Spatial Processes for Recommender Systems

Fabian Bohnert, Daniel F. Schmidt, and Ingrid Zukerman
Clayton School of Information Technology
Faculty of Information Technology, Monash University
Clayton, VIC 3800, Australia
{fabianb,dschmidt,ingrid}@infotech.monash.edu.au

Abstract

Spatial processes are typically used to analyse and predict geographic data. This paper adapts such models to the prediction of a user's interests or item ratings in recommender systems. We present the theoretical framework for a model based on Gaussian spatial processes, and discuss efficient algorithms for parameter estimation. Our model was evaluated with simulated data and a real-world dataset collected by tracking visitors in a museum, and achieves a higher predictive accuracy than a non-personalised baseline. Additionally, in the real-world scenario, the model attains a higher predictive accuracy than state-of-the-art collaborative filters.

1 Introduction

Recommender systems (RS) are designed to direct users to personally interesting items in situations where the amount of available information exceeds the users' processing capability [Resnick and Varian, 1997; Burke, 2002]. Typically, such systems (1) use information about a user to predict ratings of items that the user has not yet considered, and (2) recommend suitable items based on these predictions. *Collaborative* modelling techniques constitute one of the main model classes applied in RS [Albrecht and Zukerman, 2007]. They base their predictions upon the assumption that users who have agreed in their behaviour in the past will agree in the future.

The greatest strength of collaborative approaches is that they are independent of any representation of the items being recommended, and work well for complex objects, for which features are not readily apparent. The two main collaborative approaches are *memory-based* and *model-based*. Previous research has mainly focused on memory-based approaches, such as nearest-neighbour models (classic collaborative filtering), e. g., [Resnick *et al.*, 1994; Herlocker *et al.*, 1999], due to their intuitiveness. The main drawback of memory-based algorithms is that they operate over the entire user database to make predictions. In contrast, model-based approaches use techniques such as Bayesian networks, latent-factor models and artificial neural networks, e. g., [Breese *et al.*, 1998; Bell *et al.*, 2007], to first learn a statistical model in an offline fashion, and then use it for prediction and recommendation. This decomposition can significantly speed up the recommendation generation process.

Spatial processes (*random fields*) are a subclass of *stochastic processes* which are applied to domains that have a geospatial interpretation, e. g., [Diggle *et al.*, 1998; Banerjee *et al.*, 2004]. Typical tasks of the field of spatial statistics include modelling spatial associations between a set of observations made at certain locations, and predicting values at locations where no observations have been made.

This paper applies theory from the area of spatial statistics to the task of predicting a user's interests or item ratings in RS. We first develop a model, called *Spatial Process Model (SPM)*, which adapts a Gaussian spatial process model to the recommendation scenario. Our model is similar to (but simpler than) the Gaussian process model for preference prediction described by Schwaighofer *et al.* [2005]. We then propose a Bayesian approach based on slice Gibbs sampling [Damien *et al.*, 1999; Neal, 2003] to estimate the parameters of our model. The use of spatial processes requires a measure of distance between items in addition to user ratings. This measure is non-specific (e. g., it may be a physical or a conceptual distance), and can be readily obtained in most cases.

Our approach offers advantages over other model-based approaches in the sense that, unlike neural networks (and also memory-based techniques), our model returns the confidence in a prediction, and its parameters have a clear interpretation; unlike Bayesian networks, our model does not require a domain-specific adaptation, such as designing the network topology. In addition, the use of a distance measure endows our model with capabilities of hybrid RS [Burke, 2002; Albrecht and Zukerman, 2007] by seamlessly supporting the incorporation of other types of models (e. g., content-based). The distance measure also alleviates the *cold-start problem*. The *new-item problem* is addressed by utilising the (distance-based) correlation between this item and the other items. The *new-user problem* is similarly handled through the correlation between items rated by a user and the other items (our model can make useful personalised predictions after only one item has been rated).

Our model was evaluated with simulated data and a real-world dataset of time spans spent by museum visitors at exhibits (which we view as im-

plicit ratings). We compared the model's performance to that of (1) a baseline model which delivers a non-personalised prediction, and (2) a state-of-the-art nearest-neighbour collaborative filter incorporating performance-enhancing modifications, e. g., [James and Stein, 1961; Herlocker *et al.*, 1999]. Our results show that *SPM* outperforms both models.

The paper is organised as follows. Our spatial processes approach for modelling and predicting item ratings is described in Section 2. In Sections 3 and 4, we present the results of our model validation and evaluation, followed by our conclusions in Section 5.

2 Using Spatial Processes to Model and Predict Item Ratings in Recommender Systems

We briefly introduce stationary spatial process models (Section 2.1), before adapting a model based on Gaussian spatial processes to the RS scenario (Section 2.2). In Section 2.3, we describe an MCMC-based Bayesian approach for estimating the parameters of the model, and in Section 2.4 we outline the theory for predicting item ratings.

2.1 Stationary Spatial Process Models

Let $\boldsymbol{Y} = (Y(\boldsymbol{s}_1), \ldots, Y(\boldsymbol{s}_n))$ be a vector of observations at n sites $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$. Assuming stationarity both in the mean and variance, we can define the following basic model to capture spatial associations:

$$Y(\boldsymbol{s}_i) = \mu + \sigma w(\boldsymbol{s}_i) + \varepsilon(\boldsymbol{s}_i) \text{ for all } i = 1, \ldots, n,$$

where $w(\boldsymbol{s}_i)$ are assumed to be realisations from a stationary Gaussian spatial process $W(\boldsymbol{s})$ with mean 0, variance 1, and isotropic[1] correlation function $\rho(\|\boldsymbol{s}_i - \boldsymbol{s}_j\|; \phi, \nu)$ capturing residual spatial association; and $\varepsilon(\boldsymbol{s}_i)$ are realisations from a white-noise process with mean 0 and variance τ^2, i. e., non-spatial uncorrelated error terms. That is, we assume that $\boldsymbol{Y} = (Y(\boldsymbol{s}_1), \ldots, Y(\boldsymbol{s}_n))$ are observations from a stationary Gaussian spatial process over $\boldsymbol{s}$ with mean μ, variance $\sigma^2 + \tau^2$, and correlation function $\rho(\|\boldsymbol{s}_i - \boldsymbol{s}_j\|; \phi, \nu)$. Generally, correlation is assumed to approach 0 with increasing distance $\|\boldsymbol{s}_i - \boldsymbol{s}_j\|$. Common choices for isotropic correlation functions are the powered exponential

$$\rho(\|\boldsymbol{s}_i - \boldsymbol{s}_j\|; \phi, \nu) = \exp\left(-\left(\phi\|\boldsymbol{s}_i - \boldsymbol{s}_j\|\right)^{\nu}\right), \qquad (1)$$

where $\phi > 0$ and $0 < \nu < 2$, or correlation functions from the Matérn class [Banerjee *et al.*, 2004].[2]

The model provides a generic framework for modelling spatial associations between locations. Given estimates for μ, σ^2 and τ^2, it can make predictions for new locations. Sometimes, the observations $\boldsymbol{Y}$ would not naturally be modelled using a Gaussian distribution. For instance, they might be discrete binary variables indicating like/dislike. In a fashion similar to the way that generalised linear models extend classical Gaussian linear models, Diggle *et al.* [1998] formulate a hierarchical model for non-Gaussian observations, extending a more sophisticated version of the basic model above. That is, they replace the Gaussian model for $\boldsymbol{Y}$ by a different, more suitable member of the class of exponential models. Refer to further literature for a more thorough discussion, e. g., [Diggle *et al.*, 1998].

