Non-linear Matrix Factorization with Gaussian Processes

Neil D. Lawrence and Raquel Urtasun

University of Manchester and University of California, Berkeley

EMMDS Workshop 2009, Copenhagen, Denmark

3rd July 2009

Problem Definition

- 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 $\mathbf{Y} \in \Re^{N \times D}$.

This Talk

Gaussian Processes and Collaborative Filtering

- A split for machine learning? (inspired by reflections on NIPS 2005 keynote by Urs Hölzle “Petabyte Processing Made Easy”)
  - **Data rich**: Large amount of data, simpler models.

This Talk

Gaussian Processes and Collaborative Filtering

- A split for machine learning? (inspired by reflections on NIPS 2005 keynote by Urs Hölzle "Petabyte Processing Made Easy")
  - **Data rich**: Large amount of data, simpler models.
  - **Data scarce**: Complex models with a lot of prior knowledge encoded (e.g. Bayesian approaches)

This Talk

Gaussian Processes and Collaborative Filtering

- A split for machine learning? (inspired by reflections on NIPS 2005 keynote by Urs Hölzle "Petabyte Processing Made Easy")
  - **Data rich**: Large amount of data, simpler models.
  - **Data scarce**: Complex models with a lot of prior knowledge encoded (e.g. Bayesian approaches)
- Questions: Where are Gaussian processes in this?

This Talk

Gaussian Processes and Collaborative Filtering

- A split for machine learning? (inspired by reflections on NIPS 2005 keynote by Urs Hölzle "Petabyte Processing Made Easy")
  - **Data rich**: Large amount of data, simpler models.
  - **Data scarce**: Complex models with a lot of prior knowledge encoded (e.g. Bayesian approaches)
- Questions: Where are Gaussian processes in this?
  - Cubic scaling of inference from data!

This Talk

Gaussian Processes and Collaborative Filtering

- A split for machine learning? (inspired by reflections on NIPS 2005 keynote by Urs Hölzle "Petabyte Processing Made Easy")
  - **Data rich**: Large amount of data, simpler models.
  - **Data scarce**: Complex models with a lot of prior knowledge encoded (e.g. Bayesian approaches)
- Questions: Where are Gaussian processes in this?
  - Cubic scaling of inference from data!
  - Range of approaches to encoding prior knowledge in covariance function.

This Talk

Gaussian Processes and Collaborative Filtering

- A split for machine learning? (inspired by reflections on NIPS 2005 keynote by Urs Hölzle "Petabyte Processing Made Easy")
  - **Data rich**: Large amount of data, simpler models.
  - **Data scarce**: Complex models with a lot of prior knowledge encoded (e.g. Bayesian approaches)
- Questions: Where are Gaussian processes in this?
  - Cubic scaling of inference from data!
  - Range of approaches to encoding prior knowledge in covariance function.
- Question: Where is collaborative filtering in this?

This Talk

Gaussian Processes and Collaborative Filtering

- A split for machine learning? (inspired by reflections on NIPS 2005 keynote by Urs Hölzle "Petabyte Processing Made Easy")
  - **Data rich**: Large amount of data, simpler models.
  - **Data scarce**: Complex models with a lot of prior knowledge encoded (e.g. Bayesian approaches)
- Questions: Where are Gaussian processes in this?
  - Cubic scaling of inference from data!
  - Range of approaches to encoding prior knowledge in covariance function.
- Question: Where is collaborative filtering in this?
  - Netflix: around 100,000,000 ratings.

This Talk

Gaussian Processes and Collaborative Filtering

- A split for machine learning? (inspired by reflections on NIPS 2005 keynote by Urs Hölzle “Petabyte Processing Made Easy”)
  - **Data rich**: Large amount of data, simpler models.
  - **Data scarce**: Complex models with a lot of prior knowledge encoded (e.g. Bayesian approaches)
- Questions: Where are Gaussian processes in this?
  - Cubic scaling of inference from data!
  - Range of approaches to encoding prior knowledge in covariance function.
- Question: Where is collaborative filtering in this?
  - Netflix: around 100,000,000 ratings.
  - Purchase preference data sets are much larger!

This Talk

Gaussian Processes and Collaborative Filtering

- A split for machine learning? (inspired by reflections on NIPS 2005 keynote by Urs Hölzle "Petabyte Processing Made Easy")
  - **Data rich**: Large amount of data, simpler models.
  - **Data scarce**: Complex models with a lot of prior knowledge encoded (e.g. Bayesian approaches)
- Questions: Where are Gaussian processes in this?
  - Cubic scaling of inference from data!
  - Range of approaches to encoding prior knowledge in covariance function.
- Question: Where is collaborative filtering in this?
  - Netflix: around 100,000,000 ratings.
  - Purchase preference data sets are much larger!
- This talk: combining Collaborative filtering with Gaussian processes.

