Factorization Models for Recommender Systems and Other Applications

Part I

Lars Schmidt-Thieme, Steffen Rendle

Information Systems and Machine Learning Lab (ISMLL)
Institute of Computer Science
University of Hildesheim, Germany
—
Institute of Computer Science,
University of Konstanz, Germany

KDD 2012 — Tutorial
August 12, 2012, Bejing

Outline

Outline

PCA

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18
11	17	13	14	16	4	9	14	1	13	0	11	13	5	7	1	4	11	5
12	12	11	9	12	6	5	12	1	9	1	7	12	5	6	0	4	8	2
13	18	16	16	16	8	6	14	1	14	0	9	12	4	8	0	2	11	3
14	18	16	15	19	8	6	11	1	15	1	7	13	5	8	0	2	9	3
15	17	13	13	22	8	4	18	1	10	3	8	11	7	6	1	2	10	2
16	16	13	16	21	9	3	17	1	10	4	7	12	7	5	1	2	11	3
17	22	19	10	16	10	4	18	1	12	2	8	11	6	8	0	1	8	2
18	20	10	24	10	6	9	21	0	7	0	7	18	4	9	1	4	8	2
19	20	21	12	15	9	7	11	1	9	3	8	14	6	7	0	1	10	3
20	20	21	12	15	9	7	11	1	9	3	9	14	5	7	0	1	10	3
21	18	11	24	10	9	6	19	0	12	0	7	14	4	11	0	4	9	1
22	20	12	23	10	8	5	20	0	13	0	6	13	4	11	0	4	10	1
23	18	19	18	16	8	4	12	0	12	0	10	15	8	6	1	1	12	1
24	17	16	17	15	8	6	14	1	14	0	9	12	4	8	0	3	11	3
25	19	17	14	16	8	6	14	1	14	0	8	12	4	8	0	2	12	3
26	19	17	15	16	8	5	14	1	14	0	8	12	4	8	0	2	12	3
27	19	15	16	16	8	6	14	1	15	0	8	12	4	8	0	2	12	3
28	17	17	16	19	8	6	11	1	15	1	7	13	5	8	0	2	9	3
29	18	17	15	19	8	6	11	1	15	1	7	13	5	8	0	2	9	3
30	22	19	10	16	10	5	17	1	12	2	8	11	6	8	0	1	8	2
31	18	20	10	18	6	8	17	1	14	1	5	16	4	7	0	2	9	2
32	18	16	16	15	8	6	13	1	14	1	8	12	4	8	1	2	12	3
33	20	21	12	15	9	7	11	1	10	3	8	14	7	7	0	1	9	3
34	20	21	12	15	9	7	11	1	10	3	9	14	5	7	0	1	10	3
35	18	12	23	10	9	5	20	0	14	0	7	12	4	11	0	4	10	1
36	18	12	21	10	10	5	18	0	13	0	8	12	4	12	0	4	10	1
37	17	12	22	10	8	5	18	0	14	0	5	13	4	10	0	3	9	1
38	17	16	16	16	8	6	15	1	14	0	9	12	4	8	0	2	11	3
39	19	17	15	17	7	6	15	1	14	0	8	12	4	8	0	2	10	3
40	18	16	16	19	8	6	11	1	15	1	7	13	5	8	0	2	9	3
41	18	17	15	17	8	6	15	1	14	0	8	12	4	8	0	3	9	3
42	15	12	14	23	8	3	17	1	9	4	7	15	6	6	1	2	11	2
43	13	11	14	22	7	3	17	1	10	4	8	13	6	6	1	3	11	2
44	16	11	15	23	10	4	18	1	10	3	7	12	6	5	1	2	9	3
45	14	11	14	25	11	3	19	2	10	2	7	12	6	5	1	2	9	3
46	11	11	15	24	10	5	18	1	11	1	7	14	5	7	2	3	11	2
47	15	9	12	21	8	4	21	1	10	3	7	15	7	6	1	3	10	3
48	15	11	15	22	7	3	19	1	8	3	4	14	6	5	1	2	10	2

PCA for $x_1, \ldots, x_n \in \mathbb{R}^m$:

$$\underset{\Psi, \phi_1, \ldots, \phi_n}{\operatorname{argmin}} \sum_{i=1}^{n} ||x_i - \Psi \phi_i||^2, \quad \phi_i \in \mathbb{R}^K, \Psi \in \mathbb{R}^{m \times K}, K \leq n$$

$\rightsquigarrow$ SVD: for $X := (x_1, \ldots, x_n)^T \in \mathbb{R}^{n \times m}$:

$$X = UDV^T$$

with $U \in \mathbb{R}^{n \times m}$ orthonormal $(U^T U = I)$

$V \in \mathbb{R}^{m \times m}$ orthonormal $(V^T V = I)$

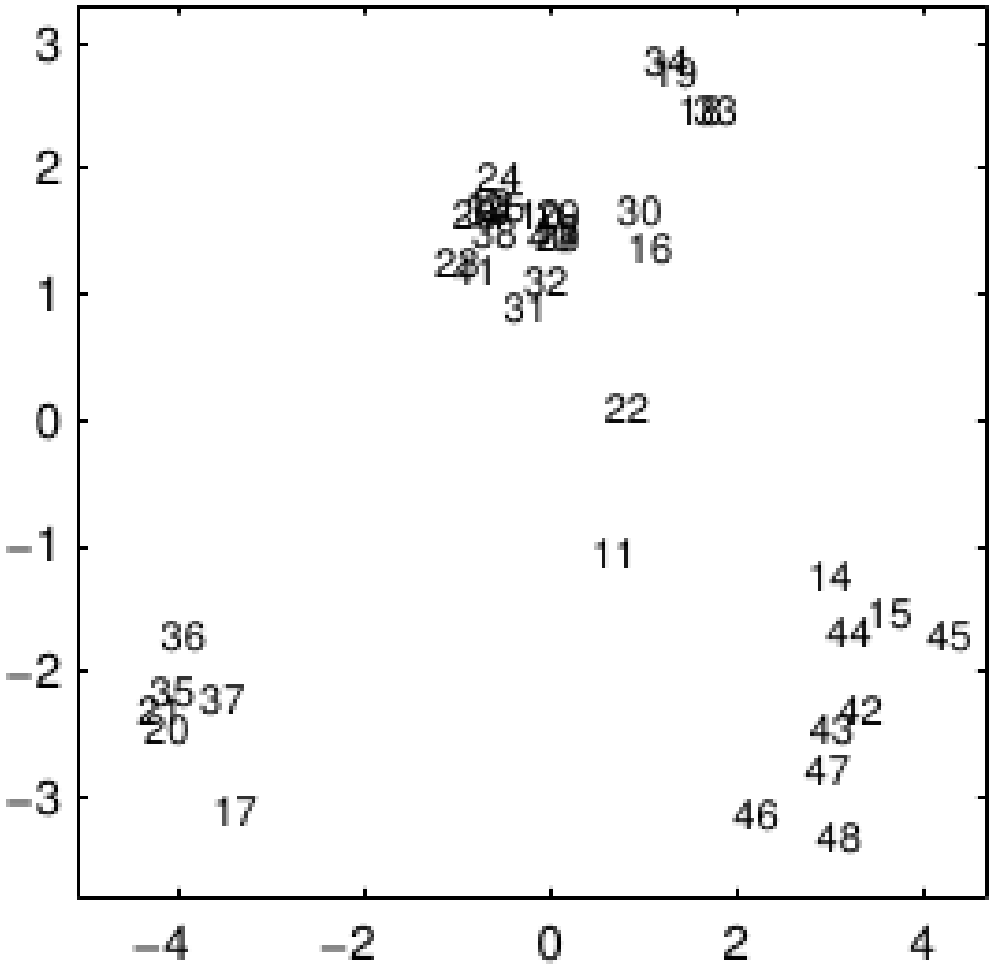
$D \in \mathbb{R}^{m \times m}$ diagonal with $D_{1,1} \geq D_{2,2} \geq \ldots \geq D_{m,m}$

yields

$$\Psi = (V_{.,1}, \ldots, V_{.,K})$$
$$\phi_i = \Psi^T x_i, \quad i = 1, \ldots, n$$

Solve SVD algebraically.

PCA



[Tipping and Bishop, 1999]

PCA

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18
11		13	14	16	4	9	14	1	13	0	11	13	5	7	1	4	11	5
12		11	9	12	6	5	12	1			7		5	6	0	4	8	2
13	18	16	16	16	8	6	14	1		0	9	12	4	8	0		11	3
14	18	16	15	19			11	1	15	1	7	13	5	8	0	2		3
15	17	13		22	8	4	18	1			8	11	7	6	1	2		2
16	16	13	16		9	3	17	1		4	7	12		5	1	2	11	
17	22	19	10		10	4	18	1	12			11	6	8		1	8	
18	20		24				21	0			7		4		1	4	8	2
19	20	21		15	9	7	11	1	9	3	8	14	6	7	0	1	10	3
20	20	21	12	15		7	11	1	9	3		14		7	0	1	10	3
21		11	24	10	9	6	19	0	12		7	14	4	11	0	4	9	
22		12	23		8	5		0	13	0	6	13	4	11		4	10	1
23	18			16	8	4	12			0	10	15	8	6	1	1	12	1
24	17	16	17	15	8	6	14	1	14	0	9	12	4	8	0	3		3
25	19	17	14	16	8	6	14	1	14		8	12	4	8	0	2		3
26	19		15	16	8	5	14	1	14	0	8	12	4		0	2	12	3
27	19	15	16	16	8	6	14	1	15	0			4		0		12	3
28	17	17	16	19	8		11	1	15	1	7	13	5		0	2	9	3
29	18	17	15	19	8	6	11	1	15	1			5	8	0		9	3
30	22	19	10	16	10		17	1	12	2	8		6	8	0	1		2
31	18	20	10					1		1	5	16			0		9	2
32	18		16		8	6	13	1	14	1	8	12		8	1	2	12	3
33	20	21		15	9				10	3	8	14	7	7	0	1	9	3
34	20	21		15	9		11		10	3		14	5		0	1	10	3
35	18	12	23	10	9	5	20	0		0	7			11		4	10	1
36		12	21	10	10	5	18	0	13	0	8			12		4	10	1
37		12	22	10		5		0			5	13		10		3	9	1
38				16		6	15	1	14	0	9	12	4	8	0	2		3
39		17	15	17			15	1	14	0	8		4	8	0	2	10	3
40		16	16			6	11		15	1	7	13	5	8	0	2	9	3
41	18		15	17	8		15	1	14	0		12	4	8	0	3	9	3
42		12	14	23	8		17	1		4	7	15	6			2	11	2
43		11	14	22		3	17	1	10	4	8	13	6	6	1	3	11	2
44	16		15	23	10	4	18	1	10	3	7	12	6	5	1	2	9	3
45			14	25	11	3		2	10	2	7	12			1	2		3
46	11	11	15	24		5	18	1	11	1	7		5	7		3		2
47	15	9	12	21	8	4	21		10	3	7	15		6			10	3
48		11	15		7	3		1	8	3	4	14	6	5	1	2		2

PCA with missing values

$$\underset{\Psi,\phi_1,\ldots,\phi_n}{\text{argmin}} \sum_{i=1}^{n} ||x_i - \Psi\phi_i||^2, \qquad \phi_i \in \mathbb{R}^K, \Psi \in \mathbb{R}^{m\times K}, K \leq n$$

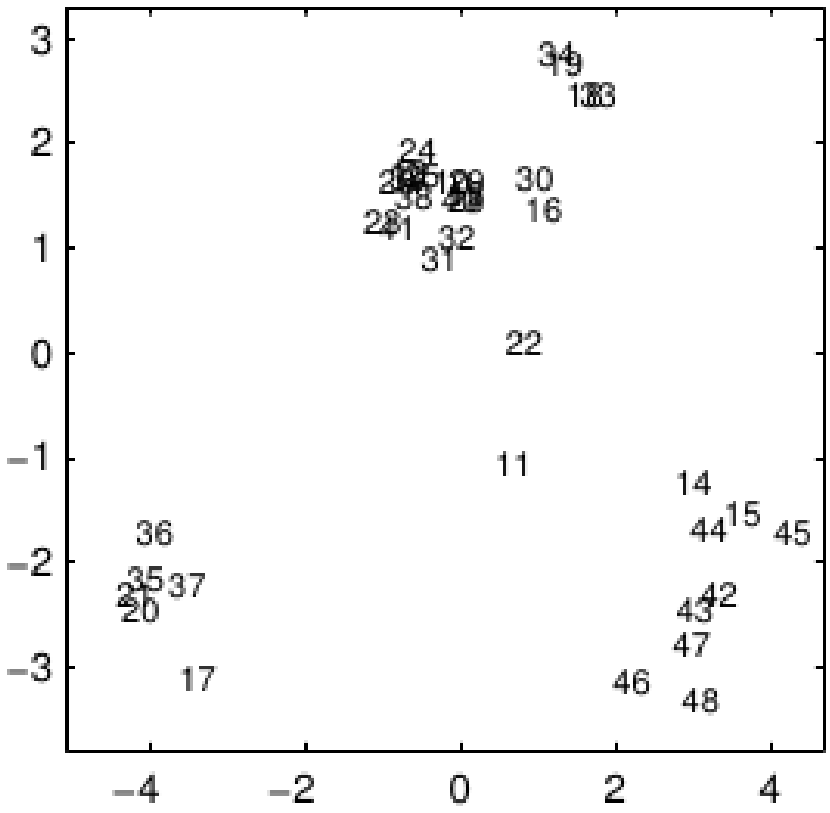
$$= \underset{\psi_1,\ldots,\psi_m,\phi_1,\ldots,\phi_n}{\text{argmin}} \sum_{i=1}^{n}\sum_{j=1}^{m}(x_{i,j} - \psi_j^T\phi_i)^2, \qquad \Psi = (\psi_1,\ldots,\psi_m)$$

$$\rightsquigarrow \underset{\psi_1,\ldots,\psi_m,\phi_1,\ldots,\phi_n}{\text{argmin}} \sum_{i=1}^{n}\sum_{j=1}^{m} w_{i,j}(x_{i,j} - \psi_j^T\phi_i)^2, \qquad w_{i,j} \in \{0,1\}$$

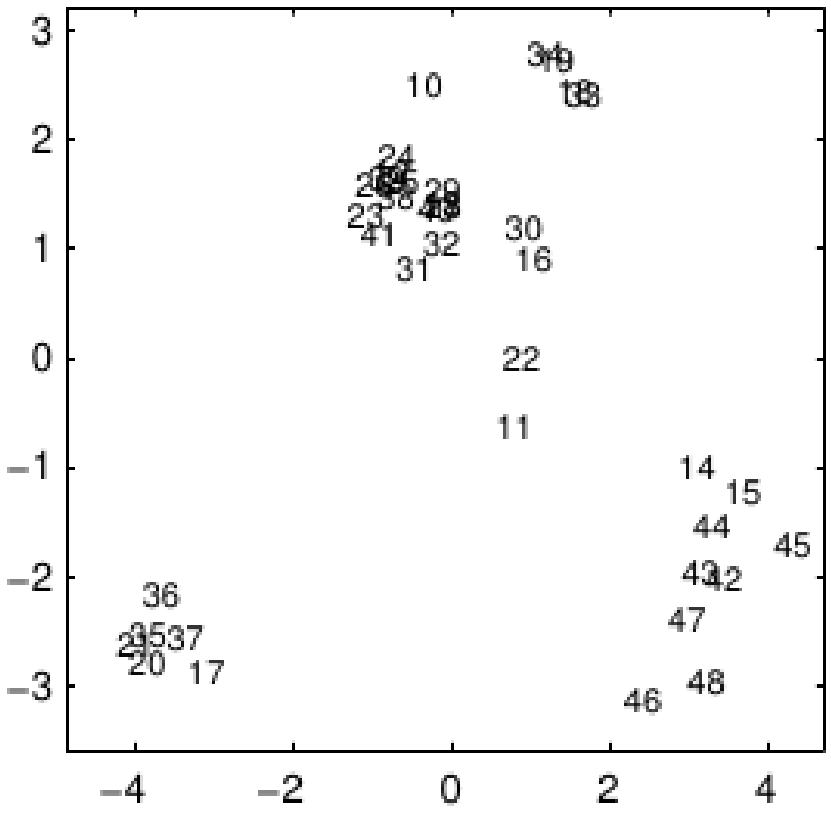
Solve numerically.

PCA

from complete data



from data with
20% missing values



[Tipping and Bishop, 1999]

The Rating Recommendation Problem

item	Toy Story	GoldenEye	Four Rooms	Get Shorty	Copycat	Shanghai Triad	Twelve Monkeys	Babe	Dead Man Walking	...
user Anna	4			3						
Berta	1									
Clara		4								
Daniel					5					
Egon								2		
Fred								4		
⋮										

The Recommendation Problem / Attributes

	item	1	2	3	4	5	6	7	8	9	...
	year	'95	'95	'95	'95	'95	'95	'95	'95	'95	
	action	−	+	−	+	−	−	−	−	−	
	children's	+	−	−	−	−	−	−	+	−	
	...										

	gender	age	...
user 1	m	24	
2	f	53	
3	m	23	
4	m	24	
5	f	33	
6	m	42	
⋮			

item	1	2	3	4	5	6	7	8	9	...
user 1	4			3						
2	1									
3		4								
4					5					
5								2		
6								4		
⋮										

Discard user/item attributes (for a start).

