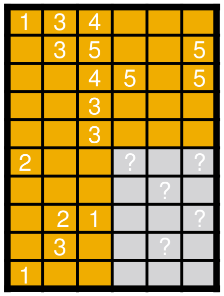
Information Storage and Retrieval

CSCE 670
Texas A&M University
Department of Computer Science & Engineering
Instructor: Prof. James Caverlee

Implicit Recommendation
27/29 March 2018

Some slides from Jure Leskovec, Julian McAuley

Recall: Matrix Factorization over Explicit Ratings



- **To solve overfitting we introduce regularization:**
  - Allow rich model where there are sufficient data
  - Shrink aggressively where data are scarce

$$\min_{P,Q} \underbrace{\sum_{training} (r_{xi} - q_i p_x)^2}_{\text{“error”}} + \underbrace{\left[\lambda_1 \sum_x \|p_x\|^2 + \lambda_2 \sum_i \|q_i\|^2 \right]}_{\text{“length”}}$$

λ_1, λ_2 … user set regularization parameters

Note: We do not care about the “raw” value of the objective function, but we care in P,Q that achieve the minimum of the objective

Putting it all together

$$r_{xi} = \mu + b_x + b_i + q_i \cdot p_x$$

Overall mean rating | Bias for user ***x*** | Bias for movie ***i*** | User-Movie interaction

- **Example:**
  - Mean rating: $\boldsymbol{\mu = 3.7}$
  - You are a critical reviewer: your ratings are 1 star lower than the mean: $\boldsymbol{b_x = -1}$
  - Star Wars gets a mean rating of *0.5* higher than average movie: $\boldsymbol{b_i = +\,0.5}$
  - Predicted rating for you on Star Wars:
    $\boldsymbol{= 3.7 - 1 + 0.5 = 3.2}$

Fitting the new model

- **Solve:**

$$\min_{Q,P} \sum_{(x,i)\in R} \left(r_{xi} - (\mu + b_x + b_i + q_i\, p_x)\right)^2$$

goodness of fit

$$+\left(\lambda_1 \sum_i \|q_i\|^2 + \lambda_2 \sum_x \|p_x\|^2 + \lambda_3 \sum_x \|b_x\|^2 + \lambda_4 \sum_i \|b_i\|^2\right)$$

regularization

λ is selected via grid-search on a validation set

- **Stochastic gradient decent to find parameters**
  - **Note:** Both biases $\boldsymbol{b_x}$, $\boldsymbol{b_i}$ as well as interactions $\boldsymbol{q_i}$, $\boldsymbol{p_x}$ are treated as parameters (we estimate them)

“Factorization meets the neighborhood”, Koren 2008

Solve this optimization:

$$min_{p,q,b} \sum_{u,i} (r_{ui} - \mu - b_u - b_i - p_u^T q_i)^2 + \lambda(||p_u||^2 + ||q_i||^2 + b_u^2 + b_i^2)$$

This is “SVD”

Then we can make our best guess:

$$\hat{r}_{ui} = b_{ui} + p_u^T q_i$$

What about implicit feedback?

“Factorization meets the neighborhood”, Koren 2008

Solve this optimization:

$$min_{p,q,b} \sum_{u,i} (r_{ui} - \mu - b_u - b_i - q_i^T (p_u + |N(u)|^{-1/2} \sum_{j \epsilon N(u)} y_i))^2 + \lambda(||p_u||^2 + ||q_i||^2 + b_u^2 + b_i^2 + \sum_{j \epsilon N(u)} ||y_i||^2)$$

Giving us the SVD++ model:

$$\hat{r}_{ui} = b_{ui} + q_i^T \left(p_u + |\mathrm{N}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{N}(u)} y_j \right)$$

$$\hat{r}_{ui} = b_{ui} + q_i^T \left(p_u + |\mathrm{N}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{N}(u)} y_j \right)$$

- p_u are the latent factors from **explicit ratings**
- **N(u)** is set of items that user u has implicit feedback on

- The fact that a user rates an item is in itself an indication of preference.
- In other words, chances that the user "likes" an item she has rated are higher than for a random not-rated item.

https://www.quora.com/Whats-the-difference-between-SVD-and-SVD++

Model	50 factors	100 factors	200 factors
SVD	0.9046	0.9025	0.9009
Asymmetric-SVD	0.9037	0.9013	0.9000
SVD++	0.8952	0.8924	0.8911

Table 1: Comparison of SVD-based models: prediction accuracy is measured by RMSE on the Netflix test set for varying number of factors (f). Asymmetric-SVD offers practical advantages over the known SVD model, while slightly improving accuracy. Best accuracy is achieved by SVD++, which directly incorporates implicit feedback into the SVD model.

