

Non-linear Matrix Factorization with Gaussian Processes

Neil D. Lawrence and Raquel Urtasun

Learning Workshop 2009, Clearwater, Florida

14th April 2009

- Collaborative filtering is the process of filtering information from different viewpoints.
- A particular use of the approach is the prediction of user tastes.
- Type of question to be answer: *What does a given user's quality rating of one item say about their likely rating for another?*
- For a data set with N items and D users we store ratings in $\mathbf{Y} \in \mathbb{R}^{N \times D}$.

Existing Approaches

Neighborhood Approach

- The **neighborhood approach**: compute similarity measure between items.
 - ▶ For a prediction, use weighted sum of “similar” items’ scores.
- Form of prediction:

$$\hat{y}_{i,j} = \mathbf{s}_{:,i}^\top \mathbf{y}_{:,j}$$

where $\mathbf{s}_{:,i}$ is normalized similarities for item i .

Existing Approaches

Matrix Factorization

- The **latent factor approach** typically involves a low rank approximation.
 - ▶ Probabilistic Variants (Salakhutdinov and Mnih, 2008b,a)
- Factorize $\mathbf{Y}$ into a lower rank form,

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

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

- Least squares fit of all $\mathbf{v}_{:,i}^T \mathbf{u}_{:,j}$ for each user j rating the i th film, $y_{i,j}$.
- For test prediction from user ℓ for item k simply compute $\mathbf{v}_{:,k}^T \mathbf{u}_{:,\ell}$.

$\mathbf{u}_{:,j}$ i th column of $\mathbf{U}$

Matrix Factorization

		<i>users</i>												
		1	2	3	4	5	6	7	8	9	10	11	12	13
<i>items</i>	1						4				4			5
	2	1		5				1					4	
	3					5				4		4		
	4								5					
	5	4				4		4				5		
	6							1					2	3
	7	3	5		3				4	4		3		
	8			4			3				5			
	9				1					2			2	
	10	5							4					

Matrix Factorization

users

1 2 3 4 5 6 7 8 9 10 11 12 13

items

1					4				4			5
2	1		5			1					4	
3				5				4		4		
4							5					
5	4			4		4				5		
6						1					2	3
7	3	5		3			4	4		3		
8			4			3			5			
9				1				2			2	
10	5						4					

Matrix Factorization

users

1 2 3 4 5 6 7 8 9 10 11 12 13

items

1					4				4			5
2	1		5			1					4	
3	?			5				4		4		
4							5					
5	4			4		4				5		
6						1					2	3
7	3	5		3			4	4		3		
8			4		3				5			
9				1				2			2	
10	5						4					

Matrix Factorization

X

V

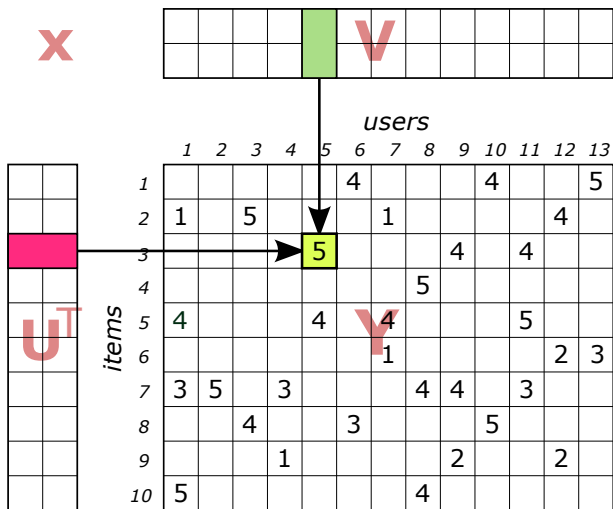
U^T

	<i>users</i>												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1						4				4			5
2	1		5				1					4	
3	?				5				4		4		
4								5					
5	4				4						5		
6												2	3
7	3	5		3				4	4		3		
8			4			3				5			
9				1					2			2	
10	5							4					