Is Rating Prediction a Regression Problem?

- Yes, one for each user,
  with the item attributes as predictors (content-based filtering),
  - but then we do **not share information between users** (collaborative information)
  - but we said we have **no item attributes** for a start.

Is Rating Prediction a Regression Problem?

- Yes, one for each user,
  with the item attributes as predictors (content-based filtering),
  - but then we do **not share information between users** (collaborative information)
  - but we said we have **no item attributes** for a start.

- Yes, one for each item,
  with the ratings for all the other items as predictors,
  - but there are lots of **missing values**,
  - but all these regression **problems are related** (multi-task learning).

Is Rating Prediction a Regression Problem?

- Yes, one for each user,
  with the item attributes as predictors (content-based filtering),
  - but then we do **not share information between users** (collaborative information)
  - but we said we have **no item attributes** for a start.

- Yes, one for each item,
  with the ratings for all the other items as predictors,
  - but there are lots of **missing values**,
  - but all these regression **problems are related** (multi-task learning).

- Yes, with just two predictors user ID and item ID,
  - but both **variables are nominal with many levels**, without interactions not leading to personalized models,
  - but with **interactions** having as many parameters as possible instances.

Collaborative Filtering (Nearest Neighbor Models)

Idea: lets use a **similarity measure** on the available data.

$$\hat{y}(u,i) := \bar{y}_{u,.} + \frac{1}{|N_u|} \sum_{v \in N_u} (y_{v,i} - \bar{y}_{v,.}), \quad N_u := \operatorname{argmax}^k_{v \in U} \operatorname{sim}(u,v)$$

$$\operatorname{sim}(u,v) := \frac{\langle y_{u|v}, y_{v|u} \rangle}{||y_{u|v}||_2 \, ||y_{v|u}||_2}$$

$$y_{u|v} := (y_{u,i})_{i \in I_u \cap I_v}, \quad I_u := \{i \in I \mid y_{u,i} \neq .\}$$

- but the choice of the similarity measure is arbitrary,
- but a nearest neighbor model is not a strong regression model.

[Goldberg et al., 1992; Resnick et al., 1994]

Matrix Factorization

Idea: **learn latent features** on the available data.

	item	1	2	3	4	5	6	7	8	9	...
ψ_1		...	...	...	...	...	...	...	...	...	
$\vdots$		...	...	...	...	...	...	...	...	...	
ψ_K		...	...	...	...	...	...	...	...	...	

	ϕ_1	...	ϕ_K
user 1	...	...	...
2	...	...	...
3	...	...	...
4	...	...	...
5	...	...	...
6	...	...	...
$\vdots$			

item	1	2	3	4	5	6	7	8	9	...
user 1	4			3						
2	1									
3		4								
4					5					
5								2		
6								4		
$\vdots$										

and assume a simple **bilinear reconstruction model**.

Matrix Factorization

Idea: **learn latent features** on the available data.

$$\hat{y}(u,i) := a + b_u + c_i + \langle \phi_u, \psi_i \rangle, \quad a, b_u, c_i \in \mathbb{R}, \phi_u, \psi_i \in \mathbb{R}^K$$

Regularized Loglikelihood:

$$f(\phi, \psi; y) := \sum_{(u,i,y) \in D} (y - \hat{y}(u,i))^2 + \lambda(||\phi||^2 + ||\psi||^2)$$

- $D := \{(u_1, i_1, y_1), \ldots, (u_N, i_N, y_N)\}$ sparse matrix data (in triples).
- can be easily learned by **stochastic gradient descent**.
- $\lambda \in \mathbb{R}_0^+$ and $K \in \mathbb{N}$ are **hyperparameters**.

Matrix Factorization

Idea: **learn latent features** on the available data.

$$\hat{y}(u,i) := a + b_u + c_i + \sum_{k=1}^{K} \phi_{u,k}\psi_{i,k}, \quad a, b_u, c_i \in \mathbb{R}, \phi_u, \psi_i \in \mathbb{R}^K$$

Regularized Loglikelihood:

$$f(\phi, \psi; y) := \sum_{(u,i,y)\in D} (y - \hat{y}(u,i))^2 + \lambda(||\phi||^2 + ||\psi||^2)$$

- $D := \{(u_1, i_1, y_1), \ldots, (u_N, i_N, y_N)\}$ sparse matrix data (in triples).
- can be easily learned by **stochastic gradient descent**.
- $\lambda \in \mathbb{R}_0^+$ and $K \in \mathbb{N}$ are **hyperparameters**.

Recommendation Quality (Matrix Factorization)

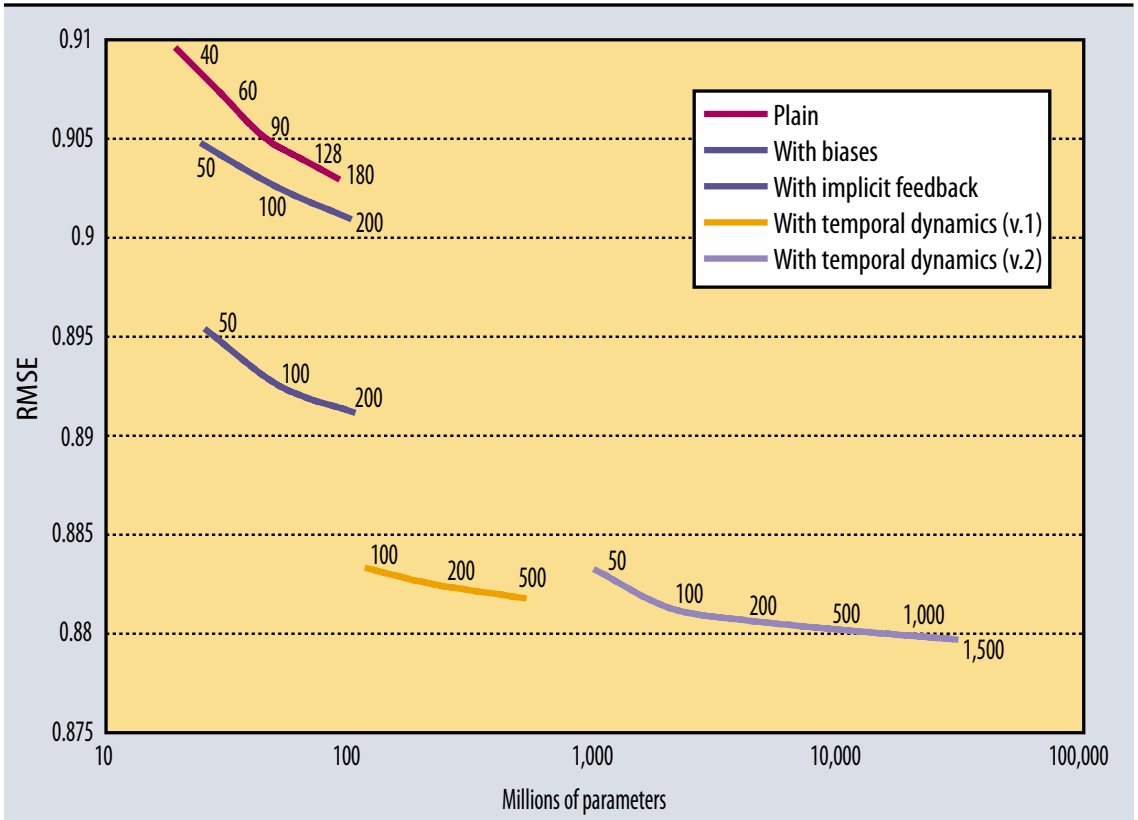


Figure 4. Matrix factorization models' accuracy. The plots show the root-mean-square error of each of four individual factor models (lower is better). Accuracy improves when the factor model's dimensionality (denoted by numbers on the charts) increases. In addition, the more refined factor models, whose descriptions involve more distinct sets of parameters, are more accurate. For comparison, the Netflix system achieves RMSE = 0.9514 on the same dataset, while the grand prize's required accuracy is RMSE = 0.8563.

[Koren et al., 2009]

Matrix Factorization / Matrix Notation

$$f(\Phi, \Psi; Y, W) := ||W \odot (Y - \Phi\Psi^T)||^2 + \lambda(||\Phi||^2 + ||\Psi||^2)$$

with

$Y \in \mathbb{R}^{U \times I}$	(partially) observed matrix
$W \in \{0,1\}^{U \times I}$	indicator for observed cells
$\Phi \in \mathbb{R}^{U \times K}$	latent user/row features
$\Psi \in \mathbb{R}^{I \times K}$	latent item/column features
$\odot$	cell-wise / Hadamard product

Matrix Factorization / General Loss and Regularizer

$$\hat{y}(u,i) := a + b_u + c_i + \langle \phi_u, \psi_i \rangle$$

$$f(a,b,c,\phi,\psi;D) := \sum_{(u,i,y)\in D} \ell(y, \hat{y}(u,i)) + \Omega(b,c,\phi,\psi)$$

with

$U, I \in \mathbb{N}$	number of users/rows, items/columns
$K \in \mathbb{N}$	number of latent features
$D := \{(u_1, i_1, y_1), \ldots, (u_N, i_N, y_N)\}$	sparse matrix data (in triples)

Matrix Factorization / General Loss and Regularizer

$$\hat{y}(u,i) := a + b_u + c_i + \langle \phi_u, \psi_i \rangle$$

$$f(a,b,c,\phi,\psi;D) := \sum_{(u,i,y)\in D} \ell(y, \hat{y}(u,i)) + \Omega(b,c,\phi,\psi)$$

with (continued)

$a \in \mathbb{R}$	global offset
$b \in \mathbb{R}^U$	user/row offset/bias
$c \in \mathbb{R}^I$	item/column offset/bias
$\phi \in \mathbb{R}^{U\times K}$	latent user/row features
$\psi \in \mathbb{R}^{I\times K}$	latent item/column features
$\ell\colon \mathbb{R}\times\mathbb{R}\to\mathbb{R}$	loss
$\Omega\colon \mathbb{R}^{U+I+UK+IK}\to\mathbb{R}$	regularization

Matrix Factorization / General Loss and Regularizer

$$\hat{y}(u,i) := a + \langle \phi_u, \psi_i \rangle$$

$$f(a, \phi, \psi; D) := \sum_{(u,i,y) \in D} \ell(y, \hat{y}(u,i)) + \Omega(\phi, \psi)$$

with one latent feature dimension fixed

$$\phi_u^1 := 1$$
$$\psi_i^2 := 1$$

thus

$$b_u := \phi_u^2$$
$$c_i := \psi_i^1$$

Unified View on MF [Singh and Gordon, 2008a]

$$\hat{y}(u,i) := g(\Phi_{u,.}^T \Psi_{i,.})$$
$$\hat{Y} := (\hat{y}(u,i))_{u \in U, i \in I}$$
$$f(\Phi, \Psi; Y) := \mathcal{D}(Y || \hat{Y}, W) + \Omega(\Phi, \Psi), \quad \Phi, \Psi \in \mathcal{C}$$

with

$Y \in \mathbb{R}^{U \times I}$	(partially) observed matrix
$W \in \{0,1\}^{U \times I}$	indicator for observed cells
$\Phi \in \mathbb{R}^{U \times K}$	latent user/row features
$\Psi \in \mathbb{R}^{I \times K}$	latent item/column features
$g : \mathbb{R} \to \mathbb{R}$	prediction link
$\mathcal{C}$	constraints on latent features
$\mathcal{D}$	Bregman divergence
Ω	regularization

Unified View on MF [Singh and Gordon, 2008a]

Method	dom X_{ij}	Link $f(\theta)$	Loss $\mathcal{D}(X\|\hat{X} = f(\Theta), W)$	W_{ij}
SVD [1]	$\mathbb{R}$	θ	$\|\|W \odot (X - \hat{X})\|\|^2_{Fro}$	1
W-SVD [24,25]	$\mathbb{R}$	θ	$\|\|W \odot (X - \hat{X})\|\|^2_{Fro}$	≥ 0
k-means [26]	$\mathbb{R}$	θ	$\|\|W \odot (X - \hat{X})\|\|^2_{Fro}$	1
k-medians	$\mathbb{R}$	θ	$\sum_{ij} \|W_{ij}\left(X_{ij} - \hat{X}_{ij}\right)\|$	1
ℓ_1-SVD [27]	$\mathbb{R}$	θ	$\sum_{ij} \|W_{ij}\left(X_{ij} - \hat{X}_{ij}\right)\|$	≥ 0
pLSI [2]	$\mathbf{1} \circ X = 1$	θ	$\sum_{ij} W_{ij}\left(X_{ij} \log \frac{X_{ij}}{\hat{X}_{ij}}\right)$	1
NMF [5]	$\mathbb{R}_+$	θ	$\sum_{ij} W_{ij}\left(X_{ij} \log \frac{X_{ij}}{\Theta_{ij}} + \Theta_{ij} - X_{ij}\right)$	1
ℓ_2-NMF [28,5]	$\mathbb{R}_+$	θ	$\|\|W \odot (X - \hat{X})\|\|^2_{Fro}$	1
Logistic PCA [29]	$\{0, 1\}$	$(1 + e^{-\theta})^{-1}$	$\sum_{ij} W_{ij}\left(X_{ij} \log \frac{X_{ij}}{\hat{X}_{ij}} + (1 - X_{ij}) \log \frac{1-X_{ij}}{1-\hat{X}_{ij}}\right)$	1
E-PCA [6]	many	many	decomposable Bregman (D_F)	1
G^2L^2M [8]	many	many	decomposable Bregman $(\mathbb{D}_F)$	1
MMMF [30]	$\{0, \ldots, R\}$	min-loss	$\sum_{r=1}^{R-1} \sum_{ij:X_{ij} \neq 0} W_{ij} \cdot h(\Theta_{ij} - B_{ir})$	1
Fast-MMMF [31]	$\{0, \ldots, R\}$	min-loss	$\sum_{r=1}^{R-1} \sum_{ij:X_{ij} \neq 0} W_{ij} \cdot h_\gamma(\Theta_{ij} - B_{ir})$	1

Unified View on MF [Singh and Gordon, 2008a]

Method	Constraints U	Constraints V	Regularizer $\mathcal{R}(U,V)$	Algorithm(s)
SVD	$U^TU = I$	$V^TV = \Lambda$	—	Gaussian Elimination, Power Method
W-SVD	—	—	—	Gradient, EM
k-means	—	$V^TV = I$, $V_{ij} \in \{0,1\}$	—	EM
k-medians	—	$V^TV = I$, $V_{ij} \in \{0,1\}$	—	Alternating
ℓ_1-SVD	—	—	—	Alternating
pLSI	$\mathbf{1}^TU\mathbf{1} = 1$ $U_{ij} \geq 0$	$\mathbf{1}^TV = \mathbf{1}$ $V_{ij} \geq 0$	—	EM
NMF	$U_{ij} \geq 0$	$V_{ij} \geq 0$	—	Multiplicative
ℓ_2-NMF	$U_{ij} \geq 0$	$V_{ij} \geq 0$	—	Multiplicative, Alternating
Logistic PCA	—	—	—	EM
E-PCA	—	—	—	Alternating
G^2L^2M	—	—	$\|\|U\|\|^2_{Fro} + \|\|V\|\|^2_{Fro}$	Alternating (Subgradient, Newton)
MMMF	—	—	$tr(UV^T)$	Semidefinite Program
Fast-MMMF	—	—	$\frac{1}{2}(\|\|U\|\|^2_{Fro} + \|\|V\|\|^2_{Fro})$	Conjugate

A Short History of Factorization Models

1901 Pearson [1901] invents PCA.

1976 Wiberg [1976] generalizes PCA to PCA with missing values.

1999 Roweis [1998] and Tipping and Bishop [1999] provide a probabilistic interpretation for PCA.

2003–2005 Srebro and Jaakkola [2003] and Srebro et al. [2005] introduce L2 regularization into matrix factorization models.

2006 Funk [2006] and Bell et al. [2007] popularize matrix factorization as leading method in the Netflix price.

2008 Singh and Gordon [2008a,b] systematize matrix factorization models

- with different losses, feature constraints, etc. as well as
- for several matrices.