2.2 Adaptation for Recommender Systems

RS help users find interesting information in a space of many options. Typically, the task is to identify items that suit the needs of a particular user given some evidence about his/her preferences. A collaborative filter, for example, utilises a set of ratings $\boldsymbol{Y}$ of users $U = \{u : u = 1, \ldots, m\}$ regarding a set of items $I = \{i : i = 1, \ldots, n\}$ to identify users who are similar to the current user. It then predicts this user's ratings on the basis of the ratings of the most similar users. Ratings of different users are usually considered to be independent, whereas ratings of related items tend to be correlated. Introducing a notion of spatial distance between items in order to functionally specify this correlation structure, we can use spatial process models (Section 2.1) for the prediction task, in a fashion similar to the Gaussian process model for preference prediction described by Schwaighofer *et al.* [2005]. The assumption made for spatial processes, that correlation between observations increases with decreasing site distance, fits well with RS, where ratings are usually more correlated the closer (i. e., more related) items are. As for the previous section, we use $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$ to denote the locations of items $i, j = 1, \ldots, n$ in a space providing such a distance measure, i. e., $\|\boldsymbol{s}_i - \boldsymbol{s}_j\|$. For example, $\|\boldsymbol{s}_i - \boldsymbol{s}_j\|$ could be computed from feature vectors representing the items, similarly to content-based RS, or from item-to-item similarities, similarly to item-to-item collaborative filtering [Sarwar *et al.*, 2001].

The following changes are necessary to adapt the model presented in the previous section to RS:

- **Multiple ratings.** Given a set of users U, we can have multiple (but independent) ratings for a given item i.
- **Non-stationarity.** Different items can have different rating means and variances. Hence, the underlying process cannot be assumed to be stationary in its mean and variance, as both μ and σ^2 depend on an item's location $\boldsymbol{s}$. We use the notation $\mu(\boldsymbol{s})$ and $\sigma^2(\boldsymbol{s})$ to indicate this.
- **Item set finiteness.** In contrast to traditional geospatial modelling, we require predictions only for a finite set of items, i. e., those at locations $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$. Hence, it is sufficient (and necessary) to know $\mu(\boldsymbol{s})$ and $\sigma^2(\boldsymbol{s})$ only at these locations. That is, we do not require a special (functional) structure for $\mu(\boldsymbol{s})$ or $\sigma^2(\boldsymbol{s})$, which is usually the case for geospatial models.[3]

[1] A correlation function is isotropic if it depends on the separation vector $\boldsymbol{s}_i - \boldsymbol{s}_j$ only through its length $\|\boldsymbol{s}_i - \boldsymbol{s}_j\|$.

[2] In our experiments (Sections 3 and 4), we use a powered exponential correlation function, as the Matérn class yielded inferior results.

[3] In order to compute predictions for a new item i without ratings, $\mu(\boldsymbol{s}_i)$ and $\sigma^2(\boldsymbol{s}_i)$ must be externally estimated

For a single user with rating vector $\boldsymbol{Y}$ with components $(\boldsymbol{Y})_i = Y(\boldsymbol{s}_i)$, we extend the basic model from Section 2.1 as follows. We set $Y(\boldsymbol{s}_i)$ to be observations from a **non-stationary** spatial process, i. e., $Y(\boldsymbol{s}_i) = \mu(\boldsymbol{s}_i) + \sigma(\boldsymbol{s}_i)w(\boldsymbol{s}_i) + \varepsilon(\boldsymbol{s}_i)$. Exploiting **item set finiteness**, $\mu(\boldsymbol{s}_i)$ and $\sigma(\boldsymbol{s}_i)$ are respectively components of $\boldsymbol{\mu} = (\mu(\boldsymbol{s}_1), \ldots, \mu(\boldsymbol{s}_n))$, the vector of mean ratings for items $1, \ldots, n$, and $\boldsymbol{\sigma} = (\sigma(\boldsymbol{s}_1), \ldots, \sigma(\boldsymbol{s}_n))$, the vector of standard deviations. Let $\boldsymbol{\theta} = \left(\boldsymbol{\mu}, \boldsymbol{\sigma}, \tau^2, \phi, \nu\right)$ be a vector collecting all model parameters, and $\boldsymbol{W} = (w(\boldsymbol{s}_1), \ldots, w(\boldsymbol{s}_n))$. Then,

$$\boldsymbol{Y} \,|\, \boldsymbol{\theta}, \boldsymbol{W} \sim \mathcal{N}\left(\boldsymbol{\mu} + \boldsymbol{\sigma} 1_n \boldsymbol{W}, \tau^2 1_n\right),$$

where 1_n is the identity matrix of dimension $n \times n$. Note that $\boldsymbol{Y} = (Y(\boldsymbol{s}_1), \ldots, Y(\boldsymbol{s}_n))$ are mutually independent given $\boldsymbol{\theta}$ and $\boldsymbol{W}$. Marginalising the model over W, we obtain[4]

$$\boldsymbol{Y} \,|\, \boldsymbol{\theta} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\sigma} 1_n H(\phi, \nu) \boldsymbol{\sigma} 1_n + \tau^2 1_n\right), \qquad (2)$$

where $H(\phi, \nu)$ denotes a correlation matrix with components $(H(\phi, \nu))_{ij} = \rho(\|\boldsymbol{s}_i - \boldsymbol{s}_j\|; \phi, \nu)$. That is, $(H(\phi, \nu))_{ij}$ represents the correlation between the ratings for items i and j.

