BPR: Bayesian Personalized Ranking from Implicit Feedback

By Steen Rendle, Christoph Freudenthaler, Zeno Gantner and Lars Schmidt-Thieme

Hoyoung Choi 2022.01.11

INDEX

Introduction

Recommender system with Implicit feedback

BPR-OPT (Bayesian Personalized Ranking Optimization)

- Generic optimization criterion for optimal personalized ranking
- Derived from maximum posterior estimator
- Analogies to maximizing AUC value
- LearnBPR and application to MF, k-NN
- Performance for personalized ranking task

Personalized Ranking

Formalization

U, I: Set of Users, Items S: Implicit feedback ($S \subseteq U \times I$)

Personalized Total Ranking $>_u \subset I^2$

- Totality: $i \neq j \Rightarrow i >_u j \lor j >_u i$
- Antisymmetry: $i >_u j \land j >_u i \Rightarrow i = j$
- Transitivity: $i >_u j \land j >_u k \Rightarrow i >_u k$

I_u^+: Set of items that the user u made an implicit feedback, $\{i \in I : (u, i) \in S\}$

U_i^+: Set of users that made an implicit feedback to the item i, $\{u \in U : (u, i) \in S\}$

Personalized Ranking

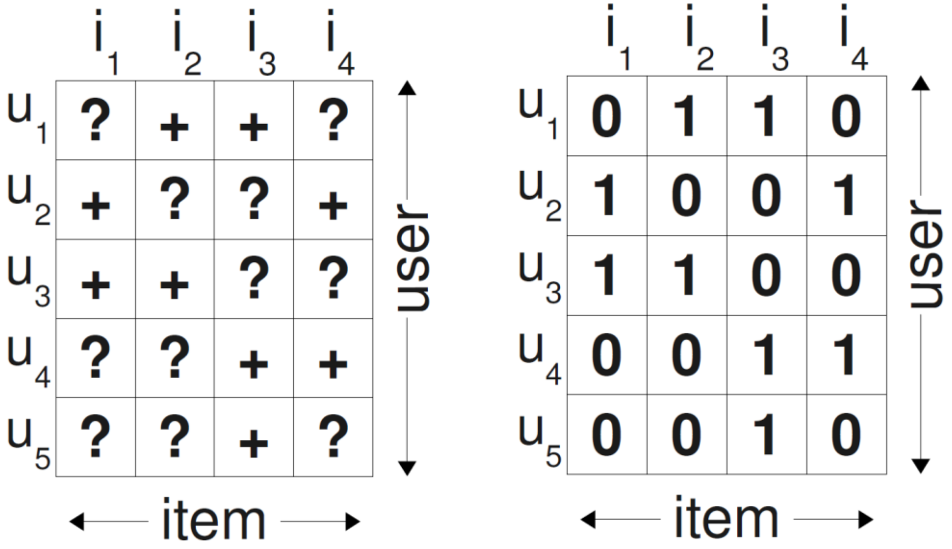
Problem Setting

Implicit feedback: Positive values + Unknown values

- Unknown values: Negative values + Missing values

Handling Implicit feedback

- Ignoring missing values?
- Predict personalized score $\hat{x}_{ui}$
- Labeling unknown values as negative
    - Is non-observed value well-predicted?
        - If it is, other reasons (Regularization)



Personalized Ranking

Problem Setting

Optimize ranking of item pairs instead of score of items

Using item pairs as training data

Assumptions:

- User prefers observed items over non-observed items
- We cannot infer preference between observed items
- We cannot infer preference between non-observed items