This Talk

Gaussian Processes and Collaborative Filtering

- A split for machine learning? (inspired by reflections on NIPS 2005 keynote by Urs Hölzle “Petabyte Processing Made Easy”)
    - **Data rich**: Large amount of data, simpler models.
    - **Data scarce**: Complex models with a lot of prior knowledge encoded (e.g. Bayesian approaches)
- Questions: Where are Gaussian processes in this?
    - Cubic scaling of inference from data!
    - Range of approaches to encoding prior knowledge in covariance function.
- Question: Where is collaborative filtering in this?
    - Netflix: around 100,000,000 ratings.
    - Purchase preference data sets are much larger!
- This talk: combining Collaborative filtering with Gaussian processes.
    - Isn’t that like putting Coca-Cola in malt whisky?

This Talk

Gaussian Processes and Collaborative Filtering

- A split for machine learning? (inspired by reflections on NIPS 2005 keynote by Urs Hölzle "Petabyte Processing Made Easy")
    - **Data rich**: Large amount of data, simpler models.
    - **Data scarce**: Complex models with a lot of prior knowledge encoded (e.g. Bayesian approaches)
- Questions: Where are Gaussian processes in this?
    - Cubic scaling of inference from data!
    - Range of approaches to encoding prior knowledge in covariance function.
- Question: Where is collaborative filtering in this?
    - Netflix: around 100,000,000 ratings.
    - Purchase preference data sets are much larger!
- This talk: combining Collaborative filtering with Gaussian processes.
    - Isn't that like putting Coca-Cola in malt whisky?

Collaborative Filtering 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.

Neighborhood Approach

users

items

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

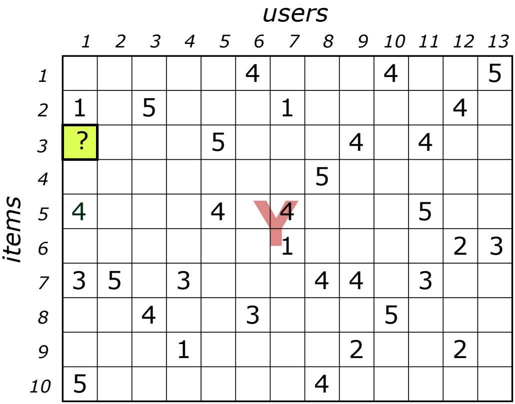
Neighborhood Approach

users

$\mathbf{Y}$

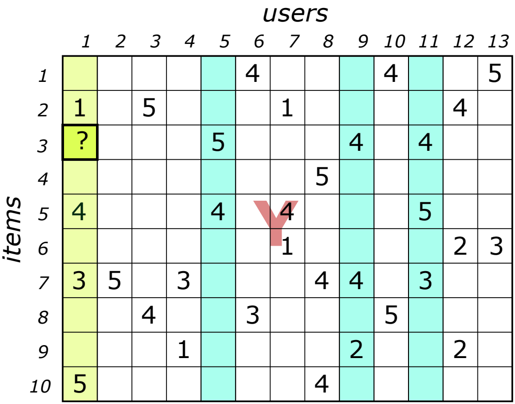
items	1	2	3	4	5	6	7	8	9	10	11	12	13
1						4				4			5
2	1		5				1					4	
3					5				4		4		
4								5					
5	4				4		4				5		
6							1					2	3
7	3	5		3				4	4		3		
8			4			3				5			
9				1					2			2	
10	5							4					

Neighborhood Approach



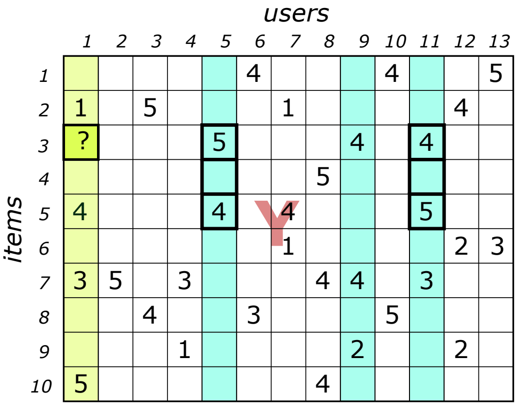
	1	2	3	4	5	6	7	8	9	10	11	12	13
1						4				4			5
2	1		5				1					4	
3	?				5				4		4		
4								5					
5	4				4		4				5		
6							1					2	3
7	3	5		3				4	4		3		
8			4			3				5			
9				1					2			2	
10	5							4					

