# A Spatio-Temporal Approach to Collaborative Filtering

Zhengdong Lu
Institute for Computational Engineering & Sciences
University of Texas at Austin
luz@cs.utexas.edu

Deepak Agarwal
Yahoo! Research
Sunnyvale,CA
dagarwal@yahoo-inc.com

Inderjit S. Dhillon
Department of Computer Sciences
University of Texas at Austin
inderjit@cs.utexas.edu

## ABSTRACT

In this paper, we propose a novel spatio-temporal model for collaborative filtering applications. Our model is based on low-rank matrix factorization that uses a spatio-temporal filtering approach to estimate user and item factors. The spatial component regularizes the factors by exploiting correlation across users and/or items, modeled as a function of some implicit feedback (e.g., who rated what) and/or some side information (e.g., user demographics, browsing history). In particular, we incorporate correlation in factors through a Markov random field prior in a probabilistic framework, whereby the neighborhood weights are functions of user and item covariates. The temporal component ensures that the user/item factors adapt to process changes that occur through time and is implemented in a state space framework with fast estimation through Kalman filtering. Our spatio-temporal filtering (ST-KF hereafter) approach provides a single joint model to simultaneously incorporate both spatial and temporal structure in ratings and therefore provides an accurate method to predict future ratings. To ensure scalability of ST-KF, we employ a mean-field approximation for inference. Incorporating user/item covariates in estimating neighborhood weights also helps in dealing with both *cold-start* and *warm-start* problems seamlessly in a single unified modeling framework; covariates predict factors for new users and items through the neighborhood. We illustrate our method on simulated data, benchmark data and data obtained from a relatively new recommender system application arising in the context of Yahoo! Front Page.

## Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining; I.2.6 [**Artificial Intelligence**]: Learning

## General Terms

Algorithms, Design

## Keywords

Collaborative filtering, matrix factorization, spatial modeling, Kalman filtering, graphical model

*RecSys'09,* October 23–25, 2009, New York, New York, USA.
Copyright 2009 ACM 978-1-60558-435-5/09/10 ...$10.00.

## 1. INTRODUCTION

Matrix factorization (MF) is an effective prediction technique that has been successfully applied to several collaborative filtering applications [17, 13, 18, 19]. However, most existing MF-based collaborative filtering algorithms do not consider the following facts:

- Real-world recommender systems are often dynamic in nature and require adaptive estimation of user and item characteristics for accurate predictions. In fact, for several well studied movie recommender problems, user factors are non-stationary over time and require an adaptive estimation procedure that emphasizes recent user ratings more than his/her past behavior. In several other applications, item characteristics, such as the popularity of a news article, may also be dynamic and change rapidly in a rather short time period. In such scenarios, "static" matrix factorization is sub-optimal. In fact, most collaborative filtering algorithms have been designed and evaluated based on random split of training and test sets without considering the temporal structure.
- In addition to the actual ratings, we often have extra information about users or items, such as implicit feedback (e.g., who rated what) and/or some side information (e.g., user demographics, browsing history). Incorporating side information becomes especially useful when available ratings for users and items are scarce.

In this paper, we provide a new method to address the above-mentioned problems through a model that simultaneously incorporates the spatial and temporal structure in rating history. We show that this joint model, referred to as spatio-temporal Kalman filtering (ST-KF), improves prediction accuracy over standard matrix factorization on synthetic data, Movielens data and the Yahoo! Front Page data.

### 1.1 Overview

Our work enhances the basic factor model for collaborative filtering that models the rating of item $i$ by user $u$ as

$$r^{(ui)} = (\mathbf{p}^{(u)})^T \mathbf{q}^{(i)} + \epsilon^{(ui)}, \tag{1}$$

where $\mathbf{p}^{(u)}$ and $\mathbf{q}^{(i)}$ are $k$-dimensional latent user and item factors respectively, and $\epsilon^{(ui)}$ denotes observation error. Estimation of these factors is obtained through some regularization on factors to prevent over-fitting; it is customary to constrain the factors by regulating the $L_2$ norm of factors. The main contribution of our work is to enhance the regularization through a spatio-temporal filtering approach to obtain better estimates of user and item factors. The spatial component, discussed in Section 2, regularizes the factors by exploiting correlations across users and/or items. The temporal component, introduced in Section 3, ensures that user/item factors adapt to process changes that occur through time. In particular, temporal change in factors is modeled in a state space framework with fast estimation through Kalman filtering. Our spatio-temporal filtering approach (named ST-KF, see Sections 4-5) provides a single joint model to simultaneously incorporate both spatial and temporal structure in ratings using a "product of experts" approach that results in an accurate method to predict future ratings. Although classical Kalman filtering is a fast estimation method for linear state-space models, simultaneous estimation of both spatial and temporal components induces a model that is computationally intensive; we ensure scalable inference for ST-KF through mean-field approximation. Our model also handles both *cold-start* and *warm-start* problems seamlessly in a single unified modeling framework by incorporating user/item covariates in estimating neighborhood weights; covariates predict factors for new users and new items through the neighborhood.

**Notation**

We use superscripts to index users, e.g., $\mathbf{p}^{(u)}$, and items, e.g., $\mathbf{q}^{(i)}$, and use subscripts for types (e.g. users vs. items) or time index in temporal modeling. We use bold upper-case letters, e.g., $\mathbf{R}$, for matrices, bold low-case letters, e.g., $\mathbf{r}$, for vectors, and italics for scalars, e.g., $r$, and the entries in a matrix or vector, e.g., $W^{(uv)}$.

## 2. SPATIAL MODEL

We first consider incorporating spatial correlation across users and/or items into the collaborative filtering model. Here we focus on the matrix factorization (MF) model, and express our prior belief on the user and item similarity as a Markov random field prior for user and item factors.

### 2.1 MF with Graph Laplacian Prior

Consider a user-item rating matrix $\mathbf{R} \in \mathbb{R}^{N\times M}$ ($N$ users and $M$ items) that is partially observed. Matrix factorization models attempt to find a rank-$k$ approximation of $\mathbf{R}$ of the form

$$\mathbf{R} \approx \mathbf{P}\mathbf{Q}^T \quad (\mathbf{P} \in \mathbb{R}^{N\times k}, \mathbf{Q} \in \mathbb{R}^{M\times k}),$$

where the rows in $\mathbf{P}$ are the *user factors* and rows in $\mathbf{Q}$ the *item factors*. The most commonly used approach for estimating $\mathbf{P}$ and $\mathbf{Q}$ is to minimize the cost function

$$\|\mathbf{B} \odot (\mathbf{R} - \mathbf{P}\mathbf{Q}^T)\|_F^2 + \lambda(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2), \qquad (2)$$

where $\mathbf{B}$ indicates the location of observed ratings with

$$B_{ui} = \begin{cases} 1 & \text{if } R_{ui} \text{ is observed,} \\ 0 & \text{if } R_{ui} \text{ is missing,} \end{cases}$$

and $\odot$ stands for the Hadamard product. Equation (2) can be equivalently interpreted as finding the maximum a posteriori (MAP) solution of $\{\mathbf{P}, \mathbf{Q}\}$ while assuming that both the user and item factors are drawn from a spherical Gaussian and $\mathbf{R}$ is contaminated with Gaussian noise.