But …

- Netflix prize data has lots of explicit ratings, so SVD++ can take advantage of both explicit and implicit (where implicit is directly derived from explicit ratings)
- But in practice there are many domains where we only have implicit feedback … no explicit ratings to build on.
- What do we do then?

Collaborative Filtering for Implicit Feedback Datasets

Yifan Hu
AT&T Labs – Research
Florham Park, NJ 07932

Yehuda Koren*
Yahoo! Research
Haifa 31905, Israel

Chris Volinsky
AT&T Labs – Research
Florham Park, NJ 07932

- Characteristics of **Implicit feedback**:
  - No negative feedback
    - If I haven't watched a show, does that mean I hate it? Or I just haven't heard about it?
    - Explicit ratings tell us a lot about what users like, but may misrepresent their overall interests
  - Implicit feedback is inherently noisy
    - Start a show, but not complete it?
  - Explicit feedback rating indicates **preference**, but implicit feedback score indicates **confidence**
    - High score for implicit feedback for a show I watch every week, but a low score for my favorite movie I watch only once
  - Evaluation is tricky — no RMSE to measure!

r_ui

- No longer an explicit rating (e.g., 1 or 4)
- Now it captures the user actions, e.g.:
  - 0.7 = user u watched 70% of a show
  - 2 = user watched a show twice

User preference

$$p_{ui} = \begin{cases} 1 & r_{ui} > 0 \\ 0 & r_{ui} = 0 \end{cases}$$

- If a user consumes an item, then there is some preference for it (regardless of rating, which recall we do not have)

- If a user does not consumer an item, then no preference

Confidence in Observing p

$$c_{ui} = 1 + \alpha r_{ui}$$

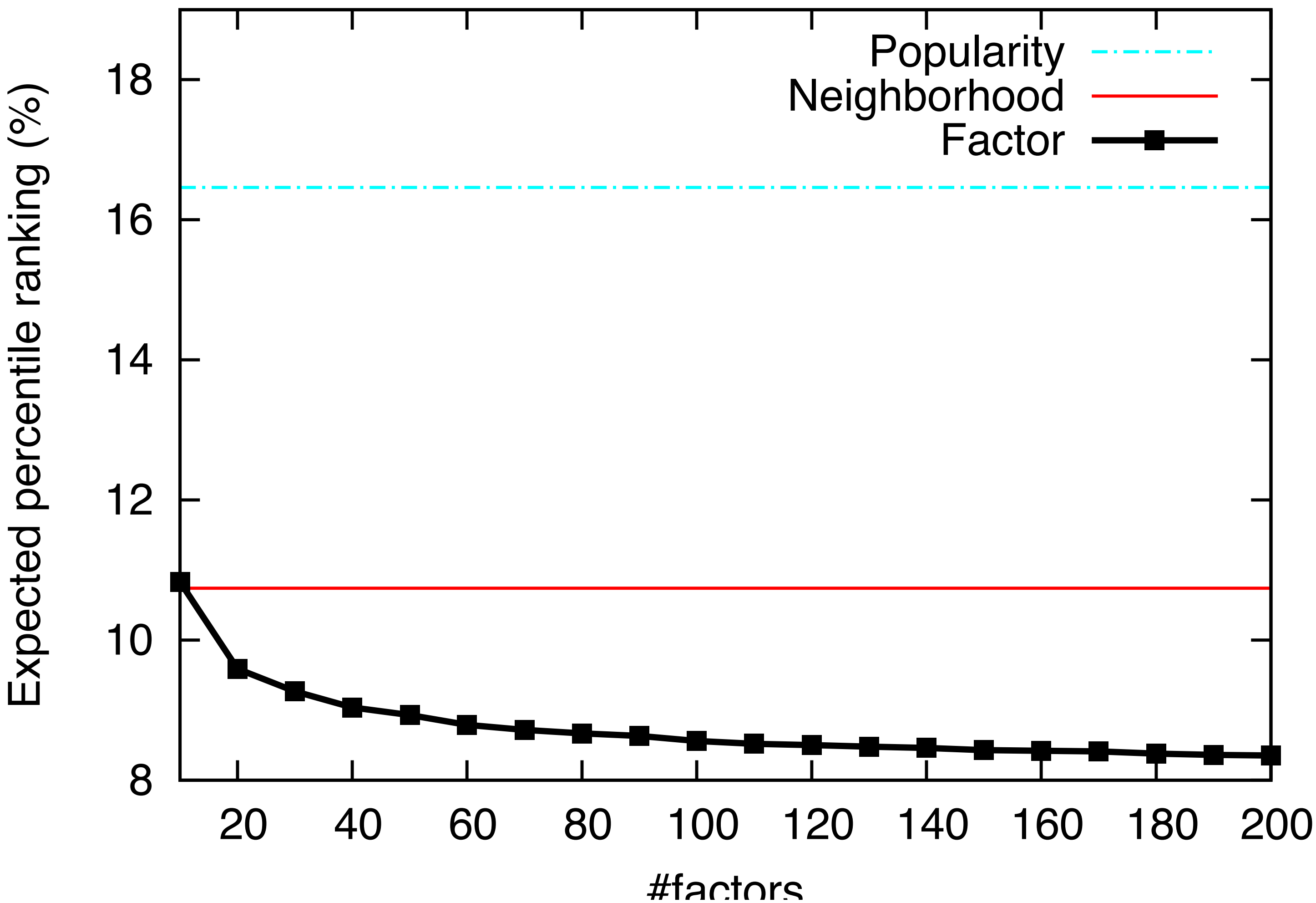
- Every item has some confidence (1) even if the user has never consumed the item
- As a user consumes an item more (r), then the confidence goes up that they prefer the item

