k,t}^u = (1 - t_u^{-\xi})\alpha_{m,k,t}^u$$
$$+ \quad t_u^{-\xi}\left(\epsilon + \iota + N \cdot E[\eta_{m,n,t}] \cdot \varphi_{m,n,k,t})\right)$$
$$\beta_{m,k,t}^u = (1 - t_u^{-\xi})\beta_{m,k,t}^u$$
$$+ \quad t_u^{-\xi}\left(\omega \cdot \frac{\alpha_{m,k,t}^z}{\beta_{m,k,t}^z} + \Omega \cdot \frac{\alpha_{m,k,t+1}^z}{\beta_{m,k,t+1}^z} + N \cdot \frac{\alpha_{n,k,t}^v}{\beta_{n,k,t}^v}\right)$$
$$\beta_{m,k,t}^z = (1 - t_u^{-\xi})\beta_{m,k,t}^z$$
$$+ \quad t_u^{-\xi}\left(\omega \cdot \frac{\alpha_{m,k,t}^u}{\beta_{m,k,t}^u} + \Omega \cdot \frac{\alpha_{m,k,t-1}^u}{\beta_{m,k,t-1}^u}\right)$$
$$\beta_m^{bz} = (1 - t_u^{-\xi})\alpha_m^{bz}$$
$$+ \quad t_u^{-\xi}\left(\frac{\sigma}{\Sigma} + \sum_k \frac{\alpha_{m,t,k}^z}{\beta_{m,k,t}^z}\right)$$

    until $t = end$ of training time period

    Update learning rates

$$t_u = t_u + 1 \qquad\qquad t_i = t_i + 1$$

    Update hyperparameters θ and κ

until held-out log likelihood converges

---

4 Results

We consider two types of time-stamped ratings data sets: explicit ratings (e.g., 1 to 5 stars) and implicit ratings (e.g., number of song plays or clicks). We fit our model to these data sets on a fixed number of time windows (Table 1) and then predict the ratings for the held-out time windows. We compare the prediction accuracy versus HCPF (Basbug and Engelhardt, 2016) and DPF (Charlin et al., 2015).

4.1 Data Sets and Discretization

1. **Active Netflix**: A subset of the Netflix Prize data set (Bennett and Lanning, 2007) filtered by selecting users and movies who are active between the first and last time periods(10/01/98 – 12/31/05) and had a minimum amount of activity across different time periods similarly to the Li et al. (2011) and Charlin et al. (2015) approaches.

2. **Active Yelp**: A subset of the Yelp Academic Challenge data set that includes customers and businesses active between 2008 and 2015.

3. **LFM Tracks/Bands**: A Last.fm dataset of binary ratings that indicate whether or not a Last.fm user played a song during the given time period (Celma, 2010). We also consider bands as items instead of tracks.

4. **LFMSum Tracks/Bands** Indicates the number of times a user listened to a song or band for a given time period.

The data set dimensions can be found in Table 1.

We split each data set into time windows using the time stamps for each rating. We kept the duration of a time window at around 3 to 9 months based on previous works and our judgment of the average pace of change for each ratings data set. Specifically, music tastes might be evolving over weeks as new songs and albums come out. Movie tastes are often cyclical over 6 months, as movies move out of theaters and onto DVDs. Interest in specific types of businesses may be static over time windows of a year or more.

Table 1: **Dataset Characteristics** Number of users, items, time slices or periods used for discretization, and average sparsity per time period.

	#USERS	#ITEMS	SLICES	SPARSITY
ACTIVE-YELP	905	6721	12	0.999348
ACTIVE-NETFLIX	10484	6513	16	0.996527
LFMTRACKS	991	1079037	16	0.999448
LFMBAND	991	107000	16	0.998666

We assign the ratings of the last 1 or 2 time windows in each data set to a held-out test set in order to quantify the performance. The choice of the number of windows to hold out depends on the number of ratings in those windows that guarantees a reasonable test set size. We then randomly sample and assign 5% of the ratings within this test set for the validation set.

