Non-linear Matrix Factorization with Gaussian Processes

Proceedings of the 26th International Conference on Machine Learning

Neil D. Lawrence
School of Computer Science, University of Manchester

Raquel Urtasun
ICSI and EECS, UC Berkeley

Presented by Eric Wang to the Duke Machine Learning Reading Group
9/11/2009

Overview

- Introduction

- Probabilistic Matrix Factorization (PMF) and Dual Probabilistic PCA (DPPCA)
  - Relationship to Bayesian PCA
  - Handling Missing Values
  - Parameter Estimation by Stochastic Gradient Descent
  - Non-Linear PMF via GP-LVMs

- Prediction of User Ratings

- Results

- Discussion and Conclusions

Introduction

- When information from different viewpoints are jointly filtered, the process is called *collaborative filtering*.
  - Neighborhood approach: Express similarity metric between items to be rated.
  - Latent factor approach: low rank approximation of the full matrix.

- This paper develops a non-linear latent factor approach for collaborative filtering which gives fully probabilistic predictions on ratings.

- The proposed model yields desirable attributes of both latent factor and neighborhood approaches. It differs from previous combined approaches because it arises naturally from Gaussian process prediction.

Introduction

- Consider a user-item dataset with *N* items and *D* users $\mathbf{Y} \in \Re^{N \times D}$.

- **Goal:** factorize **Y** into a lower rank form

$$\mathbf{Y} \approx \mathbf{U}^{\top}\mathbf{V}$$

where $\mathbf{U} \in \Re^{q \times N}$ and $\mathbf{V} \in \Re^{q \times D}$.

- A natural probabilistic interpretation of the above factorization is called *probabilistic matrix factorization* (pmf)

$$p\left(\mathbf{Y}|\mathbf{U},\mathbf{V},\sigma^2\right) = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{V}^{\top}\mathbf{u}_{:,i},\sigma^2\mathbf{I}\right)$$

where Gaussian priors are placed over **U** and **V**

$$p\left(\mathbf{U}\right) \;=\; \prod_{i=1}^{N} \prod_{j=1}^{q} \mathcal{N}\left(u_{j,i}|0,\alpha_u^{-1}\right)$$

$$p\left(\mathbf{V}\right) \;=\; \prod_{i=1}^{D} \prod_{j=1}^{q} \mathcal{N}\left(v_{j,i}|0,\alpha_v^{-1}\right)$$

PMF and Bayesian PCA

- PMF can be shown to be equivalent to probabilistic PCA. Let $\mathbf{X} \equiv \mathbf{U}^\top \in \Re^{N \times q}$ denote a matrix of latent variable and let $\mathbf{W} \equiv \mathbf{V}^\top \in \Re^{D \times q}$ be a mapping matrix . Then the previous likelihood can be written as

$$p\left(\mathbf{Y}|\mathbf{W},\mathbf{X},\sigma^2\right) = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^2\mathbf{I}\right)$$

- If the following prior is placed over **X**

$$p\left(\mathbf{X}\right) = \prod_{i=1}^{N}\prod_{j=1}^{q} \mathcal{N}\left(x_{i,j}|0,\alpha_x^{-1}\right)$$

and we marginalize it over **X**, we have

$$p\left(\mathbf{Y}|\mathbf{W},\sigma^2,\alpha_x\right) = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0},\alpha_x^{-1}\mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}\right)$$

which is the same likelihood optimized in PPCA.

DPPCA and Bayesian PCA

- Alternatively, we could marginalize **W** instead of **X** by assuming the following prior on **W**

  $$p(\mathbf{W}) = \prod_{i=1}^{D} \prod_{j=1}^{q} \mathcal{N}\left(w_{i,j} | 0, \alpha_w^{-1}\right)$$

  which, when marginalized, yields

  $$p\left(\mathbf{Y} | \mathbf{X}, \sigma^2, \alpha_w\right) = \prod_{j=1}^{D} \mathcal{N}\left(\mathbf{y}_{:,j} | \mathbf{0}, \alpha_w^{-1} \mathbf{X}\mathbf{X}^\top + \sigma^2 \mathbf{I}\right)$$

  or the marginal likelihood of a multi-output Bayesian linear regression model, where **X** is unknown and also optimized.

- Optimization here results in Dual Probabilistic PCA (DPPCA).