The model:

$$\min_{x_\star, y_\star} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right)$$

Experiments

- TV shows, r = how many times a user watched a program



BPR: Bayesian Personalized Ranking

BPR: Bayesian Personalized Ranking from Implicit Feedback

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner and Lars Schmidt-Thieme
{srendle, freudenthaler, gantner, schmidt-thieme}@ismll.de
Machine Learning Lab, University of Hildesheim
Marienburger Platz 22, 31141 Hildesheim, Germany

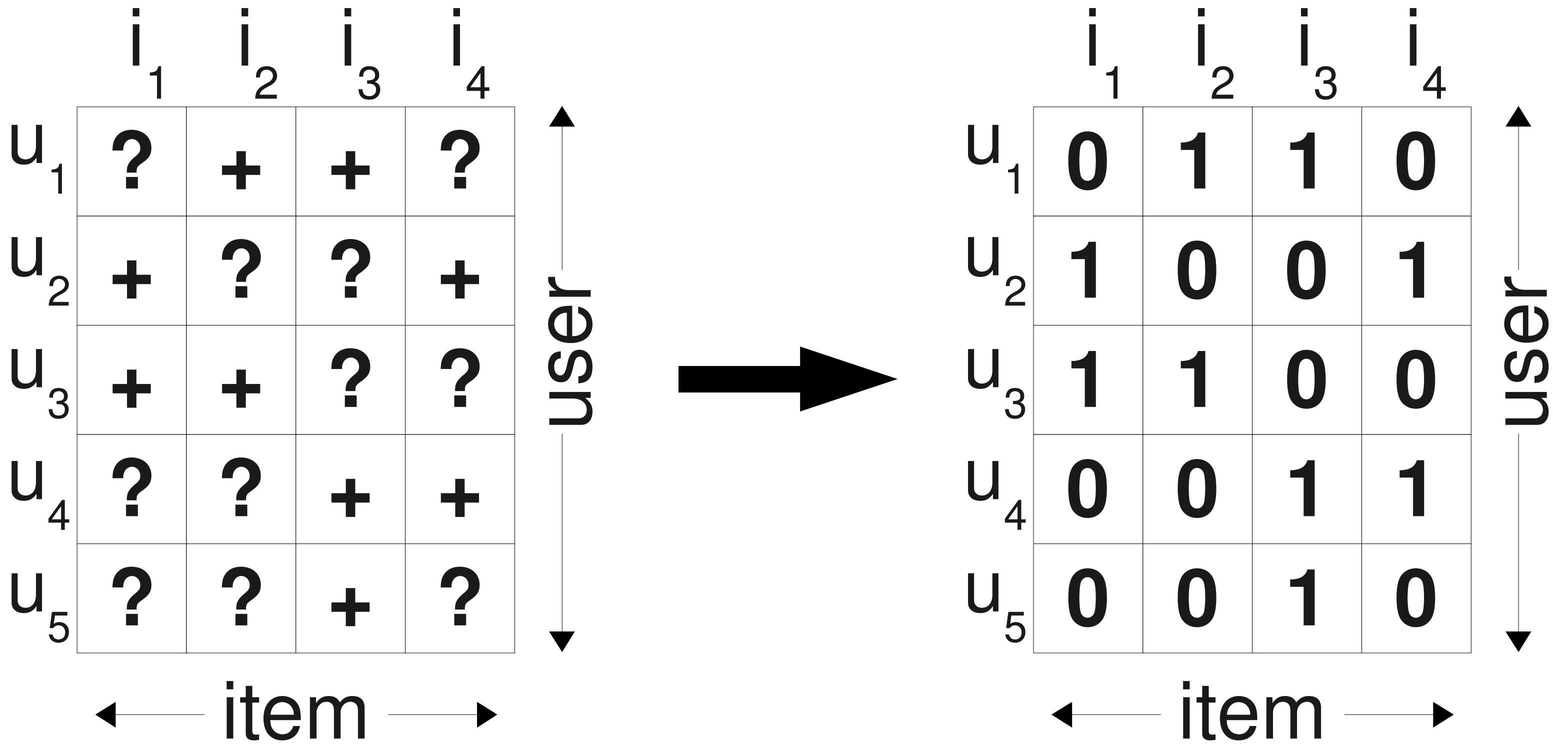


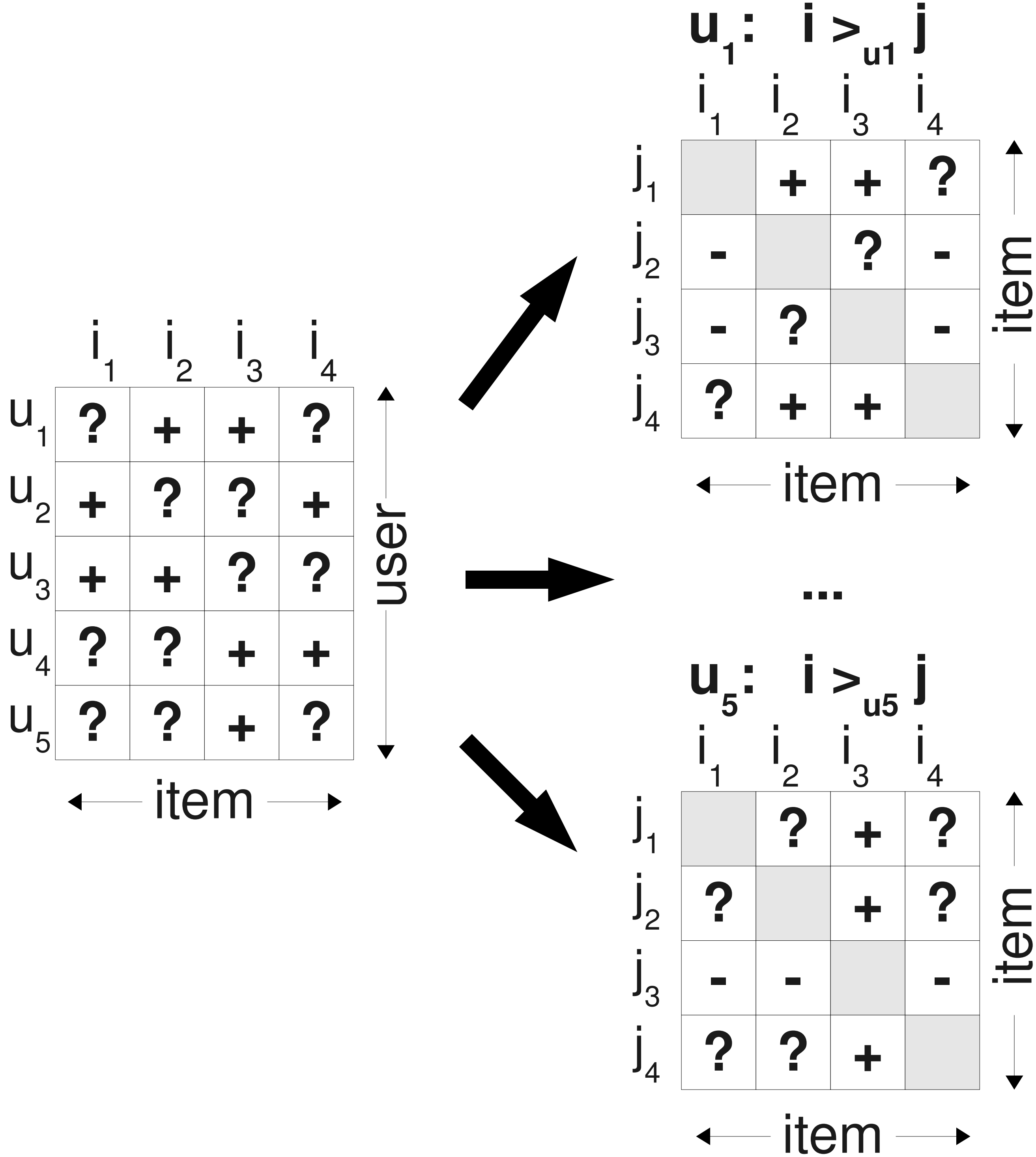
Figure 1: On the left side, the observed data S is shown. Learning directly from S is not feasible as only positive feedback is observed. Usually negative data is generated by filling the matrix with 0 values.

Goal of BPR

- For each user, estimate a personalized ranking function:

$$i >_u j$$

- x(u,i,j) —> positive if i is preferred
- x(u,i,j) —> negative if j is preferred



u1: i >u1 j
u5: i >u5 j
i1 i2 i3 i4
j1 j2 j3 j4
u1 u2 u3 u4 u5
user
item
...