4.2 Baselines

We compare the performance of DCPF to the most recent dynamic and static Poisson factorization algorithms(code provided by the corresponding authors):

- **HCPF** the most recent and accurate Poisson factorization model (Basbug and Engelhardt, 2016). To run HCPF, which is not a dynamic model, we remove the timestamps from the training, test, and validation sets and remove repeated ratings across different time periods since HCPF cannot have more than one user-item interaction for every user-item pair. In particular, we filter duplicates in our LFM data by keeping the first instance of each user-item interaction as it signals the user's initial interest in an item (keeping the final interaction had similar results). Since DCPF performs compound Poisson factorization, we compare different variants of compound Poisson generating distributions across DCPF and HCPF.

- **DPF** the most recent dynamic PF algorithm, which, in prior work (Charlin et al., 2015), was shown to outperform state-of-the-art dynamic CF algorithms, specifically, Bayesian Probabilistic Tensor Factorization (Xiong et al., 2010) and TimeSVD++ (Koren, 2010). DPF binarizes the ratings by default (the model was only tested on binary data in prior work and the provided code did not perform well on non-binary data).

4.3 Initialization

We also use the same number of latent factors ($K = 70$) for both HCPF and DCPF as a compromise between tractability and predictive accuracy. In fact, we experimented with a wide range $20 < K < 120$ and the differences were negligible on all data sets.

We initialize the parameters of the compound Poisson distribution based on the maximum likelihood estimates computed from a small sample of non-zero ratings at $t = 1$ (Basbug and Engelhardt, 2016). We also use $E[\eta_{m,n,1}]$ to initialize the rate parameter of the initial states of the chains, b_m^z and b_n^w, by assuming that the contribution of all k latent factor components for both the user and the item is equal. Therefore, $E[b_m^z] = \Delta = E[b_n^w] = \Sigma = \sqrt{E[\eta_{m,n,1}]/K}$.

In our tests, we fixed ω and Ω to 1 such that the drift of the chains depends on $\frac{\iota}{\epsilon}$. We sought to keep those values small and the ratio close to 1, so we initialized $\epsilon = 1.01$ and $\iota = 1.01$ in order to avoid numerical instability. Accordingly, only ϵ, b_m^z, and ι control user chains, where b_m^z allows each user chain to vary differently from the other users. The hyperparameters for the items were initialized in a similar fashion.

As for the SVI hyperparameters, we use 10,000 for τ the learning rate delay and 0.7 for ξ the learning rate power, which is similar to the HCPF initialization (Basbug and Engelhardt, 2016).

We initialize the variance of the Gaussian distributions of DPF with the suggested default values (Charlin et al., 2015). In fact, we attempted to tune the initialization but found no significant difference. We also tried a few time period lengths to run on both DCPF and DPF, but the intuitive choice (in Table 1) had the best performance for both.

4.4 Metrics

To understand the quality of the fit for each model, we calculate the area under the ROC curve (AUC) (Hanley and McNeil, 1982) for DPF and four variants each of DCPF and HCPF: zero-Truncated Poisson (ZTP), Gamma-Poisson (GA), Gaussian-Poisson (N), and Poisson-Poisson (PO). An example of the ROC curve can be seen in Fig. 2. The AUC compares the probability that a given user-item rating in the test set exists, after binarizing all the numeric ratings to represent whether or not the item was rated at all.

We also report the test set log likelihood for HCPF and DCPF variants fit on the full matrix similarly to the comparisons in Basbug and Engelhardt (2016).

DPF, in prior work, only computes ranking metrics which is typical for collaborative filtering models for implicit data (Charlin et al., 2015). Therefore, to compare DCPF with DPF, we generate the top L items to recommend to each user based on the predicted scores. For each user we then proceed to compute the precision @L (fraction of relevant items in the top-L items with $L = 10$ or 100). We also calculate the normalized discounted cumulative gain (NDCG) (@10 and @100) (Järvelin and Kekäläinen, 2002).

