Tomáš Horváth

RECOMMENDER SYSTEMS

Tutorial at the conference

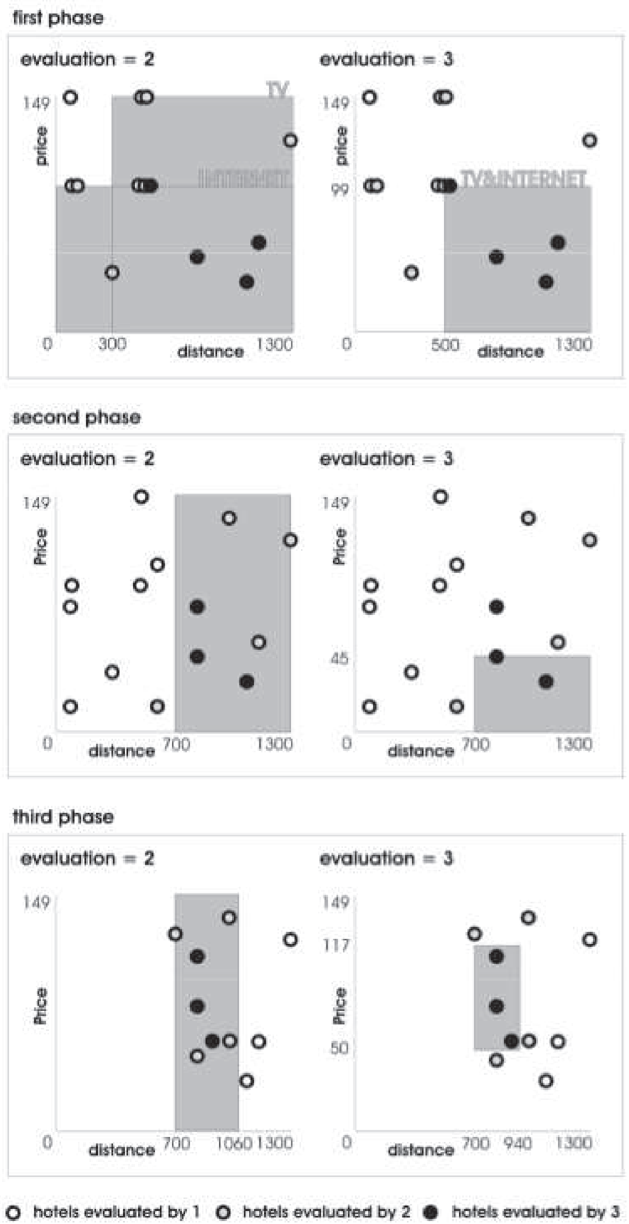
Znalosti 2012

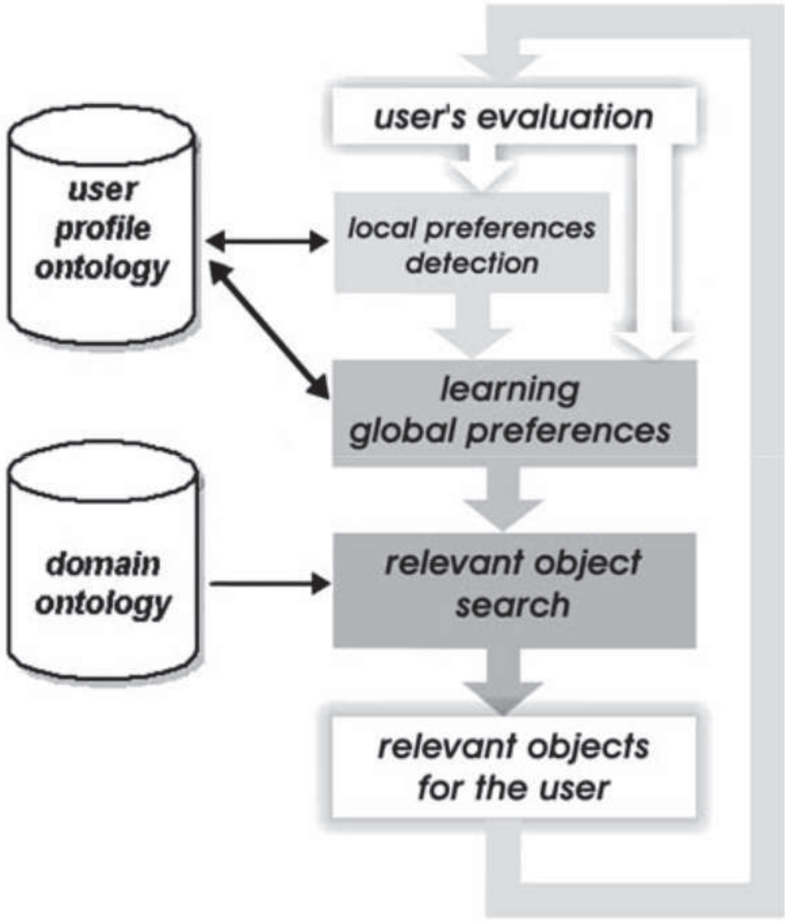
October 14-16, 2012, Mikulov, Czech Republic

Institute of Computer Science, Faculty od Science
Pavol Jozef Šafárik University in Košice, Slovak Republic