BPR: Define AUC for user u

$$\text{AUC}(u) := \frac{1}{|I_u^+|\,|I \setminus I_u^+|} \sum_{i \in I_u^+} \sum_{j \in |I \setminus I_u^+|} \delta(\hat{x}_{uij} > 0)$$

$$\left(\text{AUC} := \frac{1}{|U|} \sum_{u \in U} AUC(u) \right)$$

scoring function that compares an item *i* to an item *j* for a user *u*

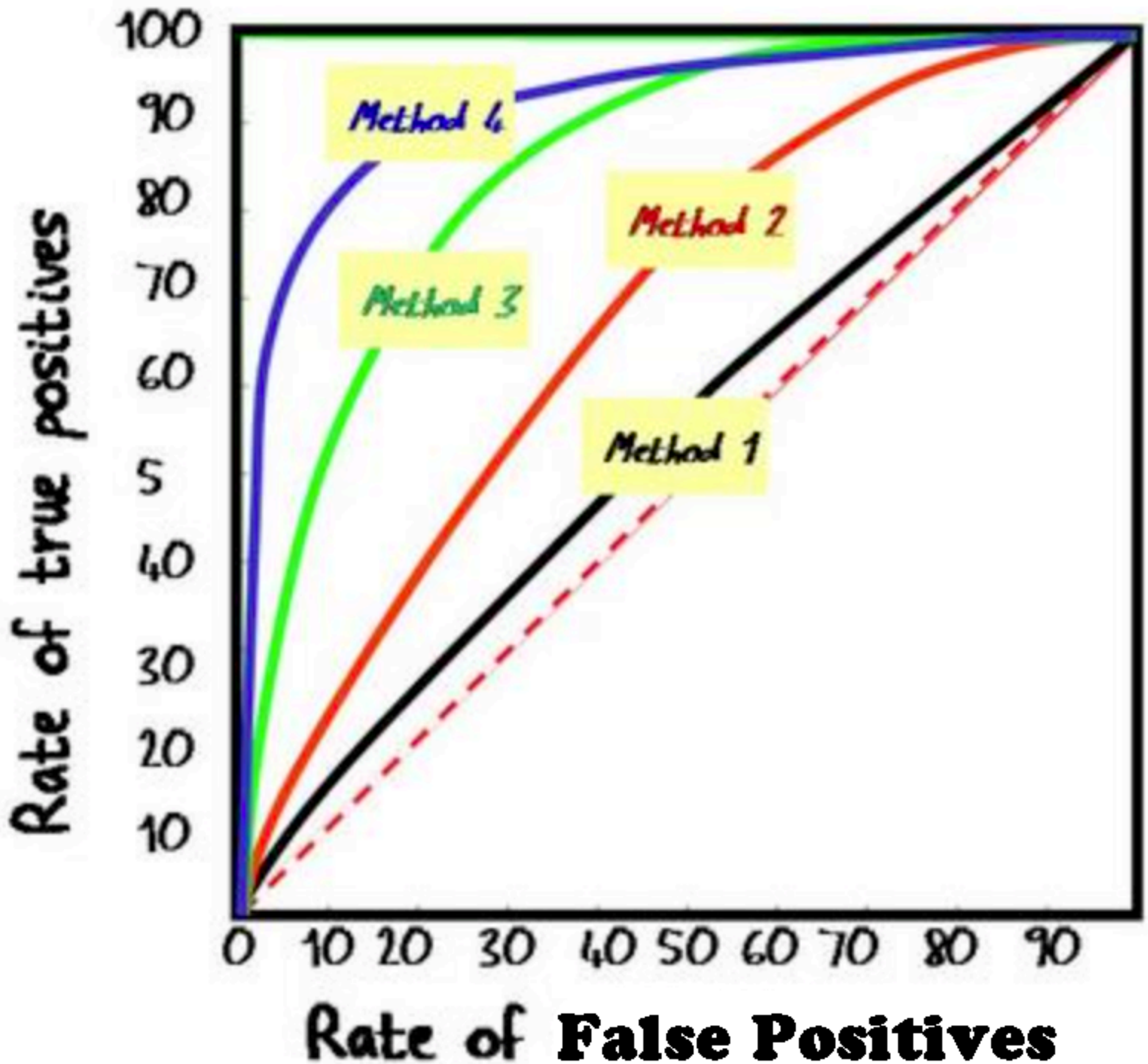
- The AUC essentially counts how many times the model correctly identifies that u prefers the item they bought (positive feedback) over the item they did not