Now suppose we have some additional similarity information for users either given *a priori*, or extracted from side information, which leads to a more informative prior for user factors $\{\mathbf{p}^{(1)}, \cdots, \mathbf{p}^{(N)}\}$, denoted $f_p(\mathbf{P})$,

$$f_p(\mathbf{P}) \propto \prod_{u,v} e^{-\frac{\alpha}{2} W_p^{(uv)} \|\mathbf{p}^{(u)} - \mathbf{p}^{(v)}\|^2}, \qquad (3)$$

where $W_p^{(uv)} \geq 0$ represents the similarity between users $u$ and $v$, and the parameter $\alpha$ controls the strength of this prior. Equation (3) defines a Gaussian Markov random field (MRF) [12] over $\{\mathbf{p}^{(1)}, \cdots, \mathbf{p}^{(N)}\}$. In fact, the conditional distribution of user factor $\mathbf{p}^{(u)}|\mathbf{p}^{(-u)}$ ($\mathbf{p}^{(-u)}$ denotes all user factors except that of user $u$) only depends on the factors in the neighborhood of $u$ and is a $k$-dimensional Gaussian given by

$$\mathbf{p}^{(u)}|\mathbf{p}^{(-u)} \sim \mathcal{N}\left(\frac{\sum_{v\in\mathcal{U}(u)} W_p^{(uv)}\mathbf{p}^{(v)}}{\sum_{v\in\mathcal{U}(u)} W_p^{(uv)}}, (\alpha \sum_{v\in\mathcal{U}(u)} W_p^{(uv)})^{-1}\mathbf{I}\right), \quad (4)$$

where $\mathcal{U}(u)$ denotes the neighbors of $u$ and $\mathbf{I}$ denotes the identity matrix. It follows from the celebrated Hammersely-Clifford theorem that for symmetric $\mathbf{W}_p$, the conditional distribution in (4) induce a unique joint distribution that is given by (3). Thus, we are modeling the joint distribution of factors by modeling the precision matrix (the inverse of covariance matrix) through similarity functions $\mathbf{W}_p$ instead of modeling the covariance matrix. All existing work on matrix factorization assumes independent priors on factors. In contrast, we impose dependencies at the outset through a joint distribution resulting in ellipsoidal constraints that are functions of the covariates. A similar prior can be imposed on item factors, for ease of exposition we assume the same $\alpha$ as in (3),

$$f_q(\mathbf{Q}) \propto \prod_{i,j} e^{-\frac{\alpha}{2} W_q^{(ij)} \|\mathbf{q}^{(i)} - \mathbf{q}^{(j)}\|^2}. \qquad (5)$$

This MRF prior, after combined with the spherical Gaussian prior in a "product of experts" fashion, gives a joint prior for user and item factors,

$$pr(\mathbf{P}, \mathbf{Q}) \propto e^{-\frac{1}{2}\lambda(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2)} f_p(\mathbf{P}) f_q(\mathbf{Q}).$$

With this prior, finding the MAP estimate of $\mathbf{P}$ and $\mathbf{Q}$ amounts to minimizing

$$\|\mathbf{B} \odot (\mathbf{R} - \mathbf{P}\mathbf{Q}^T)\|_F^2 + \lambda(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2) + \\ \alpha(\mathrm{tr}(\mathbf{P}^T\Delta_p\mathbf{P}) + \mathrm{tr}(\mathbf{Q}^T\Delta_q\mathbf{Q})) \qquad (6)$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix, $\Delta_p$ is the graph Laplacian composed from the similarity matrix $\mathbf{W}_p$ (assuming $W_p^{(uu)} = 0,\ u = 1, 2.\cdots, N$),

$$\Delta_p = \mathbf{D}_p - \mathbf{W}_p,$$

and $\mathbf{D}_p$ is the diagonal degree matrix with $D_p^{(uu)} = \sum_v W_p^{(uv)}$. It can be easily verified that $\mathrm{tr}(\mathbf{P}^T\Delta_p\mathbf{P}) = \sum_{u,v} W_p^{(uv)}\|\mathbf{p}^{(u)} - \mathbf{p}^{(v)}\|^2$ penalizes differences between similar users $u$ and $v$. The regularization for items is defined in the same way. This spatially regularized matrix factorization scheme will be referred to as SptMF.

The graph Laplacian regularized matrix factorization model in (6) can be used standalone as a way to incorporate side information about users and items or as an additional component in a more complex model. Moreover, as will be shown in Sections 4, it can also serve as an initialization step for modeling the temporal structure.

## 2.2 Estimating the Similarity Matrix

We construct the similarity matrix $\mathbf{W}_p$ and/or $\mathbf{W}_q$ by exploiting side information and ratings history. For simplicity, we will only discuss similarity construction among users, and similar discussion holds for items. We provide a few examples below of constructing such similarity measures:
**From covariates.** When the covariate is categorical (e.g., gender, occupation, etc), we use the following measure

$$W^{(uv)} = \delta(\mathbf{x}^{(u)}, \mathbf{x}^{(v)})$$

where $\mathbf{x}_u$ indicates the category of user $u$. When the covariate is numerical (e.g., age), we use the following Gaussian RBF kernel

$$W^{(uv)} = e^{-\|\mathbf{x}^{(u)} - \mathbf{x}^{(v)}\|^2/\sigma^2}$$

**From rating history.** The simplest similarity measure is the co-occurrence based on the "who-rated-what" matrix,

$$W^{(uv)} = \text{number of items rated by both } u \text{ and } v.$$

We can also combine the similarities constructed from different sources $\{\mathbf{W}_{(1)}, \cdots, \mathbf{W}_{(m)}\}$. One way is to find a convex combination $\mathbf{W} = \sum_{i=1}^{m} \alpha_i \mathbf{W}_{(i)}$ to maximize the alignment between $\mathbf{W}$ and some target correlation, for example, the Pearson correlation among users extracted from observed ratings. This alignment strategy has been proven effective for kernel learning [9] and can be readily used for our problem. Furthermore, we make our similarity matrix $\mathbf{W}$ sparse by removing weak similarities to reduce noise in estimates of neighborhoods. Such a sparse $\mathbf{W}$ reduces computational cost and enables a scalable procedure. Indeed, for both alternating least squares and stochastic gradient [1] (two commonly used methods for estimating factors with matrix factorization), estimating the factors for a particular user $u$ involves calculating the average of its nearest neighbors

$$\sum_{u' \in \mathcal{U}(u)} W^{(uu')} \mathbf{p}^{(u')} / \sum_{u' \in \mathcal{U}(u)} W^{(uu')},$$

which needs $\mathcal{O}(k)$ time if $\mathbf{W}$ is a $k$-NN graph, but $\mathcal{O}(N)$ time if $\mathbf{W}$ is dense. The advantage of having a sparse $k$-NN graph will become more obvious for the joint spatio-temporal filtering, for which we have to dynamically incorporate the user similarity prior in filtering steps.