Neighborhood Approach



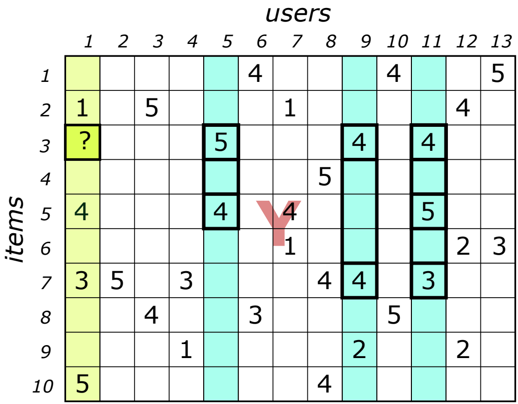
	1	2	3	4	5	6	7	8	9	10	11	12	13
1						4				4			5
2	1		5				1					4	
3	?				5				4		4		
4								5					
5	4				4		4				5		
6							1					2	3
7	3	5		3				4	4		3		
8			4			3				5			
9				1					2			2	
10	5							4					

Neighborhood Approach



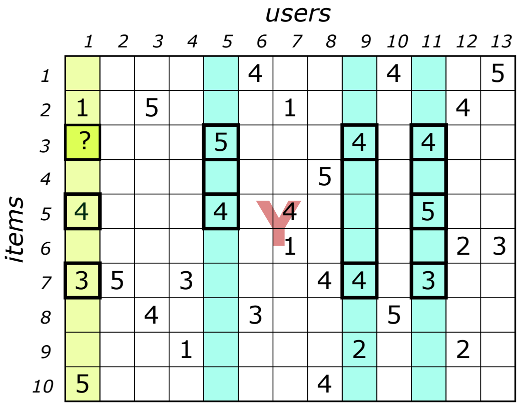
	1	2	3	4	5	6	7	8	9	10	11	12	13
1						4				4			5
2	1		5				1					4	
3	?				5				4		4		
4								5					
5	4				4		4				5		
6							1					2	3
7	3	5		3				4	4		3		
8			4			3				5			
9				1					2			2	
10	5							4					

Neighborhood Approach



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

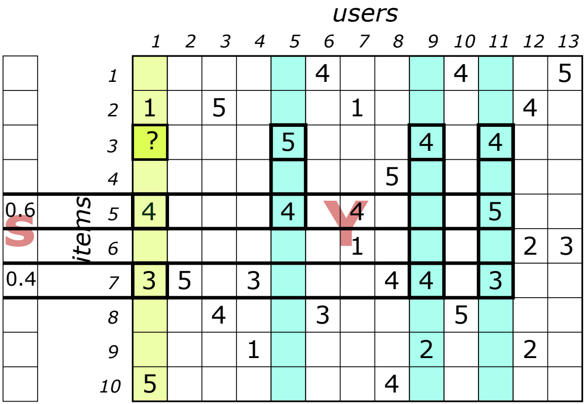
Neighborhood Approach



users

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

Neighborhood Approach

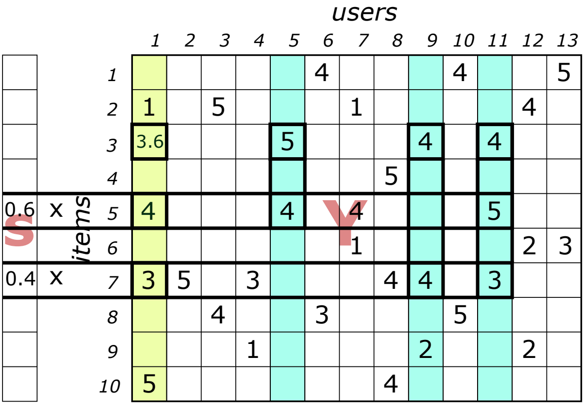


		1	2	3	4	5	6	7	8	9	10	11	12	13
	1						4				4			5
	2	1		5				1					4	
	3	?				5				4		4		
	4								5					
0.6	5	4				4		4				5		
	6							1					2	3
0.4	7	3	5		3				4	4		3		
	8			4			3				5			
	9				1					2			2	
	10	5							4					

Neighborhood Approach

compute result via inner product

$\mathbf{s}_{:,3}^{\mathsf{T}}\mathbf{y}_{:,1}$



			1	2	3	4	5	6	7	8	9	10	11	12	13
		1						4				4			5
		2	1		5				1					4	
		3	3.6				5				4		4		
		4								5					
0.6	x	5	4				4		4				5		
		6							1					2	3
0.4	x	7	3	5		3				4	4		3		
		8			4			3				5			
		9				1					2			2	
		10	5							4					