- *Marginalization and optimization with respect to both* ***W*** *and* ***X*** *results in Bayesian PCA.*

Handling Missing Values

- The special covariance structure of the above models allows straightforward marginalization of over missing values. Let the observed set over vector **y** be denoted by $\mathbf{y}_{\mathbf{i}}$ where **i** represents the indices of the observed values.

- Marginalizing over the missing values yields $\mathbf{y}_{\mathbf{i}} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{i}}, \boldsymbol{\Sigma}_{\mathbf{i},\mathbf{i}}\right)$. Thus, when the data matrix is sparse, the above marginalized likelihoods have the forms

$$p\left(\mathbf{Y}|\mathbf{W},\sigma^2,\alpha_x\right) = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{y}_{i,\mathbf{j}_i}|\mathbf{0},\alpha_x^{-1}\mathbf{W}_{\mathbf{j}_i,:}\mathbf{W}_{\mathbf{j}_i,:}^{\top}+\sigma^2\mathbf{I}\right)$$

and

$$p\left(\mathbf{Y}|\mathbf{X},\sigma^2,\alpha_w\right) = \prod_{j=1}^{D} \mathcal{N}\left(\mathbf{y}_{\mathbf{i}_j,j}|\mathbf{0},\alpha_w^{-1}\mathbf{X}_{\mathbf{i}_j,:}\mathbf{X}_{\mathbf{i}_j,:}^{\top}+\sigma^2\mathbf{I}\right)$$

DPPCA

- The authors proceed by selecting to marginalize over **W**. This is to minimize the number of parameters which must be estimated since the number of users is generally greater than the number of items.

$$p\left(\mathbf{Y}|\mathbf{X}, \sigma^2, \alpha_w\right) = \prod_{j=1}^{D} \mathcal{N}\left(\mathbf{y}_{\mathbf{i}_j,j}|\mathbf{0}, \alpha_w^{-1}\mathbf{X}_{\mathbf{i}_j,:}\mathbf{X}_{\mathbf{i}_j,:}^{\top} + \sigma^2\mathbf{I}\right)$$

- If **Y** is fully observed, then a global optimal solution can be found. If the matrix is not fully observed, either EM or stochastic gradient descent can be used to optimize the parameters and hyperparameters.

- When the item dimensionality *D* is large, EM becomes computationally expensive, making stochastic gradient descent an attractive choice.

Stochastic Gradient Descent

- To optimize the parameters, the algorithm is shown users with their observed ratings one at a time, the gradients with respect to $\mathbf{X}$, α_w and σ^2 are found for user *j*, and the parameters are updated based on the computed gradients.

- Maximization of the log likelihood is equivalent to minimizing the inverse log likelihood, which is given by

$$E_j(\mathbf{X}) = \frac{N_j}{2}\log|\mathbf{C}_j| + \frac{1}{2}\left(\mathbf{y}_{\mathbf{i}_j,j}^{\top}\mathbf{C}_j^{-1}\mathbf{y}_{\mathbf{i}_j,j}\right) + \text{const.}$$

- Differentiating with respect to X yields

$$\frac{\mathrm{d}E_j\left(\mathbf{X}\right)}{\mathrm{d}\mathbf{X}_{\mathbf{i_j},:}} = -\mathbf{G}\mathbf{X}_{\mathbf{i_j},:}$$

$$\mathbf{G} = \left(\mathbf{C}_j^{-1}\mathbf{y}_{\mathbf{i}_j,j}\mathbf{y}_{\mathbf{i}_j,j}^{\top}\mathbf{C}_j^{-1} - \mathbf{C}_j^{-1}\right)$$

$$\mathbf{C}^{-1} = \sigma^{-2}\mathbf{I} - \sigma^{-4}\mathbf{X}\left(\alpha_w\mathbf{I}\sigma^{-2} + \mathbf{X}^{\top}\mathbf{X}\right)^{-1}\mathbf{X}^{\top}$$

Non-Linear PMF

- The models discussed fall under the broader category of Gaussian Process Latent Variable Model (GP-LVM) if the covariance matrix $\mathbf{C} = \alpha_w^{-1}\mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}$ is interpreted as the covariance function of a linear Gaussian process model.