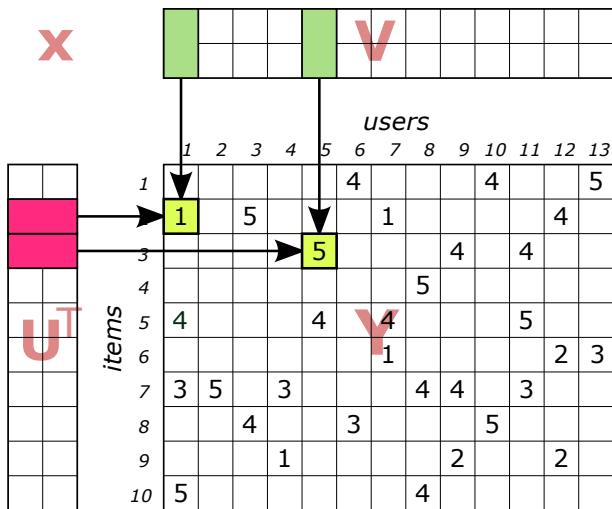
Y

items

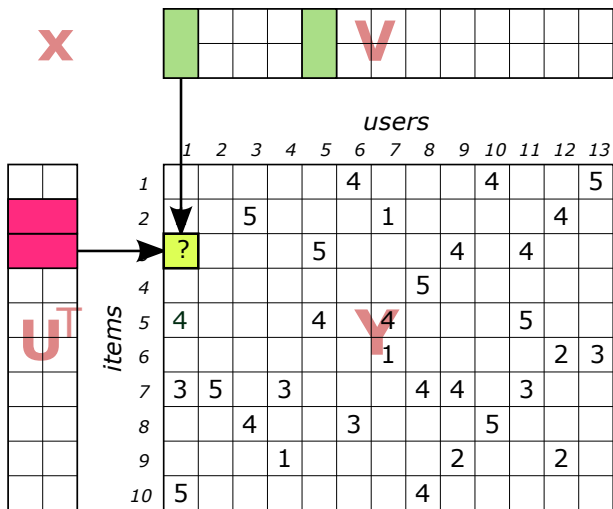
Matrix Factorization



Matrix Factorization



Matrix Factorization



- State of the Art
 - ▶ Both approaches perform well.
 - ▶ Best systems on Netflix use a combination (Koren, 2008).
- Our Contribution:
 - ▶ Relate probabilistic matrix factorization to probabilistic PCA.
 - ▶ Use GPs to non-linearize giving a probabilistic non-linear matrix factorization
 - ▶ The formula for rating test items is very similar to a neighborhood approach.

Probabilistic Matrix Factorization (PMF)

- Least squares fit has a natural probabilistic interpretation.

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

- Gaussian over $\mathbf{Y}$ with mean, $\mathbf{U}^\top \mathbf{V}$, and independent variance σ^2 .
- Missing values in $\mathbf{Y}$ are marginalized and ignored.
- In PMF a Gaussian prior is placed over $\mathbf{U}$, and $\mathbf{V}$

$$p(\mathbf{U}) = \prod_{i=1}^N \prod_{j=1}^q \mathcal{N}(u_{j,i} | 0, \alpha_u^{-1}) \quad p(\mathbf{V}) = \prod_{i=1}^D \prod_{j=1}^q \mathcal{N}(v_{j,i} | 0, \alpha_v^{-1}).$$

- Cannot marginalize both $\mathbf{U}$ and $\mathbf{V}$ — use sampling or MAP (Salakhutdinov and Mnih, 2008b,a).

PCA Equivalence

Probabilistic Matrix Factorization is Bayesian PCA

- Consider change of notation,
 - ▶ $\mathbf{X} \equiv \mathbf{U}^\top \in \mathbb{R}^{N \times q}$ (latent variables)
 - ▶ $\mathbf{W} \equiv \mathbf{V}^\top \in \mathbb{R}^{D \times q}$ (mapping matrix)

Change Notation

X

V

U^T

		<i>users</i>												
		1	2	3	4	5	6	7	8	9	10	11	12	13
<i>items</i>	1						4				4			5
	2	1		5				1					4	
	3					5				4		4		
	4								5					
	5	4				4		4				5		
	6							1					2	3
	7	3	5		3				4	4		3		
	8			4			3				5			
	9				1					2			2	
	10	5							4					

Y

Change Notation

X

WT

X

		<i>users</i>												
		1	2	3	4	5	6	7	8	9	10	11	12	13
<i>items</i>	1						4				4			5
	2	1		5				1					4	
	3					5				4		4		
	4								5					
	5	4				4		4				5		
	6							1					2	3
	7	3	5		3				4	4		3		
	8			4			3				5			
	9				1					2			2	
	10	5							4					

Y

- Using this notation:

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

cf multi-output linear regression

- Prior over $\mathbf{X}$,

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

gives

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

- Optimize wrt $\mathbf{W}$ — absorb α_x into $\mathbf{W}$.
- Probabilistic PCA (Tipping and Bishop, 1999).

