CPR: Comprehensive Personalized Ranking Using One-Bit Comparison Data

Aria Ameri
Arindam Bose
Mojtaba Soltanalian

June 04, 2019

UIC WaveOPT Lab

IEEE Data Science Workshop 2019

Table of contents

Motivation

Recommendation system

- Why do we need them?
  - To recommend relevant stuff to other people
  - To take informative decisions
- Who need them?
  - Pretty much everyone

Overview

Some context

- Earlier in the days of Netflix prize, most of the recommender systems were based on explicit data.
- *Implicit feedback data* has become more popular in both academia and industries to build robust recommender systems.

Features of implicit data

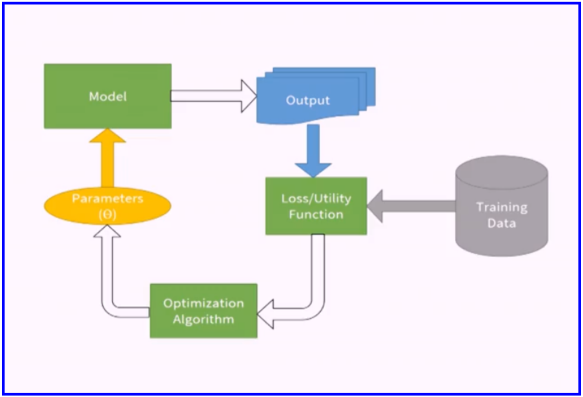
- No Negative feedback
- Inherently noisy
- Preference vs. confidence

Latent factor models

- An alternative approach to neighborhood models
- Examples: Matrix factorization, Latent semantic models, Latent dirichlet allocation

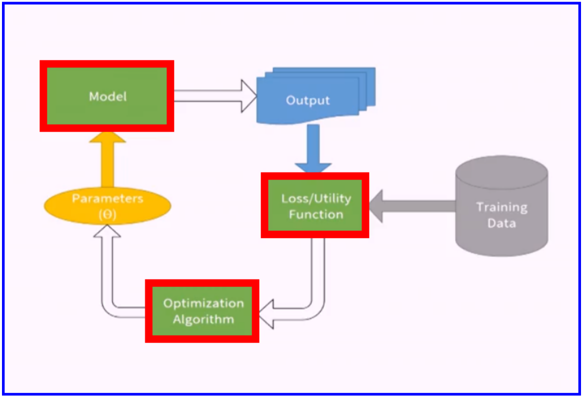
Learning recommendation systems

- The matrix factorization can be reformulated as an optimization problem with loss function and constraints
- We choose the best recommender out of a family of recommenders during the optimization process



Learning recommendation systems

- The matrix factorization can be reformulated as an optimization problem with loss function and constraints
- We choose the best recommender out of a family of recommenders during the optimization process



Learning recommendation system blocks

Model

- Can be a matrix factorization model or a linear regression model
- Has some parameters like matrices in a matrix decomposition that we would be optimizing during the process

Utility function or loss function

- θ: Parameters of our recommendation model like user and item matrices in matrix factorization
- $g(\theta)$: Loss function that we are trying to minimize

$$\arg\min_{\theta} g(\theta)$$

Optimization algorithm

- Choose anything that fits the purpose (e.g. Alternating least sqaures (ALS))

A short diversion to Matrix factorization using ALS

- What is Alternating least sqaures[1]?
- Loss function

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

The good news[2]

Inspite of the large sparsity in the dataset, the recommender system gave an AUC value of $\sim$90%

However,

The algorithm performs better in terms of finding similar items, but not very effective in recommending items to a particular user

[1] Y. Hu et al. *Collaborative filtering for implicit feedback*, 2008

[2] A. Naraparreddy, https://bit.ly/2QCEn8V, 2019

What questions ALS does and does not answer?

- ALS reduces the impact of missing data using confidence and preference metrics
- It optimizes to predict if an item is selected by a user or not
- It does not directly optimize its model parameters for ranking
- Bayesian Personalized Ranking[3]optimization criterion involves pairs of items(the user-specific order of two items) to come up with more personalized rankings for each user

[3] S Rendle et al. *BPR: Bayesian Personalized Ranking from Implicit Feedback*, 2012

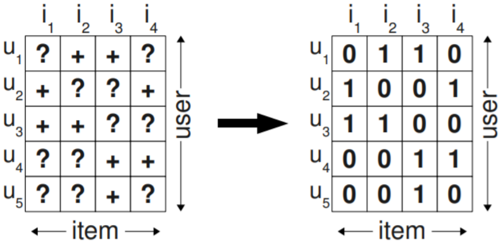
Bayesian Personalized Ranking

"First of all, it is obvious that this optimization is on instance level (one item) instead of pair level (two items) as BPR. Apart from this, their optimization is a leastsquare which is known to correspond to the MLE for normally distributed random variables. However, the task of item prediction is actually not a regression (quantitative), but a classification (qualitative) one, so the logistic optimization is more appropriate."