We are ready to generalise the model to the **multi-user** case. As above, we denote the set of users with U (cardinality m), and the set of items with I (cardinality n). Typically, for a user u in U, we have ratings for only a subset of I, say for n_u items in I. Denoting a rating by user u for item i with $Y_u(\boldsymbol{s}_i)$ and a user's rating vector with $\boldsymbol{Y}_u$, we collect all observed ratings into a vector $\boldsymbol{Y} = (\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_m)$ of dimension $\sum_{u=1}^{m} n_u$. Similarly, we structure $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ such that $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_m)$ and $\boldsymbol{\sigma} = (\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_m)$, where $\boldsymbol{\mu}_u$ and $\boldsymbol{\sigma}_u$ are the vectors of means and standard deviations for those items rated by a user u, respectively. For example, assume that $U = \{1, 2\}$ and $I = \{1, 2, 3\}$. If user 1 rated items 2 and 3, and user 2 rated items 1 and 2, then

$$\begin{aligned}
\boldsymbol{Y} &= (\boldsymbol{Y}_1, \boldsymbol{Y}_2) = (Y_1(\boldsymbol{s}_2), Y_1(\boldsymbol{s}_3); Y_2(\boldsymbol{s}_1), Y_2(\boldsymbol{s}_2)), \\
\boldsymbol{\mu} &= (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = (\mu(\boldsymbol{s}_2), \mu(\boldsymbol{s}_3); \mu(\boldsymbol{s}_1), \mu(\boldsymbol{s}_2)), \\
\boldsymbol{\sigma} &= (\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2) = (\sigma(\boldsymbol{s}_2), \sigma(\boldsymbol{s}_3); \sigma(\boldsymbol{s}_1), \sigma(\boldsymbol{s}_2)).
\end{aligned}$$

Similarly to Equation 2, $\boldsymbol{Y} \,|\, \boldsymbol{\theta}$ is multivariate normal of dimension $\sum_{u=1}^{m} n_u$, where $H(\phi, \nu)$ is block diagonal with diagonal elements $H_1(\phi, \nu), \ldots, H_m(\phi, \nu)$ (due to users $u = 1, \ldots, m$ being independent), i. e.,

$$H(\phi, \nu) = \begin{bmatrix} H_1(\phi, \nu) & & 0 \\ & \ddots & \\ 0 & & H_m(\phi, \nu) \end{bmatrix},$$

and $H_u(\phi, \nu)$ denotes the correlation matrix of dimension $n_u \times n_u$ for user u. That is, for all users $u = 1, \ldots, m$,

$$\boldsymbol{Y}_u \,|\, \boldsymbol{\theta} \sim \mathcal{N}\left(\boldsymbol{\mu}_u, \boldsymbol{\sigma}_u 1_{n_u} H_u(\phi, \nu) \boldsymbol{\sigma}_u 1_{n_u} + \tau^2 1_{n_u}\right). \quad (3)$$

Thus, given $\boldsymbol{\theta} = \left(\boldsymbol{\mu}, \boldsymbol{\sigma}, \tau^2, \phi, \nu\right)$, our model is fully specified (for $\boldsymbol{\theta}$, we set $\boldsymbol{\mu} = (\mu(\boldsymbol{s}_1), \ldots, \mu(\boldsymbol{s}_n))$ and $\boldsymbol{\sigma} = (\sigma(\boldsymbol{s}_1), \ldots, \sigma(\boldsymbol{s}_n))$).

and supplied to our model, until they can be estimated from observed data.

[4]Marginalisation over W is possible only in the Gaussian case, not in the more general case described in [Diggle *et al.*, 1998].

2.3 Parameter Estimation

This section describes efficient algorithms for estimating the $2n + 3$ model parameters $\boldsymbol{\theta} = \left(\boldsymbol{\mu}, \boldsymbol{\sigma}, \tau^2, \phi, \nu\right)$. The most popular parameter estimation strategies are maximum-likelihood and Bayesian inference. We opt for a Bayesian solution, as it offers some attractive advantages over the classic frequentist approach. For instance, prior knowledge can be formally incorporated into parameter estimation via the prior distribution $p(\boldsymbol{\theta})$, and the uncertainty about the parameters $\boldsymbol{\theta}$ is captured by the posterior distribution. Parameter estimates for $\boldsymbol{\theta}$ can be obtained from the posterior distribution

$$p(\boldsymbol{\theta}|\boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\boldsymbol{Y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}} \propto p(\boldsymbol{Y}|\boldsymbol{\theta})p(\boldsymbol{\theta}), \quad (4)$$

where $p(\boldsymbol{Y}|\boldsymbol{\theta})$ is the likelihood of $\boldsymbol{Y}$ given $\boldsymbol{\theta}$. Typically, independent priors are chosen for the different parameters, i. e., in our case, $p(\boldsymbol{\theta}) = p(\boldsymbol{\mu})p(\boldsymbol{\sigma})p(\tau^2)p(\phi)p(\nu)$.

The integrations required to calculate $p(\boldsymbol{\theta}|\boldsymbol{Y})$ in Equation 4 are generally not tractable in closed form. However, $p(\boldsymbol{\theta}|\boldsymbol{Y})$ can be approximated numerically using *Markov chain Monte Carlo (MCMC)* integration methods, such as the *Metropolis-Hastings algorithm* and the *Gibbs sampler*.[5] Following Banerjee *et al.* [2004], we use *slice sampling* [Damien *et al.*, 1999; Neal, 2003], i. e., a *slice Gibbs sampler*, to sample from the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{Y})$. This approach is favoured by Banerjee *et al.*, because it does not require tuning that is tailored to the application, and hence provides an automatic MCMC algorithm for fitting Gaussian spatial process models. Algorithm 1 summarises our sampling procedure. The algorithm consists of two iteratively applied steps: (1) slicing the likelihood, and (2) performing Gibbs updates using draws from the prior along with rejection sampling. Formally, if $L(\boldsymbol{\theta}; \boldsymbol{Y}) = p(\boldsymbol{Y}|\boldsymbol{\theta})$ denotes the likelihood function, we introduce the auxiliary variable U, such that U is uniformly distributed on $(0, L(\boldsymbol{\theta}; \boldsymbol{Y}))$ given $\boldsymbol{\theta}$ and $\boldsymbol{Y}$, i. e., $U \,|\, \boldsymbol{\theta}, \boldsymbol{Y} \sim \mathcal{U}(0, L(\boldsymbol{\theta}; \boldsymbol{Y}))$. The joint posterior distribution of $\boldsymbol{\theta}$ and U is then given by

$$p(\boldsymbol{\theta}, U|\boldsymbol{Y}) \propto p(\boldsymbol{\theta})\, \mathbb{1}\left(U < L(\boldsymbol{\theta}; \boldsymbol{Y})\right), \qquad (5)$$