Information Systems and Machine Learning Lab
University of Hildesheim, Germany

Iterative recommendation





Rating prediction – example

$sim_{pc}(i,j)$	Titanic	Pulp Fiction	Iron Man	Forrest Gump	The Mummy
Titanic	1.0	-0.956	-0.815	NaN	-0.581
Pulp Fiction	–	1.0	0.948	NaN	0.621
Iron Man	–	–	1.0	NaN	0.243
Forrest Gump	–	–	–	1.0	NaN
The Mummy	–	–	–	–	1.0

NaN values are usually converted to zero (rare in case of enough data)

$sim_{pc}(u,v)$	Joe	Ann	Mary	Steve
Joe	1.0	-0.716	-0.762	-0.005
Ann	–	1.0	0.972	0.565
Mary	–	–	1.0	0.6
Steve	–	–	–	1.0

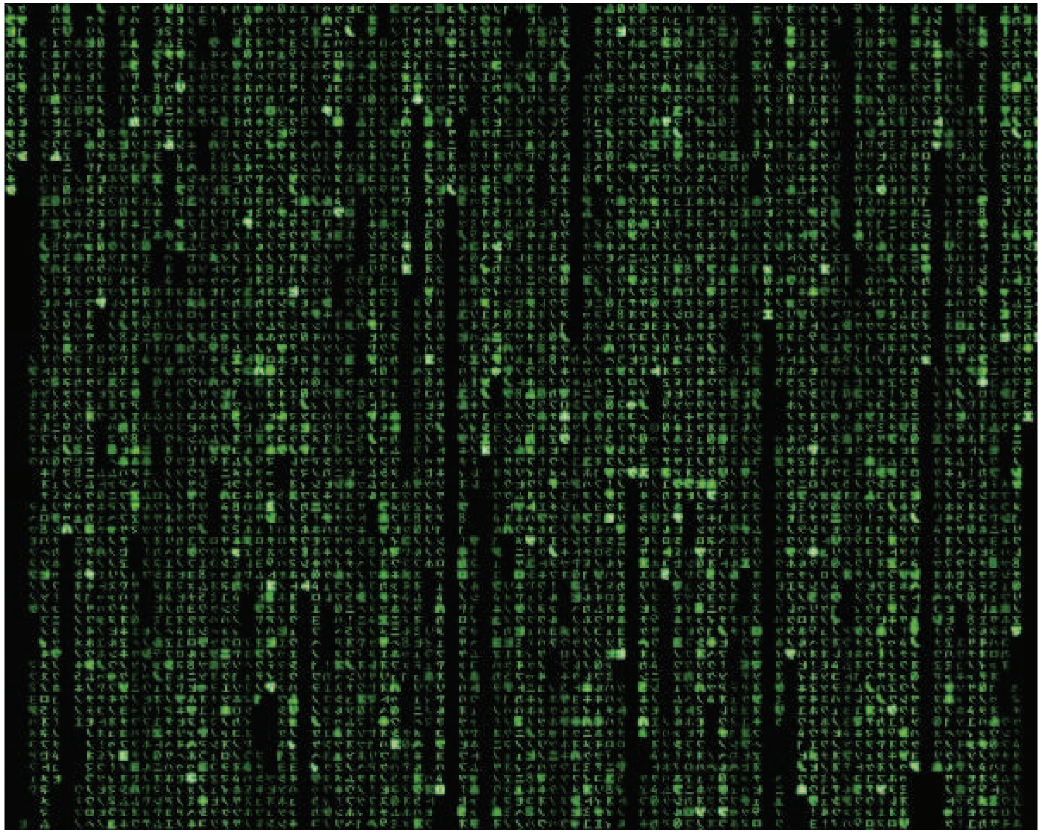
user-based

- $\mathcal{U}_{Titanic} = \{Joe, Ann, Mary\}$, $\mathcal{N}_{Titanic}^{Steve,2} = \{Mary, Ann\}$
- $\overline{\phi}_{Steve} = \frac{11}{3} = 3.67$, $\overline{\phi}_{Mary} = \frac{12}{4} = 3$, $\overline{\phi}_{Ann} = \frac{13}{4} = 3.25$
- $\hat{\phi}_{ST} = \overline{\phi}_S + \frac{s_{pc}(S,M)\cdot(\phi_{MT}-\overline{\phi}_M)+s_{pc}(S,A)\cdot(\phi_{AT}-\overline{\phi}_A)}{|s_{pc}(S,M)|+|s_{pc}(S,A)|} = 3.67 + \frac{0.6\cdot(4-3)+0.565\cdot(5-3.25)}{0.6+0.565} = 1.36$

item-based

- $\mathcal{I}_{\underline{S}teve} = \{\underline{P}ulp\ Fiction, \underline{I}ron\ Man, The\ \underline{M}ummy\}$, $\mathcal{N}_{Steve}^{\underline{T}itanic,2} = \{\underline{I}ron\ Man, The\ \underline{M}ummy\}$
- $\overline{\phi}_T = \frac{10}{3} = 3.34$, $\overline{\phi}_I = \frac{11}{3} = 3.67$, $\overline{\phi}_M = \frac{9}{3} = 3$
- $\hat{\phi}_{ST} = \overline{\phi}_T + \frac{s_{pc}(T,I)\cdot(\phi_{SI}-\overline{\phi}_I)+s_{pc}(T,M)\cdot(\phi_{SM}-\overline{\phi}_M)}{|s_{pc}(T,I)|+|s_{pc}(T,M)|} = 3.34 + \frac{-.815\cdot(4-3.67)-.581\cdot(4-3)}{0.815+0.581} = 2.73$