— Steffen Rendle et al. BPR: Bayesian Personalized Ranking from Implicit Feedback

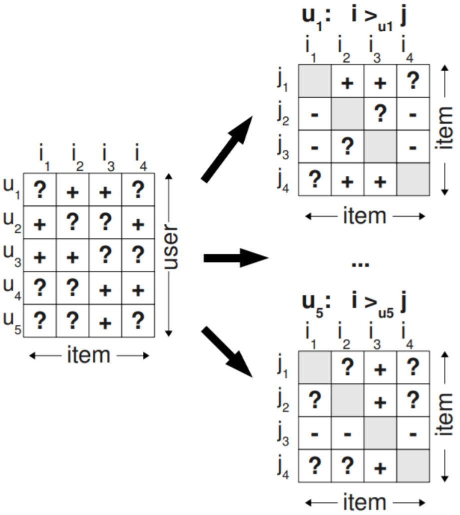
Bayesian personalized ranking approach

- The primary task of personalized ranking is to provide a user with a ranked list of items
- General implicit data representation:
  - U: set of all users
  - I: set of all items



Bayesian personalized ranking

The dataset would be considered as $(u, i, j) \in D_S$



Like in any Bayesian approach, they have a likelihood function , prior probability and posterior probability in this approach.

> *"The Bayesian formulation of finding the correct personalized ranking for all items $i \in I$ is to maximize the posterior probability $\mathbb{P}\{\Theta| >_u\}$ where Θ represents the parameter vector of an arbitrary model class (e.g. matrix factorization)."*

Our contribution

CPR: Comprehensive Personalized Ranking

- We present a similar yet deeper Bayesian framework to address the recommendation problem, which not only utilizes the one-bit item-item preference of a user, but also exploits the implicit inclination of different users towards an item.

$$(u, k, l) \in D_u$$
$$(m, i, j) \in D_m$$

- We provide a stochastic-gradient based approach to learn the system parameters.

Problem Formulation

Sets

U: the set of all users
I: the set of all items
Ω: the internal system parameter (e.g. a user/item latent matrix)

Notations

$i >_u j \quad \subset I^2$: the user u prefers item i over item j
$k >_m l \subset U^2$: user k is more likely to buy item m than user l

Identities

$$\begin{aligned} \textit{totality} &: i \neq_u j \Rightarrow i >_u j \lor j >_u i : \forall\ i, j \in I \\ \textit{antisymmetry} &: i >_u j \land j >_u i \Rightarrow i =_u j : \forall\ i, j \in I \\ \textit{transitivity} &: i >_u j \land j >_u k \Rightarrow i >_u k : \forall\ i, j, k \in I \end{aligned}$$

[same idea goes for observations $k >_m l \subset U^2$]

The Posterior, the Likelihood and the Prior functions

- The problem we are interested in:

$$\mathbb{P}\{\Omega| >_u, >_m\} = \alpha \cdot \mathbb{P}\{>_u, >_m |\Omega\} \, \mathbb{P}\{\Omega\}$$

The Posterior, the Likelihood and the Prior functions

- The problem we are interested in:

$$\mathbb{P}\{\Omega| >_u, >_m\} = \alpha \cdot \mathbb{P}\{>_u, >_m |\Omega\} \ \mathbb{P}\{\Omega\}$$

- When Ω is given, not only *the ordering of each pair of items becomes independent of rest of the orderings*, but also *two users can no longer influence other's vote*.

$$\mathbb{P}\{>_u, >_m |\Omega\} = \mathbb{P}\{>_u |\Omega\}\mathbb{P}\{>_m |\Omega\} \quad (1)$$

$$\mathbb{P}\{>_u |\Omega\} = \prod_{(k,l)\in D_u} \mathbb{P}\{k >_u l|\Omega\} \quad (2)$$

$$\mathbb{P}\{>_m |\Omega\} = \prod_{(i,j)\in D_m} \mathbb{P}\{i >_m j|\Omega\} \quad (3)$$

The Posterior, the Likelihood and the Prior functions

- The problem we are interested in:

$$\mathbb{P}\{\Omega | >_u, >_m\} = \alpha \cdot \mathbb{P}\{>_u, >_m |\Omega\} \; \mathbb{P}\{\Omega\}$$