# 3. TEMPORAL MODEL

The spatial model in Section 2 exploits correlation in covariate space to enforce smoothness across users and/or item factors. Another source of correlation is through time, which is important for dynamical modeling of the user and item factors. For simplicity, we will first assume the item characteristic to be time-invariant. Later in Section 4 we will relax this assumption and allow for on-line re-estimation of the item factors. As mentioned before, the temporal structure can be modeled in a state space framework with fast estimation through Kalman filtering [8].

## 3.1 Dynamic Model

We assume the item factors are known *a priori*, e.g., as item features given by other sources or pre-estimated by some preprocessing step such as a static MF. We assume the user factors for each user $u$ follow a random walk driven by Gaussian noise:

$$\text{dynamics:} \qquad \mathbf{p}_t^{(u)} = \mathbf{p}_{t-1}^{(u)} + \mathbf{w}_t^{(u)}, \tag{7}$$

$$\text{observation:} \qquad \mathbf{r}_t^{(u)} = \mathbf{H}_t^{(u)} \mathbf{p}_t^{(u)} + \mathbf{v}_t^{(u)}, \tag{8}$$

where $\mathbf{r}_t^{(u)}$ is the vector of ratings from user $u$ in time interval $t$, and $\mathbf{H}_t^{(u)}$ is the observation operator composed of corresponding rows of $\mathbf{Q}$ (item factors) based on the model given by (1); $\mathbf{w}_t^{(u)}$ and $\mathbf{v}_t^{(u)}$ are respectively the process noise and the observation noise, both Gaussian: $\mathbf{w}_t^{(u)} \sim \mathcal{N}(0, \Sigma_p)$, $\mathbf{v}_t^{(u)} \sim \mathcal{N}(0, \sigma_o^2 \mathbf{I})$, and uncorrelated across individual users.

Both variances $(\Sigma_p, \sigma_o)$ are either known or can be estimated from data. For example, we may assume that $\Sigma_p = \beta \mathbf{I}$ and tune $\beta$ through cross-validation.

## 3.2 Inference: Kalman Filtering

With the dynamic model described above, the user factor can be dynamically and efficiently estimated with Kalman filtering (KF). Basically, KF sequentially takes the ratings $\{\ldots, \mathbf{r}_{t-2}^{(u)}, \mathbf{r}_{t-1}^{(u)}, \mathbf{r}_t^{(u)}\}$ as observations and returns the optimal state estimate at time $t$, denoted by $\hat{\mathbf{p}}_{t|t}^{(u)}$, and the associated variance, denoted by $\Sigma_{t|t}^{(u)}$. In each KF step at time $t$, the estimate $(\hat{\mathbf{p}}_{t-1|t-1}^{(u)}, \Sigma_{t-1|t-1}^{(u)})$ from previous step is updated by incorporating the new observation $\mathbf{r}_t^{(u)}$ as follows

Function: $[\hat{\mathbf{p}}_{t|t}^{(u)}, \Sigma_{t|t}^{(u)}] = \text{KFupdate}(\mathbf{p}_{t-1|t-1}^{(u)}, \Sigma_{t-1|t-1}^{(u)}, \mathbf{r}_t^{(u)})$

**step 1: Time Update**

$$\hat{\mathbf{p}}_{t|t-1}^{(u)} \;=\; \hat{\mathbf{p}}_{t-1|t-1}^{(u)}, \quad \Sigma_{t|t-1}^{(u)} = \Sigma_{t-1|t-1}^{(u)} + \Sigma_p,$$

**step 2: Measurement Update**

$$\Sigma_{t|t}^{(u)} \;=\; \Sigma_{t|t-1}^{(u)} - \Sigma_{t|t-1}^{(u)} \mathbf{H}_t^T (\mathbf{H} \Sigma_{t|t-1}^{(u)} \mathbf{H}_t^T + \sigma_o^2 \mathbf{I})^{-1} \mathbf{H}_t \Sigma_{t|t-1}^{(u)}$$

$$\hat{\mathbf{p}}_{t|t}^{(u)} \;=\; \hat{\mathbf{p}}_{t|t-1}^{(u)} + \Sigma_{t|t}^{(u)} \mathbf{H}_t^T \sigma_o^{-2} (\mathbf{r}_t^{(u)} - \mathbf{H}_t \hat{\mathbf{p}}_{t|t-1}^{(u)}).$$

Although it is conceptually appealing to have a joint filtering model for both users and items, it renders the observation step (Equation (8)) nonlinear since it involves the dot product of the user factors and item factors. Nonlinear KF extensions such as Sigma-point Kalman filter [20] would have to be employed for this purpose. In this paper, we will focus on the linear KF, and assume the item factors are either known, or can be (dynamically) estimated by an external model (see however, Section 5).

# 4. SPATIO-TEMPORAL MODEL

In this section, we combine the modeling ideas in Sections 2 and 3 to obtain our ST-KF model that exploits both spatial and temporal correlations in the ratings.

## 4.1 Probabilistic Model

We first consider the spatio-temporal prior for the user factors $\mathbf{p}_t \equiv \{\mathbf{p}_t^{(1)}, \cdots, \mathbf{p}_t^{(N)}\}$. Let $\theta = \{\lambda, \alpha, \Sigma_p, \sigma_o\}$ specify all the parameters. At each time step $t$, the prior for $\mathbf{p}_t$ comes from two independent sources:

[1] For example, see Simon Funk's algorithm
http://sifter.org/ simon/journal/20061211.html

**Temporal Continuity:** This is expressed through the probability $p(\mathbf{p}_t|\mathbf{p}_{t-1};\theta)$, which penalizes a large deviation between $\mathbf{p}_t$ and its prediction at $t-1$. If we assume random walk dynamics as in (7), we have

$$\begin{aligned} p(\mathbf{p}_t|\mathbf{p}_{t-1};\theta) &= \prod_{u=1}^{N} p(\mathbf{p}_t^{(u)}|\mathbf{p}_{t-1}^{(u)};\theta) \\ &\propto \prod_{u=1}^{N} e^{-\frac{1}{2}(\mathbf{p}_t^{(u)}-\mathbf{p}_{t-1}^{(u)})^T \Sigma_p^{-1}(\mathbf{p}_t^{(u)}-\mathbf{p}_{t-1}^{(u)})}. \end{aligned}$$

**Spatial Similarity:** This is expressed through the time-varying Gaussian MRF prior $p(\mathbf{p}_t;\mathbf{W}_{p,t},\theta)$, where $\mathbf{W}_{p,t}$ specifies the user similarity at time $t$:

$$p(\mathbf{p}_t;\mathbf{W}_{p,t},\theta) \propto e^{-\frac{1}{2}\alpha\sum_{u,v} W_{p,t}^{(uv)}\|\mathbf{p}_t^{(u)}-\mathbf{p}_t^{(v)}\|^2},$$

where $W_{p,t}^{(uv)}$ is the $(u,v)$ entry of matrix $\mathbf{W}_{p,t}$.

The spatio-temporal prior is then given by the product

$$pr(\mathbf{p}_t) \propto p(\mathbf{p}_t|\mathbf{p}_{t-1};\theta)p(\mathbf{p}_t;\mathbf{W}_{p,t},\theta).$$

This type of spatio-temporal prior has been used in filtering tasks in other domains [15].