- *Instead* take prior over $\mathbf{W}$,

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

cf Bayesian multi-output linear regression

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

- Optimize wrt *inputs* $\mathbf{X}$ — Probabilistic Principal Coordinate Analysis (Dual PPCA, Lawrence, 2005).

- Marginalizing both $\mathbf{X}$ and $\mathbf{W}$ is Bayesian PCA.
- Not analytically tractable though we can use approximations. (Bishop, 1999a,b; Minka, 2001).
- PMF suggests marginalizing both $\mathbf{U}$ and $\mathbf{V}$.
 - ▶ Particular priors can be chosen for including information.

Gaussian Processes

Making the model non-linear

- Dual PPCA is Bayesian multi-output linear regression

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

- Or a Gaussian Process with a ‘linear’ covariance function.

$$p(\mathbf{Y}|\mathbf{X}, \sigma^2, \alpha_x) = \prod_{j=1}^D \mathcal{N}(\mathbf{y}_{:,j} | \mathbf{0}, \mathbf{K}).$$

where $\mathbf{K} = \alpha_w^{-1} \mathbf{X} \mathbf{X}^\top + \sigma^2 \mathbf{I}$.

- Make model non-linear with non-linear covariance functions e.g., RBF.
- This model is called the Gaussian process Latent Variable Model (GP-LVM).

- GP-LVM approach gives *non-linear* probabilistic matrix factorization.
- Can also be seen as “kernelization” of the algorithm.
- Different to kernel PCA.
 - ▶ Kernel PCA constructs kernel in data space.
 - ▶ Difficult to deal with missing data.
 - ▶ GP-LVM constructs kernel in latent space.

- For the product of GPs marginalizing missing values is straightforward.
- Let $\mathbf{y}_i$ be the observed subset of $\mathbf{y}$.

$$\mathbf{y}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_{i,i}),$$

- For sparse data

$$p(\mathbf{Y}|\mathbf{X}, \sigma^2, \alpha_x) = \prod_{j=1}^D \mathcal{N}(\mathbf{y}_{i_j,j} | \mathbf{0}, \mathbf{K}_{i_j,i_j}).$$

- Tipping and Bishop (1999) suggest EM for missing values:
 - ▶ For large D (Netflix is 440,000) EM too expensive.
- We suggest stochastic gradient descent.
 - ▶ Present ratings for each user one at a time.
 - ▶ Compute gradient for that user and update parameters.
- For the j th user we minimize the negative log likelihood

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

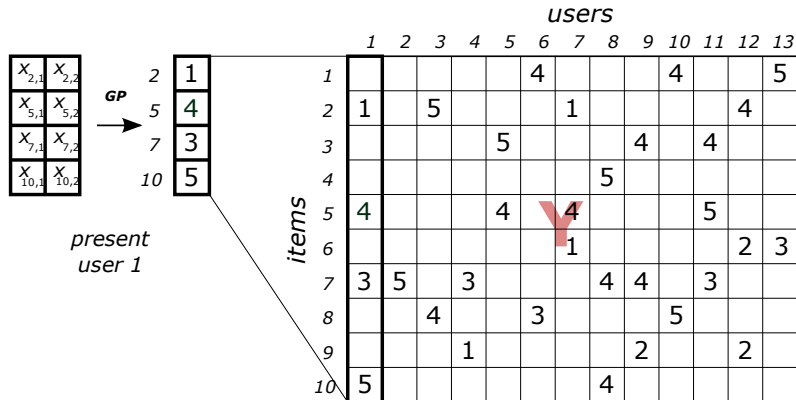
where $\mathbf{C}_j$ is covariance function (kernel) computed for j 's rated items.

Stochastic Gradient Descent

		<i>users</i>												
		1	2	3	4	5	6	7	8	9	10	11	12	13
<i>items</i>	1						4				4			5
	2	1		5				1					4	
	3					5				4		4		
	4								5					
	5	4				4		4				5		
	6							1					2	3
	7	3	5		3				4	4		3		
	8			4			3				5			
	9				1					2			2	
	10	5							4					