- When Ω is given, not only *the ordering of each pair of items becomes independent of rest of the orderings*, but also *two users can no longer influence other's vote*.

$$\mathbb{P}\{>_u, >_m |\Omega\} = \mathbb{P}\{>_u |\Omega\}\mathbb{P}\{>_m |\Omega\} \quad (1)$$

$$\mathbb{P}\{>_u |\Omega\} = \prod_{(k,l)\in D_u} \mathbb{P}\{k >_u l|\Omega\} \quad (2)$$

$$\mathbb{P}\{>_m |\Omega\} = \prod_{(i,j)\in D_m} \mathbb{P}\{i >_m j|\Omega\} \quad (3)$$

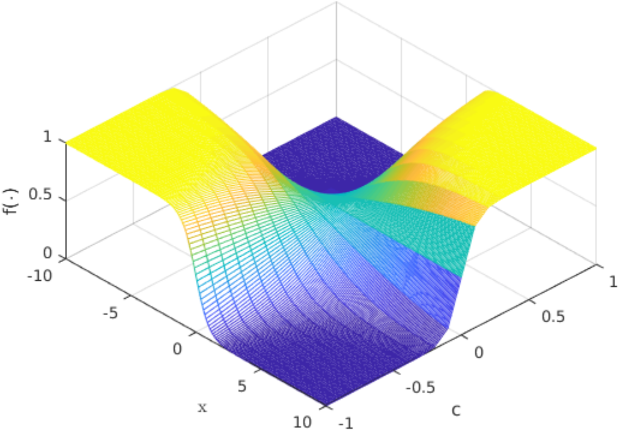
- The individual probability functions

$$\mathbb{P}\{k >_u l|\Omega\} \triangleq f(c_u, \hat{x}_{ukl}(\Omega)) \quad (4)$$

$$\mathbb{P}\{i >_m j|\Omega\} \triangleq f(c_m, \hat{x}_{ijm}(\Omega)) \quad (5)$$

The choice of $f(c, x)$

$$f(c,x) \triangleq \frac{1}{2} + \frac{1}{2}\tanh(cx)$$



The user/item entity specific functions $\hat{x}_{ijm}(\Omega)$ and $\hat{x}_{ukl}(\Omega)$

The estimates

$$\hat{x}_{ukl}(\Omega) \triangleq \hat{x}_{uk}(\Omega) - \hat{x}_{ul}(\Omega) \tag{6}$$

$$\hat{x}_{ijm}(\Omega) \triangleq \hat{x}_{im}(\Omega) - \hat{x}_{jm}(\Omega) \tag{7}$$

The user/item entity specific functions $\hat{x}_{ijm}(\Omega)$ and $\hat{x}_{ukl}(\Omega)$

The estimates

$$\hat{x}_{ukl}(\Omega) \triangleq \hat{x}_{uk}(\Omega) - \hat{x}_{ul}(\Omega) \tag{6}$$

$$\hat{x}_{ijm}(\Omega) \triangleq \hat{x}_{im}(\Omega) - \hat{x}_{jm}(\Omega) \tag{7}$$

can be modeled as $\hat{X} = PQ^T$ using matrix factorization (MF) as

$$\hat{x}_{uk} \triangleq \langle \boldsymbol{p}_u, \boldsymbol{q}_k \rangle = \boldsymbol{p}_u^T \boldsymbol{q}_k = \sum_{t=1}^{r} p_{ut} q_{tq}$$

$$\hat{x}_{im} \triangleq \langle \boldsymbol{p}_i, \boldsymbol{q}_m \rangle = \boldsymbol{p}_i^T \boldsymbol{q}_m = \sum_{t=1}^{r} p_{it} q_{tm}$$

The user/item entity specific functions $\hat{x}_{ijm}(\Omega)$ and $\hat{x}_{ukl}(\Omega)$

The estimates

$$\hat{x}_{ukl}(\Omega) \triangleq \hat{x}_{uk}(\Omega) - \hat{x}_{ul}(\Omega) \quad (6)$$

$$\hat{x}_{ijm}(\Omega) \triangleq \hat{x}_{im}(\Omega) - \hat{x}_{jm}(\Omega) \quad (7)$$

can be modeled as $\hat{X} = PQ^T$ using matrix factorization (MF) as

$$\hat{x}_{uk} \triangleq \langle \boldsymbol{p}_u, \boldsymbol{q}_k \rangle = \boldsymbol{p}_u^T \boldsymbol{q}_k = \sum_{t=1}^{r} p_{ut} q_{tq}$$

$$\hat{x}_{im} \triangleq \langle \boldsymbol{p}_i, \boldsymbol{q}_m \rangle = \boldsymbol{p}_i^T \boldsymbol{q}_m = \sum_{t=1}^{r} p_{it} q_{tm}$$