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,

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

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

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

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

Matrix Factorization

users

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

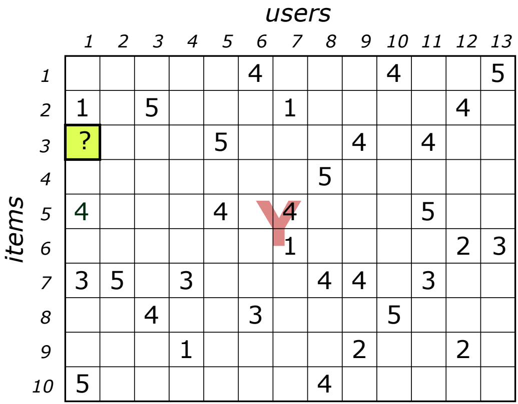
Matrix Factorization

users

Y

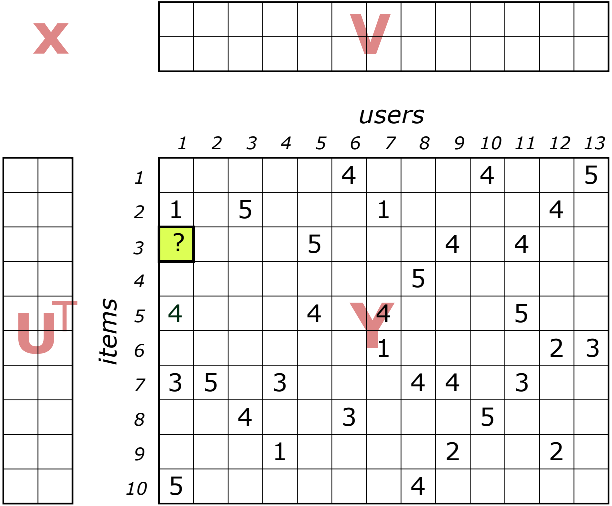
items	1	2	3	4	5	6	7	8	9	10	11	12	13
1						4				4			5
2	1		5				1					4	
3					5				4		4		
4								5					
5	4				4		4				5		
6							1					2	3
7	3	5		3				4	4		3		
8			4			3				5			
9				1					2			2	
10	5							4					

Matrix Factorization



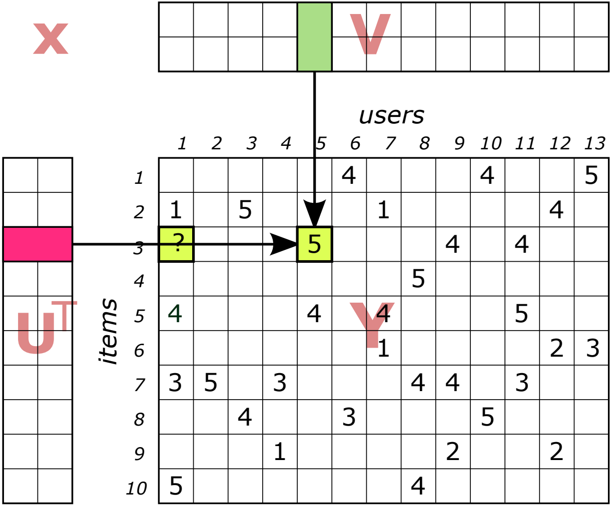
	1	2	3	4	5	6	7	8	9	10	11	12	13
1						4				4			5
2	1		5				1					4	
3	?				5				4		4		
4								5					
5	4				4		4				5		
6							1					2	3
7	3	5		3				4	4		3		
8			4			3				5			
9				1					2			2	
10	5							4					

Matrix Factorization



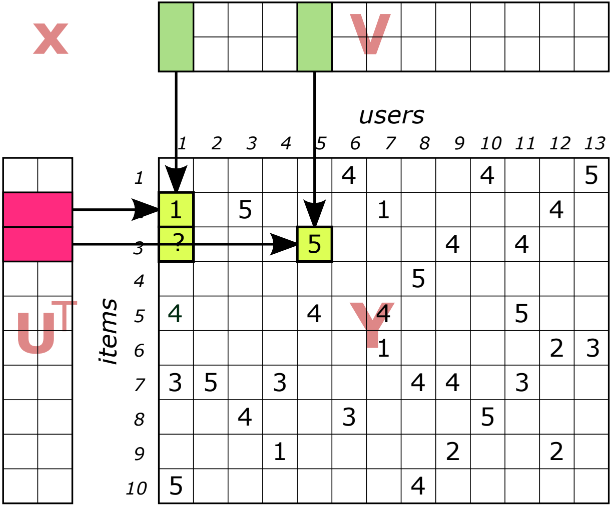
	1	2	3	4	5	6	7	8	9	10	11	12	13
1						4				4			5
2	1		5				1					4	
3	?				5				4		4		
4								5					
5	4				4		4				5		
6							1					2	3
7	3	5		3				4	4		3		
8			4			3				5			
9				1					2			2	
10	5							4					