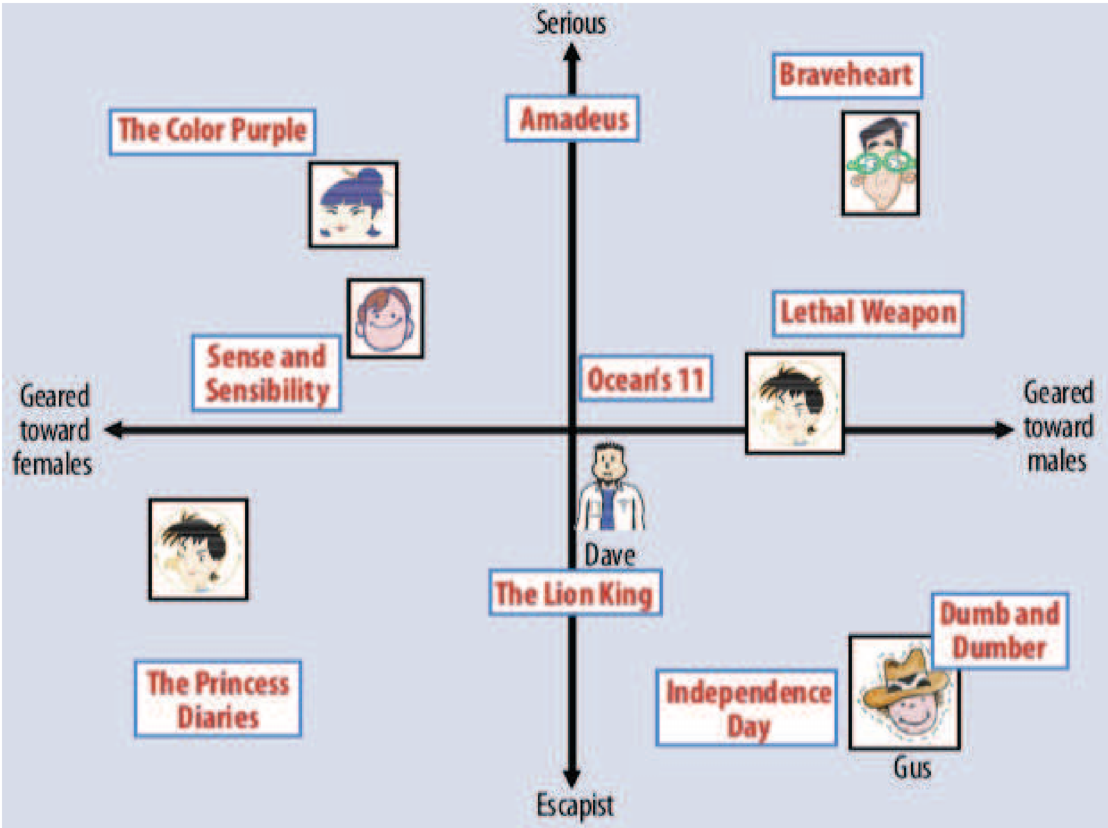
Matrix factorization



A latent space representation

Map users and items to a common latent space

- where dimensions or **factors** represent
  - items' **implicit properties**
  - users' **interest** in items' hidden properties



[1]

[1] The picture is taken from *Y. Koren et al. (2009). Matrix Factorization Techniques for Recommender Systems. Computer 42 (8).*

Known factorization models (1/2)

ϕ represented as a user-item matrix $\Phi^{n \times m}$

- n users, m items

[2] The picture is taken from wikipedia.

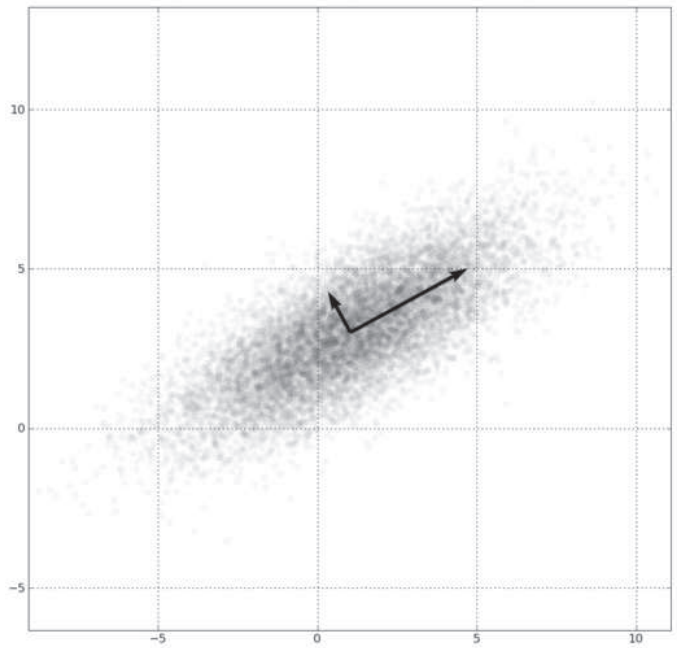
Known factorization models (1/2)

ϕ represented as a user-item matrix $\Phi^{n \times m}$

- n users, m items

Principal Component Analysis (PCA)

- transform data to a new coordinate system
  - variances by any projection of the data lies on coordinates in decreasing order



[2]

[2] The picture is taken from wikipedia.