- If the inner product $\mathbf{X}\mathbf{X}^\top$ is replaced with a Mercer kernel, the model becomes non-linear.

- If we define $f_j\left(\mathbf{x}_{i,:}\right) = \mathbf{w}_{j,:}^\top \mathbf{x}_{i,:}$, then the original probabilistic regression model from above can be written as a product of univariate Gaussians

$$p\left(\mathbf{Y}|\mathbf{W},\mathbf{X},\sigma^2\right) = \prod_{j=1}^{D}\prod_{i=1}^{N}\mathcal{N}\left(y_{i,j}|f_j\left(\mathbf{x}_{i,:}\right),\sigma^2\mathbf{I}\right)$$

Non-Linear PMF

- A zero mean GP prior can be placed over the functions **f**.

$$p\left(\mathbf{f}|\,\mathbf{X}\right) = \mathcal{N}\left(\mathbf{f}|\mathbf{0},\mathbf{K}\right)$$

- where **K** is a covariance function with members $k(\mathbf{x}_{i,:},\mathbf{x}_{j,:})$ which measures the degree of correlation between samples *i* and *j* from *p*(**f**|**X**) .

- We can denote a linear regression model as one in which

$$k\left(\mathbf{x}_{i,:},\mathbf{x}_{j,:}\right) = \mathbf{x}_{i,:}^{\top}\mathbf{x}_{j,:}$$

- For this paper, the authors chose the radial basis function as their non-linear kernel

$$k\left(\mathbf{x}_{\ell,:},\mathbf{x}_{i,:}\right) = \alpha_m \exp\left(-\frac{\gamma_m}{2}||\mathbf{x}_{\ell,:}-\mathbf{x}_{i,:}||^2\right)$$

which can be directly substituted into the marginal likelihood

$$p\left(\mathbf{Y}|\mathbf{X},\sigma^2,\boldsymbol{\theta}\right) = \prod_{j=1}^{D}\mathcal{N}\left(\mathbf{y}_{\mathbf{i}_j,j}|\mathbf{0},\mathbf{K}+\sigma^2\mathbf{I}\right)$$

Making Predictions

- Suppose we wish to predict ratings for all users for the previously unseen item ℓ . The mean of user *j*’s predicted rating is given by

  $$\boldsymbol{\mu}_{\ell,j} = \mathbf{s}^\top \mathbf{y}_{\mathbf{i}_j,:},$$

  where $\mathbf{s} = \left(\mathbf{K}_{\mathbf{i}_j,\mathbf{i}_j} + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{k}_{\mathbf{i}_j,\ell}.$ Notice that the mean prediction for a new item is simply the weighted sum of the user’s rated items, similar to neighborhood based approaches.

- Moreover, application of the GP allows a full posterior over the predictions, with variance

  $$\varsigma_{\ell,j} = k_{\ell,\ell} + \sigma^2 - \mathbf{k}_{\mathbf{i}_j,\ell}^\top \left(\mathbf{K}_{\mathbf{i}_j,\mathbf{i}_j} + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{k}_{\mathbf{i}_j,\ell}.$$

- It is shown that the variance is highly related to the number of items rated by the user.