Matrix Factorization



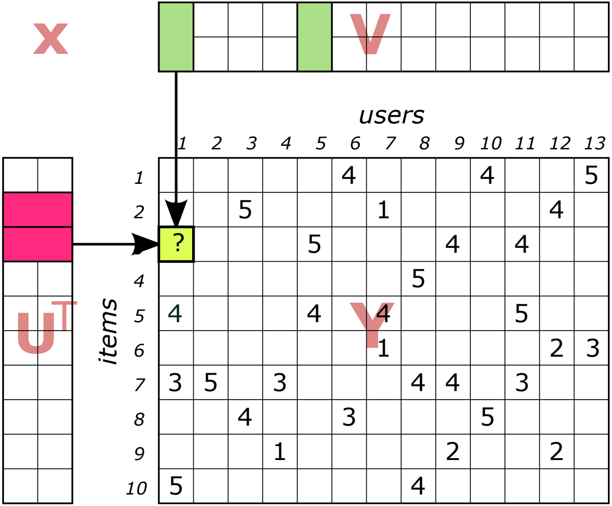
items \ users	1	2	3	4	5	6	7	8	9	10	11	12	13
1						4				4			5
2	1		5				1					4	
3	?				5				4		4		
4								5					
5	4				4		4				5		
6							1					2	3
7	3	5		3				4	4		3		
8			4			3				5			
9				1					2			2	
10	5							4					

Matrix Factorization



	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		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



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

This Talk

- 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.

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

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

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

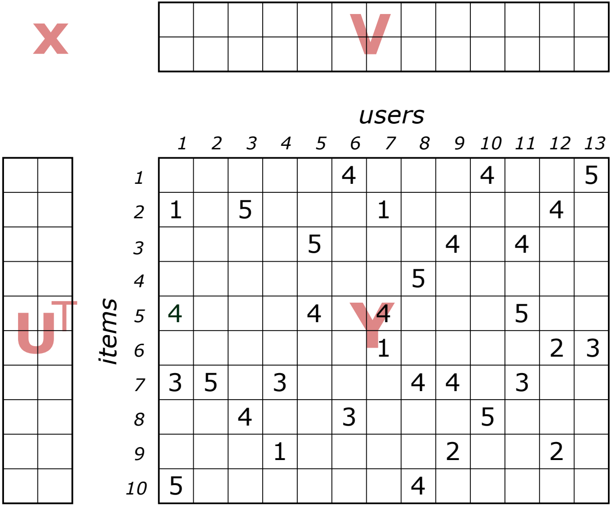
- Cannot marginalize both **U** and **V** — use sampling or MAP (Salakhutdinov and Mnih, 2008b,a).

PCA Equivalence

Probabilistic Matrix Factorization is Bayesian PCA

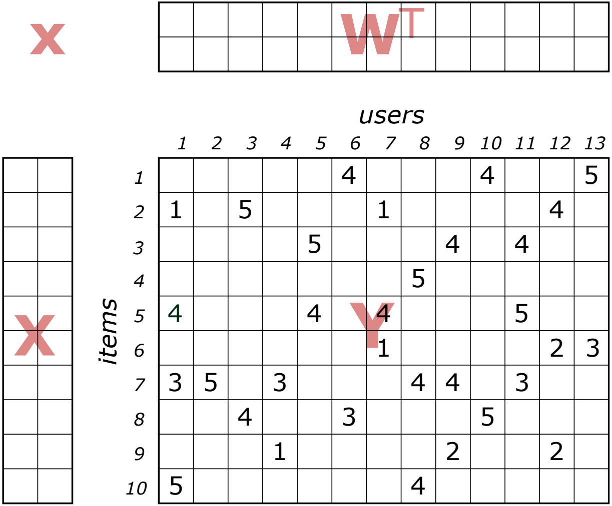
- Consider change of notation,
  - $\mathbf{X} \equiv \mathbf{U}^\top \in \Re^{N \times q}$ (latent variables)
  - $\mathbf{W} \equiv \mathbf{V}^\top \in \Re^{D \times q}$ (mapping matrix)

Change Notation



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

Change Notation



items \ users	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		4				5		
6							1					2	3
7	3	5		3				4	4		3		
8			4			3				5			
9				1					2			2	
10	5							4					

Probabilistic PCA

- Using this notation:

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

cf multi-output linear regression

- Prior over **X**,

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

gives

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

- Optimize wrt **W** — absorb α_x into **W**.
- Probabilistic PCA (Tipping and Bishop, 1999).