5 Results

We can see from Table 2 that DCPF is the best performing algorithm, in terms of AUC on four of the 6 data sets with DPF only outperforming DCPF on LFMBands and LFMTracks. Since those data sets are implicit, we have to compare the fit of DCPF and DPF on those data sets based on ranking metrics.

HCPF was outperformed by DCPF on every single data set which validates our dynamic hypothesis regarding the user and item factors. To further confirm this hypothesis we compare the test log likelihood of each DCPF and HCPF variant.

Table 2: **AUC Values for HCPF, DCPF and DPF**: Bold values indicate the best performing model across static and dynamic variants. No DPF values reported for the LFMSum data sets as DPF binarizes them.

	HCPF				DCPF				DPF
Data	PO	GA	N	ZTP	PO	GA	N	ZTP	
Active Yelp	0.740	0.743	0.744	0.735	0.827	**0.843**	0.835	0.832	0.828
Active Netflix	0.864	0.865	0.864	0.864	**0.871**	0.867	0.864	0.864	0.858
LFMSumTracks	0.885	0.884	0.885	0.883	0.903	0.900	**0.903**	0.902	-
LFMSumBands	0.913	0.912	0.912	0.913	0.913	**0.916**	0.913	0.902	-
LFMTracks	0.884	0.885	0.885	0.885	0.891	0.891	0.892	0.889	**0.900**
LFMBands	0.878	0.881	0.879	0.878	0.902	0.904	0.901	0.899	**0.937**

Table 3: **Test Set Log Likelihood Comparison of HCPF and DCPF Variants**: Dynamic variants in grey.

	LFMTRACKS	LFMBANDS	ACTIVE NETFLIX	ACTIVE YELP	LFMSUMBANDS	LFMSUMTRACKS
HCPF-PO	-0.016	-0.073	-0.238	-0.149	-0.068	-0.019
DCPF-PO	-0.004	-0.009	-0.027	**-0.006**	-0.017	**-0.004**
HCPF-GA	-0.015	-0.033	-0.228	-0.149	-0.029	-0.018
DCPF-GA	**-0.002**	**-0.002**	-0.027	-0.006	**-0.011**	-0.005
HCPF-N	-0.016	-0.039	-0.223	-0.137	-0.038	-0.020
DCPF-N	-0.002	0.002	-0.026	-0.006	-0.014	-0.005
HCPF-ZTP	-0.016	-0.086	-0.235	-0.136	-0.068	-0.020
DCPF-ZTP	-0.004	-0.008	**-0.026**	-0.006	-0.017	-0.004

5.1 Comparison of HCPF and DCPF

It is clear that, overall, DCPF outperforms HCPF in terms of AUC over all data sets (example in Fig. 2). However, if we were to pairwise compare the model on a variant-basis, we can see that it is possible for some variants of HCPF to outperform their dynamic counterparts. The ZTP static model outperforms the dynamic model on *LFMSumBands*. Additionally, there is no significant difference between the AUC values on *Active Netflix* for the N and ZTP variants.
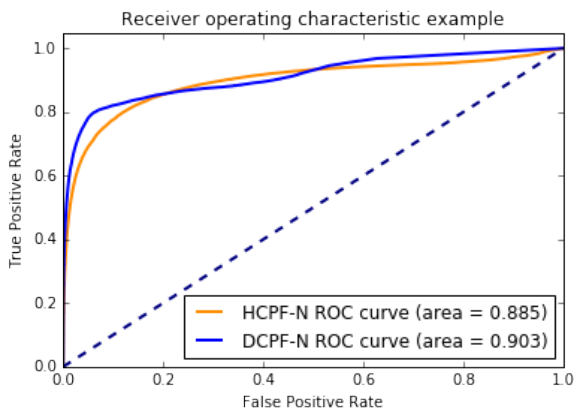

Figure 2: ROC Curve Comparison on LFMSumTracks

This rare lack of improvement over the static baseline can be linked to the imbalance, in terms of number of ratings, across time windows. In fact, *Active Netflix* has 10 times fewer ratings in the first window than in the last. Similarly, *LFMSumBands* has six times fewer ratings in the first window as compared to each window in the later half of the data. In contrast, *Active Yelp* is more balanced, in addition to being an active subset, and displays a much greater improvement.