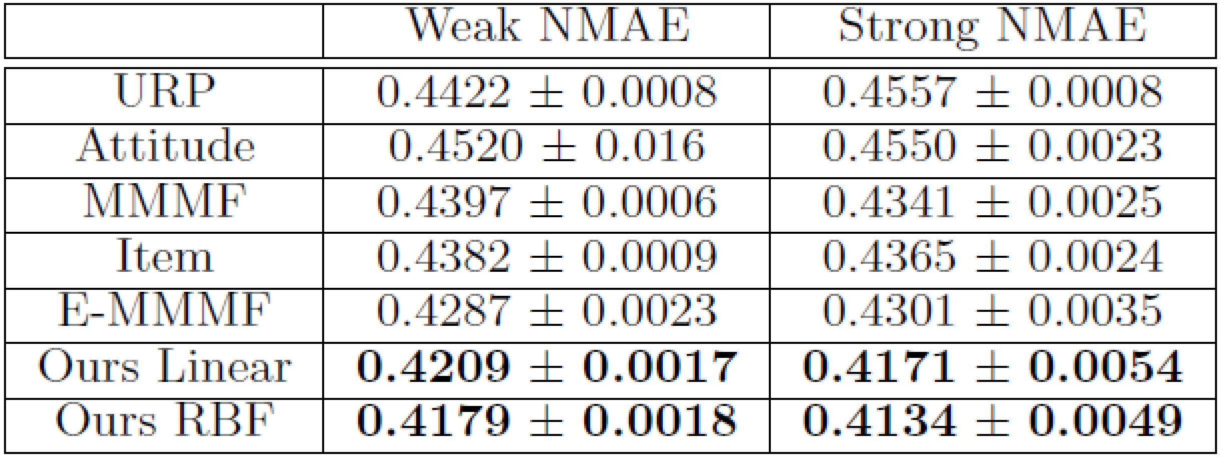
Results

- The authors show results on 3 widely used benchmark datasets
  - EachMovie: 2.6M ratings for 1,648 movies and 74,424 users.
  - 1M MovieLens: 1M ratings for 3,952 movies and 6,040 users.
  - 10M MovieLens: 10M ratings for 10,681 movies and 71,567 users.

- The normalized mean absolute error (NMAE) was used as the score metric, computed by normalizing the MAE by a factor corresponding to the score range such that random guessing yields an NMAE of 1.

- Weak generalization is the filling of missing values for movies already seen, and strong generalization is the rating prediction for previously unseen items.

Results: *EachMovie*

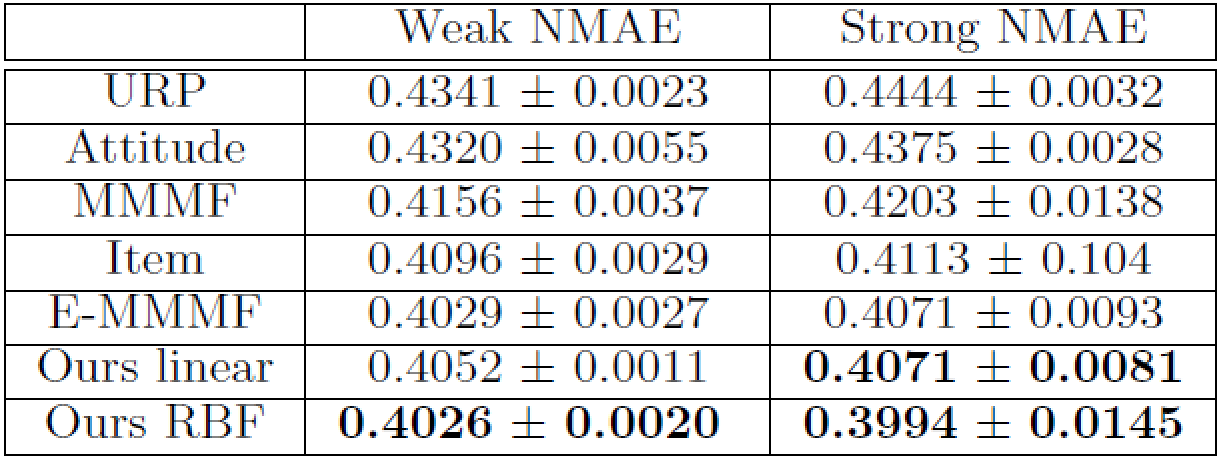


	Weak NMAE	Strong NMAE
URP	0.4422 ± 0.0008	0.4557 ± 0.0008
Attitude	0.4520 ± 0.016	0.4550 ± 0.0023
MMMF	0.4397 ± 0.0006	0.4341 ± 0.0025
Item	0.4382 ± 0.0009	0.4365 ± 0.0024
E-MMMF	0.4287 ± 0.0023	0.4301 ± 0.0035
Ours Linear	$\mathbf{0.4209 \pm 0.0017}$	$\mathbf{0.4171 \pm 0.0054}$
Ours RBF	$\mathbf{0.4179 \pm 0.0018}$	$\mathbf{0.4134 \pm 0.0049}$

- For the EachMovie dataset, 36,656 users with more than 20 ratings to their name were used. The group was split into 30,000 users for weak generalization, and the remaining users for strong generalization.

- The proposed approaches offered superior performance with 20 latent dimensions, much smaller than the 100 latent dimensions used in MMMF and E-MMMF.