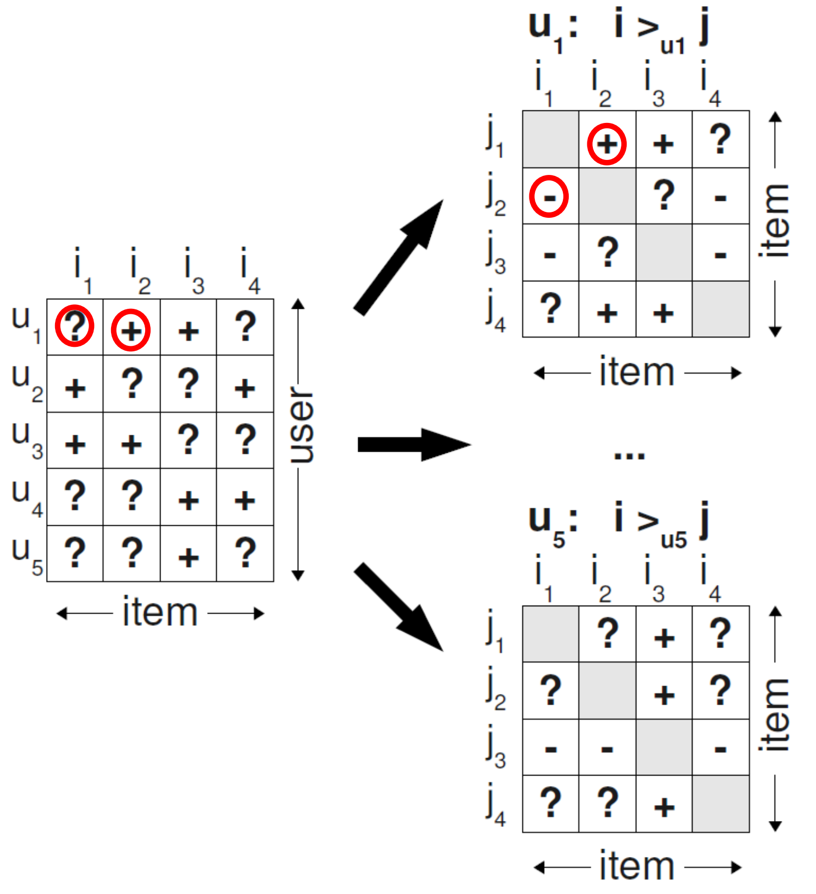
Personalized Ranking

Problem Setting

Create training data D_s as

$$D_s := \{(u,i,j) | i \in I_u^+ \wedge j \in I \setminus I_u^+\}$$

- User u prefers i over j



Advantages of pair approach

- Data contain positive, negative, missing values
- Observed subset is used as training data

BPR

Optimization Criterion

Find the parameter vector Θ that maximize the posterior probability $p(\Theta| >_u)$

$$p(\Theta| >_u) \propto p(>_u |\Theta)p(\Theta)$$

Assumptions

- All users act independently of each other
- Ordering of a pair is independent of all other pairs

BPR

Likelihood

Totality: Bernoulli distribution

Independence assumption: i.i.d

$$\prod_{u\in U} p(>_u|\Theta) = \prod_{(u,i,j)\in U\times I\times I} p(i >_u j|\Theta)^{\delta((u,i,j)\in D_S)} \cdot (1 - p(i >_u j|\Theta))^{\delta((u,i,j)\notin D_S)}$$

Antisymmetry

$$\prod_{u\in U} p(>_u|\Theta) = \prod_{(u,i,j)\in D_S} p(i >_u j|\Theta)$$

Define individual probability with real-value function $\hat{x}_{uij}$

$$p(i >_u j|\Theta) := \sigma(\hat{x}_{uij}(\Theta))$$

BPR

Prior

Normal prior with zero mean and variance-covariance matrix Σ_Θ

$$p(\Theta) \sim N(0, \Sigma_\Theta)$$

Set $\Sigma_\Theta = \lambda_\Theta I$ to reduce unknown hyperparameters

- λ_Θ is regularization parameter

BPR

Optimization Criterion

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

BPR

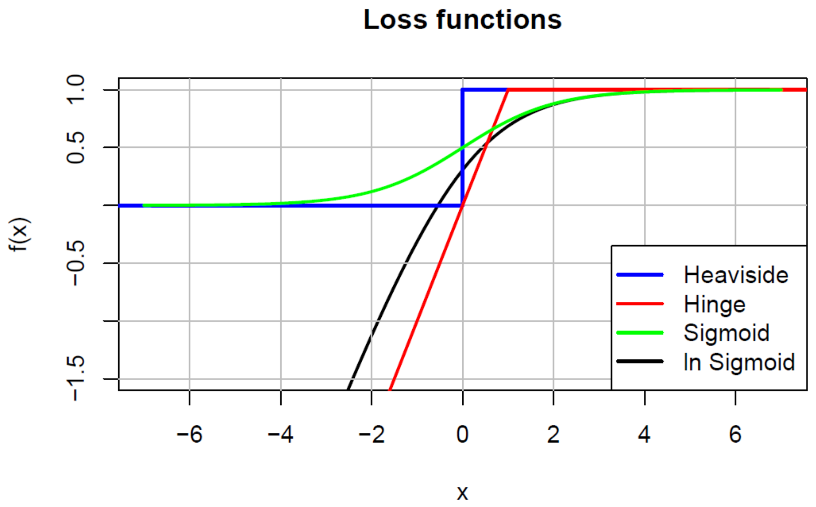
Analogies to AUC Optimization

BPR-OPT is similar to AUC Optimization

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

$$\mathrm{AUC} = \sum_{(u,i,j) \in D_s} z_u \delta(\hat{x}_{uij} > 0)$$

$$\mathrm{BPR-OPT} = \sum_{(u,i,j) \in D_s} \ln \sigma\left(\hat{x}_{uij}\right) - \lambda_\Theta \left|\left|\Theta\right|\right|^2$$