BPR: Define AUC for user u

$$\text{AUC}(u) := \frac{1}{|I_u^+|\,|I \setminus I_u^+|} \sum_{i \in I_u^+} \sum_{j \in |I \setminus I_u^+|} \delta(\hat{x}_{uij} > 0)$$

- AUC = 1.0
    - We **always** guess correctly
- AUC = 0.5
    - We guess as well as a **random pick**

ROC CURVE EXAMPLES

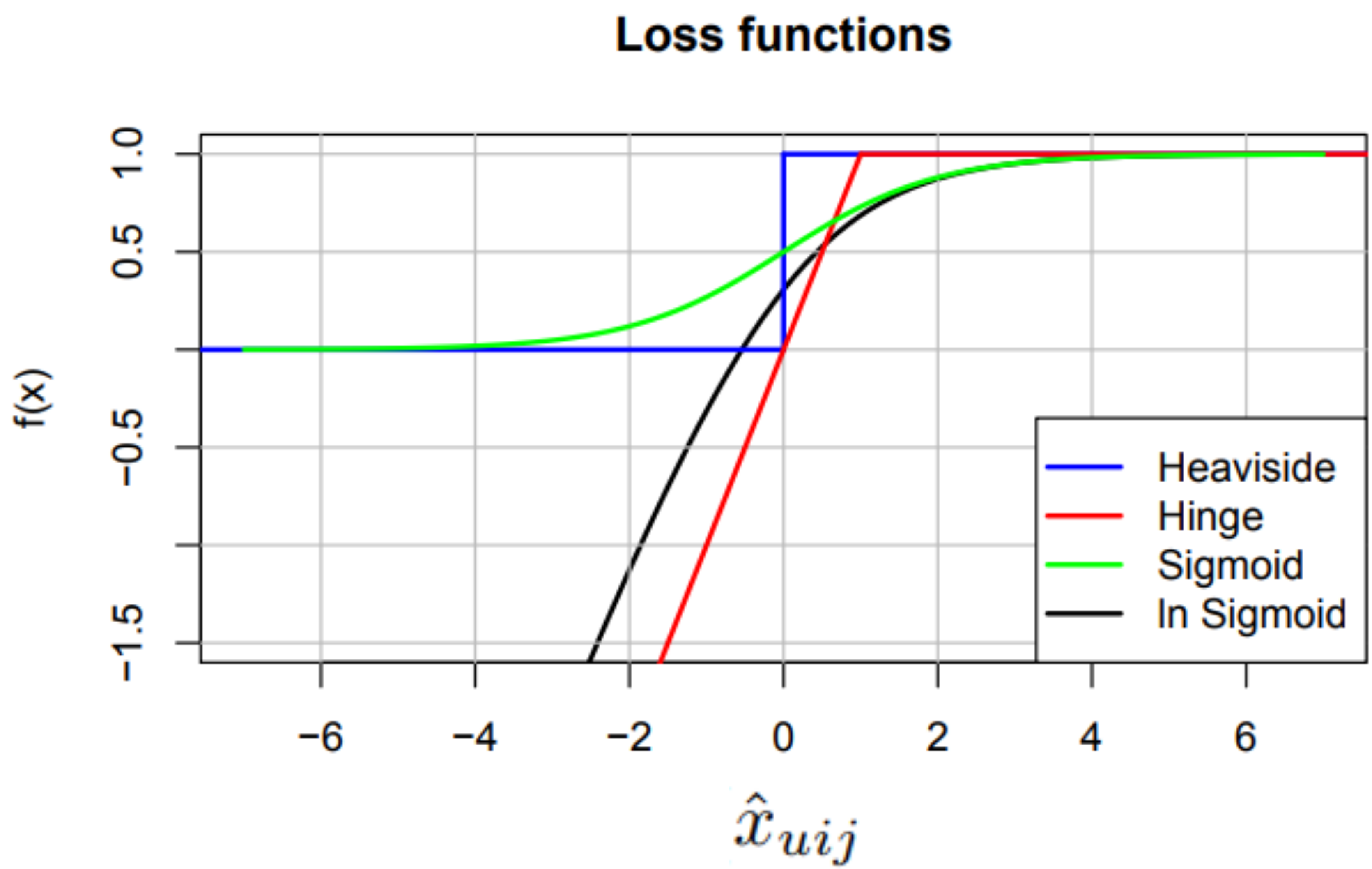


- The best classification has the largest area under the curve.
- Too sensitive to errors in the "gold standard" classification.