From the independence assumption, the likelihood of ratings given the user factors is

$$p(\mathbf{r}_t|\mathbf{p}_t;\theta) = \prod_{u=1}^{N} p(\mathbf{r}_t^{(u)}|\mathbf{p}_t^{(u)};\theta) \propto \prod_{u=1}^{N} e^{\frac{\|\mathbf{r}_t^{(u)}-\mathbf{H}_t^{(u)}\mathbf{p}_t^{(u)}\|^2}{2\sigma_o^2}} \tag{9}$$

Using $\mathbf{r}_t \equiv \{\mathbf{r}_t^{(1)},\cdots,\mathbf{r}_t^{(N)}\}$ to denote the observed rating at time $t$, the complete likelihood of $\{\mathbf{p}_\tau\}_{\tau=1}^t$ and $\{\mathbf{r}_\tau\}_{\tau=1}^t$ is given by

$$\begin{aligned} &p(\{\mathbf{p}_\tau\}_{\tau=1}^t,\{\mathbf{r}_\tau\}_{\tau=1}^t;\{\mathbf{W}_{p,\tau}\}_{\tau=1}^t,\theta) = \\ &\qquad \sum_{\tau=1}^{t} p(\mathbf{p}_\tau|\mathbf{p}_{\tau-1};\theta)p(\mathbf{r}_t|\mathbf{p}_\tau;\theta)p(\mathbf{p}_\tau;\mathbf{W}_{p,\tau},\theta) \end{aligned} \tag{10}$$

This model will be referred to as Spatio-Temporal filtering (ST-KF) in this paper. A pictorial illustration of ST-KF is given in Figure 1.

## 4.2 Inference

Due to the correlation introduced by $\mathbf{W}_{p,t}$, the estimate of each $\mathbf{p}_t^{(u)}$ cannot be done separately as in Section 3. The brute force implementation of KF update requires considering a concatenated state vector

$$\bar{\mathbf{p}}_t = [(\mathbf{p}_t^{(1)})^T \ \cdots \ (\mathbf{p}_t^{(N)})^T]^T.$$

and the corresponding covariance matrix $\bar{\Sigma}_t (\in \mathbb{R}^{Nk\times Nk})$ for the concatenated state vector. As for regular KF, we can estimate the probability $p(\bar{\mathbf{p}}_t|\{\mathbf{r}_\tau\}_{\tau=1}^t;\{\mathbf{W}_{p,\tau}\}_{\tau=1}^t,\theta)$ recursively.

$$\begin{aligned} \text{T-update:}\quad & \hat{\bar{\mathbf{p}}}_{t|t-1} = \hat{\bar{\mathbf{p}}}_{t-1|t-1},\ \bar{\Sigma}_{t|t-1} = \bar{\Sigma}_{t-1|t-1} + \Sigma_p \\ \text{M-update:}\quad & p(\bar{\mathbf{p}}_t|\{\mathbf{r}_\tau\}_{\tau=1}^t;\{\mathbf{W}_{p,\tau}\}_{\tau=1}^t,\theta) \\ & \qquad \propto p(\bar{\mathbf{p}}_t|\mathrm{est}_{t-1},\theta)p(\bar{\mathbf{p}}_t;\mathbf{W}_t,\theta)p(\mathbf{r}_t|\bar{\mathbf{p}}_t;\theta) \end{aligned}$$

where $\mathrm{est}_{t-1} = \{\hat{\bar{\mathbf{p}}}_{t|t-1},\bar{\Sigma}_{t|t-1}\}$ are the predicted mean and covariance of $\bar{\mathbf{p}}_t$ at time $t-1$.

To avoid the computation of inverse of a huge covariance matrix in the measurement update (see the function KFupdate in Section 3.2), we can use mean field approximation (MFA) [11] for inference at each time step, which turns out to be fairly cheap if $\mathbf{W}_{p,t}$ is sparse (e.g., a $k$NN graph). The naïve MFA uses the following fully factorized probability $q_t$

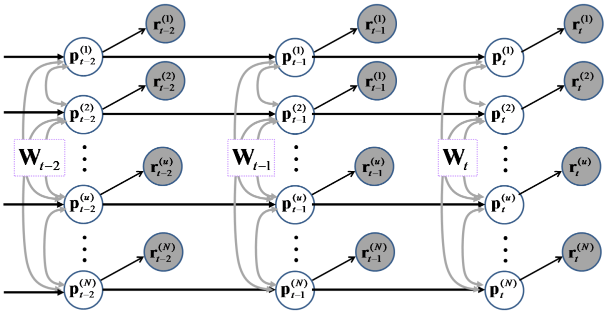

**Figure 1: The graphical model of ST-KF.**

$$q_t(\mathbf{p}_t) \equiv \prod_{u=1}^{N} q_t^{(u)}(\mathbf{p}_t^{(u)}).$$

to approximate $p(\bar{\mathbf{p}}_t|\{\mathbf{r}_\tau\}_{\tau=1}^t;\{\mathbf{W}_{p,\tau}\}_{\tau=1}^t,\theta)$. With this independence assumption, the posterior covariance can be reduced to individual ones

$$\bar{\Sigma}_{t|t} \to \{\Sigma_{t|t}^{(1)},\cdots,\Sigma_{t|t}^{(N)}\}.$$

We find $q_t(\mathbf{p}_t)$ with the minimum KL-divergence to the true posterior,

$$q_t^*(\mathbf{p}_t) = \arg\min_{q_t} D_{KL}(q_t(\mathbf{p}_t)\,||\,p(\mathbf{p}_t|\{\mathbf{r}_\tau\}_1^t;\{\mathbf{W}_{p,\tau}\}_1^t,\theta))$$

which can be recast as

$$\max_{q_t} H(q_t) + \mathbb{E}_{q_t}[\log p(\mathbf{p}_t|\{\mathbf{r}_\tau\}_1^t;\{\mathbf{W}_{p,\tau}\}_1^t,\theta)]. \tag{11}$$

where $\mathbb{E}_{q_t}[\cdot]$ stands for the expectation with respect to $q_t$. Generally (11) does not have a closed-form solution and is not even a convex problem. Instead a local optimum can be found iteratively with the following update equations: (for $u=1,2,\cdots,N$)

$$q_t^{(u)}(\mathbf{p}_t^{(u)}) \leftarrow \frac{1}{\Omega^{(u)}} e^{\mathbb{E}_{q_t}[\log p(\mathbf{p}_t|\{\mathbf{r}_\tau\}_1^t;\{\mathbf{W}_\tau\}_1^t,\theta)|\mathbf{p}_t^{(u)}]},$$

where $\mathbb{E}_{q_t}[\log p(\mathbf{p}_t|\{\mathbf{r}_\tau\}_1^t;\{\mathbf{W}_\tau\}_1^t,\theta)|\mathbf{p}_t^{(u)}]$ is the expectation of $\log p(\mathbf{p}_t|\{\mathbf{r}_\tau\}_1^t;\{\mathbf{W}_\tau\}_1^t,\theta)$ with respect to (the current estimate of) $q_t$ conditioned on $\mathbf{p}_t^{(u)}$, and

$$\Omega^{(u)} = \int_{\mathbf{p}_t^{(u)}} e^{\mathbb{E}_{q_t}[\log p(\mathbf{p}_t|\{\mathbf{r}_\tau\}_1^t;\{\mathbf{W}_\tau\}_1^t,\theta)|\mathbf{p}_t^{(u)}]} d\mathbf{p}_t^{(u)}$$

is the normalization constant for $q_t^{(u)}$. It is easy to verify that if the true posterior $p(\mathbf{p}_t|\{\mathbf{r}_\tau\}_1^t;\{\mathbf{W}_\tau\}_1^t,\theta)$ is a Gaussian, so is $q_t(\mathbf{p}_t)$. For more details of the mean field approximation, see the Appendix.