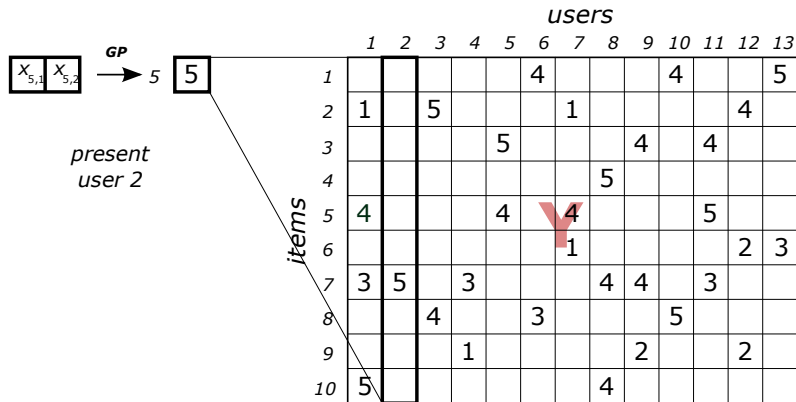
Present data a column at a time.

Stochastic Gradient Descent



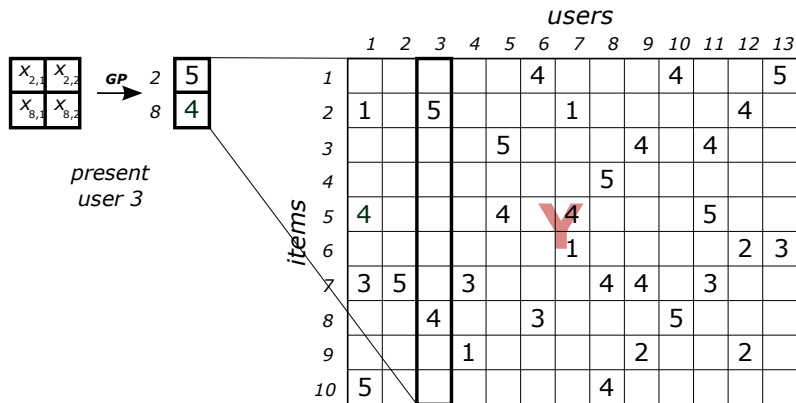
Each step updates $\mathbf{x}_{i,j,:}$.

Stochastic Gradient Descent



Complexity of GP cubic in N_j not N .

Stochastic Gradient Descent



No Sparse GP approximations required.

Model Prediction

Relation to Neighborhood Approach

- Learning: maximize likelihood wrt $\mathbf{X}$ and θ .
- Predict using standard GP formula:

$$\mu_{\ell,j} = \mathbf{s}^\top \mathbf{y}_{j,:},$$

with $\mathbf{s} = (\mathbf{K}_{\mathbf{i}_j, \mathbf{i}_j} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_{\mathbf{i}_j, \ell}$.

- ▶ Similar predictive form to neighborhood approach.
- ▶ Our “similarities” are computed in $\mathbf{X}$ space.

- For previously unseen users:
 - ▶ No need to perform new training.
 - ▶ Learned GP provides the prior and users' ratings provide the data.
- Can also compute variance of prediction:

$$s_{\ell,j} = k_{\ell,\ell} + \sigma^2 - \mathbf{k}_{i,j,\ell}^\top \mathbf{s}.$$

- We present results with:

$$k(\mathbf{x}_{i,:}, \mathbf{x}_j) = \theta_1 e^{-\frac{1}{2\theta_2} |\mathbf{x}_{i,:} - \mathbf{x}_{j,:}|^2} + \theta_{\text{bias}} + \sigma^2 \delta_{i,j}$$

or

$$k(\mathbf{x}_{i,:}, \mathbf{x}_j) = \theta_1 \mathbf{x}_{i,:}^\top \mathbf{x}_{j,:} + \theta_{\text{bias}} + \sigma^2 \delta_{i,j}$$

- Bias term is important.
 - ▶ Each user is being modeled as independent sample from this GP.
 - ▶ Bias term implies Gaussian prior over user biases.

- Followed (Marlin, 2004) in experimental setup for:
 - ▶ **EachMovie**: 2.6 million ratings for $N = 1,648$ movies and $D = 74,424$ users. Ratings range $\{1, \dots, 6\}$.
 - ▶ **1M Movielens**: 1 million ratings for 6,040 users, and 3,952 movies, with ratings ranging $\{1, \dots, 5\}$.
- Also try **10M Movielens**: 10 million ratings for $D = 71,567$ users and $N = 10,681$ movies, with ratings ranging $\{1, 1.5, 2, \dots, 5\}$.
- No Netflix!