Known factorization models (2/2)

Singular Value Decomposition (SVD)

$$\Phi = W^{n \times k} \Sigma^{k \times k} H^{n \times k^T}$$

- $W^T W = I$, $H^T H = I$
- column vectors of W are orthonormal eigenvectors of $\Phi\Phi^T$
- column vectors of H are orthonormal eigenvectors of $\Phi^T\Phi$
- Σ contains eigenvallues of W in descending order

[1] T.Raiko et al. (2007). Principal Component Analysis for Sparse High-Dimensional Data. Neural Information Processing, LNCS. 4984.

[2] A.K. Menon and Ch. Elkan (2011). Fast Algorithms for Approximating the Singular Value Decomposition. ACM Trans. Knowl. Discov. Data 5 (2).

Known factorization models (2/2)

Singular Value Decomposition (SVD)

$$\Phi = W^{n \times k} \Sigma^{k \times k} {H^{n \times k}}^T$$

- $W^T W = I$, $H^T H = I$
- column vectors of W are orthonormal eigenvectors of $\Phi\Phi^T$
- column vectors of H are orthonormal eigenvectors of $\Phi^T\Phi$
- Σ contains eigenvallues of W in descending order

PCA, SVD computed algebraically

- Φ is a **big** and **sparse** matrix
  - approximations of PCA[1], SVD[2]

[1] T.Raiko et al. (2007). Principal Component Analysis for Sparse High-Dimensional Data. Neural Information Processing, LNCS. 4984.

[2] A.K. Menon and Ch. Elkan (2011). Fast Algorithms for Approximating the Singular Value Decomposition. ACM Trans. Knowl. Discov. Data 5 (2).

MF – rating prediction (1/2)

recommendation task

- to find $\hat{\phi} : \mathcal{U} \times \mathcal{I} \to \mathbb{R}$ such that $acc(\hat{\phi}, \phi, \mathcal{T})$ is maximal

MF – rating prediction (1/2)

recommendation task

- to find $\hat{\phi} : \mathcal{U} \times \mathcal{I} \rightarrow \mathbb{R}$ such that $acc(\hat{\phi}, \phi, \mathcal{T})$ is maximal
  - acc is the **expected** accuracy on $\mathcal{T}$
  - training $\hat{\phi}$ on $\mathcal{D}$ such that the **empirical** loss $err(\hat{\phi}, \phi, \mathcal{D})$ is minimal

MF – rating prediction (1/2)

recommendation task

- to find $\hat{\phi} : \mathcal{U} \times \mathcal{I} \to \mathbb{R}$ such that $acc(\hat{\phi}, \phi, \mathcal{T})$ is maximal
  - acc is the **expected** accuracy on $\mathcal{T}$
  - training $\hat{\phi}$ on $\mathcal{D}$ such that the **empirical** loss $err(\hat{\phi}, \phi, \mathcal{D})$ is minimal

a simple, **approximative** MF **model**

- only $W^{n \times k}$ and $H^{m \times k}$
- k – the number of factors

$$\Phi^{n \times m} \approx \hat{\Phi}^{n \times m} = WH^T$$

- predicted rating $\hat{\phi}_{ui}$ of the user u for the item i

$$\hat{\phi}_{ui} = w_u h_i^T$$

MF – rating prediction (2/2)

the **loss** function $err(\hat{\phi}, \phi, \mathcal{D})$

- squared loss

$$err(\hat{\phi}, \phi, \mathcal{D}) = \sum_{(u,i)\in\mathcal{D}} e_{ui}^2 = \sum_{(u,i)\in\mathcal{D}} (\phi_{ui} - \hat{\phi}_{ui})^2 = \sum_{(u,i)\in\mathcal{D}} (\phi_{ui} - w_u h_i^T)^2$$

MF – rating prediction (2/2)

the **loss** function $err(\hat{\phi}, \phi, \mathcal{D})$

- squared loss

$$err(\hat{\phi}, \phi, \mathcal{D}) = \sum_{(u,i)\in\mathcal{D}} e_{ui}^2 = \sum_{(u,i)\in\mathcal{D}} (\phi_{ui} - \hat{\phi}_{ui})^2 = \sum_{(u,i)\in\mathcal{D}} (\phi_{ui} - w_u h_i^T)^2$$

the **objective function**

- **regularization** term $\lambda \geq 0$ to prevent overfitting
  - penalizing the magnitudes of parameters

$$f(\hat{\phi}, \phi, \mathcal{D}) = \sum_{(u,i)\in\mathcal{D}} (\phi_{ui} - w_u h_i^T)^2 + \lambda(\|W\|^2 + \|H\|^2)$$