Outline

Gradient Descent (GD)

$$\theta^* := \underset{\theta}{\operatorname{argmin}}\, f(\theta)$$

$GD(f, \eta, \sigma^2, \epsilon)$:
initialize $\theta \sim \mathcal{N}(0, \sigma^2)$
repeat
    $\theta^{old} := \theta$
    $\theta := \theta - \eta \frac{\partial f}{\partial \theta}(\theta)$
until $||\theta - \theta^{old}|| < \epsilon$
return θ

Stochastic Gradient Descent (SGD)

$$\theta^* := \underset{\theta}{\operatorname{argmin}}\, f(\theta)$$

$$f(\theta) := \sum_{c \in C} f_c(\theta)$$

```
SGD((f_c)_{c∈C}, η, σ², ε) :
initialize θ ~ N(0, σ²)
repeat
    θ^old := θ
    foreach c ∈ C in random order do
        θ := θ − η ∂f_c/∂θ (θ)
    end
until ||θ − θ^old|| < ε
return θ
```

SGD to Learn a Model

$$\theta^* := \operatorname*{argmin}_{\theta} f(\theta)$$

$$f(\theta) := \sum_{(x,y)\in D} \ell(y, \hat{y}(x;\theta)) + \lambda\Omega(\theta)$$

learn-SGD$(\ell, \hat{y}, \Omega, \eta, \sigma^2, \epsilon)$:
initialize $\theta \sim \mathcal{N}(0, \sigma^2)$
repeat
    $\theta^{old} := \theta$
    foreach $(x, y) \in D$ *in random order* **do**
        $\theta := \theta - \eta \left(\frac{\partial \ell}{\partial \hat{y}}(y, \hat{y}(x;\theta)) \frac{\partial \hat{y}}{\partial \theta}(x;\theta) + \frac{1}{|D|} \lambda \frac{\partial \Omega}{\partial \theta}(\theta) \right)$
    end
until $||\theta - \theta^{old}|| < \epsilon$
return θ

SGD to Learn a Matrix Factorization

E.g., for

$$\ell(y, \hat{y}) := (y - \hat{y})^2 \qquad \text{L}_2 \text{ / square loss}$$

$$\Omega(\theta) := ||\theta||^2 \qquad \text{L}_2 \text{ regularization}$$

$$\hat{y}(x = (u, i); \theta) := a + b_u + c_i + \langle \phi_u, \psi_i \rangle \qquad \text{matrix factorization model}$$

one easily gets:

$$\frac{\partial \ell}{\partial \hat{y}}(y, \hat{y}) = -2(y - \hat{y})$$

$$\frac{\partial \Omega}{\partial \theta}(\theta) = 2\theta$$

SGD to Learn a Matrix Factorization

E.g., for

$$\ell(y, \hat{y}) := (y - \hat{y})^2 \qquad \text{L}_2 \text{ / square loss}$$

$$\Omega(\theta) := ||\theta||^2 \qquad \text{L}_2 \text{ regularization}$$

$$\hat{y}(x = (u, i); \theta) := a + b_u + c_i + \sum_{k=1}^{K} \phi_{u,k} \psi_{i,k} \qquad \text{matrix factorization model}$$

one easily gets:

$$\frac{\partial \ell}{\partial \hat{y}}(y, \hat{y}) = -2(y - \hat{y})$$

$$\frac{\partial \Omega}{\partial \theta}(\theta) = 2\theta$$

$$\frac{\partial \hat{y}}{\partial \theta}(x = (u, i); \theta) = \begin{cases} \psi_{i,k}, & \text{for } \theta = \phi_{u,k} \\ \phi_{u,k}, & \text{for } \theta = \psi_{i,k} \\ 1, & \text{for } \theta \in \{a, b_u, c_i\} \end{cases}$$

SGD / Sparse Parameter Updates

When a component function f_c depends only on a subset of the parameters θ, only those parameters have to be updated.

For factorization models:

- $\ell(y, \hat{y}(x = (u, i); \theta))$ does only depend on ϕ_u, ψ_i.
  (not on $\phi_{u'}$, $\psi_{i'}$ for $u' \neq u, i' \neq i$)
  - update only ϕ_u, ψ_i.

SGD / Sparse Parameter Updates

When a component function f_c depends only on a subset of the parameters θ, only those parameters have to be updated.

For factorization models:

- $\ell(y, \hat{y}(x = (u, i); \theta))$ does only depend on ϕ_u, ψ_i.
  (not on $\phi_{u'}$, $\psi_{i'}$ for $u' \neq u, i' \neq i$)
  - update only ϕ_u, ψ_i.

- but beware: the regularization term depends on all parameters !
  - update whenever we have to touch the parameter anyway, e.g.,

$$\phi_{u,k} := \phi_{u,k} - \eta \left(-2(y - \hat{y}(x; \theta))\psi_{i,k} + \frac{1}{|D|} \frac{|D|}{\mathrm{freq}(u; D)} \lambda 2 \phi_{u,k} \right)$$

$$\mathrm{freq}(u; D) := |\{(u', i', y') \in D \mid u' = u\}|$$

  - this normalization often is missing, leading to stronger regularization for parameters involved in more components.

Block Coordinate Descent

$$\theta^* := \operatorname*{argmin}_{\theta} f(\theta)$$

$$\theta \in \Theta := \prod_{c \in C} \Theta_c$$

Denote the function with parameters outside block c fixed to θ by

$$f|_{\overline{\Theta_c},\theta}(\tilde{\theta}_c) := f(\theta|_{\prod_{c'<c}\Theta_{c'}}, \tilde{\theta}_c, \theta|_{\prod_{c'>c}\Theta_{c'}}), \quad \tilde{\theta}_c \in \Theta_c$$

$BCD(f, (\Theta_c)_{c \in C}, \sigma^2, \epsilon)$:
initialize $\theta \sim \mathcal{N}(0, \sigma^2)$
repeat
    $\theta^{old} := \theta$
    foreach $c \in C$ *in random order* **do**
        $\theta|_{\Theta_c} := \operatorname{argmin}_{\tilde{\theta}_c} f|_{\overline{\Theta_c},\theta}(\tilde{\theta}_c)$
    end
until $||\theta - \theta^{old}|| < \epsilon$
return θ

Example for Fixing Function Arguments

$$f(\theta_1, \theta_2, \theta_3) := 10 + \theta_1 + \theta_2 \cdot \theta_3$$
$$\Theta := \mathbb{R}^3$$

Example for Fixing Function Arguments

$$
\begin{aligned}
f(\theta_1, \theta_2) &:= f(\theta_{1,1}, \theta_{1,2}, \theta_2) := 10 + \theta_{1,1} + \theta_{1,2} \cdot \theta_2 \\
\Theta &:= \mathbb{R}^3 \\
\Theta_1 &:= \mathbb{R}^2, \quad \Theta_2 := \mathbb{R} \\
\theta &:= ((3,7), -5)
\end{aligned}
$$

$$
\begin{aligned}
f|_{\overline{\Theta_1}, \theta}(\tilde{\theta}_1) &:= f(\theta|_{\prod_{c'<1} \Theta_{c'}}, \tilde{\theta}_1, \theta|_{\prod_{c'>1} \Theta_{c'}}), \qquad \tilde{\theta}_1 \in \Theta_1 = \mathbb{R}^2 \\
&= f(\tilde{\theta}_1, \theta_2) \\
&= f(\tilde{\theta}_{1,1}, \tilde{\theta}_{1,2}, -5) = 10 + \tilde{\theta}_{1,1} + \tilde{\theta}_{1,2} \cdot 5
\end{aligned}
$$

Example for Fixing Function Arguments

$$
\begin{aligned}
f(\theta_1, \theta_2) &:= f(\theta_{1,1}, \theta_{1,2}, \theta_2) := 10 + \theta_{1,1} + \theta_{1,2} \cdot \theta_2 \\
\Theta &:= \mathbb{R}^3 \\
\Theta_1 &:= \mathbb{R}^2, \quad \Theta_2 := \mathbb{R} \\
\theta &:= ((3,7), -5)
\end{aligned}
$$

$$
\begin{aligned}
f|_{\overline{\Theta_1}, \theta}(\tilde{\theta}_1) &:= f(\theta|_{\prod_{c'<1} \Theta_{c'}}, \tilde{\theta}_1, \theta|_{\prod_{c'>1} \Theta_{c'}}), \qquad \tilde{\theta}_1 \in \Theta_1 = \mathbb{R}^2 \\
&= f(\tilde{\theta}_1, \theta_2) \\
&= f(\tilde{\theta}_{1,1}, \tilde{\theta}_{1,2}, -5) = 10 + \tilde{\theta}_{1,1} + \tilde{\theta}_{1,2} \cdot 5
\end{aligned}
$$

$$
\begin{aligned}
f|_{\overline{\Theta_2}, \theta}(\tilde{\theta}_2) &:= f(\theta|_{\prod_{c'<1} \Theta_{c'}}, \tilde{\theta}_2, \theta|_{\prod_{c'>1} \Theta_{c'}}), \qquad \tilde{\theta}_2 \in \Theta_2 = \mathbb{R} \\
&= f(\theta_1, \tilde{\theta}_2) \\
&= f(3, 7, \tilde{\theta}_2) = 10 + 3 + 7 \cdot \tilde{\theta}_2 = 13 + 7 \cdot \tilde{\theta}_2
\end{aligned}
$$

Block Coordinate Descent

Useful if the component problems

$$\operatorname*{argmin}_{\tilde{\theta}_c} f|_{\overline{\Theta_c},\theta}(\tilde{\theta}_c)$$

are simpler to solve.

For matrix factorization:

$$\Theta_U := ((b_u)_{u \in U}, (\phi_{u,.})_{u \in U})$$
$$\Theta_I := ((c_i)_{i \in I}, (\psi_{i,.})_{i \in I})$$

Then the component problems are just regression problems:

$$\theta_U|\theta_I := \operatorname*{argmin}_{b_u,\phi_u} \sum_{(u,i,y)\in D} \ell(y, a + b_u + c_i + \langle \phi_u, \psi_i \rangle) + \lambda\Omega(\theta_U)$$

$$\theta_I|\theta_U := \operatorname*{argmin}_{c_i,\psi_i} \sum_{(u,i,y)\in D} \ell(y, a + b_u + c_i + \langle \phi_u, \psi_i \rangle) + \lambda\Omega(\theta_I)$$

Block Coordinate Descent

Especially for L_2 loss and L_2 regularization: ridge regression.

$$\theta^* := \underset{\theta}{\operatorname{argmin}} \sum_{(x,y)\in D} (y - \langle x, \theta \rangle)^2 + \lambda ||\theta||^2$$

Could be solved by solving a linear system of equations:

$$\begin{aligned}(X^T X + \lambda I)\,\theta^* =& X^T Y,\\ \text{with } X :=& (x_1^T, \ldots, x_N^T)^T,\ Y := (y_1, \ldots, y_N)^T,\\ D :=& \{(x_1, y_1), \ldots, (x_N, y_N)\}\end{aligned}$$

This specific block coordinate descent is called **Alternating Least Squares** in the literature [Bell and Koren, 2007].

Coordinate Descent

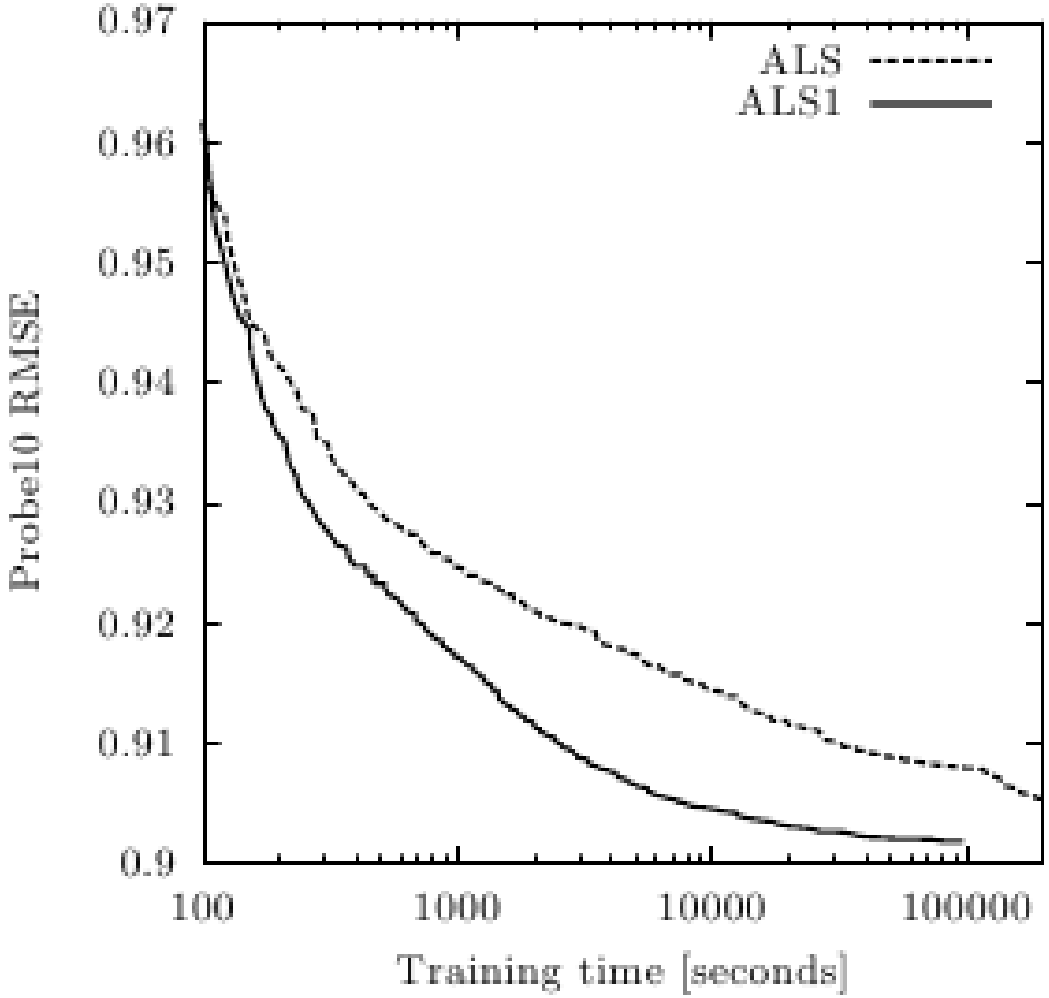
Often for coordinate descent blocks of dimension 1 (scalars) are best, esp. if the 1-dimensional component problem can be solved analytically.

For 1-dimensional ridge-regression:

$$\theta^* = (X^T X + \lambda I)^{-1}(X^T Y) = \frac{\sum_{i=1}^N x_i y_i}{\sum_{i=1}^N x_i^2 + \lambda}$$

Coordinate descent for matrix factorization models with L2 loss and L2 regularization is called ALS1 [Pilászy et al., 2010].

Coordinate Descent (ALS1) / Convergence Speed



[Pilászy et al., 2010]

Damped Newton Algorithm

$Newton(f, \eta, \sigma^2, \epsilon)$:
initialize $\theta \sim \mathcal{N}(0, \sigma^2)$
repeat
    $\theta^{old} := \theta$
    $d := solve_d \left(\frac{\partial^2 f}{\partial \theta^2}(\theta) + \eta I \right) d = -\frac{\partial f}{\partial \theta}(\theta)$
    $\theta := \theta + d$
until $||\theta - \theta^{old}|| < \epsilon$
return θ

Newton for Factorization Models

$$f(\Phi, \Psi; Y) := ||W \odot (Y - \Phi\Psi^T)||^2 + \lambda(||\Phi||^2 + ||\Psi||^2)$$

$$\begin{aligned}
\frac{\partial^2 f}{\partial \phi_u \partial \phi_v} &= 2\delta(u = v)(\Psi^T \mathrm{diag}(W_{u,.})^2 \Psi + 2\lambda I) \\
\frac{\partial^2 f}{\partial \phi_u \partial \psi_i} &= 2w_{u,i}(\phi_u \psi_i^T + I(\phi_u^T \psi_i - Y_{u,i})) \\
\frac{\partial^2 f}{\partial \psi_i \partial \psi_j} &= 2\delta(i = j)(\Phi^T \mathrm{diag}(W_{.,i})^2 \Phi + 2\lambda I)
\end{aligned}$$

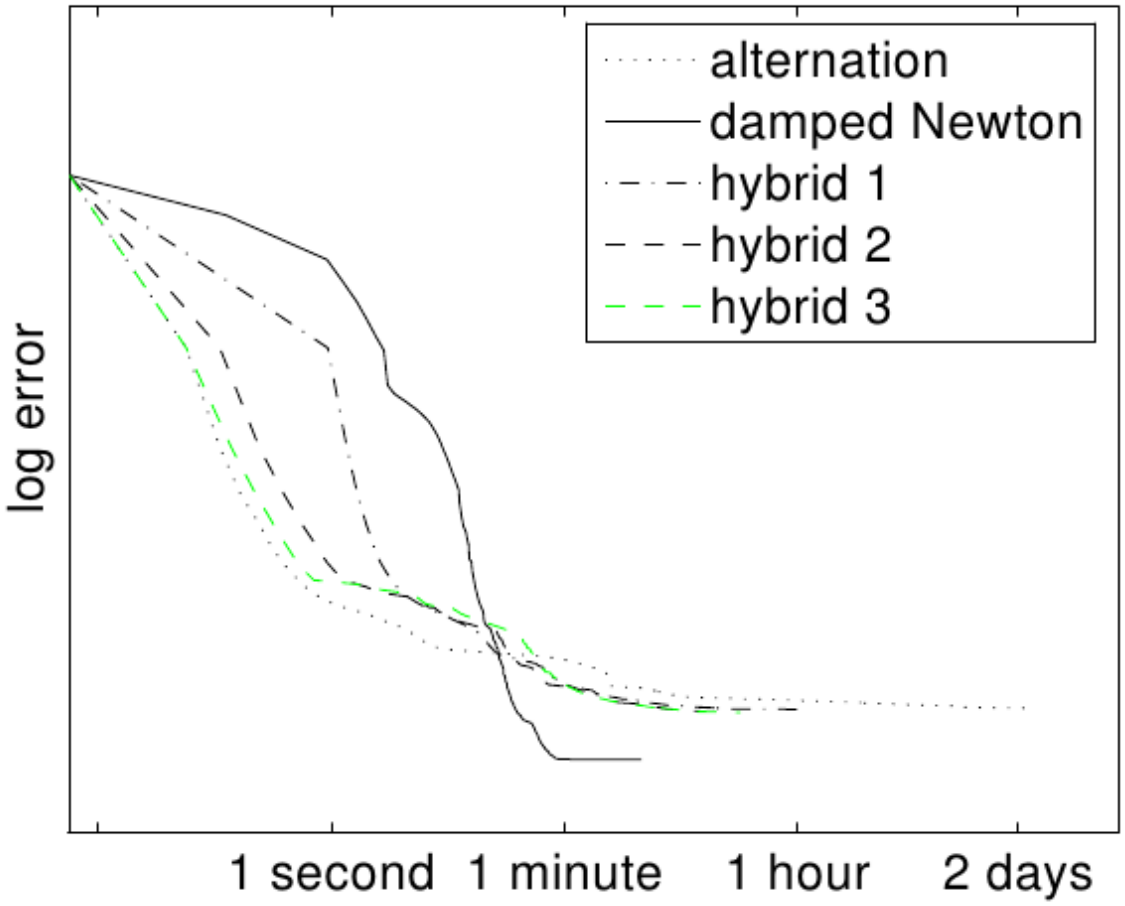
[Buchanan and Fitzgibbon, 2005]

Notation:

$$\delta(A) := \begin{cases} 1, & \text{if } A \text{ is true} \\ 0, & \text{else} \end{cases}$$

Newton for Factorization Models

e.g., for a small 72×319 matrix with 28% given values:



[Buchanan and Fitzgibbon, 2005]

Newton for Factorization Models / Variants

FNMA Fast Nonnegative Matrix Approximation [Kim et al., 2007]:

- alternating least squares
- solving each component problem via a Quasi-Newton method (BFGS).