Dual Probabilistic PCA

- *Instead* take prior over **W**,

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

cf Bayesian multi-output linear regression

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

- Optimize wrt *inputs* **X** — Probabilistic Principal Coordinate Analysis (Dual PPCA, Lawrence, 2005).

Bayesian PCA and PMF

- 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

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

- Or a Gaussian Process with a 'linear' covariance function.

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

  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).

Non-Linear Matrix Factorization

- 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.

Non-Linear Matrix Factorization

- The marginal likelihood of DPPCA is that of a Bayesian linear regression

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

- Replace inner product matrix with covariance function for non-linear model.

Non-Linear Matrix Factorization

- The marginal likelihood of DPPCA is that of a Bayesian linear regression

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

- Replace inner product matrix with covariance function for non-linear model.

Missing values

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

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

- For sparse data

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

Stochastic Gradient Descent

- 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 jth user we minimize the negative log likelihood

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

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

Stochastic Gradient Descent

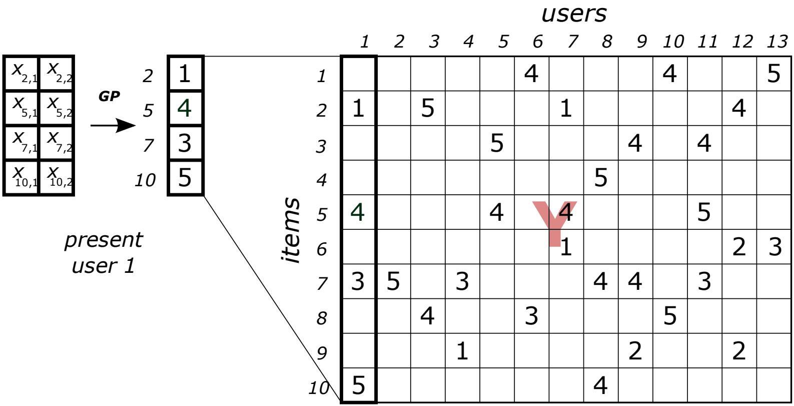
users

Y

items	*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		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



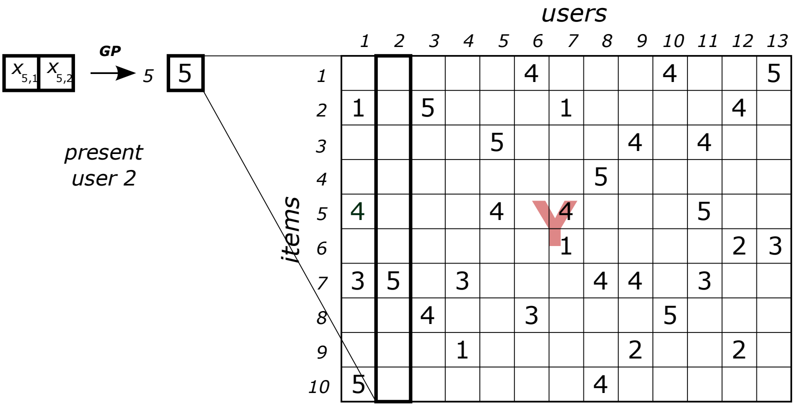
$x_{2,1}$	$x_{2,2}$	
$x_{5,1}$	$x_{5,2}$	
$x_{7,1}$	$x_{7,2}$	
$x_{10,1}$	$x_{10,2}$	

2	1
5	4
7	3
10	5

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

Each step updates $\mathbf{X}_{\mathbf{i}_j,:}$.

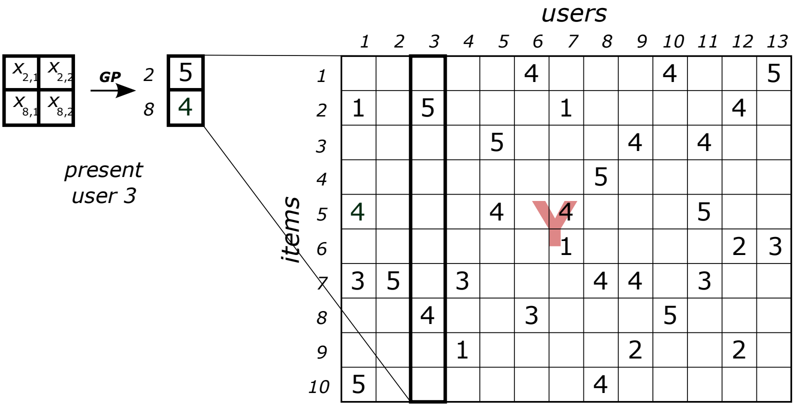
Stochastic Gradient Descent



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

Complexity of GP cubic in N_j not N.