Results: *1M MovieLens*



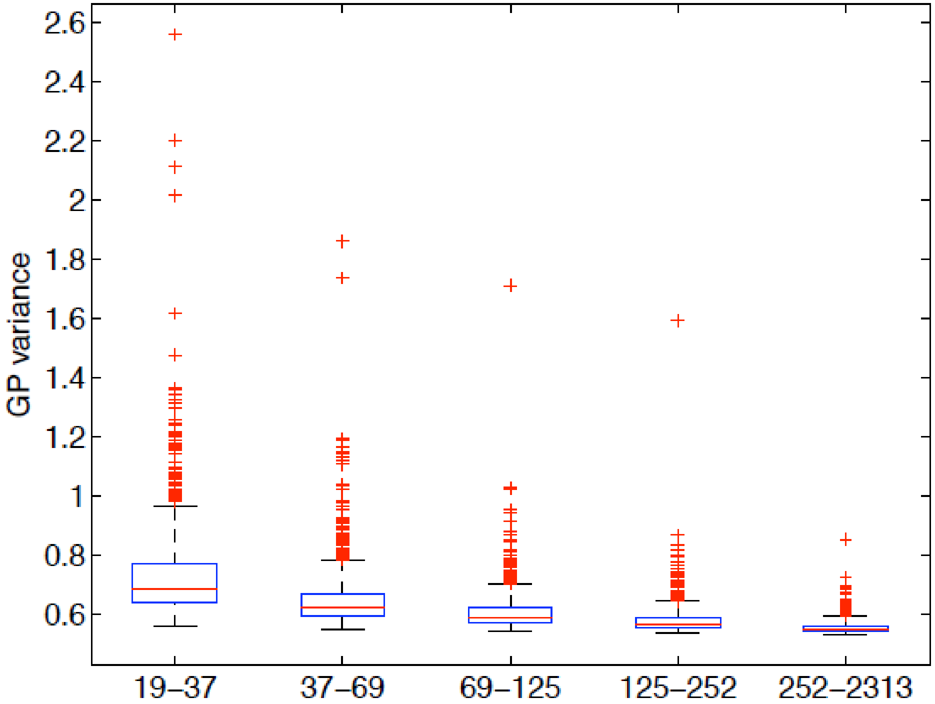
	Weak NMAE	Strong NMAE
URP	0.4341 ± 0.0023	0.4444 ± 0.0032
Attitude	0.4320 ± 0.0055	0.4375 ± 0.0028
MMMF	0.4156 ± 0.0037	0.4203 ± 0.0138
Item	0.4096 ± 0.0029	0.4113 ± 0.104
E-MMMF	0.4029 ± 0.0027	0.4071 ± 0.0093
Ours linear	0.4052 ± 0.0011	**0.4071 ± 0.0081**
Ours RBF	**0.4026 ± 0.0020**	**0.3994 ± 0.0145**

- Here, 5000 users were used for weak generalization and the remainder were used for strong generalization.

- The latent dimensionalities ranged from 10 to 11 for weak generalization, and 14 to 15 in strong generalization.

- When an ensemble approach of the author’s model was set up (similar to E-MMMF), NMAE was reduced to (0.3987 ± 0.0013).