MF – rating prediction (2/2)

the **loss** function $err(\hat{\phi}, \phi, \mathcal{D})$

- squared loss

$$err(\hat{\phi}, \phi, \mathcal{D}) = \sum_{(u,i)\in\mathcal{D}} e_{ui}^2 = \sum_{(u,i)\in\mathcal{D}} (\phi_{ui} - \hat{\phi}_{ui})^2 = \sum_{(u,i)\in\mathcal{D}} (\phi_{ui} - w_u h_i^T)^2$$

the **objective function**

- **regularization** term $\lambda \geq 0$ to prevent overfitting
  - penalizing the magnitudes of parameters

$$f(\hat{\phi}, \phi, \mathcal{D}) = \sum_{(u,i)\in\mathcal{D}} (\phi_{ui} - w_u h_i^T)^2 + \lambda(\|W\|^2 + \|H\|^2)$$

The task is to find parameters W and H such that, given λ, the objective function $f(\hat{\phi}, \phi, \mathcal{D})$ is minimal.

Gradient descent

How to find a minimum of an "objective" function $f(\Theta)$*?*

- in case of MF, $\Theta = W \cup H$, and
- $f(\Theta)$ refers to the error of approximation of Φ by WH^T

Gradient descent

How to find a minimum of an "objective" function $f(\Theta)$?

- in case of MF, $\Theta = W \cup H$, and
- $f(\Theta)$ refers to the error of approximation of Φ by WH^T

Gradient descent

input: f, α, Σ^2, *stopping criteria*
initialize $\Theta \sim \mathcal{N}(0, \Sigma^2)$
repeat
  $\Theta \leftarrow \Theta - \alpha \frac{\partial f}{\partial \Theta}(\Theta)$
until approximate minimum is reached
return Θ

Gradient descent

How to find a minimum of an "objective" function $f(\Theta)$?

- in case of MF, $\Theta = W \cup H$, and
- $f(\Theta)$ refers to the error of approximation of Φ by WH^T

<u>Gradient descent</u>

input: f, α, Σ^2, *stopping criteria*
initialize $\Theta \sim \mathcal{N}(0, \Sigma^2)$
repeat
$\quad \Theta \leftarrow \Theta - \alpha \frac{\partial f}{\partial \Theta}(\Theta)$
until approximate minimum is reached
return Θ

stopping criteria

- $|\Theta^{old} - \Theta| < \epsilon$
- maximum number of iterations reached
- a combination of both

Stochastic gradient descent

if f can be written as

$$f(\Theta) = \sum_{i=1}^{n} f_i(\Theta)$$

Stochastic gradient descent

if f can be written as

$$f(\Theta) = \sum_{i=1}^{n} f_i(\Theta)$$

Stochastic gradient descent (SGD)

input: f_i, α, Σ^2, *stopping criteria*
initialize $\Theta \sim \mathcal{N}(0, \Sigma^2)$
repeat
  for all i in random order **do**
    $\Theta \leftarrow \Theta - \alpha \frac{\partial f_i}{\partial \Theta}(\Theta)$
  end for
until approximate minimum is reached
return Θ

MF with SGD

updating parameters **iteratively** for each data point ϕ_{ui} in the opposite direction of the **gradient** of the objective function at the given point until a **convergence** criterion is fulfilled.

- updating the vectors w_u and h_i for the data point $(u, i) \in D$

MF with SGD

updating parameters **iteratively** for each data point ϕ_{ui} in the opposite direction of the **gradient** of the objective function at the given point until a **convergence** criterion is fulfilled.

- updating the vectors w_u and h_i for the data point $(u, i) \in D$

$$\frac{\partial f}{\partial w_u}(u, i) = -2(e_{ui}h_i - \lambda w_u)$$

$$\frac{\partial f}{\partial h_i}(u, i) = -2(e_{ui}w_u - 2\lambda h_i)$$

$$w_u(u, i) \leftarrow w_u - \alpha \frac{\partial f}{\partial w_u}(u, i) = w_u + \alpha(e_{ui}h_i - \lambda w_u)$$

$$h_i(u, i) \leftarrow h_i - \alpha \frac{\partial f}{\partial h_i}(u, i) = h_i + \alpha(e_{ui}w_u - \lambda h_i)$$

where $\alpha > 0$ is a **learning rate**.

MF with SGD – Algorithm

Hyper-parameters: $k, iters$ *(the max number of iteration)*, $\alpha, \lambda, \Sigma^2$

$W \leftarrow \mathcal{N}(0, \Sigma^2)$

$H \leftarrow \mathcal{N}(0, \Sigma^2)$

for $iter \leftarrow 1, \ldots, iters \cdot |\mathcal{D}|$ **do**

  draw randomly (u, i) from $\mathcal{D}$

  $\hat{\phi}_{ui} \leftarrow 0$

  for $j \leftarrow 1, \ldots, k$ **do**

    $\hat{\phi}_{ui} \leftarrow \hat{\phi}_{ui} + W[u][j] \cdot H[i][j]$

  end for

  $e_{ui} = \phi_{ui} - \hat{\phi}_{ui}$

  for $j \leftarrow 1, \ldots, k$ **do**

    $W[u][j] \leftarrow W[u][j] + \alpha * (e_{ui} * H[i][j] - \lambda * W[u][j])$

    $H[i][j] \leftarrow H[i][j] + \alpha * (e_{ui} * W[u][j] - \lambda * H[i][j])$

  end for

end for

return $\{W, H\}$

MF with SGD – Example[2]