Stochastic Newton [Singh and Gordon, 2008b]:

- alternating least squares
- solving each component problem via a stochastic Quasi-Newton method.

LM_+ [Phan et al., 2011]:

- extension to tensors.

Outline

Two Different Recommendation Tasks

predict rating values / votes:

item	Toy Story	GoldenEye	Four Rooms	Get Shorty	Copycat	Shanghai Triad	Twelve Monkeys	Babe	Dead Man	...
user Anna	4		3							
Berta	1									
Clara		4								
Daniel				5						
Egon								2		
Fred								4		
⋮										

- model votes of non-rated items as missing values?

predict (rated) items:

item	Toy Story	GoldenEye	Four Rooms	Get Shorty	Copycat	Shanghai Triad	Twelve Monkeys	Babe	Dead Man	...
user Anna	1		1							
Berta	1									
Clara		1								
Daniel				1						
Egon								1		
Fred								1		
⋮										

- there are only positive, no (explicit) negative examples.

Weighted Regularized Matrix Factorization

idea: assign a small weight to the negative examples.

$$\hat{y}(u,i) := \langle \phi_u, \psi_i \rangle, \quad \phi_u, \psi_i \in \mathbb{R}^K$$

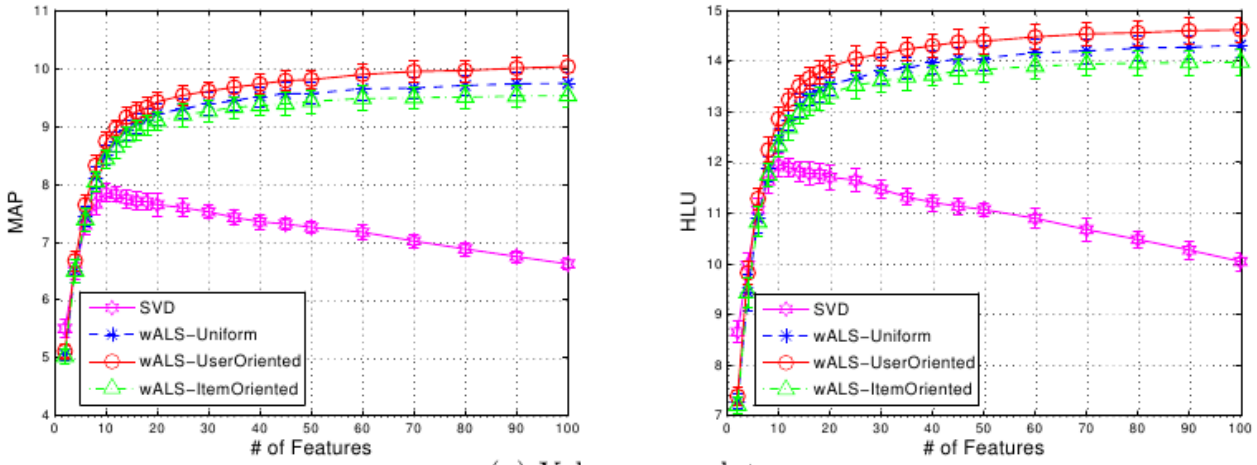
$$f(\phi,\psi) := \sum_{(u,i)\in U\times I} w_{u,i}(\delta((u,i)\in D) - \hat{y}(u,i))^2 + \lambda(||\phi|| + ||\psi||)$$

$w_{u,i}$ (for $(u,i)\in D$)	$w_{u,i}$ (for $(u,i)\notin D$)	reference
$1+\alpha r_{u,i}$	1	[Hu et al., 2008]
1	α	[Pan et al., 2008]
1	αn_u	[Pan et al., 2008]
1	$\alpha(U-n_i)$	[Pan et al., 2008]

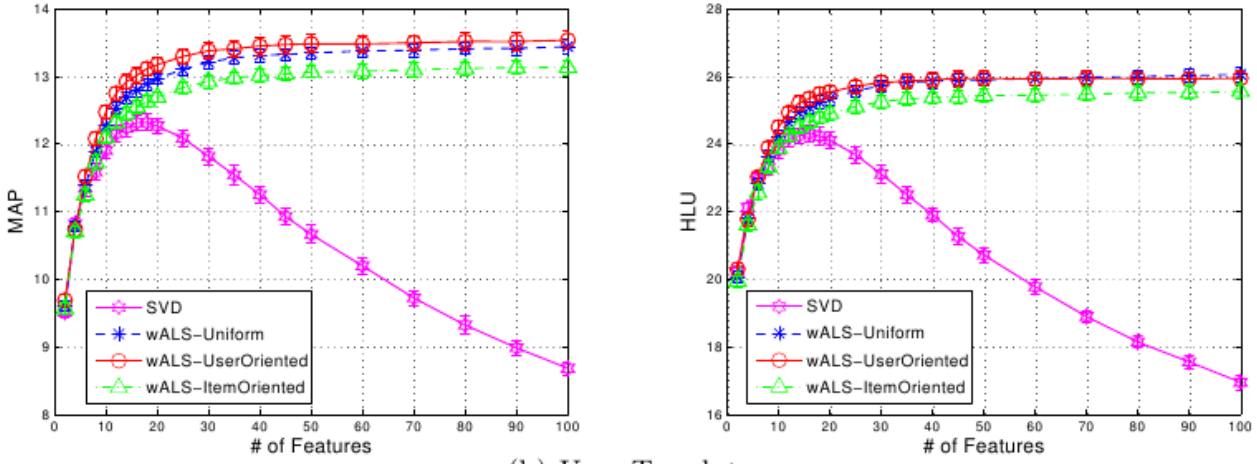
where

- $D \subseteq U \times I$ is the training data,
- $r_{u,i}$ (additionally observed) ratings,
- $n_u := |\{(u',i') \in D \mid u = u'\}|$ number of events for a user,
- α a hyperparameter (e.g., $\alpha = 40$ in [Hu et al., 2008]).

WR-MF / Recommendation Quality



(a) Yahoo news data



(b) User-Tag data

[Pan et al., 2008]

BPR / Data Interpretation

Absolute preference:

	i_1	i_2	i_3	i_4
u_1	?	+	+	?
u_2	+	?	?	+
u_3	+	+	?	?
u_4	?	?	+	+
u_5	?	?	+	?

user ↕ item ↔

→

	i_1	i_2	i_3	i_4
u_1	0	1	1	0
u_2	1	0	0	1
u_3	1	1	0	0
u_4	0	0	1	1
u_5	0	0	1	0

user ↕ item ↔

$$S \subseteq U \times I$$

BPR / Data Interpretation

Absolute preference:

	i_1	i_2	i_3	i_4
u_1	?	+	+	?
u_2	+	?	?	+
u_3	+	+	?	?
u_4	?	?	+	+
u_5	?	?	+	?

user ↕ item ↔

	i_1	i_2	i_3	i_4
u_1	0	1	1	0
u_2	1	0	0	1
u_3	1	1	0	0
u_4	0	0	1	1
u_5	0	0	1	0

user ↕ item ↔

$$S \subseteq U \times I$$

Relative preference:

	i_1	i_2	i_3	i_4
u_1	?	+	+	?
u_2	+	?	?	+
u_3	+	+	?	?
u_4	?	?	+	+
u_5	?	?	+	?

user ↕ item ↔

u_1: $i >_{u1} j$

	i_1	i_2	i_3	i_4
j_1		+	+	?
j_2	-		?	-
j_3	-	?		-
j_4	?	+	+	

item ↕ item ↔

...

u_5: $i >_{u5} j$

	i_1	i_2	i_3	i_4
j_1		?	+	?
j_2	?		+	?
j_3	-	-		-
j_4	?	?	+	

item ↕ item ↔

$$D_S := \{(u,i,j) \in U \times I \times I \mid (u,i) \in S, (u,j) \notin S\}$$

Bayesian Personalized Ranking (BPR)

Relative preferences, modeled by matrix factorization:

$$\begin{aligned}\hat{p}(i >_u j) :=& \sigma(\hat{y}_{u,i,j}) \\ =& \sigma(\hat{y}_{u,i} - \hat{y}_{u,j}) \\ =& \sigma(\langle \phi_u, \psi_i \rangle - \langle \phi_u, \psi_j \rangle)\end{aligned}$$

$$\sigma(x) := \frac{1}{1+e^{-x}}$$

$$\hat{y}_{u.i,j} := \hat{y}_{u,i} - \hat{y}_{u,j}$$

$$\hat{y}_{u,i} := \langle \phi_u, \psi_i \rangle$$

Regularized Loglikelihood:

$$\text{BPR-Opt}(\Theta; D_s) := -\sum_{(u,i,j) \in D_s} \ln \hat{p}(i >_u j) + \lambda_\phi ||\phi||^2 + \lambda_\psi ||\psi||^2$$

with parameters $\Theta := (\phi, \psi)$.

BPR / Learning BPR via SGD

LearnBPR(D_S, η, λ):
initialize $\Theta := (\phi, \psi)$
repeat
    draw (u, i, j) *from* D_S
    $\Theta \leftarrow \Theta + \eta \left(\frac{e^{-\hat{y}_{u,i,j}}}{1+e^{-\hat{y}_{u,i,j}}} \cdot \frac{\partial}{\partial \Theta} \hat{y}_{u,i,j} - \lambda_\Theta \cdot \Theta \right)$
until *convergence*
Return Θ

$$\frac{\partial}{\partial \phi_u} \hat{y}_{u,i,j} = \psi_i - \psi_j$$
$$\frac{\partial}{\partial \psi_i} \hat{y}_{u,i,j} = \phi_u$$
$$\frac{\partial}{\partial \psi_j} \hat{y}_{u,i,j} = -\phi_u$$

BPR / Evaluation

Datasets

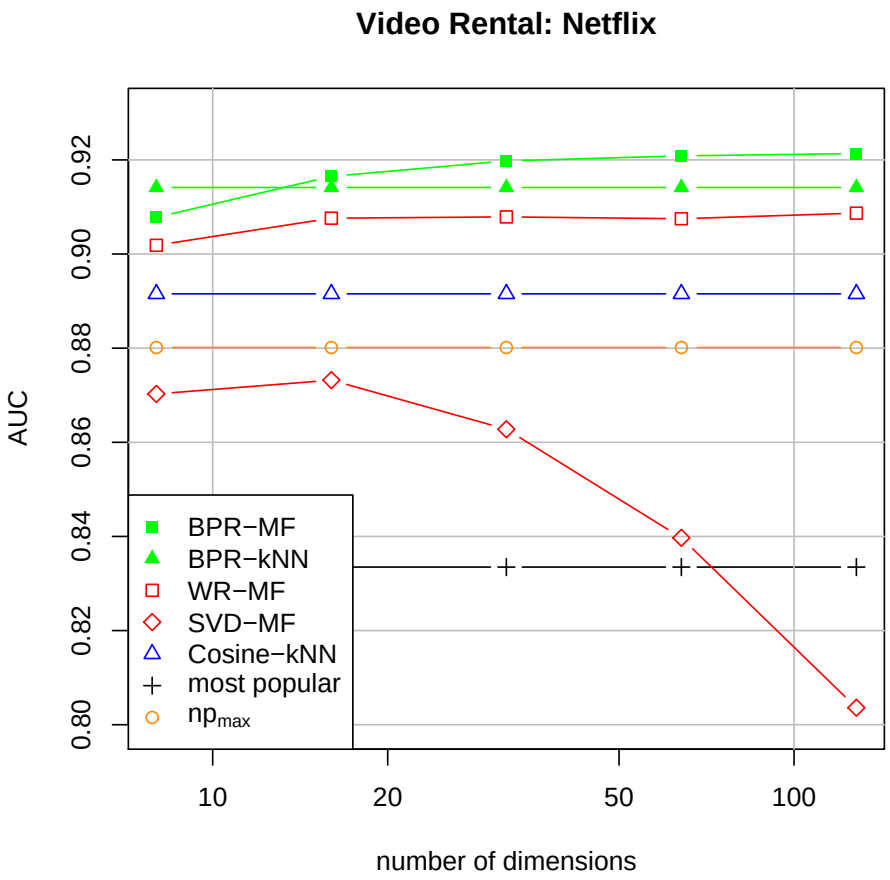
dataset	users $\lvert U \rvert$	items $\lvert I \rvert$	transactions $\lvert S \rvert$
Rossmann	10,000	4000	426,612
Netflix sample	10,000	5000	565,738

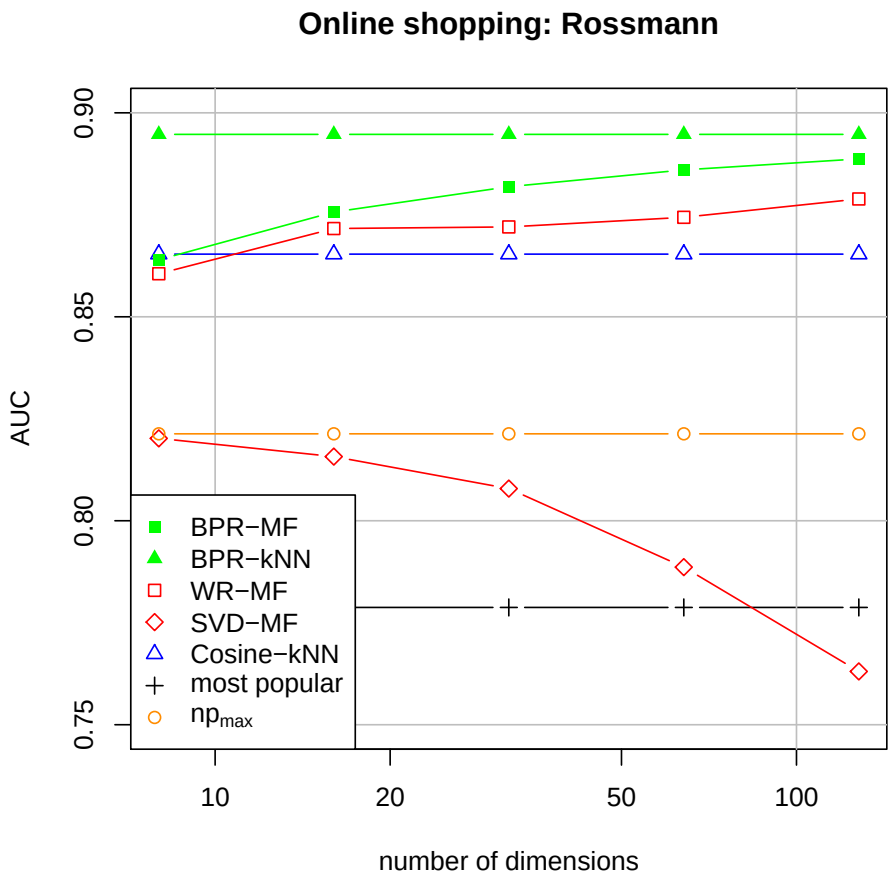
Evaluated methods

- SVD-MF
- WR-MF [Pan et al., 2008; Hu et al., 2008]
- nearest neighbor with cosine similarity
- constant (most popular)
- upper bound on AUC for non-personalized models

Netflix sample: target: rating event; every user and item occurs in at least 10 transactions.

BPR / Recommendation Quality





[Rendle et al., 2009]

Also see Ning and Karypis [2011] for an evtl. improved model (Sparse Linear Method, SLIM).

Ordinal MF

Targets often are ordinal.