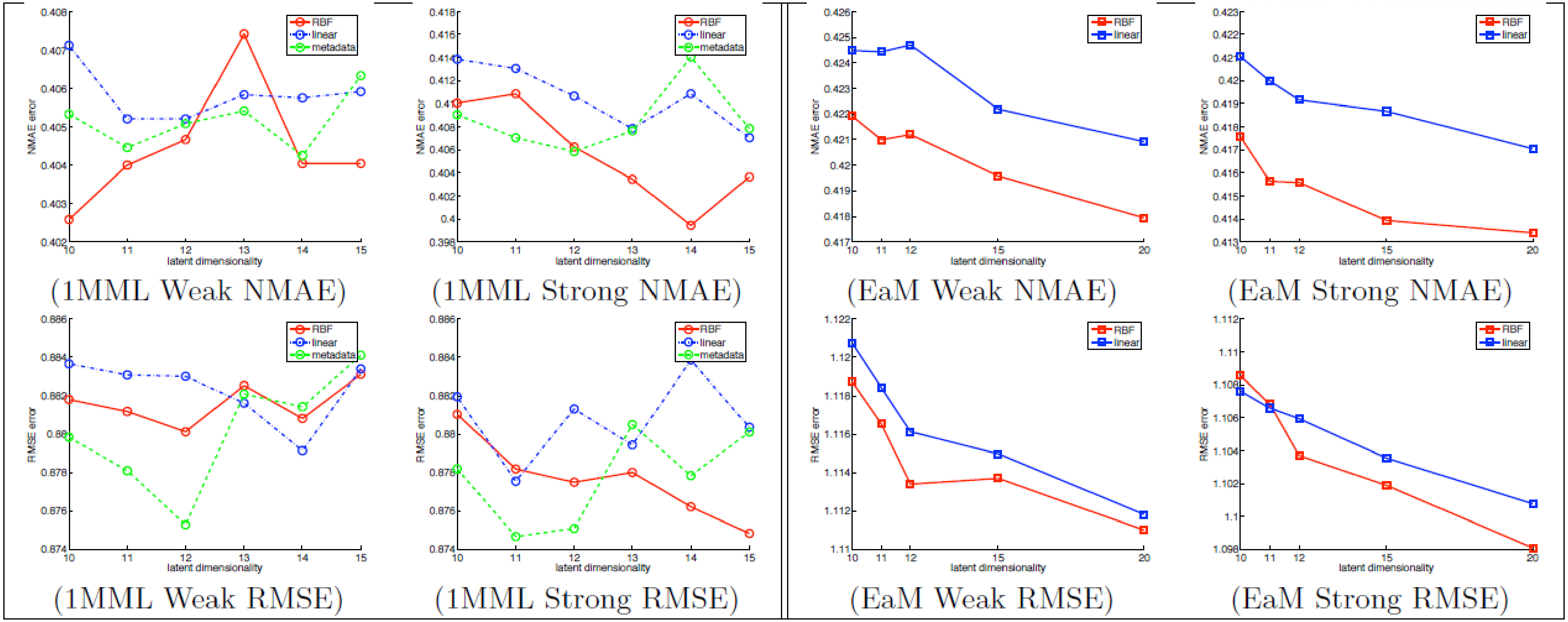
Results: *GP Variance*



- Here, the authors plot the prediction variance as a function of the number of movies rated, using a 10D RBF model learned for 1M MovieLens Weak.

- The prediction variance is a good indicator of model uncertainty, and decreases (as expected) with the number of movies rated by the user, and thus the amount of data seen by the model.

Results: *Latent Dimensionality*



(1MML Weak NMAE) (1MML Strong NMAE) (EaM Weak NMAE) (EaM Strong NMAE)

(1MML Weak RMSE) (1MML Strong RMSE) (EaM Weak RMSE) (EaM Strong RMSE)

- Here, the authors show the NMAE and RMSE for both strong and weak generalization as a function of the latent dimensionality. The colors represent the different kernels used. The metadata kernel computed the covariance matrix using a binary vector over the genres that defines each movie in the database.

Results: *10M MovieLens*

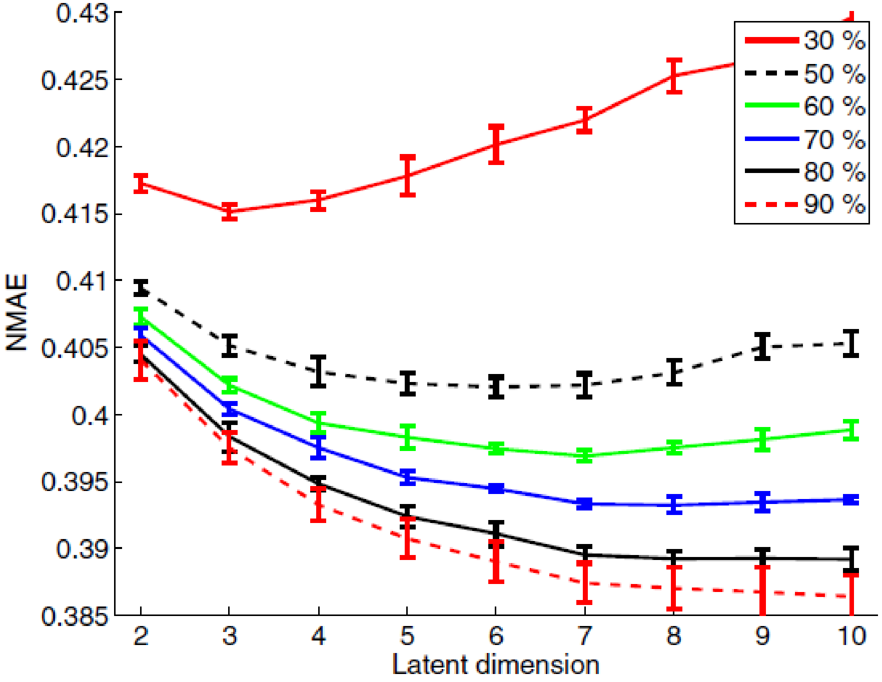
- The 10M MovieLens dataset is very new and at the time of publication there were published results to compare against.