# 5. IMPLEMENTATION & ALGORITHMS

In this section we give some implementation details, and then pseudo-code for our ST-KF approach.

**Initialization**

We often need to perform a static matrix factorization (e.g., MF or SptMF) before the filtering, in order to obtain item factors as well as initial user factors. In practice it is often the case that we encounter new users and new items during the filtering process. For new users, we need to have an initial guess of user factors before seeing any ratings, and therefore the initial factors for new users will be obtained purely based on the covariates. One such solution can be naturally derived from the conditional distribution of the Markov random field prior, or equivalently the objective function in (6). Without loss of generality, we can assume that the $N^{th}$ user is new and has no ratings. It is easy to verify that given $\{\mathbf{p}^{(1)}, \cdots, \mathbf{p}^{(N-1)}\}$, the solution of $\mathbf{p}^{(N)}$ to (6) is simply a regularized version of nearest neighbors interpolation

$$\widetilde{\mathbf{p}}^{(N)} = \frac{\sum_{u\in\mathcal{U}(N)} W_p^{(uN)}\mathbf{p}^{(u)}}{\sum_{u'\in\mathcal{U}(N)} W_p^{(u'N)} + \lambda/\alpha}. \tag{12}$$

**New Items**

Handling new items is more involved since we assume item factors are known. The same difficulty arises when there are too few ratings for an item for reliable factors fitting, or the item factors also change with time. Here, we resort to the following approximation. At each time step $t$, we update the item factors $\mathbf{q}_t^{(i^*)}$ by finding an approximate solution to the following optimization

$$\mathbf{q}_t^{(i)} = \arg\min_{\mathbf{q}}\{\sum_{\tau=1}^{t}\sum_{u\in\mathcal{F}_\tau(i^*)} \frac{||r_\tau^{(ui^*)} - (\mathbf{p}_\tau^{(u)})^T\mathbf{q}||^2}{2\sigma_o^2} + \sum_{\tau=1}^{t}\sum_{j\in\mathcal{U}(i^*)} \frac{\alpha_i W_q^{(ij)}||\mathbf{q} - \mathbf{q}_\tau^{(j)}||^2}{2}\} \tag{13}$$

where the first term on the right hand side is the rating square error, and the second term is the regularization from the item similarity. $\mathcal{F}_\tau(i)$ stands for the indices of users who have rated item $i$ at time $\tau$, and $\mathcal{U}(i)$ contains the indices of the items that are nearest neighbors of item $i$ (according to $\mathbf{W}_q$) and also has been reliably estimated (with enough number of ratings of it). In practice, all the reliably estimated items are stored in a stack, which is also dynamically updated. It is easy to see that when there are no ratings for item $i$, the solution for $\mathbf{q}_t^{(i)}$ becomes a neighborhood interpretation that has the same form as (12)

## 5.1 Pseudo-code for ST-KF

```
Algorithm 1: Spatio-Temporal Filtering (ST-KF)
Input: $\mathbf{R}_0$, the ratings before time 0, and the sequence of
    ratings $\{\mathbf{R}_1, \cdots, \mathbf{R}_T\}$ to present.
Output: User factors $\{\mathbf{p}_t^{(1)}, \cdots, \mathbf{p}_t^{(N)}\}$ for $t = 1, \cdots, T$.
step 0: Initialization.
        1. Initialize user factors $\{\hat{\mathbf{p}}_{0|0}^{(1)}, \cdots, \hat{\mathbf{p}}_{0|0}^{(N)}\}$ and obtain
           $\{\mathbf{q}^{(1)}, \cdots, \mathbf{q}^{(M)}\}$ through a static matrix factorization over $\mathbf{R}_0$.
        2. $t \leftarrow 1$; Set initial variances $\{\mathbf{P}_{0|0}^{(u)}\}$ for all $u$.
step 1: Individual Kalman filtering
    (To initialize mean-field approximation.)
    for $i = 1 : N$
        $[\hat{\mathbf{p}}_{t|t}^{(u)}, \Sigma_{t|t}^{(u)}] = \mathsf{KFupdate}(\mathbf{p}_{t-1|t-1}^{(u)}, \Sigma_{t-1|t-1}^{(u)}, \mathbf{r}_t^{(u)})$.
    end
step 2: Mean Field Approximation
        $(\{\hat{\mathbf{p}}_t^{(u)}\}, \{\Sigma_{t|t}^{(u)}\}) = \mathsf{MFA}(\{\mathbf{p}_t^{(u)}\}, \{\Sigma_{t|t}^{(u)}\}, \mathbf{R}_t, \mathbf{W}_{p,t})$
step 3: Update the item factors and stack.
step 4: $t = t + 1$, go to step 1.

function:$[\{\hat{\mathbf{p}}_{t|t}^{(u)}\}, \{\Sigma_{t|t}^{(u)}\}] = \mathsf{MFA}(\{\mathbf{p}_{t|t}^{(u)}\}, \{\Sigma_{t|t}^{(u)}\}, \mathbf{R}_t, \mathbf{W}_{p,t})$
step 0: Initialization.
    for $u = 1 : N$
        $q^{(u)} \leftarrow \mathcal{N}(\hat{\mathbf{p}}_{t|t}^{(u)}, \Sigma_{t|t}^{(u)})$
    end
step 1: $q^{(u)}(\tilde{\mathbf{p}}_t^{(u)}) \leftarrow \frac{1}{\Omega_u} e^{\mathbb{E}_q[\log p(\{\tilde{\mathbf{p}}_t^{(u)}\}_1^N | \mathbf{r}_t^{(u)}, \mathbf{W}_{p,t}) | \tilde{\mathbf{p}}_t^{(u)}]}$,
    (see Appendix for details).
step 2: If converged, then return the mean and covariance
    for each $q^{(u)}$; otherwise go to step 1.
```

# 6. EXPERIMENTS

We tested the proposed algorithms on synthetic data, the MovieLens data, and Yahoo! Front Page data. On synthetic data and MovieLens, the task is to predict the missing entries based on the (sparsely) observed ones. The prediction performance is measured as the *root of mean square error* on test set,

$$\mathsf{RMSE} = \sqrt{\frac{\sum_{(u,i)\in test}(\hat{r}^{(ui)} - r^{(ui)})^2}{|test|}},$$

where *test* stands for the set for all the entries in test set. For Yahoo! Front Page data, the error is measured by the more relevant ROC curve for "click-or-not" prediction.