Experimental Setup

- **Weak** generalization: fill in missing values in $\mathbf{Y}$.
- **Strong** generalization: inference on previously unseen columns of $\mathbf{Y}$.
- Results in terms of
 - ▶ Normalized mean absolute error (NMAE).
 - ▶ Root mean squared error (RMSE).
- Selected optimization parameters on Movielens 100k (results not presented).
- Used stochastic gradient descent with momentum 0.9 and learn rate $5e-4$.
- Initialize $\mathbf{X}$ with Gaussian random values, std $1e-3$.
- Initialize parameters to 1 apart from $\theta_{\text{bias}} = 0.11$ and $\sigma^2 = 5$.

Influence of latent space dimensionality

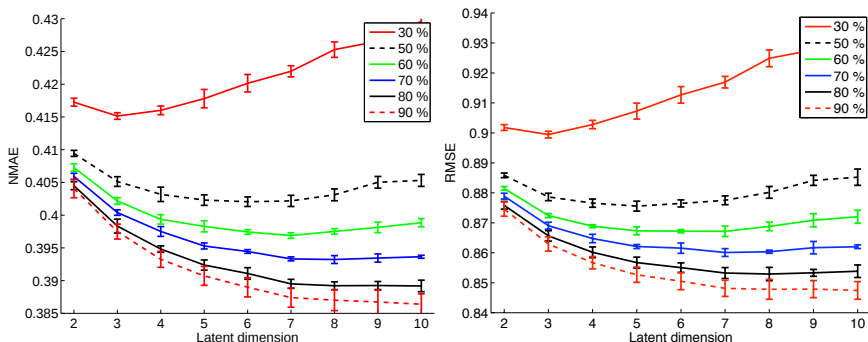


Figure: 1M MovieLens: NMAE and RMSE errors as a function of the latent space dimensionality for different percentages of the database used as training, i.e., 30-90 %.

Table: EachMovie Best results vs URP and Attitude algorithms of (Marlin, 2004), the MMMF of (Rennie and Srebro, 2005), E-MMMF of DeCoste (2007), and the Item-based approach of (Park and Pennock, 2007)[Park et al. 07].

	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	0.4209 ± 0.0017	0.4171 ± 0.0054
Ours RBF	0.4179 ± 0.0018	0.4134 ± 0.0049

Table: 1M MovieLens data set. Best results vs the URP and Attitude algorithms of (Marlin, 2004), the MMMF of (Rennie and Srebro, 2005), E-MMMF of DeCoste (2007), and the Item-based approach of (Park and Pennock, 2007) when using a non-linear latent space.

	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

- 10M Movielens: results in a NMAE of (**0.3968** $\pm$ 0.0165), and a RMSE of (**0.8740** $\pm$ 0.0278) using a 10D latent space.

- Promising approach on standard benchmarks.
- Could also make a map of users and do stochastic gradient descent over items if $D > N$
- Also exploring:
 - ▶ Side information (film genre etc.).
 - ▶ Have $\mathbf{X}$ represent users rather than items.
 - ★ Expect this to be useful when there are more items than users.
 - ▶ Using EP to model discrete outputs.
- Ruslan and Andriy have been looking at similar models!

- C. M. Bishop. Bayesian PCA. In M. J. Kearns, S. A. Solla, and D. A. Cohn, editors, *NIPS*, volume 11, pages 482–388, Cambridge, MA, 1999a. MIT.
- C. M. Bishop. Variational principal components. In *Proceedings Ninth International Conference on Artificial Neural Networks, ICANN'99*, volume 1, pages 509–514, 1999b.
- D. DeCoste. Collaborative prediction using ensembles of maximum margin matrix factorization. In Z. Ghahramani, editor, *ICML*, volume 24. Omnipress, 2007. [[Google Books](#)] .
- Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008. [[PDF](#)].

- N. D. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *JMLR*, 6:1783–1816, 11 2005.
- B. Marlin. Collaborative filtering: A machine learning perspective. Master's thesis, University of Toronto, 2004.
- T. P. Minka. Automatic choice of dimensionality for PCA. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *NIPS*, volume 13, pages 598–604, Cambridge, MA, 2001. MIT.
- S.-T. Park and D. M. Pennock. Applying collaborative filtering techniques to movie search for better ranking and browsing. In *13th ACM SIGKDD*, 2007. [PDF].
- J. D. M. Rennie and N. Srebro. Fast maximum margin factorization for collaborative prediction. In L. de Raedt and S. Wrobel, editors, *ICML*, volume 22, pages 713–719, 2005. [[Google Books](#)] . [PDF].