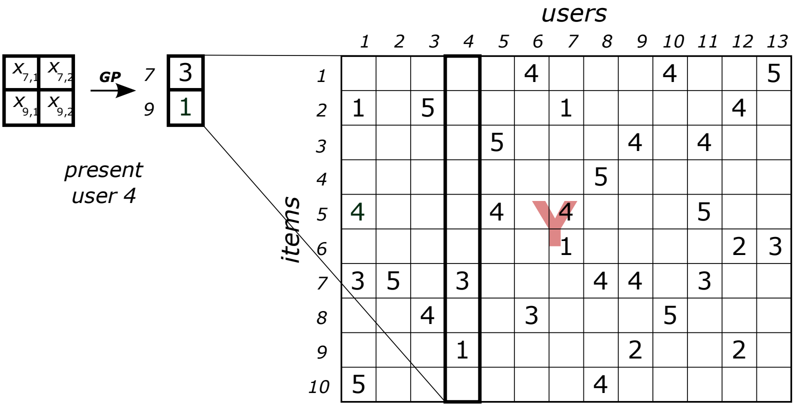
Stochastic Gradient Descent



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

No Sparse GP approximations required.

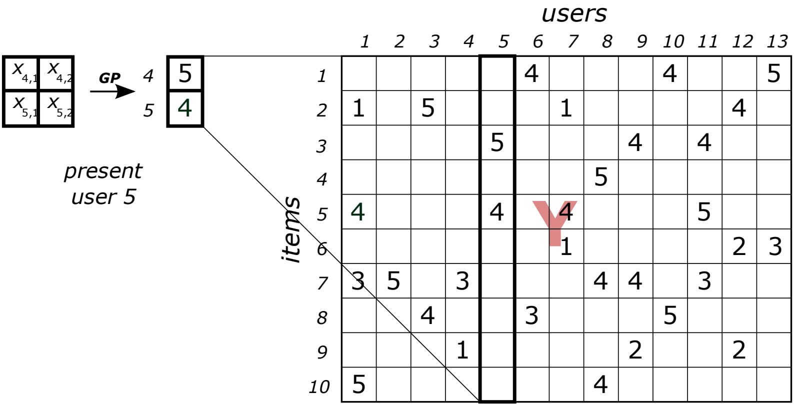
Stochastic Gradient Descent



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

No Sparse GP approximations required.

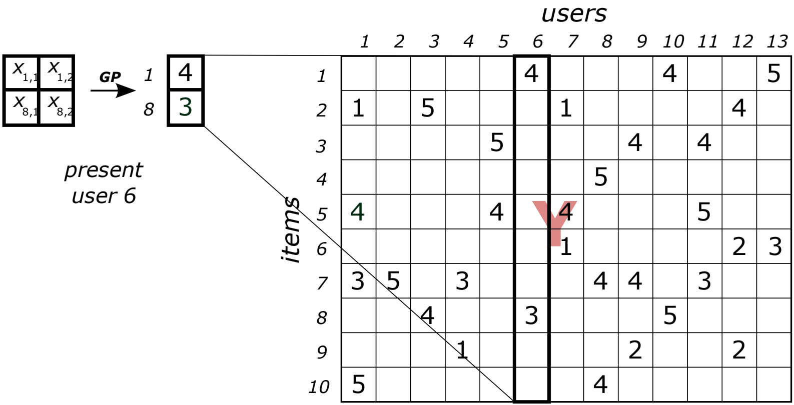
Stochastic Gradient Descent



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

No Sparse GP approximations required.

Stochastic Gradient Descent



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

No Sparse GP approximations required.

Model Prediction

Relation to Neighborhood Approach

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

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

  with $\mathbf{s} = \left(\mathbf{K}_{\mathbf{i}_j,\mathbf{i}_j} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{k}_{\mathbf{i}_j,\ell}$.
    - Similar predictive form to neighborhood approach.
    - Our "similarities" are computed in $\mathbf{X}$ space.