## 6.1 Synthetic Data

The experiments on synthetic data are designed to show that both spatial correlation and temporal structure individually help in finding more accurate user factors, and their strengths can be combined by using spatio-temporal filtering. For data, we generate a rank-5 100 × 50 rating matrix $\mathbf{R} = \mathbf{P}\mathbf{Q}^T$, where entries of $\mathbf{P}$ and $\mathbf{Q}$ are independently drawn from a uniform distribution on interval $[0, 1]$. In the MF model, we also set the number of factors to be 5.

**Spatial Model**

We considered two types of artificially generated similarities between users (Similar for item similarity generation):

- **Weak**: $W_p^{(uv)} = \max(0, \text{corr}(u, v))$ where $\text{corr}(u, v)$ is the Pearson correlation between $u$ and $v$ estimated from *all* the ratings.
- **Strong**: $W_p^{(uv)} = e^{-\frac{\|\mathbf{p}^{(u)} - \mathbf{p}^{(v)}\|^2}{\sigma^2}}$ with the *true* user factors $\mathbf{p}^{(u)}$ and $\mathbf{p}^{(v)}$. We keep only the 5 nearest neighbor for each node.

We randomly selected 10% of the entries in the rating matrix for training and used the remaining entries for testing. We follow the spatial objective function given in (6) and use stochastic gradient for optimization. Figure 2 shows the RMSE on test set achieved by different settings of $\lambda$ and $\alpha$ in training. We note that when using the weak similarity, the best performance is achieved with a balanced $\lambda$ and $\alpha$, while with strong similarity, the performance is the best when the Laplacian regularization term dominates.

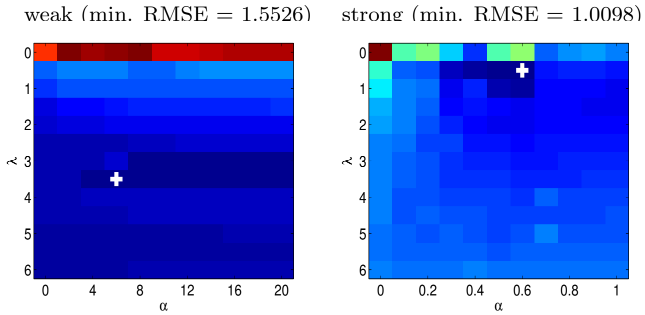

**Figure 2: The RMSE of SptMF with strong and weak similarity and different $(\lambda, \alpha)$. The white cross indicates the location of optimal $(\lambda, \alpha)$. With only Frobenius norm regularization ($\alpha = 0$), the minimum RMSE = 1.5862.**

### Temporal Model

We assume that the user factors change linearly with time,

$$\mathbf{p}_t^{(u)} = \mathbf{p}_0^{(u)} + t\mathbf{g}^{(u)} + \epsilon_t^{(u)}, \qquad \mathbf{g}^{(u)} \sim \mathcal{N}(0, 0.05\mathbf{I}), \qquad (14)$$

while the item factors keep unchanged. In this experiment we generated 10% of entries with the initial user factor $\mathbf{p}_0$, and then let the user factors evolve with time as in (14), while generating 2% of entries observable at each time step. The remaining 70% of the entries are generated with $\mathbf{p}_{10}$, but will be held out for testing.

To get $\mathbf{Q}$ and initial $\mathbf{P}$ with the static MF (step 0 in Algorithm 1), two strategies can be taken:

- **I:** use only the first 10% of the ratings. This corresponds to the scenario where little is known at the beginning and the model fitting has to be done in an online fashion.
- **II:** use all the observed 30% of ratings. This corresponds to the scenario where ratings over a significant time duration are available, and we need to retrace the change of user factors to obtain the most updated estimate.

Strategy II , if applicable, often works better since it usually gives a more accurate fitting of the item factors. One question associated with Strategy II, however, is "at which time step should we start evolving user factors?". Intuitively, we should go back to time 0 as we do with strategy I, but we observed that the results are often better when starting in the middle of the training duration (e.g. step 6 in this case). Figure 3 (left panel) compares the two initialization strategies as well as the two choices of starting points on the synthetic data. See Section 6.2 for similar results on MovieLens data.

### Spatio-Temporal Model

We take the same data from last experiment, but for each step, we also assume there is a noisy and sparse similarity measure for $\mathbf{p}_t$, generated as $W_{p,t}^{(uv)} = e^{-\frac{\|\mathbf{p}_t^{(u)} - \mathbf{p}_t^{(v)}\|^2}{\sigma^2}} + \epsilon_t^{(uv)}$ with a proper $\sigma$, where we also assume only 50% of the similarities (randomly chosen at each time step) are observed. We pruned the graph to be 5-NN, which further sparsifies the similarity $\mathbf{W}_{p,t}$.

In Figure 3 (right panel) we compared ST-KF and KF with two different initialization strategies. Clearly, under all configurations the filtering reduces the test RMSE from initial MF. With each of the initialization strategies, ST-KF outperforms KF by large margin.

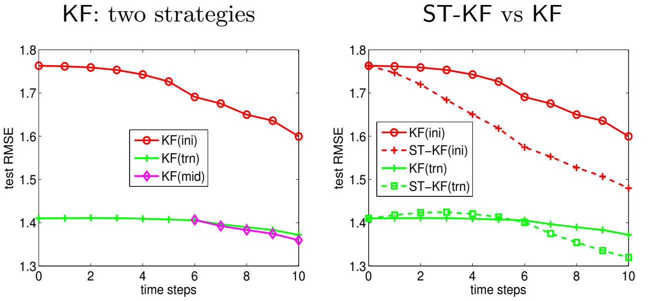

**Figure 3: Kalman filtering on synthetic data. Legend: (ini): initialize with the first 10% of the ratings; (trn): initialize with all the 30% of ratings collected in 10 time steps; (mid): initialization with all the 30% of ratings, but start filtering after step 6.**

## 6.2 MovieLens Data

We used the MovieLens data [2] with 6040 users, 3952 movies and around one million ratings. It is a commonly used data set for missing entry prediction, where typically some randomly selected entries are assumed to be observed and the rest are for testing. We argue that a more proper setting of the problem is to predict the ratings at time $t+1$ and after based on ratings obtained until time $t$. For the MovieLens data, we have ratings (from different users) for 1083 days, and we use the ratings in the first 420 days (95% of all the ratings) for training and the ratings in the remaining 663 days (5% of the ratings) for testing. This particular split of the data makes it much harder than what we know from [16, 1], in that (1) the user characteristic might have drifted away from it was in training phase after up to 20 months, and (2) only 813 users have ratings in the test set, and the particular training/testing split for those users is around 80/20.

For MovieLens data, in addition to the ratings, we also have the movie genres and the user demographic information: age, gender, occupation, and geo-location. We find a convex combination of user similarities composed from the four sources via a similarity-target alignment, as described in Section 2.2. We compared four different settings for this rating prediction task: (1) standard MF (for initialization) + standard KF (for filtering), (2) standard MF + ST-KF, (3) SptMF + standard KF, and finally (4) SptMF + ST-KF. We use all the data before day 420 for training and start the filtering at day 250 [3], as suggested in Section 6.1. The result is shown in Figure 4. We have the following three observations

- For the initial matrix factorization model, SptMF (test RMSE = 0.9073) outperforms standard MF (0.9163).
- Despite the limited temporal structure in the training set (420 days only), the time-domain filtering always reduces the test RMSE except for (MF + KF), with which the test RMSE keeps almost unchanged.