Let's have the following hyper-parameters:
$K = 2,\ \alpha = 0.1,\ \lambda = 0.15,\ iter = 150,\ \sigma^2 = 0.01$

$\Phi =$

1	4	5		3
5	1		5	2
4	1	2	5	
	3	4		4

Results are:

$W =$

1.1995242	1.1637173
1.8714619	-0.02266505
2.3267753	0.27602595
2.033842	0.539499

$H^T =$

1.6261001	1.1259034	2.131041	2.2285593	1.6074764
-0.40649664	0.7055319	1.0405376	0.39400166	0.49699315

Results[1] are:

$\hat{\Phi} =$

1.477499	2.171588	3.767126	3.131717	2.506566
3.052397	2.091094	3.964578	4.161733	2.997066
3.671365	2.814469	5.245668	5.294111	3.877419
3.087926	2.670543	4.895569	4.745101	3.537480

[1] Note, that these hyper-parameters are just picked up in an ad-hoc manner. One should search for the "best" hyper-parameter combinations using e.g. grid-search (a brute-force approach).

[2] Thanks to my colleague Thai-Nghe Nguyen for computing an example.

Biased MF

baseline estimate

- user-item bias

$$b_{ui} = \mu + b'_u + b''_i$$

  - μ – average rating across the whole $\mathcal{D}$
  - b', b'' – vectors of user and item biases, respectively

Biased MF

baseline estimate

- user-item bias

$$b_{ui} = \mu + b'_u + b''_i$$

  - μ – average rating across the whole $\mathcal{D}$
  - b', b'' – vectors of user and item biases, respectively

prediction

$$\hat{\phi}_{ui} = \mu + b'_u + b''_i + w_u h_i$$

Biased MF

baseline estimate

- user-item bias

$$b_{ui} = \mu + b'_u + b''_i$$

    - μ – average rating across the whole $\mathcal{D}$
    - b', b'' – vectors of user and item biases, respectively

prediction

$$\hat{\phi}_{ui} = \mu + b'_u + b''_i + w_u h_i$$

objective function to minimize

$$f(\phi, \hat{\phi}, \mathcal{D}) = \sum_{(u,i)\in\mathcal{D}} (\phi_{ui} - \mu - b'_u - b''_i - w_u h_i)^2 + \lambda(\|W\|^2 + \|H\|^2 + b'^2 + b''^2)$$

Biased MF with SGD

similar to unbiased MF

- initialize average and biases

$$\mu = \frac{\sum_{(u,i)\in\mathcal{D}}}{|\mathcal{D}|}$$

$$b' \leftarrow (\overline{\phi}_{u_1}, \ldots, \overline{\phi}_{u_n})$$

$$b'' \leftarrow (\overline{\phi}_{i_1}, \ldots, \overline{\phi}_{i_m})$$

Biased MF with SGD

similar to unbiased MF

- initialize average and biases

$$\mu = \frac{\sum_{(u,i)\in\mathcal{D}}}{|\mathcal{D}|}$$

$$b^{'} \leftarrow (\overline{\phi}_{u_1}, \ldots, \overline{\phi}_{u_n})$$

$$b^{''} \leftarrow (\overline{\phi}_{i_1}, \ldots, \overline{\phi}_{i_m})$$

- update average and biases

$$\mu \leftarrow \mu - \frac{\partial f}{\partial \mu}(u,i) = \mu + \alpha e_{ui}$$

$$b^{'} \leftarrow b^{'} - \frac{\partial f}{\partial b^{'}}(u,i) = b^{'} + \alpha(e_{ui} - \lambda b^{'})$$

$$b^{''} \leftarrow b^{''} - \frac{\partial f}{\partial b^{''}}(u,i) = b^{''} + \alpha(e_{ui} - \lambda b^{''})$$

MF – item recommendation

to predict a personalized **ranking score**[1] $\hat{\phi}_{ui}$

- how the item i is preferred to other items for the user u
- to find W and H such that $\hat{\Phi} = WH^T$

$$\hat{\phi}_{ui} = w_u h_i^T$$

[1] S. Rendle et al. (2009). BPR: Bayesian Personalized Ranking from Implicit Feedback. 25th Conference on Uncertainty in Artificial Intelligence.

MF – item recommendation

to predict a personalized **ranking score**[1] $\hat{\phi}_{ui}$

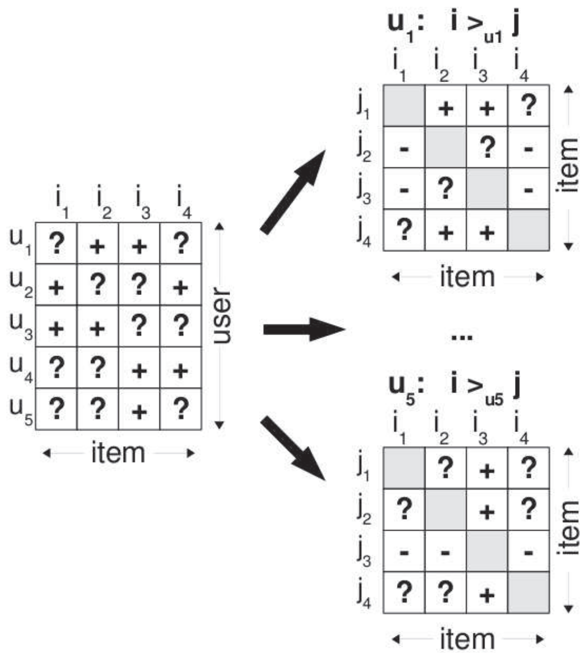
- how the item i is preferred to other items for the user u
- to find W and H such that $\hat{\Phi} = WH^T$