BPR

Summary: Goal is to count how many times we identified ***i*** as being more preferable than ***j*** for a user ***u***

$$\delta(\hat{x}_{uij} > 0)$$



Idea: Replace the counting function $\delta(\hat{x}_{uij} > 0)$ by a smooth function

$$\sigma(\hat{x}_{uij})$$

$\hat{x}_{uij}$ is any function that compares the compatibility of *i* and *j* for a user *u*

e.g. could be based on matrix factorization:

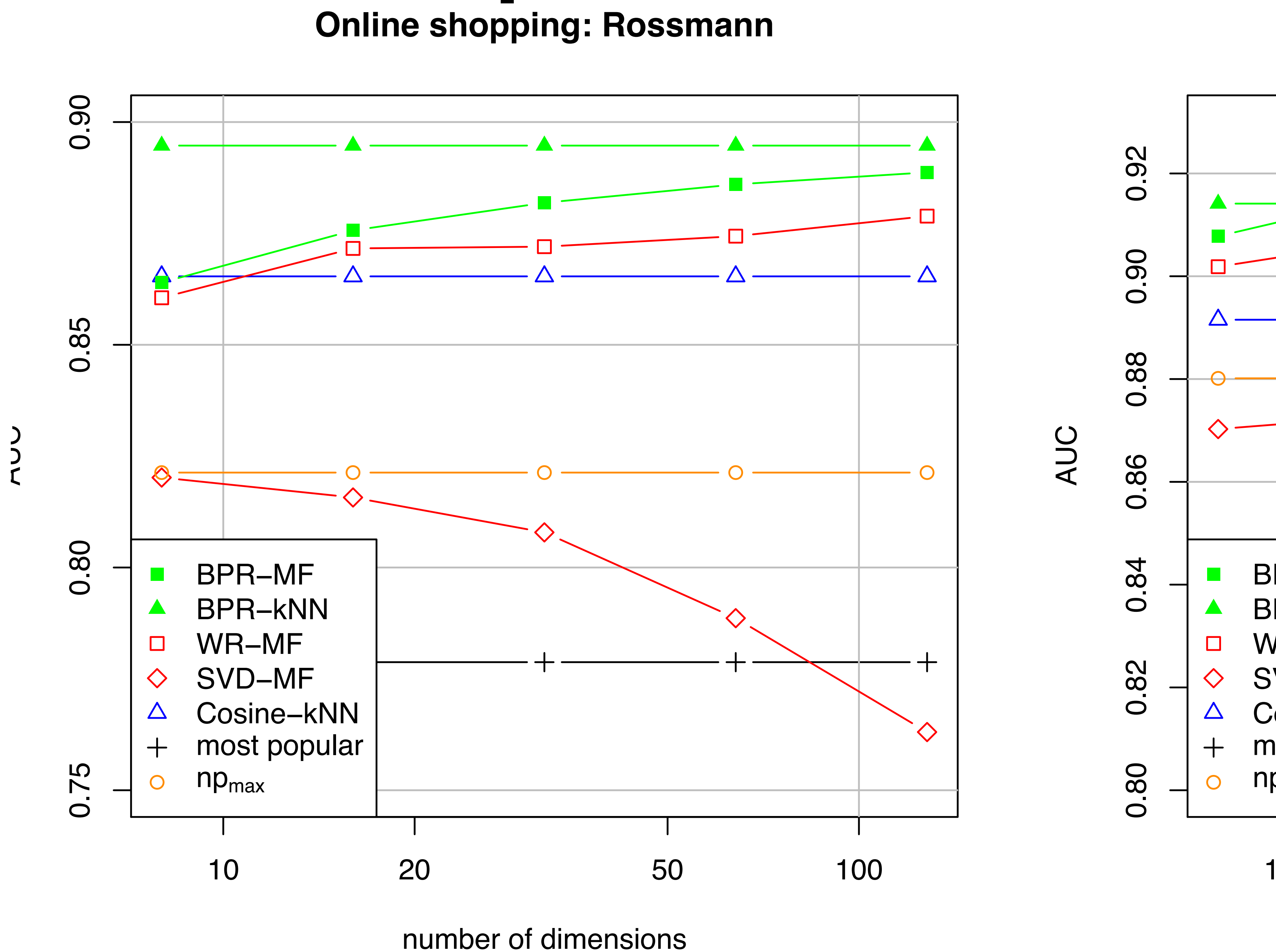
$$x_{uij} = \gamma_u * \gamma_i - \gamma_u * \gamma_j$$

$$
\begin{aligned}
\textrm{BPR-}\textsc{Opt} :=& \ln p(\Theta| >_u) \\
=& \ln p(>_u |\Theta)\, p(\Theta) \\
=& \ln \prod_{(u,i,j)\in D_S} \sigma(\hat{x}_{uij})\, p(\Theta) \\
=& \sum_{(u,i,j)\in D_S} \ln \sigma(\hat{x}_{uij}) + \ln p(\Theta) \\
=& \sum_{(u,i,j)\in D_S} \ln \sigma(\hat{x}_{uij}) - \lambda_\Theta ||\Theta||^2
\end{aligned}
$$