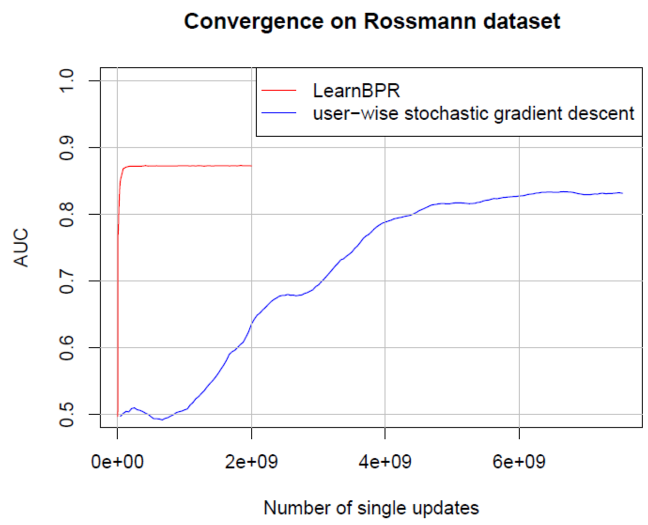
Only difference in the loss function

- Substitute non-differentiable Heaviside function with differentiable function log-sigmoid

BPR

BPR Learning Algorithm

Optimization criterion for BPR is differentiable: Gradient Descent



Full gradient descent?

- Computing full gradient is not feasible
- Skewness in training pairs (most of items are not observed)

Stochastic gradient descent?

- Consecutive updates on same user-item pairs

SGD with bootstrapping

- Triples are selected randomly
- number of same user-item pairs would decrease

procedure $\text{LEARNBPR}(D_S, \Theta)$
  initialize Θ
  repeat
    draw (u, i, j) from D_S
    $\Theta \leftarrow \Theta + \alpha \left(\frac{e^{-\hat{x}_{uij}}}{1+e^{-\hat{x}_{uij}}} \cdot \frac{\partial}{\partial \Theta} \hat{x}_{uij} + \lambda_\Theta \cdot \Theta \right)$
  until convergence
  return $\hat{\Theta}$
end procedure

Learning Models with BPR

Basic Methodology

Decomposition of estimator

$$\hat{x}_{uij} \coloneqq \hat{x}_{ui} - \hat{x}_{uj}$$

Application is available for any CF model that predicts $\hat{x}_{ui}$

Use the same model with different optimization criterion

- BPR-OPT is optimal for ranking system: better results for ranking tasks

Learning Models with BPR

Matrix Factorization with BPR

Estimate $X = U \times I$ with two low-rank matrices: $\hat{X} = WH^T$ (W: $|U| \times k, H: |I| \times k$)

$$\hat{x}_{ui} = \sum_{f=1}^{k} w_{uf} \cdot h_{if}$$

$$\hat{x}_{uij} = \sum_{f=1}^{k} w_{uf} \cdot (h_{if} - h_{jf})$$

$$\frac{\partial}{\partial\theta}\hat{x}_{uij} = \begin{cases} (h_{if} - h_{jf}) & if\ \theta = w_{uf} \\ w_{uf} & if\ \theta = h_{if} \\ -w_{uf} & if\ \theta = h_{jf} \\ 0 & else \end{cases}$$

3 regularization constraints: one for user features, 2 for item features

Learning Models with BPR

k-Nearest-Neighbor with BPR

Item-based kNN

Scores of items are calculated by the similarities with observed items

$$\hat{x}_{ui} = \sum_{l \in I_u^+ \wedge l \neq i} c_{il}$$

$$\frac{\partial}{\partial\theta}\hat{x}_{uij} = \begin{cases} +1 & if\ \theta \in \{c_{il}, c_{li}\} \wedge l \in I_u^+ \wedge l \neq i \\ -1 & if\ \theta \in \{c_{jl}, c_{lj}\} \wedge l \in I_u^+ \wedge l \neq j \\ 0 & else \end{cases}$$

2 regularization constraints: one for updating c_{il}, one for updating c_{jl}

Relations to other methods

Weighted Regularized Matrix Factorization

Introduce weights for user-item pairs to prevent overfitting and increase impact of positive feedback

Considered confidence and preference of implicit feedback

$$\sum_{u \in U} \sum_{i \in I} c_{ui}(\langle w_u, h_i \rangle - p_{ui})^2 + \lambda ||W||_f^2 + \lambda ||H||_f^2$$

Weakness of WR-MF

- Instance level optimization
- Least-square less suitable for ranking task

Strength of WR-MF

- Learned in $O(\text{iter } (|S|k^2 + k^3 \; (|I| + |U|)))$
- BPR-OPT converge faster