- i.e., ratings not continuous in $[1, 5]$, but discrete in $R := \{1, 2, 3, 4, 5\}$.

goals:

- improved rating prediction
  - use standard regression evaluation measures:

$$\mathrm{RMSE}(D, \hat{r}) := \left(\frac{1}{|D|} \sum_{(u,i,y) \in D} (y - \hat{r}(u, i))^2 \right)^{\frac{1}{2}}$$

$$\mathrm{MAE}(D, \hat{r}) := \frac{1}{|D|} \sum_{(u,i,y) \in D} |y - \hat{r}(u, i)|$$

- better fit to application
  - optimize ranking evaluation measures:

$$\mathrm{FCP}(D, \hat{y}) := \frac{\sum_{(u,i,y),(u,i',y') \in D} \delta(y > y') \delta(\hat{y}(u, i) > \hat{y}(u, i'))}{\sum_{(u,i,y),(u,i',y') \in D} \delta(y > y')}$$

Ordinal MF / Maximum Margin MF (MMMF)

idea: push score $\hat{y}$ into ordinal level thresholds ρ with margin 1:

$$-\infty =: \rho_0 < \rho_1 < \ldots < \rho_{R-1} < \rho_R := \infty$$

$$\begin{aligned}
\rho_{y-1} + 1 &\leq \hat{y} \leq \rho_y - 1 \\
\rho_r + 1 &\leq \hat{y}, && \text{for } r < y && \rightsquigarrow [1 - (\hat{y} - \rho_r)]_+ \text{ min.} \\
\hat{y} &\leq \rho_y - 1 && \text{for } r \geq y && \rightsquigarrow [1 - (\rho_r - \hat{y})]_+ \text{ min.}
\end{aligned}$$

summation yields the ordinal MMMF loss:

$$\ell(y, \hat{y}; \rho) := \sum_{r=1}^{R-1} \left[1 - (-1)^{y>r}(\rho_r - \hat{y})\right]_+$$

$$\hat{r}(\hat{y}; \rho) := \min\{r \in R \mid \hat{y} < \rho_r\}$$

[Rennie and Srebro, 2005]

Ordinal MF / OrdRec

similar idea as MMMF, but just using the most strict thresholds:

$$\begin{aligned} \rho_{y-1} &\leq \hat{y} & &\rightsquigarrow \rho_{y-1} - \hat{y} \quad \text{min.} \\ \hat{y} &\leq \rho_y & &\rightsquigarrow \hat{y} - \rho_y \quad \text{min.} \end{aligned}$$

Wrapping terms with a sigmoid and summing yields the OrdRec loss:

$$\ell(y, \hat{y}; \rho) := \sigma(\rho_{y-1} - \hat{y}) + \sigma(\hat{y} - \rho_y)$$

To ensure strict monotonicity of ρ, the parametrization $(\rho_0, \alpha_1, \ldots, \alpha_{R-2})$ is used with $\rho_{r+1} := \rho_r + e^{\alpha_r}$.

[Koren and Sill, 2011]

The same ordinal model is used by Paquet et al. [2012] (but with the probit instead of the logistic function and embedded in a hierarchical Bayesian model).

Ranking

- some ranking measures decompose into pairwise comparisons
  - e.g., Fraction of Concordant Pairs (FCP):

$$\mathrm{FCP}(D, \hat{y}) := \frac{\sum_{(u,i,y),(u,i',y') \in D} \delta(y > y') \delta(\hat{y}(u,i) > \hat{y}(u,i'))}{\sum_{(u,i,y),(u,i',y') \in D} \delta(y > y')}$$

- others are not decomposable
  - e.g., Normalized Discounted Cumulative Gain at k (NDCG@k):

$$\mathrm{DCG}_k(y, \pi) := \sum_{j=1}^{k} \frac{2^{y_{\pi(j)}} - 1}{\log_2(j+1)}$$

$$\mathrm{NDCG}_k(y, \pi) := \frac{\mathrm{DCG}_k(y, \pi)}{\mathrm{DCG}_k(y, \operatorname{argsort} y)}$$

$$\mathrm{NDCG}_k(D, \hat{y}) := \frac{1}{U} \sum_{u=1}^{U} \mathrm{NDCG}_k(D_u, \operatorname{argsort} \hat{y}(u, .))$$

Ranking / CoFiRank-ordinal

idea: use a pairwise loss with margin 1.

$$\hat{y}_j + 1 \leq \hat{y}_j, \qquad \text{for } y_j < y_i \qquad \rightsquigarrow [1 - (\hat{y}_j - \hat{y}_i)]_+ \text{ min.}$$

averaging yields the CoFiRank ordinal loss:

$$\ell(y, \hat{y}) := \frac{2}{|I_u|^2 - \sum_{r \in 1}^{R} |I_u^r|^2} \sum_{\substack{i,j \in I_u: \\ y_i > y_j}} c(y_i, y_j)[1 - (\hat{y}_i - \hat{y}_j)]_+$$

where

- $y, \hat{y} \in R^I$ are vectors of ratings for all items by a user and
- $c(y, y')$ is the cost for confusing the ranks of items at levels y, y'.
- CofiRank is a ranking model and does not provide rating predictions $\hat{r}$!

Notation:
$I_u := \{i \in I \mid (u, i, y) \in D\}$ are all items actually rated by a user,
$I_u^r := \{i \in I \mid (u, i, y) \in D, y = r\}$ are all items actually rated as r by a user.

[Weimer et al., 2008a,b]

Ranking / CoFiRank-NDCG

$$\ell(y, \hat{y}; c) := \max_{\pi} (1 - NDCG(y, \pi) + \langle c, \hat{y}^{\pi} - \hat{y} \rangle)$$

- convex upper bound for

$$1 - NDCG(y, \operatorname{argsort} \hat{y})$$

- but non-decomposable
  - requires the structured prediction framework and a special learning algorithm (see Weimer et al. [2008a] for details).

Notation:
$y^{\pi} := (y_{\pi(1)}, y_{\pi(2)}, \ldots, y_{\pi(n)}), \quad y \in \mathbb{R}^n, \pi \in S_n$
Weimer et al. [2008a] use $c_i := (i+1)^{-0.25}$.

Ranking / CoFiRank

	Method	N=10	N=20	N=50
EachMovie	CoFi^{Rank}-NDCG	$\mathbf{0.6367 \pm 0.001}$	$\mathbf{0.6619 \pm 0.0022}$	$\mathbf{0.6771 \pm 0.0019}$
	GPR	0.4558 ± 0.015	0.4849 ± 0.0066	0.5375 ± 0.0089
	CGPR	0.5734 ± 0.014	0.5989 ± 0.0118	0.6341 ± 0.0114
	GPOR	0.3692 ± 0.002	0.3678 ± 0.0030	0.3663 ± 0.0024
	CGPOR	0.3789 ± 0.011	0.3781 ± 0.0056	0.3774 ± 0.0041
	MMMF	0.4746 ± 0.034	0.4786 ± 0.0139	0.5478 ± 0.0211
MovieLens	CoFi^{Rank}-NDCG	$\mathbf{0.6237 \pm 0.0241}$	$\mathbf{0.6711 \pm 0.0065}$	0.6455 ± 0.0103
	GPR	0.4937 ± 0.0108	0.5020 ± 0.0089	0.5088 ± 0.0141
	CGPR	0.5101 ± 0.0081	0.5249 ± 0.0073	0.5438 ± 0.0063
	GPOR	0.4988 ± 0.0035	0.5004 ± 0.0046	0.5011 ± 0.0051
	CGPOR	0.5053 ± 0.0047	0.5089 ± 0.0044	0.5049 ± 0.0035
	MMMF	0.5521 ± 0.0183	0.6133 ± 0.0180	$\mathbf{0.6651 \pm 0.0190}$

Evaluation measure: NDCG@10. N is number of ratings/test user.
GPR, CGPR, GPOR, CGPOR are Gaussian Process Models from Yu et al. [2006].

[Weimer et al., 2008a]

Ranking / CoFiRank

	Method	N=10	N=20	N=50
EachMovie	CoFi^{RANK}-NDCG	0.6562 ± 0.0012	0.6644 ± 0.0024	0.6406 ± 0.0040
	CoFi^{RANK}-Ordinal	$\mathbf{0.6727 \pm 0.0309}$	$\mathbf{0.7240 \pm 0.0018}$	$\mathbf{0.7214 \pm 0.0076}$
	CoFi^{RANK}-Regression	0.6114 ± 0.0217	0.6400 ± 0.0354	0.5693 ± 0.0428
MovieLens	CoFi^{RANK}-NDCG	0.6400 ± 0.0061	0.6307 ± 0.0062	0.6076 ± 0.0077
	CoFi^{RANK}-Ordinal	0.6233 ± 0.0039	0.6686 ± 0.0058	$\mathbf{0.7169 \pm 0.0059}$
	CoFi^{RANK}-Regression	$\mathbf{0.6420 \pm 0.0252}$	0.6509 ± 0.0190	0.6584 ± 0.0187
	MMMF	0.6061 ± 0.0037	$\mathbf{0.6937 \pm 0.0039}$	0.6989 ± 0.0051
Netflix	CoFi^{RANK}-NDCG	0.6081	0.6204	
	CoFi^{RANK}-Regression	0.6082	0.6287	

Evaluation measure: NDCG@10. N is number of ratings/test user.

[Weimer et al., 2008a]

Unary and Ordinal Targets / Summary

Two different problems:

- ordinal regression $\rightsquigarrow$ models with ordinal binning losses.
  - bin thresholds are parameters that can be learned along with the other parameters.
  - MMMF, OrdRec, and the model in Paquet et al. [2012].

- ranking $\rightsquigarrow$ models with pair losses or non-decomposable losses.
  - no additional parameters.
  - models with pair contrast can be learned by SGD etc., models with more complex losses require more complex learning algorithms.
  - unary targets usually handled as ranking problem.
  - CoFiRank, BPR.

Outline

Example: RS with Attributes

		item	1	2	3	4	5	6	7	8	9	...
		year	'95	'95	'95	'95	'95	'95	'95	'95	'95	
		action	−	+	−	+	−	−	−	−	−	
		children's	+	−	−	−	−	−	−	+	−	
		...										

	gender	age	...
user 1	m	24	
2	f	53	
3	m	23	
4	m	24	
5	f	33	
6	m	42	
⋮			

item	1	2	3	4	5	6	7	8	9	...
user 1	4			3						
2	1									
3		4								
4					5					
5								2		
6								4		
⋮										

Can we build better models using the attributes?

Example: RS with Item Attributes

Idea: factorize both matrices

$$
\begin{aligned}
\hat{y}(u,i) &:= \langle \phi_u^Y, \psi_i^Y \rangle \\
\hat{z}(i,a) &:= \langle \phi_i^Z, \psi_a^Z \rangle \\
f(\phi^Y, \psi^Y, \phi^Z, \psi^Z) &:= \sum_{(u,i,y) \in D^Y} \ell^Y(y, \hat{y}(u,i)) + \sum_{(i,a,z) \in D^Z} \ell^Z(z, \hat{z}(i,a)) \\
&\quad + \Omega(\phi^Y, \psi^Y, \phi^Z, \psi^Z)
\end{aligned}
$$

Example: RS with Item Attributes

Idea: factorize both matrices sharing features

$$\hat{y}(u,i) := \langle \phi_u^Y, \psi_i^Y \rangle$$
$$\hat{z}(i,a) := \langle \phi_i^Z, \psi_a^Z \rangle$$
$$\psi_i^Y = \phi_i^Z$$
$$f(\phi^Y, \psi^Y, \phi^Z, \psi^Z) := \sum_{(u,i,y) \in D^Y} \ell^Y(y, \hat{y}(u,i)) + \sum_{(i,a,z) \in D^Z} \ell^Z(z, \hat{z}(i,a)) + \Omega(\phi^Y, \psi^Y, \phi^Z, \psi^Z)$$

Example: RS with Item Attributes

Idea: factorize both matrices sharing features

$$
\begin{aligned}
\hat{y}(u,i) &:= \langle \phi^U_u, \phi^I_i \rangle \\
\hat{z}(i,a) &:= \langle \phi^I_i, \phi^A_a \rangle \\
f(\phi^U, \phi^I, \phi^A) &:= \sum_{(u,i,y) \in D^Y} \ell^Y(y, \hat{y}(u,i)) + \sum_{(i,a,z) \in D^Z} \ell^Z(z, \hat{z}(i,a)) \\
&\quad + \Omega(\phi^U, \phi^I, \phi^A)
\end{aligned}
$$

Example: RS with Item Attributes

Idea: factorize both matrices sharing features, weighting the auxiliary loss

$$\hat{y}(u,i) := \langle \phi^U_u, \phi^I_i \rangle$$
$$\hat{z}(i,a) := \langle \phi^I_i, \phi^A_a \rangle$$
$$f(\phi^U, \phi^I, \phi^A) := \sum_{(u,i,y) \in D^Y} \ell^Y(y, \hat{y}(u,i)) + \alpha \sum_{(i,a,z) \in D^Z} \ell^Z(z, \hat{z}(i,a)) + \Omega(\phi^U, \phi^I, \phi^A)$$

Relation Z has a regularization effect on ϕ^I:
such ϕ^I are preferred that are useful to also reconstruct Z.

Multi-relational Prediction Problem (Matrices)

Let

- $\mathcal{E}$ be a set of entity classes, each $E \in \mathcal{E}$ a set of entities,
- $\mathcal{Z}$ be a set of matrices, each $Z \in \mathcal{Z}$ a matrix with
  - dimensions $s(Z) \in \mathcal{E}^2$ and
  - observed cells $Z \subseteq \text{dom}(Z) \times \mathbb{R}$, with $\text{dom}(Z) := s(Z)_1 \times s(Z)_2$.
- $Y \in \mathcal{Z}$ called **target matrix**, $Z \in \mathcal{Z}, Z \neq Y$ **auxiliary matrix**.

Sought is a model

$$\hat{y} : \text{dom}(Y) \to \mathbb{R}$$

with small prediction error for unseen test instances $Y_{\text{test}} \subseteq \text{dom}(Y) \times \mathbb{R}$ of the target matrix:

$$\ell(\hat{y}; Y_{\text{test}}) := \sum_{(x_1, x_2, y) \in Y_{\text{test}}} \ell^Y(y, \hat{y}(x_1, x_2))$$

Multi-relational Prediction Problem (Tensors)

Let

- $\mathcal{E}$ be a set of entity classes, each $E \in \mathcal{E}$ a set of entities,
- $\mathcal{Z}$ be a set of tensors, each $Z \in \mathcal{Z}$ a tensor with
  - dimensions $s(Z) \in \mathcal{E}^*$ and
  - observed cells $Z \subseteq \mathrm{dom}(Z) \times \mathbb{R}$, with $\mathrm{dom}(Z) := \prod_{i=1}^{|s(Z)|} s(Z)_i$.
- $Y \in \mathcal{Z}$ called **target tensor**, $Z \in \mathcal{Z}, Z \neq Y$ **auxiliary tensor**.

Sought is a model

$$\hat{y} : \mathrm{dom}(Y) \to \mathbb{R}$$

with small prediction error for unseen test instances $Y_{\text{test}} \subseteq \mathrm{dom}(Y) \times \mathbb{R}$ of the target tensor:

$$\ell(\hat{y}; Y_{\text{test}}) := \sum_{(x_1, x_2, y) \in Y_{\text{test}}} \ell^Y(y, \hat{y}(x_1, x_2))$$

Multi-relational Prediction Problem

Matrices (and tensors) can be used to encode positive and negative examples of relations:

$$\begin{aligned} \text{KB} \vDash R(x_1, x_2) \quad &\Longleftrightarrow: \quad Z(x_1, x_2) = 1 \\ \text{KB} \vDash \neg R(x_1, x_2) \quad &\Longleftrightarrow: \quad Z(x_1, x_2) = 0 \end{aligned}$$

In the same way certainty factors could be encoded.

When only positive relation facts are contained in the knowledge base (in recsys sometimes called implicit feedback),

- vanilla factorization models cannot work,
- but see section 3 on unary targets about what to do.

Multi-relational Factorization Model (Matrices)

$$\hat{Z}(x) := \langle \phi_{x_1}^{s(Z)_1}, \phi_{x_2}^{s(Z)_2} \rangle, \qquad \forall Z \in \mathcal{Z}, x \in \mathrm{dom}(Z)$$

$$f((\phi^E)_{E \in \mathcal{E}}; \mathcal{Z}) := \sum_{Z \in \mathcal{Z}} \alpha_Z \sum_{(x,z) \in Z} \ell^Z(z, \hat{Z}(x)) \quad + \Omega((\phi^E)_{E \in \mathcal{E}})$$

- Usually $\alpha_Y := 1$, the other α's are hyperparameters.
- ℓ^Y is given by the problem, $\ell^Z, Z \neq Y$ can be chosen ("structural hyperparameter").

Called

- Collective Matrix Factorization [Singh and Gordon, 2008b] (also allows prediction link function and latent feature constraints).
- Multi-relational Matrix Factorization [Lippert et al., 2008] (without the α weights).