This is an expected weakness of DCPF since emerging users or items (first observation at $t > 1$) can display inadequate chain updates. This is due to the great difference between the initialization and the update at $t > 1$ where a sudden shift in interest or activity cannot be perfectly captured by a smooth transition. Additionally, this kind of shift is rarely repeated over time and is the reason dynamic CF approaches use "active" subset of the data sets (Charlin et al., 2015; Xiong et al., 2010; Li et al., 2011). However, our positive correlation might bias future updates towards higher values to match this shift even though the shift usually occurs only once when first observing a user or item.

We further compare the test log likelihood of the static and dynamic variants. This allows us to evaluate not only the binary accuracy but also the general model fit that is more relevant to non-binary data.

Accordingly, DCPF substantially outperforms HCPF for every compound-Poisson generating distribution on the *Active Netflix* and *Active Yelp* data sets according to Table 3. A similar margin of improvement is seen in the *LFM* data sets (Table 3). This suggests that DCPF can effectively exploit data with repeated ratings over time–LastFm has repeated user-item ratings across time windows. This repeated structure allows for a more accurate estimate of the user interests and

Table 4: **Ranking metrics comparison of DCPF and DPF** NDCG at 10 and 100, and precision at 10 and 100 are reported for DPF and DCPF

DATA SET	MODEL	PREC@10	PREC@100	NDCG@10	NDCG@100
ACTIVE YELP	DCPF	**0.007**	**0.004**	**0.014**	**0.043**
	DPF	0.006	0.003	0.011	0.030
ACTIVE NETFLIX	DCPF	**0.044**	**0.024**	**0.059**	**0.091**
	DPF	0.030	0.017	0.041	0.077
LFMTRACKS	DCPF	**0.169**	**0.103**	**0.193**	**0.121**
	DPF	0.056	0.036	0.063	0.043
LFMBANDS	DCPF	**0.283**	**0.165**	**0.358**	**0.232**
	DPF	0.069	0.042	0.080	0.076

item attributes, as we have multiple opportunities to estimate parameters based on non-zero observations along the chain for a given user-item pair.

5.2 Comparison of DCPF and DPF

The reported AUC values for DCPF were lower than those of DPF on *LFMBands* and *LFMTracks*. This was expected on these binary data sets since DPF only samples non-missing entries and is specialized to the task of binary prediction.(Charlin et al., 2015) This advantage over sampling from the whole matrix was extensively discussed in Basbug and Engelhardt (2016).

However, binary accuracy does not translate well into predicting rankings which matter more when fitting on implicit ratings data sets. In most cases, we care about modeling the scale of the interest a user has for an item. Therefore, we computed the NDCG and precision values reported in Table 4. This should allow us to compare how well each model predicts the top 10 or 100 items for each user in the test set.

As seen in Table 4, DCPF outperforms DPF on all data sets based on all metrics. This contrast to the AUC comparison is due to the fact that the DPF model binarizes all observations and thus ignores how strongly a user favors an item that has been rated. Additionally, long-tailed gamma priors have been proven to better capture the user and item factors in CF settings, than Gaussian priors (Gopalan et al., 2013). This difference is clearest on the *LFM* data sets which confirms the advantage of scaled ratings in assessing how much a user would rate specific items. In this case, the number of times a user plays a song is a scaled indicator of future interest in that song or similar songs.

6 Conclusion

In this paper, we proposed a novel conjugate and computationally stable dynamic matrix factorization (DCPF) that models the smoothly changing latent factors over time using gamma-Markov chains. Preserv-

ing the gamma-Poisson structure allows us to leverage the long-tailed gamma prior's better fit to user and item factors in sparse matrices. Our conjugate chain construction guarantees a straightforward scalable inference and potentially enables the combination with other conjugate structures or models. We also offset the main computational concern of non-negative multiplicative gamma chains, the numerical instability, with the use of Poisson compounding. This allows us to both provide flexibility for the observation data and control the gamma chains' growth.