Relations to other methods

Maximum Margin Matrix Factorization

Designed for explicit feedback, but can be applied to implicit feedback

$$\sum_{(u,i,j)\in D_S} \max(0, 1 - \langle w_u, h_i - h_j \rangle) + \lambda ||W||_f^2 + \lambda ||H||_f^2$$

Similar to BPR-OPT

Difference in loss function: Can only be applied to MF

Weakness of MMMF

- Designed for sparse dataset: Inappropriate for dense dataset of implicit feedback

Evaluation

Models and Datasets

Models

- MF: SVD-MF, WR-MF, BPR-MF
- kNN: Cosine-kNN, BPR-kNN
- Most popular (User-independent)
- Theoretical upper bound for non-personalized ranking

Datasets

- Rossmann: Buying history of 10,000 users, 4,000 items, 426,612 purchases
    - Predict a personalized list of items the user wants to buy next
- Netflix: Rating behavior of 10,000 users, 5,000 items, 565,738 rating actions
    - Predict a personalized list of items the user is more likely to rate

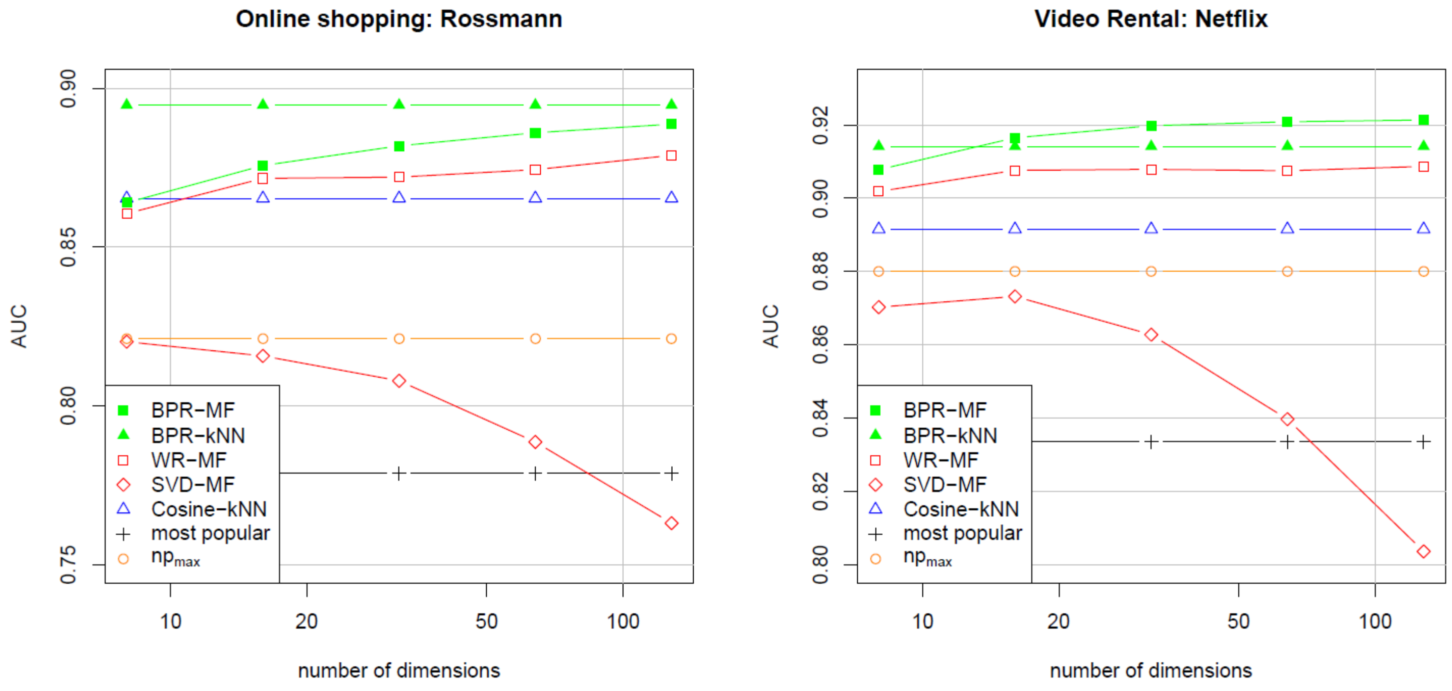
Evaluation

Evaluation Methodology and Results

Evaluation criterion: Average AUC value

Results

- BPR outperform other models
- Importance of optimization methods
- Personalized vs. Non-personalized



Conclusion

Presented a generic optimization criterion and learning algorithm

- BPR-OPT
- LearnBPR

Applied BPR to MF, kNN

Models learned by BPR outperform previous works

- Production quality not only depends on model, but also on the optimization criterion

Thank you