[2] http://www.grouplens.org/taxonomy/term/14

[3] The filtering starting at day 0 returns significantly worse result, see Figure 4 for details.

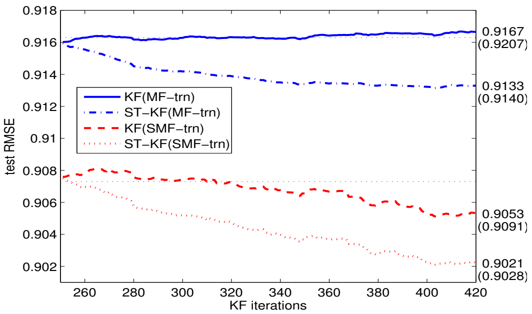

**Figure 4: The result on MovieLens. Legend: (MF-trn): initialize with a standard MF. (SMF-trn): initialize with SptMF. The numbers on the right y-axis indicate the test RMSE achieved at day 420, while the numbers in parentheses are the those obtained with starting point time 0.**

- Spatial prior helps the temporal filtering. It is obvious since (MF+ST-KF) beats (MF+KF), and (SptMF + ST-KF) beats (SptMF + KF), both by large margin.

Most saliently, the most sophisticated method (SptMF + ST-KF) reduces RMSE from the that of baseline matrix factorization method by 1.4% (0.9163 → 0.9022).

## 6.3 Yahoo! FrontPage Application

We now describe a relatively new recommender system application that arise in the context of Yahoo! Front Page. This application has been studied recently (see [4, 7] for a detailed description). The application involves recommending content items on a module (Today Module) that is published on Yahoo! Front Page to maximize overall click-rates. The module consists of four tabs (Featured, Entertainment, Sports and Video), our goal is to recommend the best stories from the available content pool to fill up the four slots on the featured tab. The available content in this application is programmed by human editors and typically consists of a few items (few tens) at any given point in time. However, the pool is dynamic and changes over time, editors push new stories and take out old ones. Hence, each content item has a short lifetime.

### Data Characteristics

There are several characteristics that makes this application different from movie recommender problem. First, short article lifetime means item factors have to be learnt online, the items in the test period have no overlaps with those in the training period. Second, we have rich meta-data on users which includes age, gender, geo-location, and browsing history. For items, we have content features in the form of content categories that are assigned by editors manually. We note that such problems are commonplace in web applications; sites like Digg, Top Picks on MSN, Yahoo! Buzz, Yahoo! Finance, HotFeeder etc face similar content recommendation problems.

We created a data set that consists of about 2M binary ratings (click or no click) for about 30K users over a six month period that covered about 4.3K items. Other than age, gender and geo-location, our user features include browsing behavior that are inferred based on a user's network wide activity (search, ad-clicks, page views, subscriptions, etc.) on the Yahoo! portal. In fact, a user is assigned an intensity score in a few thousand content categories based on his/her activity pattern in the recent past; we reduced these to a few hundred features by conducting a principal components analysis on the training data.

### Algorithm Details

For Yahoo! Front Page data, besides the user factor and item factor, we also considered a time-varying overall bias $\mu$, the user bias $b$, and the news popularity bias $c$. So at time $t$, the rating (click it or not) is modeled as

$$r_t^{(ui)} = \mu_t + b_t^{(u)} + c_t^{(i)} + (\mathbf{p}_t^{(u)})^T \mathbf{q}^{(i)} + \epsilon_t^{(ui)}, \qquad (15)$$

where $\mu_t$ , $b_t^{(u)}$, and $c_t^{(i)}$ are respectively the overall bias, the bias of user $u$ and the bias of item $i$ at time $t$. The user bias and the popularity bias can be expressed with same rating model in (1 ) with the following extension. Let

$$\bar{\mathbf{p}}_t^{(u)} = [(\mathbf{p}_t^{(u)})^T \;\; b_t^{(u)} \;\; 1]^T, \quad \bar{\mathbf{q}}_t^{(i)} = [(\mathbf{q}_t^{(i)})^T \;\; 1 \;\; c_t^{(i)}]^T,$$

then it is easy to see $(\bar{\mathbf{p}}_t^{(u)})^T \bar{\mathbf{q}}^{(i)} = b_t^{(u)} + c_t^{(i)} + (\mathbf{p}_t^{(u)})^T \mathbf{q}^{(i)}$.

### Results on Y! Front Page Data

To illustrate the performance of ST-KF, we compare against three baseline models that are (a) Cov-Only: This is a linear regression based only on user and item covariates. (b) Fact-Only: This is the usual matrix factorization model without any spatial or temporal smoothing. Since most of the items in the test set are new, the item popularity and item factors are estimated as zeros and hence this reduces to a model that is based only on global and user popularity terms, i.e., $\mu_t + b_t^{(u)}$. (c) Item-Cov: In this model, we estimate the user and item factors on the training data using the usual matrix factorization model. To avoid cold-start in the test data, we further estimate a linear regression model based on user and item covariates to estimate the respective factors obtained from matrix factorization. On the test set, we use the regression to predict factors for new user/item. Figure 5 shows the ROC curves comparing the methods. First, all our methods are better than the straw man baseline that predicts a constant score for all test cases, the curve for this model is given by the 45 degree straight line. Using covariates alone does not provide good performance, showing the need to learn user and item specific models for this application; in fact, the Fact-Only model is better than Cov-Only. As expected, the Item-Cov model is better than Fact-Only; however, incorporating both spatial and temporal smoothing through ST-KF provides the best performance.

## 7. RELATED WORK AND DISCUSSION

There has been some work on incorporating user and item covariates or side-information into collaborative filtering. The most well-studied direction is to use kernel-based classification or regression models, where the covariates-based kernel serves either as a "basic" kernel in a kernel combination [6, 1] or as an initial kernel for later kernel fitting [21]. Another direction is to treat collaborative filtering as a parameterized regression problem, where the covariates (or features induced from other side-information) become part of the regression parameters [5]. The work that most resembles our spatial model is proposed independently in [14] for relational data analysis, where the links between entities are treated

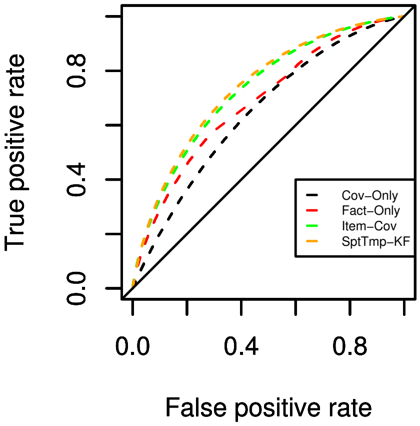

**Figure 5: ROC curves comparing different methods on Y! FP data**

as the auxiliary similarity used to regularize a matrix factorization model.