$$\hat{\phi}_{ui} = w_u h_i^T$$

problem: positive feedback only

- **pairwise ranking** data

$$\mathcal{D}_p = \{(u,i,j) \in \mathcal{D} | i \in \mathcal{I}_u \wedge j \in \mathcal{I} \setminus \mathcal{I}_u\}$$



[1] S. Rendle et al. (2009). BPR: Bayesian Personalized Ranking from Implicit Feedback. 25th Conference on Uncertainty in Artificial Intelligence.

MF – Bayesian Personalized Ranking (1/3)

Bayesian formulation of the problem

- $\succ$ – the unknown preference structure (ordering)
  - we use the derived pairwise ranking data $\mathcal{D}_p$
- Θ – parameters of an arbitrary prediction model
  - in case of MF, $\Theta = W \cup H$

$$p(\Theta| \succ) \propto p(\succ |\Theta)p(\Theta)$$

MF – Bayesian Personalized Ranking (1/3)

Bayesian formulation of the problem

- $\succ$ – the unknown preference structure (ordering)
  - we use the derived pairwise ranking data $\mathcal{D}_p$
- Θ – parameters of an arbitrary prediction model
  - in case of MF, $\Theta = W \cup H$

$$p(\Theta | \succ) \propto p(\succ | \Theta) p(\Theta)$$

prior probability

- assume independence of parameters
- assume, $\Theta \sim N(0, \frac{1}{\lambda} I)$

$$p(\Theta) = \prod_{\theta \in \Theta} \sqrt{\frac{\lambda}{2\pi}} \, e^{-\frac{1}{2}\lambda\theta^2}$$

MF – Bayesian Personalized Ranking (2/3)

likelihood

- assume users' feedbacks are independent
- assume, ordering of each pair is independent

$$p(\succ|\Theta) = \prod_{u \in \mathcal{U}} p(\succ_u|\Theta) = \prod_{(u,i,j) \in \mathcal{D}_p} p(i \succ_u j|\Theta)$$

MF – Bayesian Personalized Ranking (2/3)

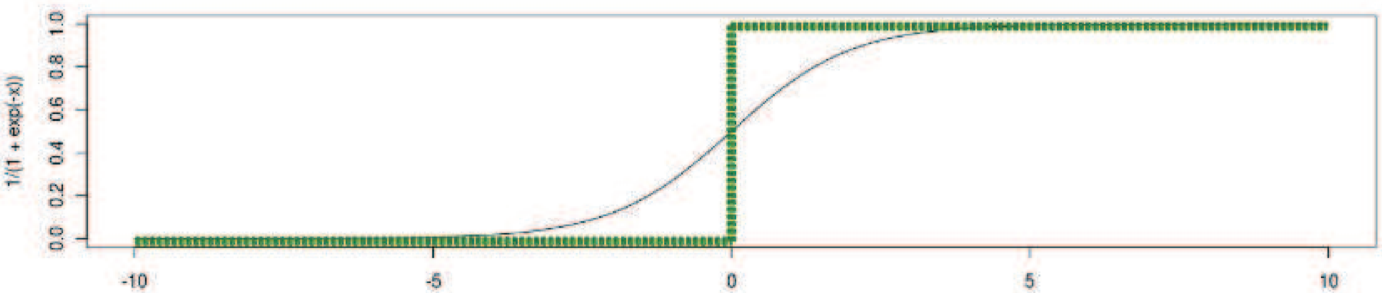
likelihood

- assume users' feedbacks are independent
- assume, ordering of each pair is independent

$$p(\succ|\Theta) = \prod_{u \in \mathcal{U}} p(\succ_u|\Theta) = \prod_{(u,i,j) \in \mathcal{D}_p} p(i \succ_u j|\Theta)$$

- using the ranking scores $\hat{\phi}$

$$p(i \succ_u j|\Theta) = p(\hat{\phi}_{ui} - \hat{\phi}_{uj} > 0) = \sigma(\hat{\phi}_{ui} - \hat{\phi}_{uj}) = \frac{1}{1 + e^{-(\hat{\phi}_{ui} - \hat{\phi}_{uj})}}$$