where $\mathbb{1}$ denotes the indicator function. From Equation 5 it is obvious that U slices the likelihood, defining a slice $S = \{\boldsymbol{\theta} : U < L(\boldsymbol{\theta}; \boldsymbol{Y})\}$ for $\boldsymbol{\theta}$ (the name *slice sampling* derives from this fact). Running a Gibbs sampler drawing $U \,|\, \boldsymbol{\theta}, \boldsymbol{Y}$ followed by $\boldsymbol{\theta} \,|\, U, \boldsymbol{Y}$ at each iteration, we can obtain samples from $p(\boldsymbol{\theta}, U|\boldsymbol{Y})$, and hence from the marginal posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{Y})$ of $\boldsymbol{\theta}$. Drawing $U \,|\, \boldsymbol{\theta}, \boldsymbol{Y}$ is routine, and a component θ_k of $\boldsymbol{\theta}$ is updated by drawing from its prior $p(\theta_k)$ restricted to S, given the other components of $\boldsymbol{\theta}$, and U and $\boldsymbol{Y}$. It is generally advisable to compute the negative log-likelihood $l(\boldsymbol{\theta}; \boldsymbol{Y}) = -\log L(\boldsymbol{\theta}; \boldsymbol{Y})$ rather than $L(\boldsymbol{\theta}; \boldsymbol{Y})$ in order to increase numerical stability. To implement this strategy, we can use the auxiliary variable $V = -\log U$ instead of U. Then, $(V - l(\boldsymbol{\theta}; \boldsymbol{Y}))$ is exponentially distributed with mean 1 given $\boldsymbol{\theta}$ and $\boldsymbol{Y}$, i. e., $(V - l(\boldsymbol{\theta}; \boldsymbol{Y})) \,|\, \boldsymbol{\theta}, \boldsymbol{Y} \sim \mathrm{Exp}(1)$. This transforms the slice S into $S = \{\boldsymbol{\theta} : l(\boldsymbol{\theta}; \boldsymbol{Y}) < V\}$.

[5]Refer to [Andrieu *et al.*, 2003] for an excellent introduction to MCMC for machine learning.

Algorithm 1 Slice Gibbs sampling algorithm

Initialise $\boldsymbol{\theta}$, e. g., by drawing $\boldsymbol{\theta}$ from $p(\boldsymbol{\theta})$.
repeat
  Updating of $V \,|\, \boldsymbol{\theta}, \boldsymbol{Y}$***.***
  Draw $Z \sim \text{Exp}\,(1)$, and set $V = l\,(\boldsymbol{\theta}; \boldsymbol{Y}) + Z$.
  Component-wise updating of $\boldsymbol{\theta} \,|\, V, \boldsymbol{Y}$***.***
  for $k = 1, \ldots, |\boldsymbol{\theta}|$ **do**
    repeat
      Draw the k-th component θ_k of $\boldsymbol{\theta}$ from $p(\theta_k)$, using shrinkage sampling to truncate the domain of $p(\theta_k)$ after each iteration.
    until $l\,(\boldsymbol{\theta}; \boldsymbol{Y}) < V$.
  end for
  Keep acquired sample of $\boldsymbol{\theta}$.
until the number of MCMC samples of $\boldsymbol{\theta}$ from $p(\boldsymbol{\theta}|\boldsymbol{Y})$ is sufficiently large.

When updating a component θ_k of $\boldsymbol{\theta}$, we use *shrinkage sampling* to reduce the number of draws required before a point in the slice S is found [Neal, 2003]. That is, if $\theta_{k,2}$ drawn from $p(\theta_k)$ is outside the slice, i. e., $l\,(\boldsymbol{\theta}; \boldsymbol{Y}) \geq V$, and is larger (smaller) than the current value $\theta_{k,1}$, then the next draw is made from $p(\theta_k)$ with its domain truncated such that the upper (lower) bound is $\theta_{k,2}$. The truncated interval shrinks with each rejection until a point in the slice is found. This may result in higher autocorrelations compared to the simple rejection scheme [Banerjee *et al.*, 2004], but significantly speeds up the Gibbs sampler.

For our model (Equation 3), the marginal negative log-likelihood $l(\boldsymbol{\theta}; \boldsymbol{Y}) = -\log p(\boldsymbol{Y}|\boldsymbol{\theta})$ associated with $\boldsymbol{Y}$ is

$$l(\boldsymbol{\theta}; \boldsymbol{Y}) = \tfrac{1}{2} \sum_{u=1}^{m} \log |\Sigma_u| + \tfrac{1}{2} \sum_{u=1}^{m} (\boldsymbol{Y}_u - \boldsymbol{\mu}_u)^T \Sigma_u^{-1} (\boldsymbol{Y}_u - \boldsymbol{\mu}_u) + C,$$

where $\Sigma_u = \Sigma_u(\boldsymbol{\sigma}_u, \tau^2, \phi, \nu) = \boldsymbol{\sigma}_u 1_{n_u} H_u(\phi, \nu) \boldsymbol{\sigma}_u 1_{n_u} + \tau^2 1_{n_u}$, and $C \equiv const.$ independent of $\boldsymbol{\theta}$. Given $\boldsymbol{\sigma}_u$, ϕ and ν, computing the eigen decomposition of positive-definite $\boldsymbol{\sigma}_u 1_{n_u} H_u(\phi, \nu) \boldsymbol{\sigma}_u 1_{n_u}$ simplifies the calculation of $\log |\Sigma_u|$ and Σ_u^{-1}. It also speeds up the sampling procedure when updating τ^2 at a given iteration of the slice Gibbs sampling algorithm. To minimise the number of eigen decompositions, we update ϕ and ν together. We proceed similarly for the components of $\boldsymbol{\sigma}_u$, and hence for the components of $\boldsymbol{\sigma}$.

In the remainder of the paper, we set $\sigma(\boldsymbol{s}_i) = \sqrt{\sigma^2_{\boldsymbol{Y}}(\boldsymbol{s}_i) - \tau^2}$, where $\sigma^2_{\boldsymbol{Y}}(\boldsymbol{s}_i)$ denotes the sample variance of the ratings for item i (at site $\boldsymbol{s}_i$), calculated from the ratings $Y_u(\boldsymbol{s}_i)$. This reduces the number of free model parameters from $2n + 3$ to $n + 3$, i. e., $\boldsymbol{\theta} = \big(\boldsymbol{\mu}, \tau^2, \phi, \nu\big)$, and significantly speeds up the slice Gibbs sampler.