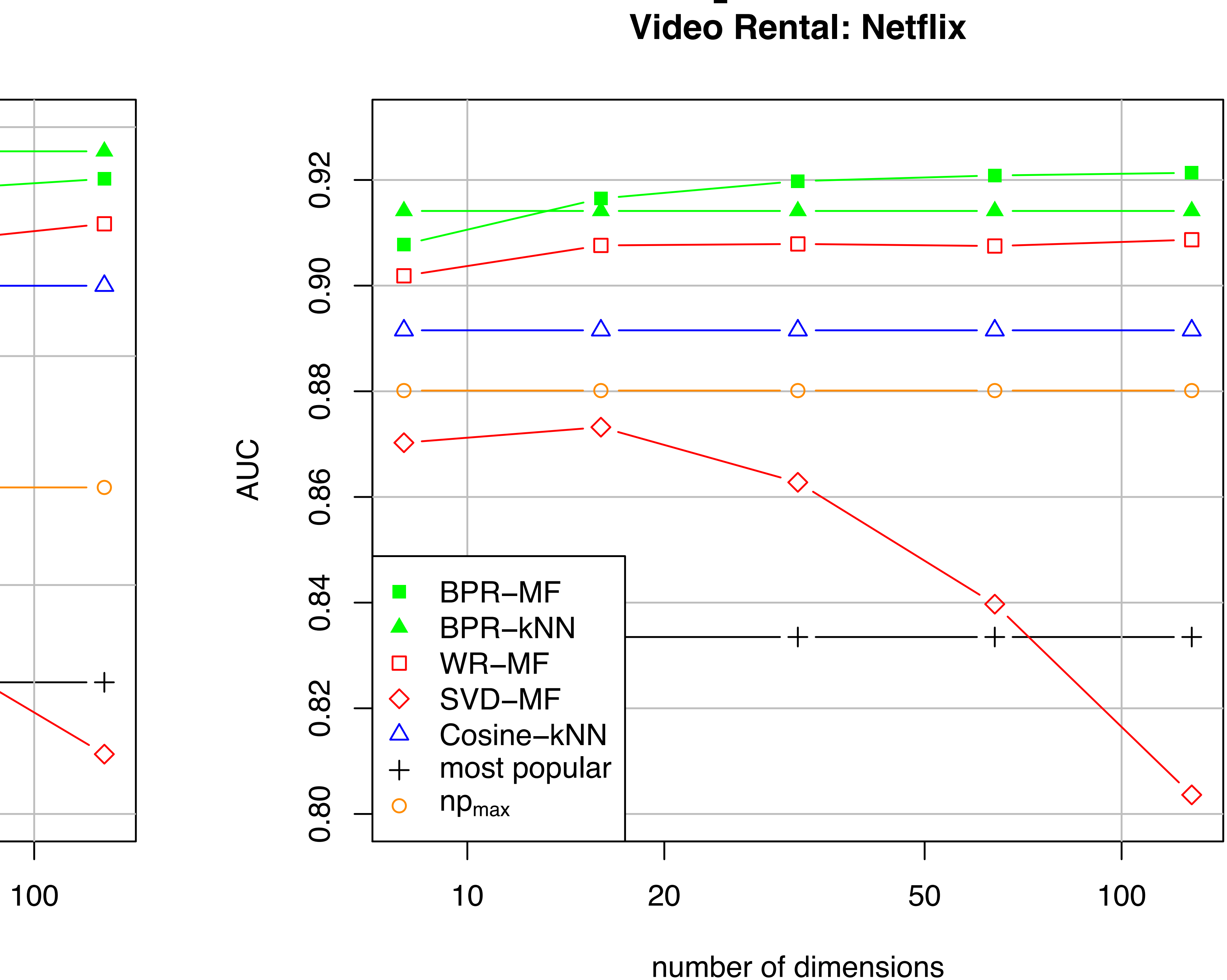
$$\hat{x}_{uij} := \hat{x}_{ui} - \hat{x}_{uj}$$

$$\hat{x}_{ui} = \langle w_u, h_i \rangle = \sum_{f=1}^{k} w_{uf} \cdot h_{if}$$

BPR: Experiments



BPR: Experiments



BPR: Summary

- Given a "one-class" prediction task (like purchase prediction) we might want to optimize a ranking function rather than trying to factorize a matrix directly
- The AUC is one such measure that counts among a users *u*, items they consumed *i*, and items they did not consume, *j*, **how often we correctly guessed that *i* was preferred by *u***
- We can optimize this approximately by maximizing $\sigma(\hat{x}_{uij})$ where $\hat{x}_{uij} = \gamma_u \cdot \gamma_i - \gamma_u \cdot \gamma_j$