Multi-relational Factorization Model / Learning

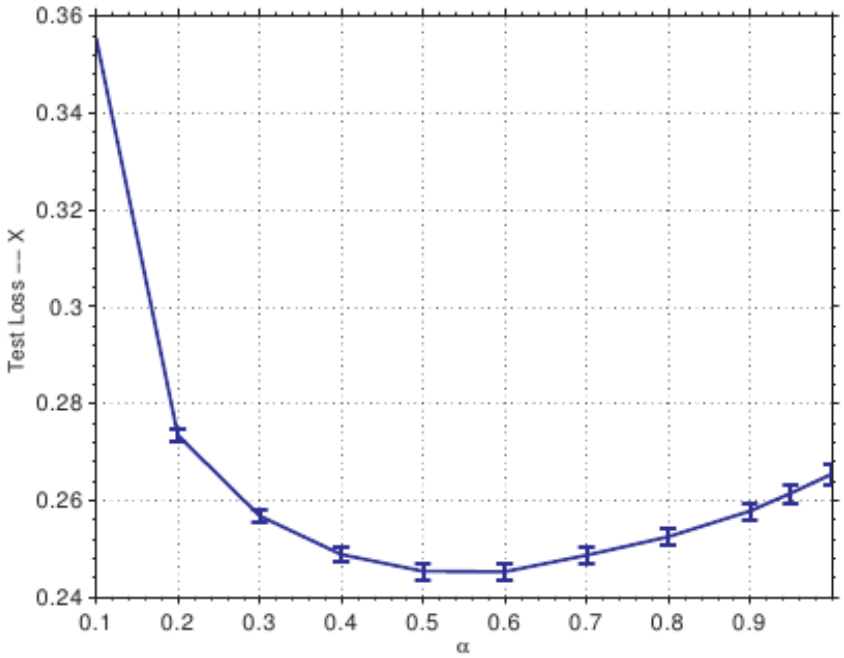
All learning algorithms for matrix factorization models can be carried over to multi-relational matrix factorization models:

- sequentially:
  - first learn auxiliary relations, then target relation.
  - beware: factorization of the auxiliary relations not informed by the target relation!
    - corresponds to regression on unsupervised dimensionality reduction (e.g., PCA regression).
- jointly:
  - alternate steps for different relations.
  - ensure all relations are converged.

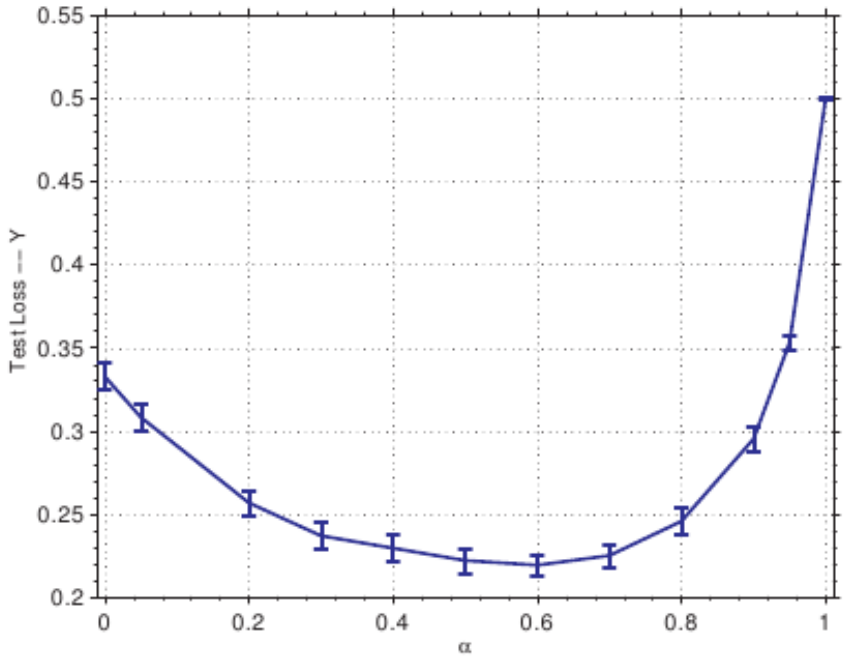
Multi-relational Factorization Model / Learning

SGD-MRFM$(\mathcal{Z}, \alpha, \eta, \sigma^2, n, \epsilon)$:
initialize $\phi^E \sim \mathcal{N}(0, \sigma^2)$ *for all* $E \in \mathcal{E}$
repeat
  $\phi^{old} := \phi$
  for $i = 1, \ldots, n$ **do**
    for $Z \in \mathcal{Z}$ **do**
      $E := s(Z)_1, F := s(Z)_2$
      draw $(x, y) \in Z$ *randomly*
      $\phi^E_{x_1} := \phi^E_{x_1} - \eta \frac{\partial}{\partial \phi^E_{x_1}} (\alpha_Z \ell^Z(y, \langle \phi^E_{x_1}, \phi^F_{x_2} \rangle) + \frac{1}{freq_E(x_1)} \Omega(\phi^E_{x_1}))$
      $\phi^F_{x_2} := \phi^F_{x_2} - \eta \frac{\partial}{\partial \phi^F_{x_2}} (\alpha_Z \ell^Z(y, \langle \phi^E_{x_1}, \phi^F_{x_2} \rangle) + \frac{1}{freq_F(x_2)} \Omega(\phi^F_{x_2}))$
    end
  end
until $||\phi - \phi^{old}|| < \epsilon$
return ϕ

Multi-relational Factorization Model / Evaluation



(a) Ratings

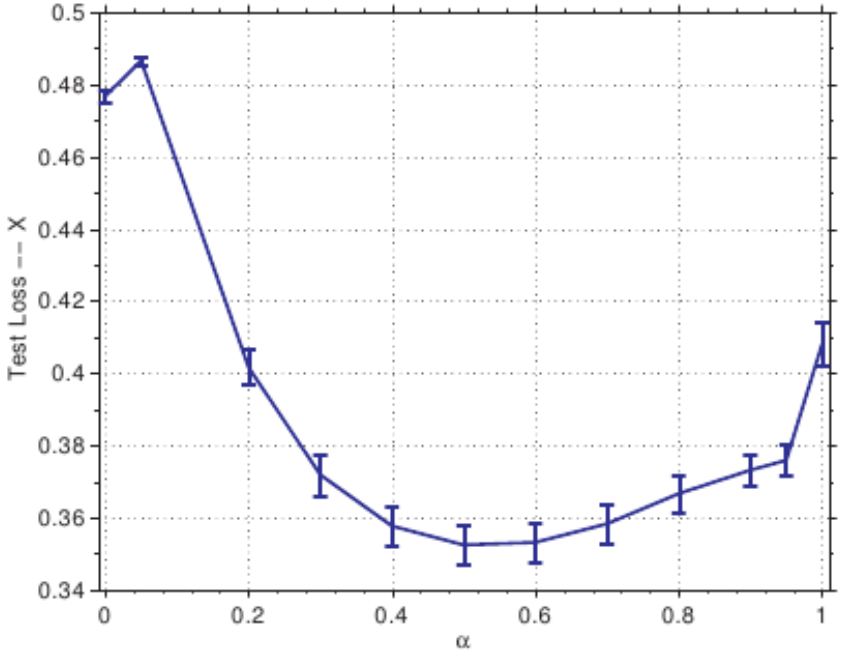


(b) Genres

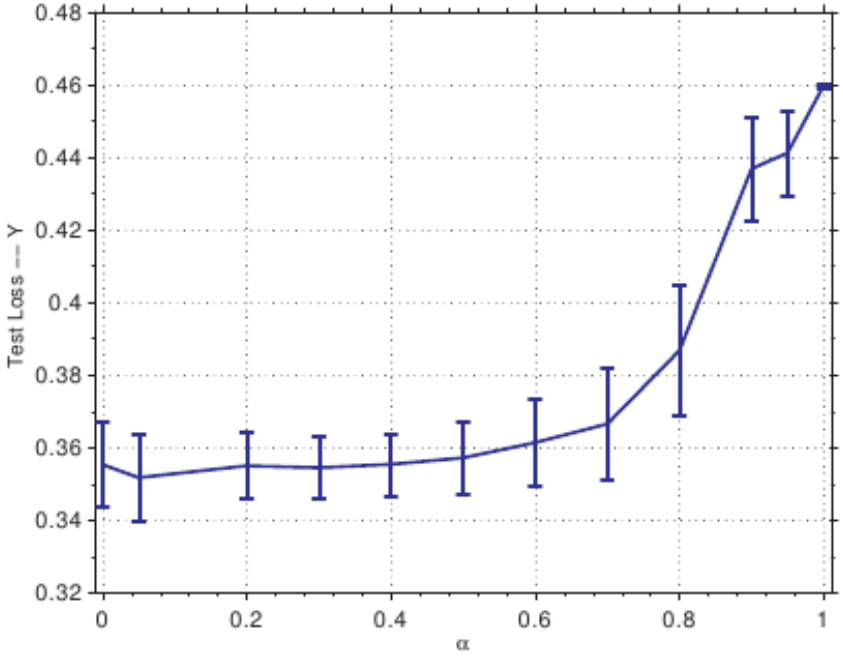
Netflix dataset (sample):
500 users, 3000 movies, 63500 ratings; k=20.
Relations "is_rated" (Netflix) and "has_genre" (IMDB). MAE.

[Singh and Gordon, 2008b]

Multi-relational Factorization Model / Evaluation



(a) Ratings



(b) Genres

Netflix dataset (sample):
750 users, 1000 movies, 7500 ratings; k=20.
Relations "is_rated" (Netflix) and "has_genre" (IMDB). MAE.

[Singh and Gordon, 2008b]

Multi-relational Factorization Model / Evaluation

Evaluations so far answered most important questions:

1. Joint factorization of auxiliary relations can improve the prediction accuracy of the target relation.
2. Gains are larger for sparse (target) relations.

Multi-relational Factorization Model / Evaluation

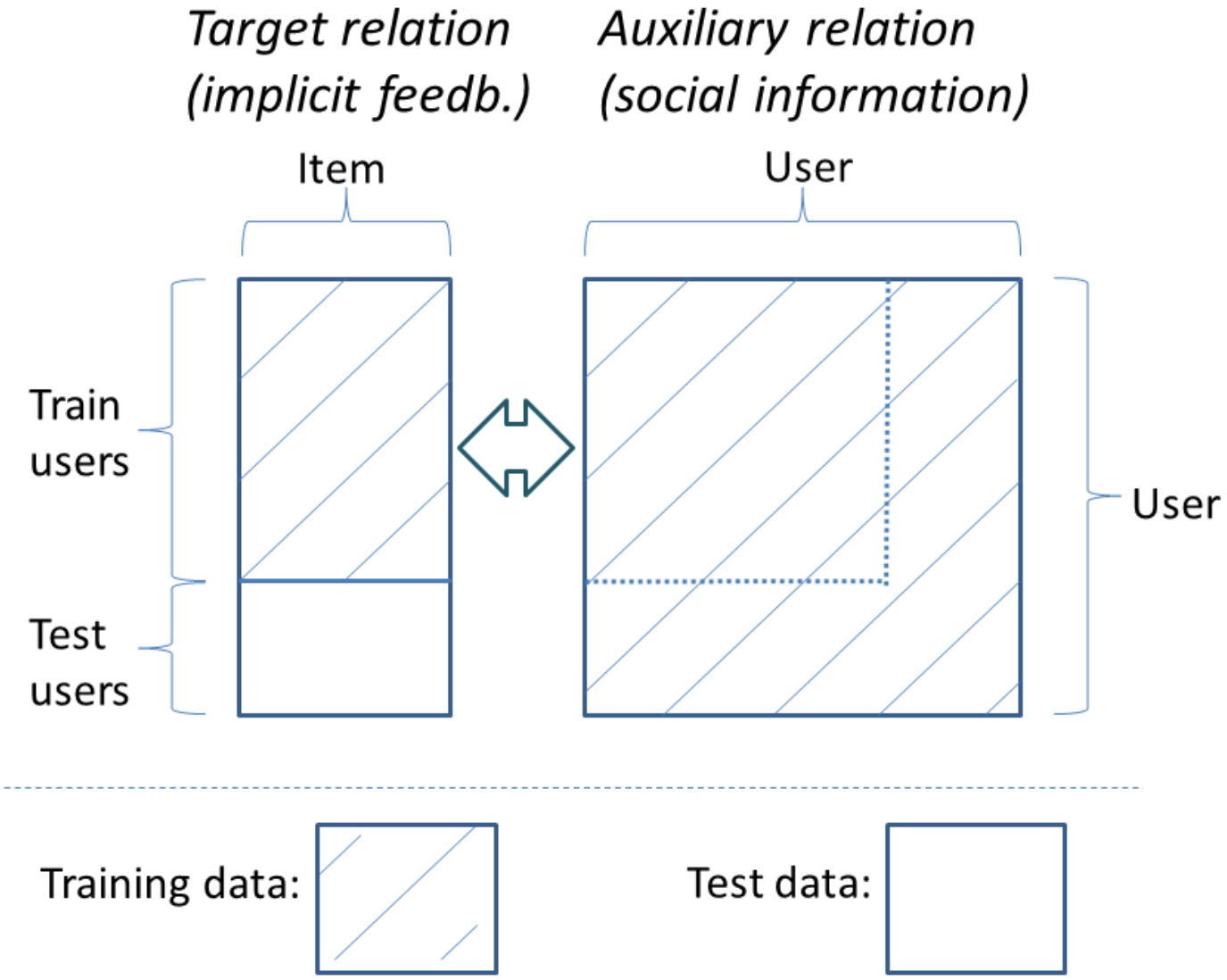
Evaluations so far answered most important questions:

1. Joint factorization of auxiliary relations can improve the prediction accuracy of the target relation.
2. Gains are larger for sparse (target) relations.

Open questions:

1. Do multirelational factorization models exploit the auxiliary information better than other models?
   - Evtl. even than models designed with the specific application task in mind?
2. Can the hyperparameters α be identified on the training set?
   - By ex-post picking a value for a hyperparameter that introduces some bias we often can see improvements over the base model.

Example: Coldstart Problem



Example: Coldstart Problem / State-of-the-Art Approaches

- First pre-process the auxiliary relation
  - Dimensionality Reduction
  - Feature extraction
  - ...
- Then learn a prediction model on the target relation

Example: Coldstart Problem / Multirelational BPR

$\mathcal{E}$ a set called **entity classes**.

X_E a set called **entity extensions** (for every $E \in \mathcal{E}$).

$\mathcal{R}$ a set called (binary) **relations**.

$s_R \in \mathcal{E}^2$ the **signature** of relation $R \in \mathcal{R}$.

$S_R \subseteq X_E \times X_F \times \mathbb{R}$ a set called **relation extensions** of relation $R \in \mathcal{R}$, $(E, F) := s_R$.

$$\hat{y}^R(e, f) := \langle \phi_e^E, \phi_f^F \rangle, \quad R \in \mathcal{R}, (E, F) := s_R, (e, f) \in X_E \times X_F$$

regularized loss:

$$f(\phi; S) := \sum_{R \in \mathcal{R}} \alpha_R \sum_{(e,f,y) \in S_R} \ell^R(y, \hat{y}^R(e, f)) + \sum_{E \in \mathcal{E}} \lambda_E ||\phi^E||^2$$

For positive-only relations ($y \equiv 1$):

$$\sum_{(e,f,y) \in S_R} \ell^R(y, \hat{y}^R(e, f)) \overset{BPR}{\rightsquigarrow} - \sum_{(e,f^+,f^-) \in D_{S_R}} \ln \sigma(\hat{y}^R(e, f^+) - \hat{y}^R(e, f^-))$$

Example: Coldstart Problem / Learning Multirelational BPR

$$\text{MR-BPR}(\phi; D) := -\sum_{R \in \mathcal{R}} \alpha_R \sum_{(e,f^+,f^-) \in D_{S_R}} \ln \sigma(\hat{y}^R(e, f^+) - \hat{y}^R(e, f^-)) + \sum_{E \in \mathcal{E}} \lambda_E ||\phi^E||^2$$

LearnMR-BPR($\mathcal{E}, \mathcal{R}, D, \lambda, \alpha, \eta$):
initialize $\hat{\Theta} := (\phi_E)_{E \in \mathcal{E}}$
repeat
  for $R \in \mathcal{R}$ **do**
    draw (e, f^+, f^-) *from* D_R
    $\hat{\Theta} \leftarrow \hat{\Theta} + \eta \left(\alpha_R \frac{e^{-\hat{y}^R(e,f^+,f^-)}}{1+e^{-\hat{y}^R(e,f^+,f^-)}} \cdot \frac{\partial}{\partial \hat{\Theta}} \hat{y}^R(e, f^+, f^-) - \lambda_\Theta \cdot \hat{\Theta} \right)$
  end
until *convergence*
return $\hat{\Theta}$