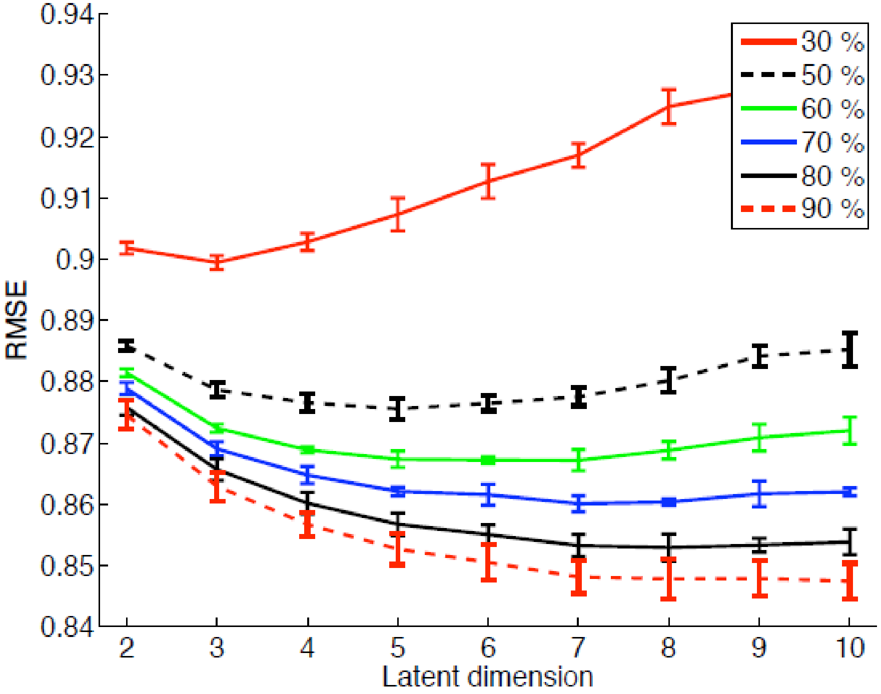
- The RMSE of author's method on the 10M Movielens dataset is 0.8740 ± 0.0278.

- For the 1M Movielens dataset, the weak RBF RMSE was $(\mathbf{0.8801} \pm 0.0082)$, the weak linear RMSE was $(\mathbf{0.8791} \pm 0.0080)$ The strong RBF RMSE was $(\mathbf{0.8748} \pm 0.0268)$ and the strong linear RMSE was $(\mathbf{0.8775} \pm 0.0239)$.

- For the EachMovie dataset, the weak RBF RMSE was $(\mathbf{1.1110} \pm 0.0028)$, the weak linear RMSE was $(\mathbf{1.1118} \pm 0.0022)$ The strong RBF RMSE was $(\mathbf{1.0981} \pm 0.0077)$ and the strong linear RMSE was $(\mathbf{1.1008} \pm 0.0080)$.

Results





- These two plots show the NMAE and RMSE as a function of the latent space dimensionality. The different curves correspond to various percentages of the database used for training. In general, as the training set increases, the latent dimensionality also increases.

Conclusions

- The authors have proposed a non-linear GP-LVM to perform collaborative filtering.

- The proposed model shows desirable traits of both neighborhood based and latent factor based approaches, and the predictive equations of the model are very similar to neighborhood based approaches.

- Besides offering state of the art performance, a particular advantage of the model is its ability to compute a fully probabilistic predictions.

- Parameter estimation was performed using stochastic gradient descent.