Predictions

- 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:

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

Covariance Functions Used

- We present results with:

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

  or

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

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

Experimental evaluation: Datasets

- Followed (Marlin, 2004) in experimental setup for:
  - **EachMovie**: 2.6 million ratings for $N = 1,648$ movies and $D = 74,424$ users. Ratings range $\{1, \cdots, 6\}$.
  - **1M Movielens**: 1 million ratings for $6,040$ users, and $3,952$ movies, with ratings ranging $\{1, \cdots, 5\}$.
- Also try **10M Movielens**: 10 million ratings for $D = 71,567$ users and $N = 10,681$ movies, with ratings ranging $\{1, 1.5, 2, \cdots, 5\}$.
- 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

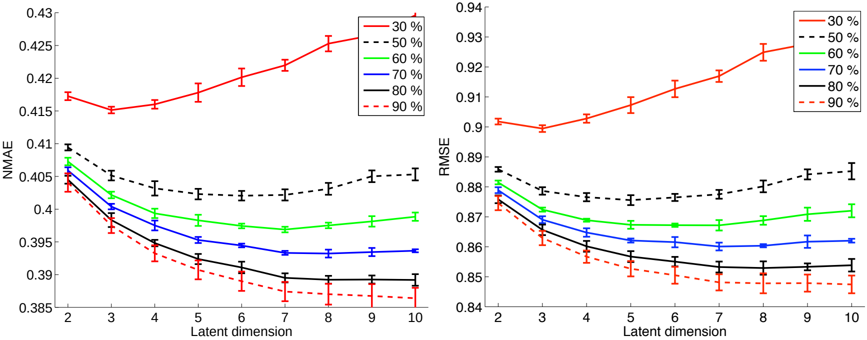


Figure: **1M MovieLens:** NMAE and RMSE errors as a function of the latent space dimensionality for different percentages of the database used as training, i.e., 30-90 %.

Results EachMovie

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**

Results Movielens

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** ± 0.0165), and a RMSE of (**0.8740** ± 0.0278) using a 10D latent space.

Conclusions

- 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 on Netflix.

References I

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].

References II

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].

References III

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

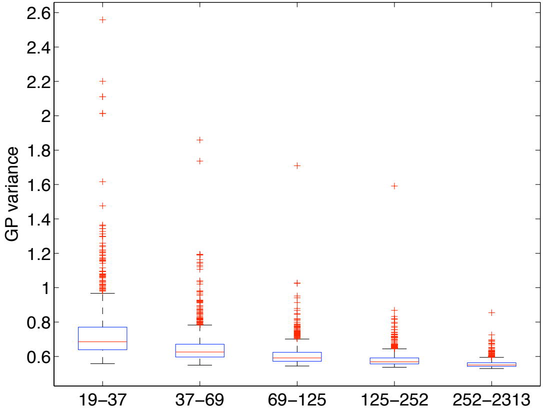


Figure: **GP variance:** as a function of the number of movies rated, for a 10D latent space learned on 1M MovieLens Weak. The variance of the GP is a good indicator of the uncertainty in the model, its value decreases with the amount of movies rated.

Other extensions and results

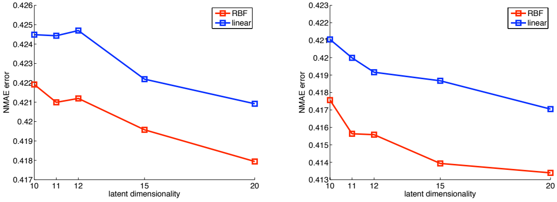
- Adding movie meta-data:
  - There might be additional data about the movie that we might want to include, for example with the 1M MovieLens data, there is information about the genre of the movie (*e.g.*, comedy and western).
  - This can be encoded in a binary vector that defines the genre for each 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,

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

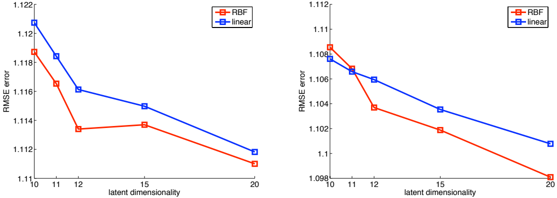
  - This can be combined with the covariance function defined in $\mathbf{x}$-space through a tensor product,

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

Kernel comparison EachMovie



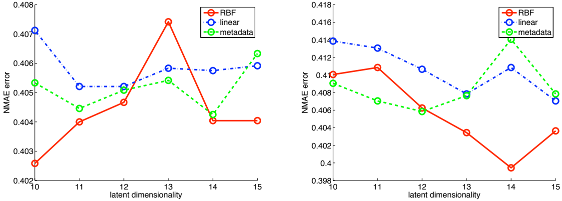
(Weak NMAE) (Strong NMAE)



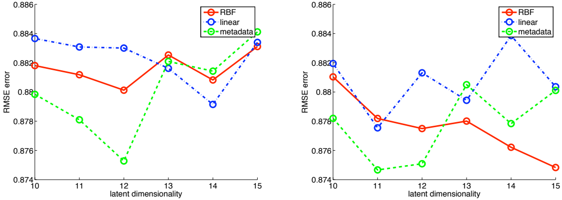
(Weak RMSE) (Strong RMSE)

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



(Weak NMAE) (Strong NMAE)



(Weak RMSE) (Strong RMSE)

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