We applied DCPF to six different time-stamped ratings data sets, and showed that DCPF achieved a higher predictive performance than a state-of-the-art static factorization model (HCPF) in terms of AUC and test log likelihood. We also demonstrated an improvement over DPF in terms of ranking accuracy.

We also illustrated a drawback of this approach relative to the static model in the presence of imbalanced recommendations across time, or emerging and inactive users or items. This can be addressed with an "active" subset of the data sets, which we already used with mitigating effects for some of the data sets. Additionally, our approach lacks the flexibility of classic time series techniques. Our smoothness assumption allows us to model the dynamic latent factors as gamma-Markov chains but ignores cyclical or seasonal behavior that can be prevalent in shopping or music listening habits. Another extension can use gamma processes to avoid tuning the latent dimension similarly to the concurrent work by Schein et al. (2016).

However, we believe this conjugate, stable, scalable, and potentially modular framework will open doors for more complicated dynamic Bayesian models.

References

Ayan Acharya, Joydeep Ghosh, and Mingyuan Zhou. Nonparametric bayesian factor analysis for dynamic count matrices. In *AISTATS*, 2015.

Mehmet E Basbug and Barbara Engelhardt. Hierarchical compound poisson factorization. *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

James Bennett and Stan Lanning. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35, 2007.

David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

Oscar Celma. *Music Recommendation and Discovery in the Long Tail*. Springer, 2010.

Ali Taylan Cemgil. Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience*, 2009, 2009.

Ali Taylan Cemgil and Onur Dikmen. Conjugate gamma markov random fields for modelling nonstationary sources. In *Independent Component Analysis and Signal Separation*, pages 697–705. Springer, 2007.

Laurent Charlin, Rajesh Ranganath, James McInerney, and David M Blei. Dynamic poisson factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 155–162. ACM, 2015.

Freddy Chong Tat Chua, Richard J Oentaryo, and Ee-Peng Lim. Modeling temporal adoptions using dynamic matrix factorization. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 91–100. IEEE, 2013.

Prem Gopalan, Jake M Hofman, and David M Blei. Scalable recommendation with poisson factorization. *arXiv preprint arXiv:1311.1704*, 2013.

Prem K Gopalan, Laurent Charlin, and David Blei. Content-based recommendations with poisson factorization. In *Advances in Neural Information Processing Systems*, pages 3176–3184, 2014.

San Gultekin and John Paisley. A collaborative kalman filter for time-evolving dyadic processes. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 140–149. IEEE, 2014.

James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Yehuda Koren. The bellkor solution to the netflix grand prize. *Netflix prize documentation*, 81, 2009.

Yehuda Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Bin Li, Xingquan Zhu, Ruijiang Li, Chengqi Zhang, Xiangyang Xue, and Xindong Wu. Cross-domain collaborative filtering over time. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 2293–2298. AAAI Press, 2011.

Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887. ACM, 2008.

Aaron Schein, Hanna Wallach, and Mingyuan Zhou. Poisson-gamma dynamical systems. In *Advances in Neural Information Processing Systems*, pages 5006–5014, 2016.

John Z Sun, Kush R Varshney, and Karthik Subbian. Dynamic matrix factorization: A state space approach. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 1897–1900. IEEE, 2012.

Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff G Schneider, and Jaime G Carbonell. Temporal collaborative

filtering with bayesian probabilistic tensor factorization. In *SDM*, volume 10, pages 211–222. SIAM, 2010.

Kai Yu, Xiaowei Xu, Martin Ester, and Hans-Peter Kriegel. Feature weighting and instance selection for collaborative filtering: An information-theoretic approach*. *Knowledge and Information Systems*, 5 (2):201–224, 2003.