2.4 Prediction

Given $\boldsymbol{\theta}$, the prediction of a user u's ratings of unseen items, say $\boldsymbol{Y}_{u,1}$, from a vector of observed ratings $\boldsymbol{Y}_{u,2}$ is straightforward. That is, we can use standard multivariate normal theory, because $\boldsymbol{Y}_u = (\boldsymbol{Y}_{u,1}, \boldsymbol{Y}_{u,2}) \,|\, \boldsymbol{\theta}$ is normally distributed (Section 2.2, similarly to Equation 2). If we use the following notation

$$\begin{bmatrix} \boldsymbol{Y}_{u,1} \\ \boldsymbol{Y}_{u,2} \end{bmatrix} |\, \boldsymbol{\theta} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_{u,1} \\ \boldsymbol{\mu}_{u,2} \end{bmatrix}, \begin{bmatrix} \Sigma_{u,11} & \Sigma_{u,12} \\ \Sigma_{u,12}^T & \Sigma_{u,22} \end{bmatrix} \right),$$

then the conditional distribution $p\,(\boldsymbol{Y}_{u,1}|\boldsymbol{Y}_{u,2}, \boldsymbol{\theta})$ is normal with mean vector and covariance matrix

$$\begin{aligned} \mathbb{E}\,(\boldsymbol{Y}_{u,1}|\boldsymbol{Y}_{u,2}, \boldsymbol{\theta}) &= \boldsymbol{\mu}_{u,1} + \Sigma_{u,12}\Sigma_{u,22}^{-1}\,(\boldsymbol{Y}_{u,2} - \boldsymbol{\mu}_{u,2})\,, \\ \text{Cov}\,(\boldsymbol{Y}_{u,1}|\boldsymbol{Y}_{u,2}, \boldsymbol{\theta}) &= \Sigma_{u,11} - \Sigma_{u,12}\Sigma_{u,22}^{-1}\Sigma_{u,12}^T. \end{aligned}$$

The expectation $\mathbb{E}\,(\boldsymbol{Y}_{u,1}|\boldsymbol{Y}_{u,2}, \boldsymbol{\theta})$ represents a personalised prediction of ratings $\boldsymbol{Y}_{u,1}$, and a measure of confidence can be easily derived from $\text{Cov}\,(\boldsymbol{Y}_{u,1}|\boldsymbol{Y}_{u,2}, \boldsymbol{\theta})$.

3 Model Validation with a Simulated Dataset

In order to validate our *Spatial Process Model (SPM)* (Section 2.2) and algorithms for parameter estimation and prediction (Sections 2.3 and 2.4), we use simulated data from a model with known parameters. This sections reports on the results we obtained.

3.1 Dataset

In this artificial toy study, we reduce the spatial dimension to 1, i. e., the one-dimensional interval $[0, 1]$, and place 10 equally-spaced sites over this interval, i. e., $\boldsymbol{s} = 0, 1/9, \ldots, 1$. We let a rating $Y_u(\boldsymbol{s}) = \mu(\boldsymbol{s}) + \sigma(\boldsymbol{s}) w_u(\boldsymbol{s}) + \varepsilon(\boldsymbol{s})$ for all users u and all sites $\boldsymbol{s}$ (Section 2.2), where $w_u(\boldsymbol{s})$ are samples from a stationary Gaussian spatial process $W(\boldsymbol{s})$ with mean 0, variance 1 and powered exponential correlation function $\rho(\|\boldsymbol{s}_i - \boldsymbol{s}_j\|; \phi, \nu)$ (Equation 1), and $\varepsilon(\boldsymbol{s})$ are realisations from a white-noise process with mean 0 and variance τ^2. The true model parameters are

$$\begin{aligned} \boldsymbol{\mu} &= (1.3, 5.1, 3.6, 7.0, 8.6, 3.2, 4.9, 5.5, 9.0, 6.7)\,, \\ \boldsymbol{\sigma}^2 &= (1.2, 2.1, 0.9, 2.4, 2.2, 2.7, 1.6, 1.2, 1.7, 1.8)\,, \\ \tau^2 &= 0.5, \\ (\phi, \nu) &= (4.0, 1.6)\,. \end{aligned}$$

We sampled 100 complete rating vectors $\boldsymbol{Y}_u$ (i. e., 100 independent complete user profiles) from a Gaussian spatial process with the above parameters. Hence, in total, our simulated dataset consists of 1000 ratings.

3.2 Parameter Estimation

To validate our model fitting algorithm, we ran a slice Gibbs sampler (Algorithm 1) on the complete dataset of 1000 samples. For each of the 13 free model parameters,[6] we used (uninformative) independent uniform prior distributions. We stopped the sampling procedure after 8000 iterations, and used every 20-th sample after a burn-in phase of 1000 iterations as a sample from the posterior distribution of $\boldsymbol{\theta}$. Thus, in total, we obtained 350 samples of $\boldsymbol{\theta}$ from $p(\boldsymbol{\theta}|\boldsymbol{Y})$. Figure 1 shows the results of this run. In Figure 1, θ_1 to θ_{10} correspond to μ_1 to μ_{10}, θ_{11} to θ_{20} correspond to σ_1^2 to σ_{10}^2, and θ_{21} to θ_{23} correspond to τ^2, ϕ and

[6] We set $\sigma(\boldsymbol{s}_i) = \sqrt{\sigma^2_{\boldsymbol{Y}}(\boldsymbol{s}_i) - \tau^2}$ to speed up the sampling process (Section 2.3), which reduces the number of free parameters from 23 to 13.

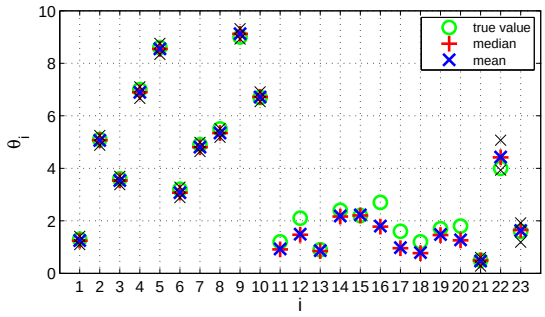

Figure 1: Parameter fit for simulated data

ν respectively. For each of the parameters, the plot depicts the true value (green $\circ$), the sample median of the parameter's posterior distribution (red $+$), and the sample mean (blue $\times$). For $\boldsymbol{\mu}$, τ^2, ϕ and ν, we also show a 95% credible interval estimated from the sample of the posterior distribution (lower and upper bounds marked by a black $\times$). We do not show these intervals for $\boldsymbol{\sigma}$, as we approximated $\boldsymbol{\sigma}$'s components by $\sigma(\boldsymbol{s}_i) = \sqrt{\sigma^2_{\boldsymbol{Y}}(\boldsymbol{s}_i) - \tau^2}$ (Section 2.3) — hence, no posterior distributions were estimated for the site variances. The results verify that the estimates of all free model parameters fall within a 95% credible interval. This confirms the algorithm's ability to fit the model accurately for our simulated dataset.

3.3 Predictive Model Performance

Experimental Setup

To evaluate *SPM*'s predictive performance with the simulated data, we implemented a baseline, called *Mean Model (MM)*, which predicts the rating of item i to be its (non-personalised) mean rating $\mu(\boldsymbol{s}_i)$. Due to the small size of the dataset, we used leave-one-out cross validation. That is, for each user, we trained the models with the data from 99 of the 100 users, and used the withheld data for testing. For *SPM* (as in the previous section), we stopped the slice Gibbs sampler after 8000 iterations for each of the 100 runs, and used every 20-th sample after 1000 iterations as a sample from the posterior distribution of the parameter vector $\boldsymbol{\theta}$. Predictions were computed by conditioning a multivariate normal distribution (Section 2.4), using the posterior mean of $p(\boldsymbol{\theta}|\boldsymbol{Y})$ to instantiate the model.