- R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using MCMC. In S. Roweis and A. McCallum, editors, *ICML*, volume 25. Omnipress, 2008a. [PDF]. In Press.
- R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *NIPS*, volume 20, pages 1257–1264, Cambridge, MA, 2008b. MIT. [PDF]. In press.
- M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *JRSSb*, 6(3):611–622, 1999. [DOI].

Variance as indicator of uncertainty

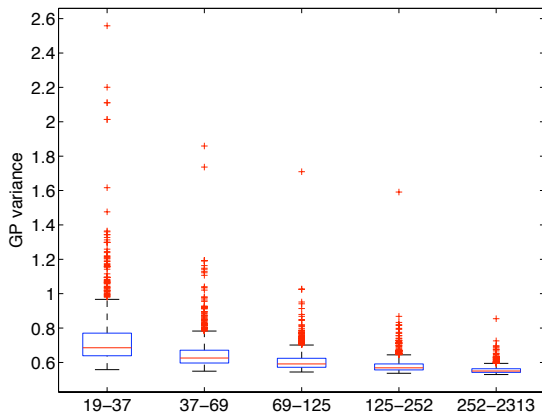


Figure: GP variance: as a function of the number of movies rated, for a 10D latent space learned on 1M MovieLens Weak. The variance of the GP is a good indicator of the uncertainty in the model, its value decreases with the amount of movies rated.

- Adding movie meta-data:

- ▶ There might be additional data about the movie that we might want to include, for example with the 1M MovieLens data, there is information about the genre of the movie (e.g., comedy and western).
- ▶ This can be encoded in a binary vector that defines the genre for each movie in the database.
- ▶ We can include this information in the kernel matrix. If the meta-data for a particular movie is given in a vector $\mathbf{m}_{i,:}$, then a covariance function can be created from the meta-data,

$$k_m(\mathbf{m}_{i,:}, \mathbf{m}_{j,:}) = \alpha_m \exp\left(-\frac{\gamma_m}{2} \|\mathbf{m}_i - \mathbf{m}_j\|^2\right).$$

- ▶ This can be combined with the covariance function defined in $\mathbf{x}$ -space through a tensor product,

$$k_{i,j} = k_m(\mathbf{m}_{i,:}, \mathbf{m}_{j,:}) k_x(\mathbf{x}_{i,:}, \mathbf{x}_{j,:}).$$

Kernel comparison EachMovie

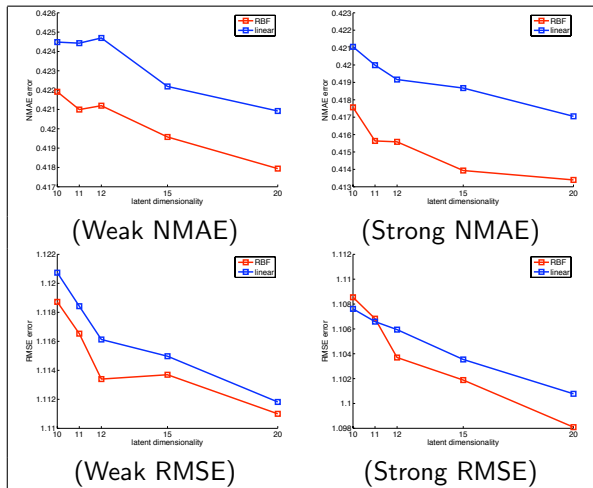


Figure: EachMovie: NMAE and RMSE errors for different kernels. Note that in general non-linear latent spaces result in better performance.

Kernel comparison 1M Movielens

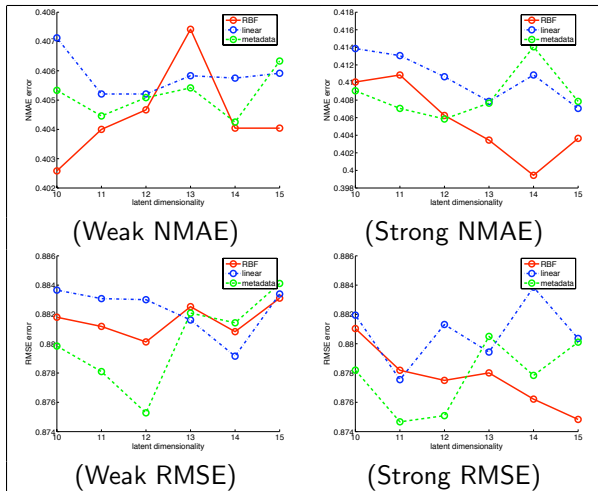


Figure: 1M MovieLens: NMAE and RMSE errors for different kernels. Note that in general non-linear latent spaces result in better performance.