Example: Coldstart Problem / Evaluation

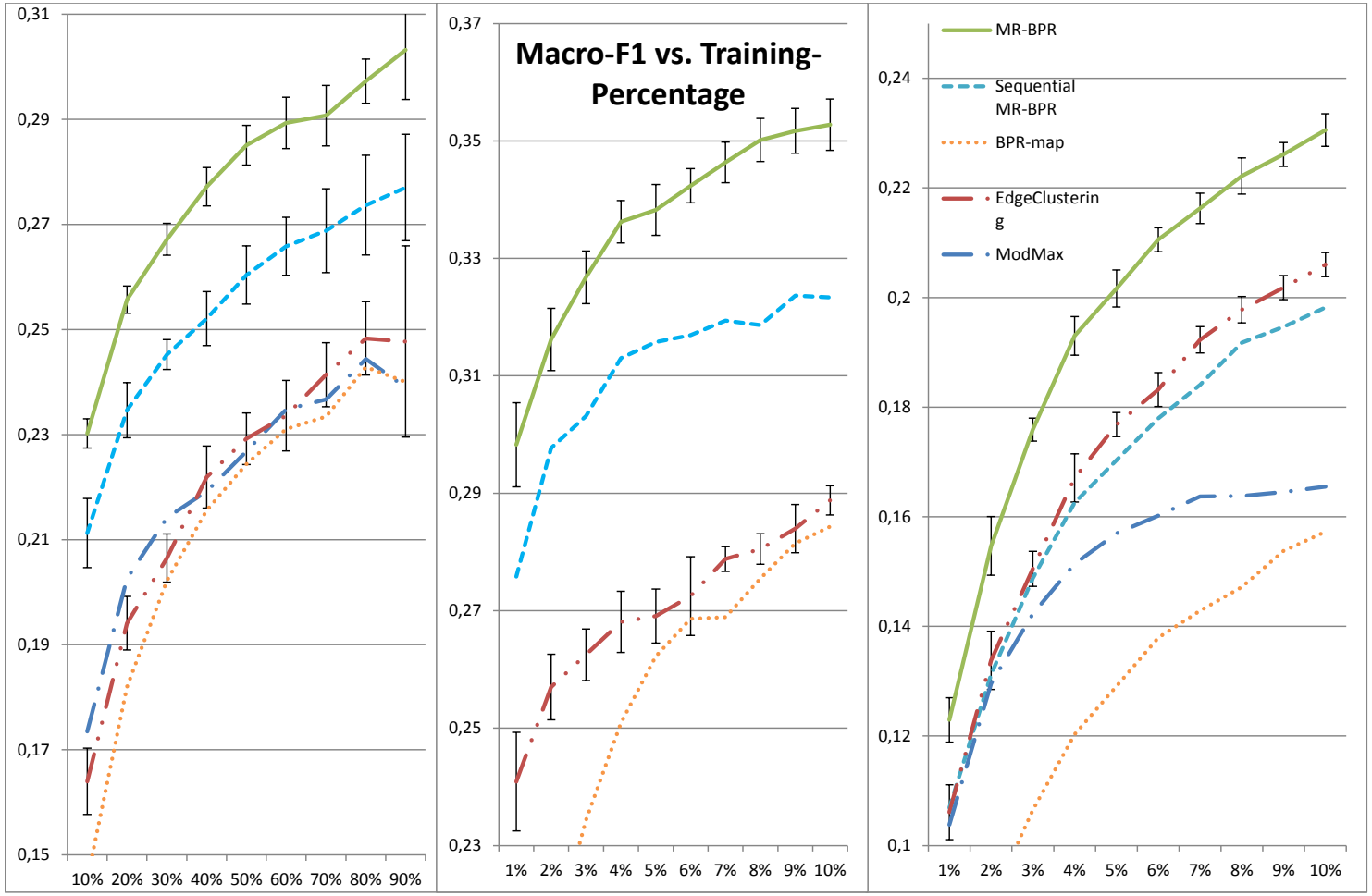
Datasets

dataset	relation	dimensionality	# instances	avg. instances /user	avg. instances /label	sparsity
Blogcatalog	target	10312x39	14,476	1.4	371.2	96.40%
	auxiliary	10312^2	667,966	64.7	-	99.37%
Flickr	target	80513x195	107,741	1.3	552.5	99.31%
	auxiliary	80513^2	11,799,764	146.6	-	99.82%
YouTube	target	31,703x47	50,691	1.6	1,078.5	96.60%
	auxiliary	$1,138,499^2$	5,980,886	5.3	-	99,9995%

Evaluated methods

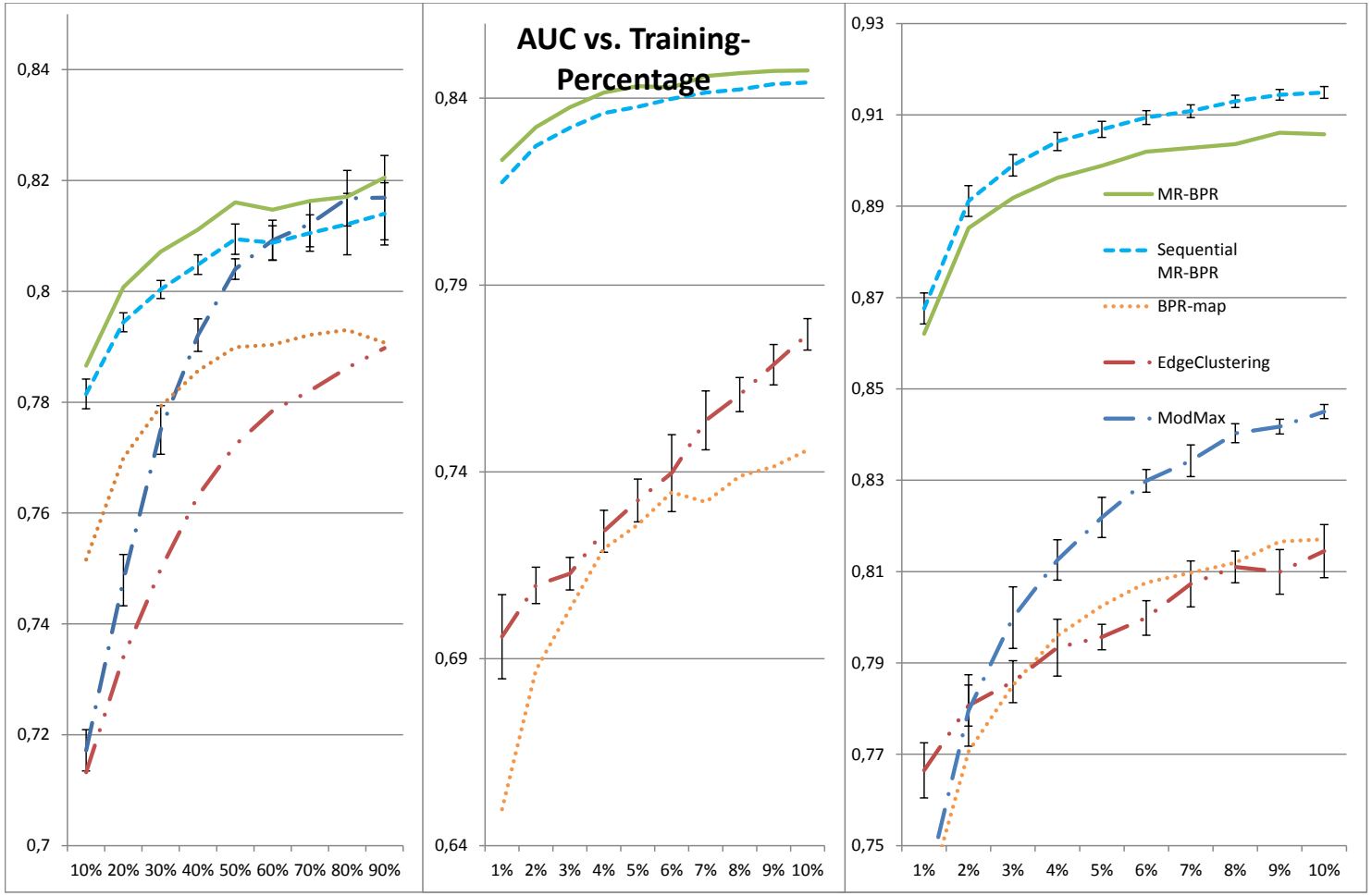
- modularity maximization [Tang and Liu, 2009a]
- edge clustering [Tang and Liu, 2009b]
- BPR-map [Gantner et al., 2010]

Example: Coldstart Problem / Evaluation (Macro-F1)



[Krohn-Grimberghe et al., 2012]

Example: Coldstart Problem / Evaluation (AUC)



[Krohn-Grimberghe et al., 2012]

Multi-relational Factorization Models / Summary

- Factorization models can be learned for several relations (with some common modes) at once
  - latent features are associated to modes / entity classes (so that the factorizations share latent features)
  - the loss of auxiliary relations has to be weighted down (hyperparameters)

- Parameters of multi-relational factorization models should not be learned sequentially, but jointly
  - e.g., interleaving update steps of different relations of iterative learning algorithms such as SGD or ALS.

- Many problems can be modeled by multi-relational factorization problems, e.g., attribute-aware recommender systems, link prediction with node and link features.

Outline

Probabilistic MF

Probabilistic interpretation for MF with L2 loss and L2 regularization:

$$p(Y \mid \Phi, \Psi, \alpha) := \prod_{(u,i,y) \in Y} \mathcal{N}(y \mid \Phi_u^T \Psi_i, \alpha^{-1})$$

$$p(\Phi \mid \alpha_\Phi) := \prod_{u=1}^{U} \mathcal{N}(\Phi_u \mid 0, \alpha_\Phi^{-1} I)$$

$$p(\Psi \mid \alpha_\Psi) := \prod_{i=1}^{I} \mathcal{N}(\Psi_i \mid 0, \alpha_\Psi^{-1} I)$$

Then the negative loglikelihood is just

$$f(\Phi, \Psi \mid Y, \alpha) := -\log p(Y \mid \Phi, \Psi, \alpha)\ p(\Phi \mid \alpha_\Phi)\ p(\Psi \mid \alpha_\Psi)$$

$$\propto \alpha \sum_{(u,i,y) \in Y} (y - \Phi_u^T \Psi_i)^2 + \alpha_\Phi \sum_{u=1}^{U} \Phi_u^T \Phi_u + \alpha_\Psi \sum_{i=1}^{I} \Psi_i^T \Psi_i$$

$$\propto \sum_{(u,i,y) \in Y} (y - \Phi_u^T \Psi_i)^2 + \lambda_\Phi ||\Phi||^2 + \lambda_\Psi ||\Psi||^2, \quad \lambda_\Phi := \frac{\alpha_\Phi}{\alpha}, \lambda_\Psi := \frac{\alpha_\Psi}{\alpha}$$

Bayesian Probabilistic MF

Adding hyperpriors:

$$p(Y \mid \Phi, \Psi, \alpha) := \prod_{(u,i,y) \in Y} \mathcal{N}(y \mid \Phi_u^T \Psi_i, \alpha^{-1})$$

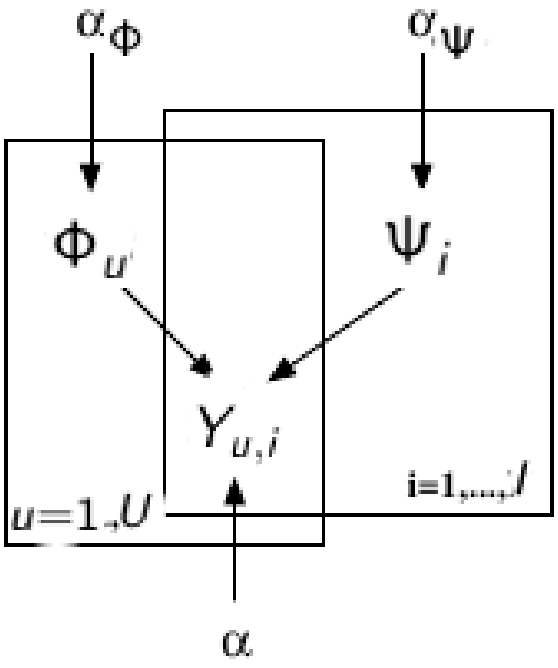
$$p(\Phi \mid \mu_\Phi, \Lambda_\Phi) := \prod_{u=1}^{U} \mathcal{N}(\Phi_u \mid \mu_\Phi, \Lambda_\Phi)$$

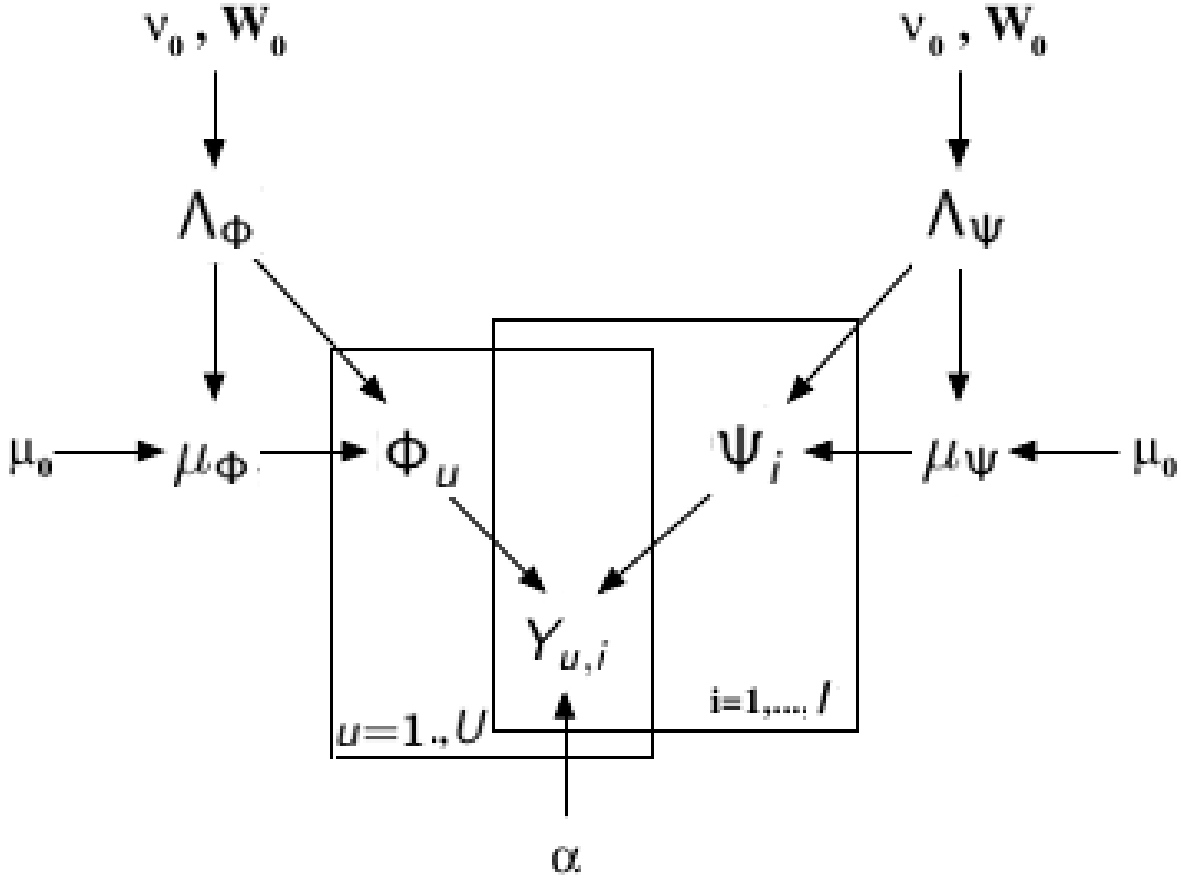
$$p(\Psi \mid \mu_\Psi, \Lambda_\Psi) := \prod_{i=1}^{I} \mathcal{N}(\Psi_i \mid \mu_\Psi, \Lambda_\Psi)$$

$$p(\mu_\Phi, \Lambda_\Phi \mid \mu_0, \nu_0, \beta_0, W_0) := \mathcal{N}(\mu_\Phi \mid \mu_0, (\beta_0 \Lambda_\Phi)^{-1})\ \mathcal{W}(\lambda_\Phi \mid W_0, \nu_0)$$

$$p(\mu_\Psi, \Lambda_\Psi \mid \mu_0, \nu_0, \beta_0, W_0) := \mathcal{N}(\mu_\Psi \mid \mu_0, (\beta_0 \Lambda_\Psi)^{-1})\ \mathcal{W}(\lambda_\Psi \mid W_0, \nu_0)$$

Bayesian Probabilistic MF





adapted from Salakhutdinov and Mnih [2008]

Bayesian Probabilistic MF / Inference

As usual, one cannot analytically compute

$$p(Y_{u,i} \mid Y, \alpha, \mu_0, \nu_0, \beta_0, W_0) := \int p(Y_{u,i} \mid \Phi_u, \Psi_i, \alpha)\; p(\Phi_u, \Psi_i \mid Y, \mu_\Phi, \Lambda_\Phi, \mu_\Psi, \Lambda_\Psi)$$
$$p(\mu_\Phi, \Lambda_\Phi \mid \mu_0, \nu_0, \beta_0, W_0)\; p(\mu_\Psi, \Lambda_\Psi \mid \mu_0, \nu_0, \beta_0, W$$
$$d(\Phi_u, \Psi_i, \mu_\Phi, \Lambda_\Phi, \mu_\Psi, \Lambda_\Psi)$$

Bayesian Probabilistic MF / Gibbs

But one can draw from conditional distributions sequentially (Gibbs):