We performed two types of experiments: *Individual Item* and *Progressive Observations*.

- ***Individual Item (II).*** *II* evaluates predictive performance for a single item. For each user-item pair (u, i), we removed the rating $Y_u(\boldsymbol{s}_i)$ from the user's rating vector, and computed a prediction $\hat{Y}_u(\boldsymbol{s}_i)$ from the other ratings. This experiment is lenient in the sense that all available ratings except the rating for item i are kept in a user's profile.
- ***Progressive Observations (PO).*** *PO* evaluates performance with increasing number of ratings. For each user, we started with an empty rating vector, and iteratively added one of the ratings to the profile (the order was chosen randomly). We then predicted the ratings of all yet unadded items.

For both experiments, we used the *mean absolute error (*MAE*)* to measure predictive accuracy as follows:

$$\text{MAE} = \frac{1}{\sum_{u \in U} |I_u|} \sum_{u \in U} \sum_{i \in I_u} |Y_u(\boldsymbol{s}_i) - \hat{Y}_u(\boldsymbol{s}_i)|,$$

where I_u denotes a user u's set of items for which predictions were computed. For *II*, we calculated the total MAE for all users and all items; and for *PO*, we computed the MAE across the yet unadded items and all users for each number of observed ratings.

Table 1: Predictive model performance (MAE) for the *II* experiment with simulated data

	MAE	Stderr
Mean Model (MM)	1.0930	0.0264
Spatial Process Model (SPM)	0.8479	0.0196

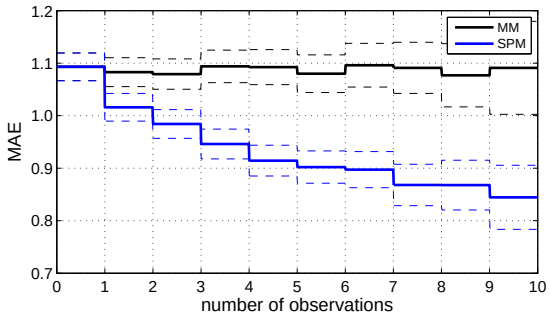

Figure 2: Predictive model performance (MAE) for the *PO* experiment with simulated data

Results

Table 1 shows the results for the *II* experiment. *SPM* achieves an MAE of 0.8479 (standard error 0.0196), whereas *MM* attains an MAE of 1.0930 (standard error 0.0264). The performance difference is statistically significant with $p \ll 0.01$.

Figure 2 depicts the performance of *SPM* in comparison to *MM* for the *PO* experiment, showing also a standard error band for both models (dashed lines). As expected, with increasing number of ratings, *SPM*'s MAE clearly decreases, whereas *MM* remains at a constant performance level. The performance difference is statistically significant from one rating onwards ($p < 0.05$).

4 Evaluation with a Real-World Dataset from the Museum Domain

This section reports on the results of an evaluation performed with a real-world dataset from the museum domain, including comparison with a state-of-the-art collaborative filter.

Our setting is motivated by the need to automatically recommend exhibits to visitors of physical educational spaces, such as museums. A first step in this process is the inference of visitors' interests from non-intrusive observations of their actions in the space. An RS can then combine these inferred interests with models that predict visitors' pathways through the physical museum [Bohnert *et al.*, 2008] to suggest exhibits that visitors would be interested in but are likely to overlook.

Table 2: Museum dataset statistics

	Mean	Stddev	Min	Max
Visit length (hrs)	1:50:39	0:47:54	0:28:23	4:42:12
Viewing time (hrs)	1:31:09	0:42:05	0:14:09	4:08:27
Exhibit areas / visitor	52.70	20.69	16	103
Visitors / exhibit area	66.09	25.36	6	117

In an information-seeking context, viewing time correlates positively with preference and interest. This observation was used in [Bohnert *et al.*, 2008] to propose a formulation of visitors' interests based on viewing times of exhibits. In this paper, we evaluate the predictive accuracy of our model on the basis of its predictions of viewing times.

4.1 Dataset and Model Justification

We obtained the dataset by manually tracking visitors to Melbourne Museum (Melbourne, Australia) from April to June 2008. In general, visitors do not require recommendations to travel between individual, logically related exhibits in close physical proximity. Hence, with the help of museum staff, we grouped Melbourne Museum's exhibited collection, comprising a few thousand exhibits, into 126 coherent exhibit areas. We restricted ourselves to tracking first-time adult visitors travelling on their own, to ensure that neither prior knowledge about the museum nor other visitors' interests influenced a visitor's decisions about what to look at. The resulting dataset comprises 158 complete visitor pathways in the form of time-annotated sequences of visited exhibit areas, with a total visit length of 291:22:37 hours, and a total viewing time of 240:00:28 hours.[7] A total of 8327 exhibit areas were viewed, yielding 52.7 areas per visitor on average (41.8% of the exhibit areas). Hence, 58.2% entries are missing from the viewing time (rating) matrix. This indicates that there is a potential for pointing a visitor to personally relevant but unvisited exhibit areas. Table 2 summarises further statistics of the dataset.

Clearly, the deployment of an automated RS in a museum requires suitable positioning technology to non-intrusively track visitors, and models to infer visitors' high-level activities (i. e., which exhibits are being viewed). Although our dataset was obtained manually, it provides information that is of the same type as information inferable from sensing data. Additionally, the results obtained from experiments with this dataset are essential for model development, as they provide an upper bound for the predictive performance of our model.

The museum space is carefully themed by curatorial staff, such that closely-related exhibits are in physical proximity. Based on this observation, we hypothesise that physical walking distance between exhibits is inversely proportional to their (content) similarity. Thus, in our experiments, we use physical walking distance for measuring (content) distances between exhibits. To calculate walking distances, we employed an SVG file-based representation of Melbourne Museum's site map, mapped onto a graph structure which preserves the physical layout of the museum (i. e., preventing paths from passing through walls or ceilings), and normalised the distances to the interval $[0, 1]$.

We used the *Bayesian Information Criterion (BIC)* for model selection, i. e., to select the most appropriate family of probability distributions for approximating the distribution of viewing times at each exhibit area. We tested exponential, gamma, normal, log-normal and Weibull distributions. The log-normal family fitted best, with respect to both number of best fits and average BIC score (averaged over all exhibit areas). Hence, we transformed all viewing times to their log-equivalent to obtain normally distributed data.