Incorporating temporal component and covariate into recommender system problems have been studied very recently in several papers [4, 3, 2]. Of these, [10] is most related to our work where the authors regularize the factors through a regression and capture temporal variation through random walk priors on each parameter. We note that it is easy to incorporate their model in our framework by replacing $\lambda(||\mathbf{P}||_F^2 + ||\mathbf{Q}||_F^2)$ in Equation 6 with $\lambda(||\mathbf{P} - \mathbf{G}\mathbf{X}_p||_F^2 + ||\mathbf{Q} - \mathbf{F}\mathbf{X}_q||_F^2)$ where $\mathbf{G}$ and $\mathbf{F}$ are unknown regression coefficient matrices, and $\mathbf{X}_p$, $\mathbf{X}_q$ are user and item covariate vectors respectively. Our spatio-temporal prior provides additional regularization by inducing dependencies in the factors a-prior through a Markov random field.

## 8. CONCLUSIONS

We presented an efficient spatio-temporal approach to collaborative filtering, and showed its efficacy on both synthetic and real-world data sets. Our future work will focus on better models for joint and dynamical estimation of user factors and item factors, including bilinear filtering model based on Sigma-point Kalman filter, and parameterized regression models as in [10].

## 9. ACKNOWLEDGMENTS

This research is partially supported by NSF grant CCF-0431257 and a gift from Yahoo! research. ZL is supported by an ICES postdoctoral fellowship at UT Austin.

## 10. REFERENCES

[1] J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert. A new approach to collaborative filtering: Operator estimation with spectral regularization. *JMLR*, 2009.

[2] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *KDD*, 2009.

[3] D. Agarwal, B.-C. Chen, and P. Elango. Spatio-temporal models for estimating click-rate. In *WWW*, 2009.

[4] D. Agarwal, B.-C. Chen, P. Elango, R. Ramakrishnan, N. Motgi, S. Roy, and J. Zachariah. Online models for content optimization. In *NIPS(21)*, 2009.

[5] D. Agarwal and S. Merugu. Predictive discrete latent factor models for large scale dyadic data. In *KDD*, 2007.

[6] J. Basilico and T. Hofmann. Unifying collaborative and content-based filtering. In *ICML*, 2004.

[7] D. Chakrabarti, D. Agarwal, and V. Josifovski. Contextual advertising by combining relevance with click feedback. In *WWW*, 2008.

[8] C. Chui and G. Chen. *Kalman Filtering for Real Time Application*. Springer-Verlag, 1999.

[9] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor. On kernel-target. In *NIPS(14)*, 2002.

[10] D.Stern, R.Herbrich, and G.Thore. Matchbox: Large scale online Bayesian recommendations. In *WWW*, 2009.

[11] T. S. Jaakkola. Tutorial on variational approximation methods. In *Advanced Mean Field Methods: Theory and Practice*, pages 129–159. MIT Press, 2000.

[12] J.Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Roy. Stat. Soc. B*, 36(2):192–236, 1974.

[13] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS(12)*, 2000.

[14] W. Li and D. Y. Relation regularized matrix factorization. In *IJCAI*, 2009.

[15] Z. Lu, M. A. Carreira-Perpinan, and C. Sminchisescu. People tracking with the Laplacian eigenmaps latent variable model. In *NIPS(20)*, 2008.

[16] B. M. Marlin, R. S. Zemel, S. Roweis, and M. Slaney. Collaborative filtering and the missing at random assumption. In *UAI*, 2007.

[17] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS(20)*, 2008.

[18] N. Srebro and T. Jaakkola. Weighted low-rank approximations. In *ICML*, 2003.

[19] N. Srebro, J. Rennie, and T. Jaakkola. Maximum margin matrix factorization. In *NIPS(17)*, 2005.

[20] R. van der Merwe. *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models.* PhD thesis, OGI, OHSU, 2004.

[21] K. Yu and W. Chu. Gaussian process models for link analysis and transfer learning. In *NIPS(19)*, 2007.

# APPENDIX

## Mean Field Approximation

From Bayes rule we have the Markovian property

$$p(\mathbf{p}_t|\{\mathbf{r}_n\}_1^t; \{\mathbf{W}_{p,\tau}\}_1^t, \theta) \propto p(\mathbf{p}_t; \mathrm{est}_{t-1}, \theta) p(\mathbf{r}_t|\mathbf{p}_t; \theta) p(\mathbf{p}_t; \mathbf{W}_{p,t}, \theta).$$

Also we assume the temporal prior factorizes,

$$p(\mathbf{p}_t; \mathrm{est}_{t-1}, \theta) = \prod_{u=1}^{N} p(\mathbf{p}_t^{(u)}; \mathrm{est}_{t-1}^{(u)}, \theta),$$

which is a result of the MFA at $t-1$. Easy to verify

$$\mathbb{E}_{q_t}[\log p(\mathbf{p}_t|\{\mathbf{r}_n\}_1^t; \{\mathbf{W}_{p,\tau}\}_1^t, \theta)|\mathbf{p}_t^{(u)} = \log p(\mathbf{p}_t^{(u)}; \mathrm{est}_{t-1}, \theta)$$
$$+ \log p(\mathbf{r}_t^{(u)}|\mathbf{p}_t^{(u)}; \theta) + \mathbb{E}_{q_t}[\log p(\mathbf{p}_t; \mathbf{W}_{p,t}, \theta)|\mathbf{p}_t^{(u)}] + c,$$

where $c$ is a constant. Easy to see that for each $u$

$$\log p(\mathbf{p}_t^{(u)}; \mathrm{est}_{t-1}, \theta) =$$
$$c_1 - \frac{1}{2}(\mathbf{p}_t^{(u)} - \hat{\mathbf{p}}_{t|t-1}^{(u)})^T (\Sigma_{t-1|t-1}^{(u)} + \Sigma_p)^{-1} (\mathbf{p}_t^{(u)} - \hat{\mathbf{p}}_{t|t-1}^{(u)});$$
$$\log p(\mathbf{r}_t^{(u)}|\mathbf{p}_t^{(u)}; \theta) = c_2 - \frac{\|\mathbf{r}_t^{(u)} - \mathbf{H}_t^{(u)}\mathbf{p}_t^{(u)}\|^2}{2\sigma_o^2};$$
$$\mathbb{E}_{q_t}[\log p(\mathbf{p}_t; \mathbf{W}_{p,t}, \theta)|\mathbf{p}_t^{(u)}] = c_3 - \frac{\alpha}{2}\sum_{v \neq u} W_t^{(uv)} (\mathbf{q}_t^{(u)} - \mathbb{E}_{q_t^{(v)}}[\mathbf{p}_t^{(v)}]).$$

When $\mathbf{W}_{p,t}$ is given by a $k$NN graph, $\mathbb{E}_{q_t}[\log p(\mathbf{p}_t; \mathbf{W}_{p,t}, \theta)$ $|\mathbf{p}_t^{(u)}]$ can be evaluated efficiently as

$$c_3 - \frac{\alpha}{2}(\sum_{v \in \mathcal{U}(u)} W_{p,t}^{(uv)} \mathbf{p}^{(u)} - \sum_{v \in \mathcal{U}(u)} W_t^{(uv)} \mathbb{E}_{q_t^{(v)}}[\mathbf{p}_t^{(v)}])$$

where $\mathcal{U}(u)$ is the set of nearest neighbors of $u$.