1. draw new user features:

$$p(\Phi_u \mid Y, \Psi, \mu_\Phi, \Lambda_\Phi, \alpha) := \mathcal{N}(\Phi_u \mid \mu_u^*, (\Lambda_u^*)^{-1})$$

$$\text{with } \Lambda_u^* := \Lambda_\Phi + \alpha \sum_{(u',i,y) \in Y : u' = u} \Psi_i \Psi_i^T$$

$$\mu_u^* := (\Lambda_u^*)^{-1} (\alpha \sum_{(u',i,y) \in Y : u' = u} \Psi_i y) + \Lambda_\Phi \mu_\Phi$$

2. draw new user feature precision matrices:

$$p(\Lambda_\Phi \mid \Phi, \mu_0, \nu_0, \beta_0, W_0) := \mathcal{W}(\Lambda_\Phi \mid W_0^*, \nu_0^*)$$

$$\text{with } (W_0^*)^{-1} := W_0^{-1} + U\bar{S} + \frac{\beta_0 U}{\beta_0 + U}(\mu_0 - \bar{\Phi})(\mu_0 - \bar{\Phi})^T$$

$$\bar{S} := \frac{1}{U} \sum_{u=1}^{U} \Phi_u \Phi_u^T, \quad \bar{\Phi} := \frac{1}{U} \sum_{u=1}^{U} \Phi_u, \quad \nu_0^* := \nu_0 + U$$

Bayesian Probabilistic MF / Gibbs

But one can draw from conditional distributions sequentially (Gibbs):

3. draw new user feature means:

$$p(\mu_\Phi \mid \Phi, \Lambda_\Phi, \mu_0, \nu_0, \beta_0, W_0) := \mathcal{N}(\mu_\Phi \mid \mu_0^*, (\beta_0^* \Lambda_\Phi)^{-1})$$

$$\text{with } \mu_0^* := \frac{\beta_0 \mu_0 + U\bar{\Phi}}{\beta_0 + U}, \beta_0^* := \beta_0 + U,$$

1b, 2b, 3b. draw new item features Ψ_i, item feature means μ_Ψ and item feature precision matrices Λ_Ψ
(same formulas, just swap Φ and Ψ and u and i)

[Salakhutdinov and Mnih, 2008]

Bayesian Probabilistic MF / Gibbs

SGD-BPMF$(Y, K, T_0, T, \alpha, \mu_0, \nu_0, \beta_0, W_0)$:
$(\Phi^1, \Psi^1) :=$ *learn-MF*(Y, K)
for $t = 1, \ldots, T_0 + T$ **do**
  $\Lambda_\Phi^t \sim p(\Lambda_\Phi \mid \Phi^t, \mu_0, \nu_0, \beta_0, W_0)$
  $\mu_\Phi^t \sim p(\mu_\Phi \mid \Phi^t, \Lambda_\Phi^t, \mu_0, \nu_0, \beta_0, W_0)$
  $\Lambda_\Psi^t \sim p(\Lambda_\Psi \mid \Psi^t, \mu_0, \nu_0, \beta_0, W_0)$
  $\mu_\Psi^t \sim p(\mu_\Psi \mid \Psi^t, \Lambda_\Psi^t, \mu_0, \nu_0, \beta_0, W_0)$
  for $u = 1, \ldots, U$ **do**
    $\Phi_u^{t+1} \sim p(\Phi_u \mid Y, \Psi^t, \mu_\Phi^t, \Lambda_\Phi^t, \alpha)$
  end
  for $i = 1, \ldots, I$ **do**
    $\Psi_i^{t+1} \sim p(\Psi_i \mid Y, \Phi^{t+1}, \mu_\Psi^t, \Lambda_\Psi^t, \alpha)$
  end
end
return $(\Phi^t, \Psi^t)_{t=T_0+1,\ldots,T_0+T}$

Bayesian Probabilistic MF / Gibbs

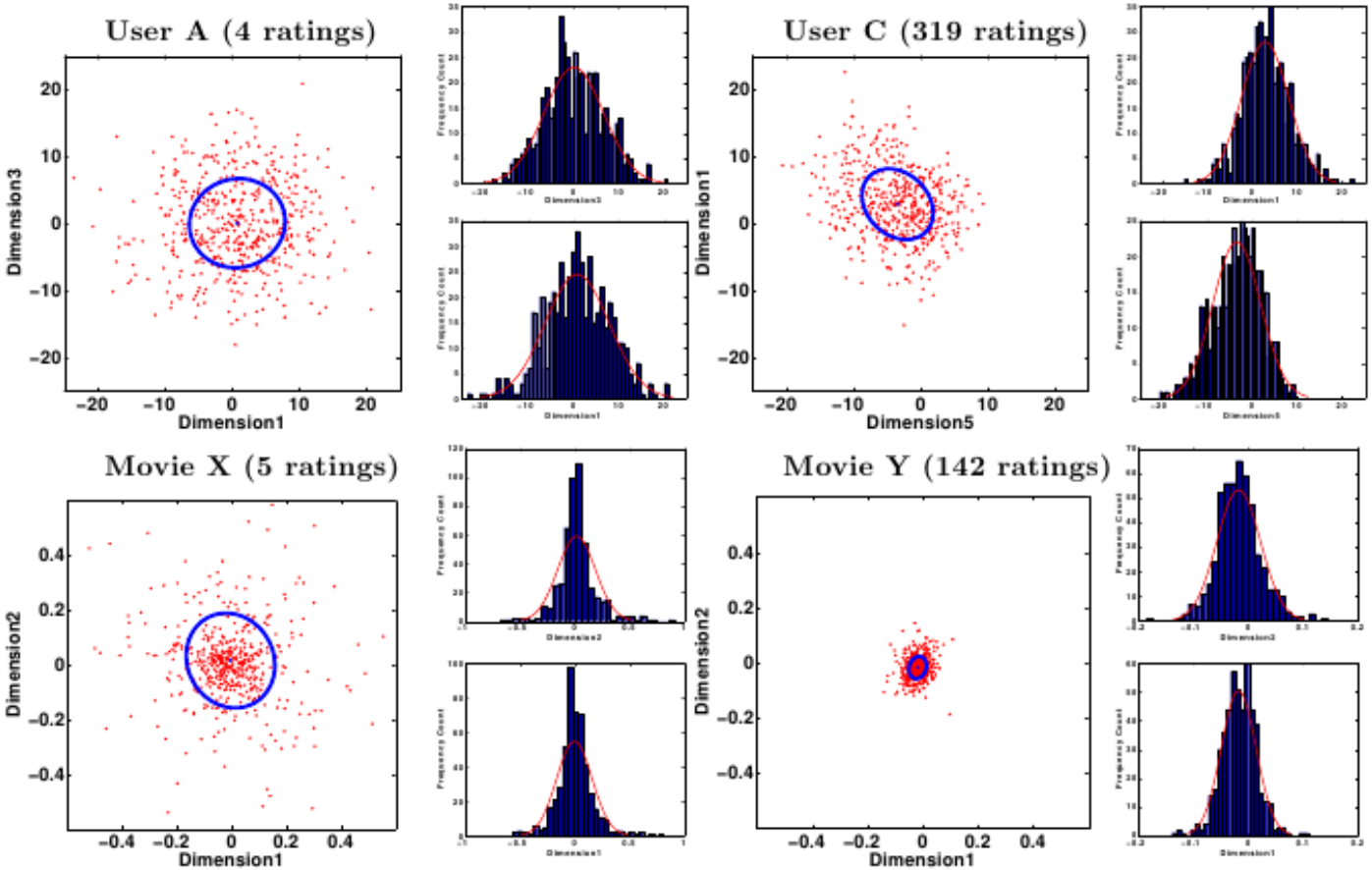
- hyperhyperparameters (used by Salakhutdinov and Mnih [2008]):

$$\mu_0 = 0, \quad \nu_0 = K, \quad W_0 = I, \quad \alpha = 2, \quad \beta_0 \stackrel{??}{=} 1$$

- prediction:

$$\hat{y}(u, i; (\Phi^t, \Psi^t)_{t=1,\ldots,T}) := \frac{1}{T} \sum_{t=1}^{T} \langle \Phi_u^t, \Psi_i^t \rangle$$

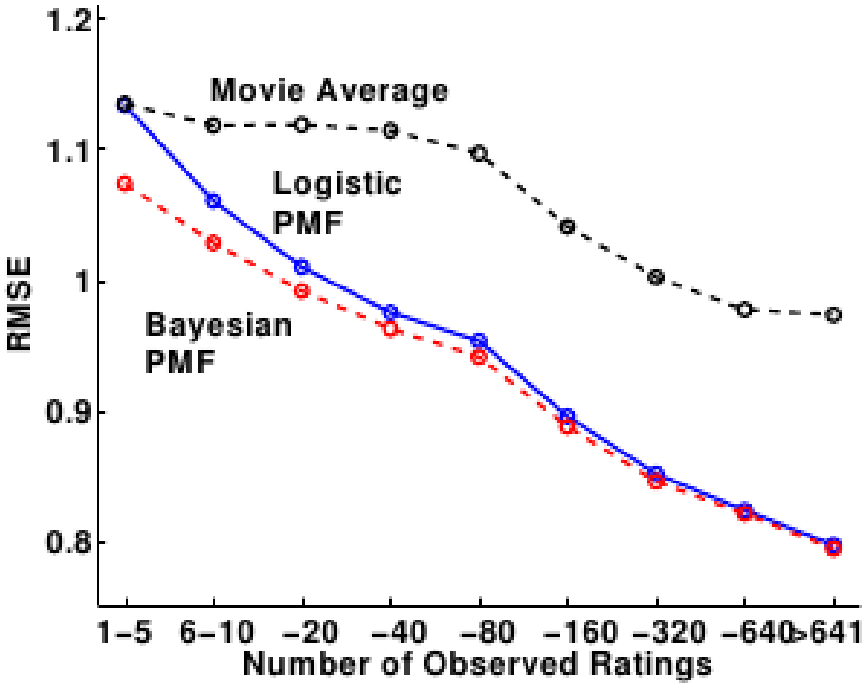
Bayesian Probabilistic MF / Evaluation



[Salakhutdinov and Mnih, 2008]

Bayesian Probabilistic MF / Evaluation

D	Valid. RMSE PMF	Valid. RMSE BPMF	% Inc.	Test RMSE PMF	Test RMSE BPMF	% Inc.
30	0.9154	0.8994	1.74	0.9188	0.9029	1.73
40	0.9135	0.8968	1.83	0.9170	0.9002	1.83
60	0.9150	0.8954	2.14	0.9185	0.8989	2.13
150	0.9178	0.8931	2.69	0.9211	0.8965	2.67
300	0.9231	0.8920	3.37	0.9265	0.8954	3.36



[Salakhutdinov and Mnih, 2008]

Bayesian Factorization Models / Summary

- Factorization models with L2 loss and L2 regularization can be interpreted as probabilistic models with normal targets and latent features.
- Priors — mean and precision of latent features — can be added resulting in a hierarchical Bayesian model.
- For conjugate priors (normal mean, Wishart precision), conditional distributions can be analytically computed and parameters can be learned by Gibbs sampling.
- Bayesian models usually provide better performance than a single model instance (comparable to an ensemble of model instances).

End of Part I

References I

R. M Bell, Y. Koren, and C. Volinsky. The BellKor solution to the netflix prize. *KorBell Team's Report to Netflix*, 2007.

R.M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, page 43–52, 2007.

A. M. Buchanan and A. W. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, page 316–322, 2005. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1467459`.

Simon Funk. Netflix update: Try this at home, 2006. URL `http://sifter.org/~simon/journal/20061211.html`.

Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. pages 176–185. IEEE, December 2010. ISBN 978-1-4244-9131-5. doi: 10.1109/ICDM.2010.129. URL `http://www.computer.org/csdl/proceedings/icdm/2010/4256/00/4256a176-abs.html`.

Zeno Gantner, Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. MyMediaLite: a free recommender system library. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, page 305–308, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0683-6. doi: 10.1145/2043932.2043989. URL `http://doi.acm.org/10.1145/2043932.2043989`.

D. Goldberg, D. Nichols, B. M Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Eighth IEEE International Conference on Data Mining, 2008. ICDM'08*, page 263–272, 2008.

D. Kim, S. Sra, and I. S. Dhillon. Fast newton-type methods for the least squares nonnegative matrix approximation problem. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, page 343–354, 2007. URL `http://www.cs.utexas.edu/~inderjit/public_papers/ls_nnma_siam07.pdf`.

Yehuda Koren and Joe Sill. OrdRec: an ordinal model for predicting personalized item rating distributions. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, page 117–124, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0683-6. doi: 10.1145/2043932.2043956. URL `http://doi.acm.org/10.1145/2043932.2043956`.

References II

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37, 2009. ISSN 0018-9162. URL http://dx.doi.org/10.1109/MC.2009.263.

Artus Krohn-Grimberghe, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, page 173–182, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-0747-5. doi: 10.1145/2124295.2124317. URL http://doi.acm.org/10.1145/2124295.2124317.

C. Lippert, S. H Weber, Y. Huang, V. Tresp, M. Schubert, and H. P Kriegel. Relation-prediction in multi-relational domains using matrix-factorization. In *NIPS Workshop: Structured Input-Structured Output*, 2008.

X. Ning and G. Karypis. SLIM: sparse linear methods for top-n recommender systems. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, page 497–506, 2011.

Rong Pan, Yunhong Zhou, Bin Cao, N.N. Liu, R. Lukose, M. Scholz, and Qiang Yang. One-class collaborative filtering. In *Eighth IEEE International Conference on Data Mining, 2008. ICDM '08*, pages 502 –511, 2008. doi: 10.1109/ICDM.2008.16.

Ulrich Paquet, Blaise Thomson, and Ole Winther. A hierarchical model for ordinal matrix factorization. *Statistics and Computing*, 22(4):945–957, 2012. ISSN 0960-3174. doi: 10.1007/s11222-011-9264-x. URL http://www.springerlink.com/content/n6w36236766m9605/abstract/.

K Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):572, 559, 1901.

A. H Phan, P. Tichavsk, and A. Cichocki. Fast damped gauss-newton algorithm for sparse and nonnegative tensor factorization. 2011.

István Pilászy, Dávid Zibriczky, and Domonkos Tikk. Fast als-based matrix factorization for explicit and implicit feedback datasets. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, page 71–78, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-906-0. doi: 10.1145/1864708.1864726. URL http://doi.acm.org/10.1145/1864708.1864726.

S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009)*, pages 452–461, 2009.

References III

J. D.M Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, page 719, 2005.

P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, page 175–186, 1994.

Sam Roweis. EM algorithms for PCA and SPCA. In *Proceedings of the 1997 conference on Advances in neural information processing systems 10*, NIPS '97, page 626–632, Cambridge, MA, USA, 1998. MIT Press. ISBN 0-262-10076-2. URL `http://dl.acm.org/citation.cfm?id=302528.302762`.

Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning*, pages 880–887, Helsinki, Finland, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390267. URL `http://portal.acm.org/citation.cfm?id=1390156.1390267&coll=portal&dl=ACM&CFID=83114244&CFTOKEN=43516963`.

A. Singh and G. Gordon. A unified view of matrix factorization models. *Machine Learning and Knowledge Discovery in Databases*, page 358–373, 2008a. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.141.5438&rep=rep1&type=pdf`.

Ajit P. Singh and Geoffrey J. Gordon. Relational learning via collective matrix factorization. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 650–658, Las Vegas, Nevada, USA, 2008b. ACM. ISBN 978-1-60558-193-4. doi: 10.1145/1401890.1401969. URL `http://portal.acm.org/citation.cfm?id=1401890.1401969`.

N. Srebro and T. Jaakkola. Weighted low-rank approximations. In *Proceedings of the 25th international conference on Machine learning*, volume 20, pages 720–727, Helsinki, Finland, 2003.

N. Srebro, J. D.M Rennie, and T. S Jaakkola. Maximum-margin matrix factorization. *Advances in neural information processing systems*, 17:1329–1336, 2005.

Lei Tang and Huan Liu. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 817–826, Paris, France, 2009a. ACM. ISBN 978-1-60558-495-9. doi: 10.1145/1557019.1557109. URL `http://portal.acm.org/citation.cfm?id=1557019.1557109&coll=GUIDE&dl=GUIDE&CFID=81119910&CFTOKEN=49152139`.

References IV

Lei Tang and Huan Liu. Scalable learning of collective behavior based on sparse social dimensions. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, page 1107–1116, New York, NY, USA, 2009b. ACM. ISBN 978-1-60558-512-3. doi: 10.1145/1645953.1646094. URL http://doi.acm.org/10.1145/1645953.1646094.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999. URL http://onlinelibrary.wiley.com/doi/10.1111/1467-9868.00196/abstract.

M. Weimer, A. Karatzoglou, Q. Le, and A. Smola. Cofi rank-maximum margin matrix factorization for collaborative ranking. *Advances in neural information processing systems*, 20:1593–1600, 2008a.

M. Weimer, A. Karatzoglou, and A. Smola. Improving maximum margin matrix factorization. *Machine Learning*, 72(3):263–276, 2008b.

T. Wiberg. Computation of principal components when data are missing. In *Proc. Second Symp. Computational Statistics*, page 229–236, 1976.

Shipeng Yu, Kai Yu, Volker Tresp, and Hans-Peter Kriegel. Collaborative ordinal regression. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, page 1089–1096, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143981. URL http://doi.acm.org/10.1145/1143844.1143981.