4.2 Parameter Estimation

We ran a slice Gibbs sampler (Algorithm 1) on the complete dataset of 8327 log viewing times, and on 158 reduced datasets — each with the data of one visitor withheld.[8] For each of the 129 free model parameters,[9] we used (uninformative) independent uniform prior distributions. Similarly to above, we stopped the sampling procedure after 8000 iterations, and used every 20-th sample after a burn-in phase of 1000 iterations as a sample from the posterior distribution of $\boldsymbol{\theta}$. Thus, in total, we obtained 350 samples of $\boldsymbol{\theta}$ from $p(\boldsymbol{\theta}|\boldsymbol{Y})$ for each of the 159 runs.

For visualisation purposes, we used the output of the run on the complete dataset to obtain point estimates for ϕ and ν (i. e., the posterior means 28.6334 for ϕ and 0.2583 for ν). Figure 3 depicts a plot of $\rho(\|\boldsymbol{s}_i - \boldsymbol{s}_j\|; \phi, \nu)$ for this parameterisation, showing the shape of the fitted powered exponential correlation function. The dashed lines indicate the correlation functions obtained with ϕ and ν set to the lower and upper bounds of their 95% credible intervals respectively. The correlation function rapidly drops to values around 0.4, and then more slowly approaches 0.1 as $\|\boldsymbol{s}_i - \boldsymbol{s}_j\|$ approaches 1.0. The shape of this function confirms the existence of a relatively high correlation between viewing durations at exhibit areas in close physical proximity. For the complete dataset, the posterior mean estimate of τ^2 is 0.1578 (we omit the estimates for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ due to space limitations).

4.3 Predictive Model Performance

Experimental Setup

To evaluate *SPM*'s predictive performance on the museum dataset, we implemented two additional models: *Mean Model (MM)* and *Collaborative Filter (CF)*. *MM*, which we use as a baseline, predicts the log viewing time of an exhibit area i to be its (non-personalised) mean log viewing time $\mu(\boldsymbol{s}_i)$. For *CF*, we implemented a nearest-neighbour collaborative filtering algorithm, and added modifications from

[7] For our experiments, we ignore travel time between exhibit areas, and collapse multiple viewing events of one area into one event.

[8] To speed up the convergence of the Markov chain, we used the output of an initial long run of the slice Gibbs sampler on the complete dataset as the starting values for the runs on the reduced datasets.

[9] We set $\sigma(\boldsymbol{s}_i) = \sqrt{\sigma^2_{\boldsymbol{Y}}(\boldsymbol{s}_i) - \tau^2}$ to speed up the sampling process (Section 2.3), which reduces the number of free parameters from 255 to 129.

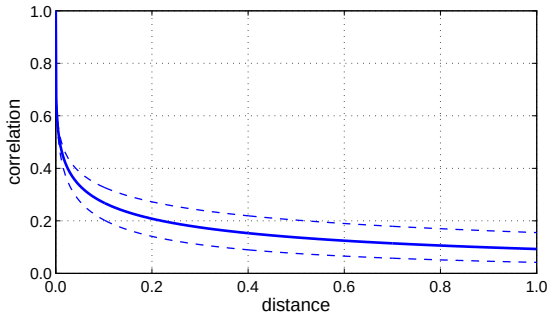

Figure 3: Correlation function $\rho(\|\boldsymbol{s}_i - \boldsymbol{s}_j\|; \phi = 28.6334, \nu = 0.2583)$, fitted with the museum dataset

the literature that improve its performance, such as shrinkage to the mean [James and Stein, 1961] and significance weighting [Herlocker *et al.*, 1999]. Additionally, to ensure that varying exhibit area complexity does not affect the similarity computation for selecting the nearest neighbours (viewing time increases with exhibit complexity), we transformed the log viewing times into z-scores by normalising the values for each of the exhibit areas separately. Visitor-to-visitor differences with respect to their mean viewing durations were neutralised by transforming predictions to the current visitor's viewing-time scale [Herlocker *et al.*, 1999]. In total, we tested several thousand different parameterisations. In this paper, we report only on the performance of the best one.

Due to the relatively small dataset, we used leave-one-out cross validation to evaluate the performance of the different models. That is, for each visitor, we trained the models with the data from 157 of the 158 visit trajectories, and used the withheld visitor pathway for testing. For *CF*, predictions were computed from the ratings of the nearest neighbours. For *SPM* (as above), we used the the posterior mean of $p(\boldsymbol{\theta}|\boldsymbol{Y})$ from the appropriate model training to instantiate the model, and computed predictions by conditioning a multivariate normal distribution (Section 2.4).

We performed two types of experiments: *Individual Exhibit* and *Progressive Visit* (the *Individual Exhibit* experiment is identical to the *II* experiment in Section 3, and the *Progressive Visit* experiment is similar to the *PO* experiment).

- ***Individual Exhibit (IE).*** *IE* evaluates predictive performance for a single exhibit. For each observed visitor-exhibit area pair (u, i), we removed the observation $Y_u(\boldsymbol{s}_i)$ from the vector of visitor u's log viewing durations, and computed a prediction $\hat{Y}_u(\boldsymbol{s}_i)$ from the other observations.
- ***Progressive Visit (PV).*** *PV* evaluates performance as a museum visit progresses, i. e., as the number of viewed exhibit areas increases. For each visitor, we started with an empty visit, and iteratively added each viewed exhibit area to the visit history, together with its log viewing time. We then predicted the log viewing times of all yet unvisited exhibit areas.

As above, for both experiments, we used the *mean absolute error (*MAE*)* to measure predictive accuracy.

Table 3: Predictive model performance (MAE) for the *IE* experiment with the museum dataset

	MAE	**Stderr**
Mean Model (MM)	0.8618	0.0071
Collaborative Filter (CF)	0.7868	0.0068
Spatial Process Model (SPM)	0.7548	0.0066

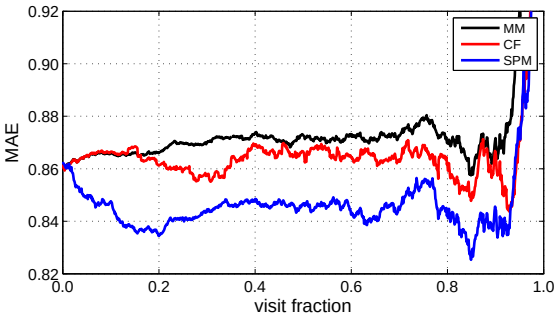

Figure 4: Predictive model performance (MAE) for the *PV* experiment with the museum dataset

For *IE*, we calculated the total MAE for all observed visitor-exhibit area pairs; and for *PV*, we computed the MAE across the yet unvisited exhibit areas and all visitors for each time fraction of a visit (to account for different visit lengths, we normalised all visits to a length of 1).

Results

Table 3 shows the results for the *IE* experiment, where *SPM* achieves an MAE of 0.7548 (stderr 0.0066), outperforming both *MM* and *CF*. The performance difference between *SPM* and the other models is statistically significant with $p \ll 0.01$.