MF – Bayesian Personalized Ranking (3/3)

maximum **a posteriori estimation** of Θ

$$\arg\max_{\Theta} p(\Theta, \succ) =$$

MF – Bayesian Personalized Ranking (3/3)

maximum **a posteriori estimation** of Θ

$$\arg\max_{\Theta} p(\Theta, \succ) =$$

$$\arg\max_{\Theta} p(\succ |\Theta) p(\Theta) =$$

MF – Bayesian Personalized Ranking (3/3)

maximum **a posteriori estimation** of Θ

$$\arg\max_{\Theta} p(\Theta, \succ) =$$

$$\arg\max_{\Theta} p(\succ |\Theta) p(\Theta) =$$

$$\arg\max_{\Theta} ln\, p(\succ |\Theta) p(\Theta) =$$

MF – Bayesian Personalized Ranking (3/3)

maximum **a posteriori estimation** of Θ

$$\arg\max_{\Theta} p(\Theta, \succ) =$$

$$\arg\max_{\Theta} p(\succ |\Theta) p(\Theta) =$$

$$\arg\max_{\Theta} ln\, p(\succ |\Theta) p(\Theta) =$$

$$\arg\max_{\Theta} ln \prod_{(u,i,j)\in\mathcal{D}_p} \sigma(\hat{\phi}_{ui} - \hat{\phi}_{uj}) \sqrt{\frac{\lambda}{2\pi}}\, e^{-\frac{1}{2}\lambda\theta^2}$$

MF – Bayesian Personalized Ranking (3/3)

maximum **a posteriori estimation** of Θ

$$\arg\max_{\Theta} p(\Theta, \succ) =$$

$$\arg\max_{\Theta} p(\succ|\Theta)p(\Theta) =$$

$$\arg\max_{\Theta} ln\, p(\succ|\Theta)p(\Theta) =$$

$$\arg\max_{\Theta} ln \prod_{(u,i,j)\in\mathcal{D}_p} \sigma(\hat{\phi}_{ui} - \hat{\phi}_{uj}) \sqrt{\frac{\lambda}{2\pi}}\, e^{-\frac{1}{2}\lambda\theta^2}$$

$$\arg\max_{\Theta} \underbrace{\sum_{(u,i,j)\in\mathcal{D}_p} ln\, \sigma(\hat{\phi}_{ui} - \hat{\phi}_{uj}) - \lambda\|\Theta\|^2}_{BPR\text{–}OPT}$$

Finding parameters for BPR-OPT

Stochastic gradient ascent

$$\frac{\partial BPR-OPT}{\partial \Theta} \propto \sum_{(u,i,j)\in \mathcal{D}_p} \frac{e^{-(\hat{\phi}_{ui}-\hat{\phi}_{uj})}}{1+e^{-(\hat{\phi}_{ui}-\hat{\phi}_{uj})}} \frac{\partial}{\partial \Theta}(\hat{\phi}_{ui} - \hat{\phi}_{uj}) - \lambda \Theta$$

Finding parameters for BPR-OPT

Stochastic gradient ascent

$$\frac{\partial BPR-OPT}{\partial \Theta} \propto \sum_{(u,i,j)\in \mathcal{D}_p} \frac{e^{-(\hat{\phi}_{ui}-\hat{\phi}_{uj})}}{1+e^{-(\hat{\phi}_{ui}-\hat{\phi}_{uj})}} \frac{\partial}{\partial \Theta}(\hat{\phi}_{ui}-\hat{\phi}_{uj}) - \lambda\Theta$$

$$\frac{\partial}{\partial \theta}(\hat{\phi}_{ui}-\hat{\phi}_{uj}) = \begin{cases} (h_i - h_j) & if\ \theta = w_u \\ w_u & if\ \theta = h_i \\ -w_u & if\ \theta = h_j \\ 0 & else \end{cases}$$

Finding parameters for BPR-OPT

Stochastic gradient ascent

$$\frac{\partial BPR-OPT}{\partial \Theta} \propto \sum_{(u,i,j)\in\mathcal{D}_p} \frac{e^{-(\hat{\phi}_{ui}-\hat{\phi}_{uj})}}{1+e^{-(\hat{\phi}_{ui}-\hat{\phi}_{uj})}} \frac{\partial}{\partial \Theta}(\hat{\phi}_{ui}-\hat{\phi}_{uj}) - \lambda\Theta$$

$$\frac{\partial}{\partial \theta}(\hat{\phi}_{ui}-\hat{\phi}_{uj}) = \begin{cases} (h_i - h_j) & if\ \theta = w_u \\ w_u & if\ \theta = h_i \\ -w_u & if\ \theta = h_j \\ 0 & else \end{cases}$$

<u>LearnBPR</u>

input: $f_i, \alpha, \Sigma^2, stopping\ criteria$
initialize $\Theta \sim \mathcal{N}(0, \Sigma^2)$
repeat
  draw $(u, i, j) \in \mathcal{D}_p$ randomly
  $\Theta \leftarrow \Theta + \alpha \frac{\partial BPR-OPT}{\partial \Theta}(\Theta)$
until approximate maximum is reached
return Θ

BPR-OPT vs AUC

Area under the ROC curve (AUC)

- probability that the ranking of a randomly drawn pair is correct

$$AUC = \sum_{u \in \mathcal{U}} AUC(u) = \frac{1}{|\mathcal{U}|} \frac{1}{|\mathcal{I}_u|\, |\mathcal{I} \setminus \mathcal{I}_u|} \sum_{(u,i,j) \in \mathcal{D}_p} \delta(\hat{\phi}_{ui} \succ \hat{\phi}_{uj})$$

- $\delta(\hat{\phi}_{ui} \succ \hat{\phi}_{uj}) = 1$ if $\hat{\phi}_{ui} \succ \hat{\phi}_{uj}$, and 0, else