The performance of *SPM*, *CF* and the baseline *MM* for the *PV* experiment is depicted in Figure 4. *CF* outperforms *MM* slightly (statistically significantly for visit fractions 0.191 to 0.374 and for several shorter intervals later on, $p < 0.05$). We were happy to see a further, significant improvement in performance for *SPM*, compared to both *MM* and *CF* (statistically significant for visit fractions 0.019 to 0.922, $p < 0.05$). Drawing attention to the initial portion of the visits, *SPM*'s MAE decreases rapidly, whereas the MAE for *MM* and *CF* remains at a higher level. Generally, the faster a model adapts to a visitor's interests, the more likely it is to quickly deliver (personally) useful recommendations. Such behaviour in the early stages of a museum visit is essential in order to build trust in the RS, and to guide a visitor in a phase of his/her visit where such guidance is most likely needed. As expected, *MM* performs at a relatively constant MAE level. For *CF* and *SPM*, we expected to see a relative improvement in performance as the number of visited exhibit areas increases. However, this trend is rather subtle. Additionally, for all three models, there is a performance drop towards the end of a visit. We postulate that these phenomena may be explained, at least partially, by the increased influence of outliers on the MAE as the number of exhibit areas remaining to be viewed is reduced with the progression of a visit. This influence in turn offsets potential gains in per-

formance obtained from additional observations. Our hypothesis is supported by a widening in the standard error bands for all models as a visit progresses, in particular towards the end (not shown in Figure 4 for clarity of presentation). However, this behaviour requires further, more rigorous investigation.

To give an indication as to whether *SPM* can discover unvisited but personally interesting exhibit areas, we predicted the log viewing times of all unvisited exhibit areas for each visitor, given their complete visit history. We then counted the predicted log viewing durations that were significantly above the corresponding exhibit area's average log viewing time $\mu(\boldsymbol{s}_i)$. For this purpose, we used the 95% credible interval around $\mu(\boldsymbol{s}_i)$. On average, *SPM* predicts 23.6 such exhibit areas per visitor, which corresponds to an average of 30.1% of the predictions per visitor. This indicates that our model discovers exhibit areas in which visitors appear to be interested, but were not viewed during their visit.

5 Conclusions and Future Work

In this paper, we utilised the theory of spatial processes to develop a model-based approach for predicting users' interests or item ratings in RS. Our model was validated with simulated data, and evaluated with a real-world dataset from the museum domain. We showed that in our application scenario, our model attains a higher predictive accuracy than state-of-the-art nearest-neighbour collaborative filters. Additionally, under the realistic *Progressive Visit* setting, our model rapidly adapts to a user's rating vector (starting from as little as one rating), thus alleviating the *new-user problem* common to collaborative filtering.

Our dataset is relatively small compared to other real-world RS applications. Although a high number of ratings per user slows down the slice Gibbs sampler due to repeated inversion of matrices of high dimension, employing our model with larger datasets should not represent a problem in practice. This is because the number of ratings per user is usually small compared to the number of users and items, and the computational complexity of evaluating the likelihood function depends only linearly on the number of users in the database.

In the future, we propose to hybridise our model by incorporating content-based item features into our distance measure. In addition, we plan to extend our model to fit item ratings that are not Gaussian, e. g., [Diggle *et al.*, 1998; Yu *et al.*, 2006]; and to consider negative correlations between items.

Acknowledgments

This research was supported in part by grant DP0770931 from the Australian Research Council. The authors thank Carolyn Meehan and her team from Museum Victoria for fruitful discussions and their support of this research. Thanks also go to David Abramson for making available his computer cluster, and to Jeff Tan and Blair Bethwaite for their assistance with using it to run our experiments.

References

[Albrecht and Zukerman, 2007] D.W. Albrecht and I. Zukerman. Introduction to the special issue on statistical and probabilistic methods for user modeling. *User Modeling and User-Adapted Interaction*, 17(1-2):1–4, 2007.

[Andrieu *et al.*, 2003] C. Andrieu, N. de Freitas, A. Doucet, and M.I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50(1):5–43, 2003.

[Banerjee *et al.*, 2004] S. Banerjee, B.P. Carlin, and A.E. Gelfand. *Hierarchical Modeling and Analysis for Spatial Data*. Chapman & Hall/CRC, 2004.

[Bell *et al.*, 2007] R. Bell, Y. Koren, and C. Volinsky. Chasing $1,000,000: How we won the Netflix progress prize. *ASA Statistical and Computing Graphics Newsletter*, 18(2):4–12, 2007.

[Bohnert *et al.*, 2008] F. Bohnert, I. Zukerman, S. Berkovsky, T. Baldwin, and L. Sonenberg. Using interest and transition models to predict visitor locations in museums. *AI Communications*, 21(2-3):195–202, 2008.

[Breese *et al.*, 1998] J.S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. of the 14th Intl. Conf. on Uncertainty in Artificial Intelligence (UAI-98)*, pages 42–52, 1998.

[Burke, 2002] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[Damien *et al.*, 1999] P. Damien, J. Wakefield, and S. Walker. Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(2):331–344, 1999.

[Diggle *et al.*, 1998] P.J. Diggle, J.A. Tawn, and R.A. Moyeed. Model-based geostatistics. *Applied Statistics*, 47(3):299–350, 1998.

[Herlocker *et al.*, 1999] J.L. Herlocker, J.A. Konstan, A. Borchers, and J.T. Riedl. An algorithmic framework for performing collaborative filtering. In *Proc. of the 22th Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR-99)*, pages 230–237, 1999.

[James and Stein, 1961] W. James and C.M. Stein. Estimation with quadratic loss. In *Proc. of the Fourth Berkeley Symp. on Mathematical Statistics and Probability, Vol. 1*, pages 361–379, 1961.

[Neal, 2003] R.M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–767, 2003.

[Resnick and Varian, 1997] P. Resnick and H.R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[Resnick *et al.*, 1994] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J.T. Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In *Proc. of the 1994 ACM Conf. on Computer Supported Cooperative Work (CSCW-94)*, pages 175–186, 1994.

[Sarwar *et al.*, 2001] B. Sarwar, G. Karypis, J.A. Konstan, and J.T. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proc. of the 10th Intl. Conf. on World Wide Web (WWW-01)*, pages 285–295, 2001.

[Schwaighofer *et al.*, 2005] A. Schwaighofer, V. Tresp, and K. Yu. Learning Gaussian process kernels via hierarchical Bayes. In *Advances in Neural Information Processing Systems 17 (NIPS-04)*, pages 1209–1216. 2005.

[Yu *et al.*, 2006] S. Yu, K. Yu, V. Tresp, and H.-P. Kriegel. Collaborative ordinal regression. In *Proc. of the 23rd Intl. Conf. on Machine Learning (ICML-06)*, pages 1089–1096, 2006.