Hence

$$\hat{x}_{ukl} = \boldsymbol{p}_u^T (\boldsymbol{q}_k - \boldsymbol{q}_l) \quad (8)$$

$$\hat{x}_{ijm} = (\boldsymbol{p}_i - \boldsymbol{p}_j)^T \boldsymbol{q}_m \quad (9)$$

The likelihood function

Hence

$$\mathbb{P}\{>_u, >_m | \Omega\} = \prod_{u=1}^{|U|} \prod_{(k,l)\in D_u} f(c_u, \hat{x}_{ukl}(\Omega)) \times \prod_{m=1}^{|I|} \prod_{(i,j)\in D_m} f(c_m, \hat{x}_{ijm}(\Omega))$$

The prior function

Assume, the system parameters: $\Omega \triangleq [P^T \mid Q^T] = [\omega_1 \cdots \omega_N]$ are *independent normalized multivariate normal random* variables with known covariance matrices $\{\Sigma_n\}_{n=1}^N$ where N is the number of parameter vectors in Ω.

The prior function

Assume, the system parameters: $\Omega \triangleq [P^T \mid Q^T] = [\omega_1 \cdots \omega_N]$ are *independent normalized multivariate normal random* variables with known covariance matrices $\{\Sigma_n\}_{n=1}^N$ where N is the number of parameter vectors in Ω.

The prior

$$\mathbb{P}\{\Omega\} = \frac{1}{(2\pi)^{\frac{N}{2}} \prod_n |\Sigma_n|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\sum_n \omega_n^T \Sigma_n^{-1} \omega_n\right\} \tag{10}$$

Comprehensive Personalized Ranking (CPR)

Finally

$$\text{CPR} \triangleq \ln \mathbb{P}\{\Omega| >_u, >_m\}$$

Comprehensive Personalized Ranking (CPR)

Finally

$$\begin{aligned} \text{CPR} &\triangleq \ln \mathbb{P}\{\Omega| >_u, >_m\} \\ &\simeq \ln \mathbb{P}\{>_u, >_m |\Omega\} \, \mathbb{P}\{\Omega\} \end{aligned}$$

Comprehensive Personalized Ranking (CPR)

Finally

$$\begin{aligned}
\text{CPR} &\triangleq \ln \mathbb{P}\{\Omega| >_u, >_m\} \\
&\simeq \ln \mathbb{P}\{>_u, >_m |\Omega\}\ \mathbb{P}\{\Omega\} \\
&\simeq \sum_u \sum_{(k,l)\in D_u} \ln f(c_u, \hat{x}_{ukl}(\Omega)) \\
&\quad + \sum_m \sum_{(i,j)\in D_m} \ln f(c_m, \hat{x}_{ijm}(\Omega)) \\
&\quad - \frac{1}{2}\sum_n \omega_n^T \Sigma_n^{-1} \omega_n
\end{aligned} \tag{11}$$

Learning the CPR

$$\frac{\partial}{\partial \Omega} \ln f(c, \hat{x}) = c(1 - \tanh(c\hat{x})) \frac{\partial}{\partial \Omega} \hat{x}$$

Learning the CPR

$$\frac{\partial}{\partial \Omega} \ln f(c, \hat{x}) = c(1 - \tanh(c\hat{x}))\ \frac{\partial}{\partial \Omega} \hat{x}$$

$$\frac{\partial}{\partial \Omega} \hat{x}_{ijm} = \begin{cases} (p_{it} - p_{jt}), & \omega_t = q_{tm}, \\ q_{tm}, & \omega_t = p_{it}, \\ -q_{tm}, & \omega_t = p_{jt}, \\ 0, & \text{otherwise} \end{cases}$$

Learning the CPR

$$\frac{\partial}{\partial \Omega} \ln f(c, \hat{x}) = c(1 - \tanh(c\hat{x})) \frac{\partial}{\partial \Omega} \hat{x}$$

$$\frac{\partial}{\partial \Omega} \hat{x}_{ijm} = \begin{cases} (p_{it} - p_{jt}), & \omega_t = q_{tm}, \\ q_{tm}, & \omega_t = p_{it}, \\ -q_{tm}, & \omega_t = p_{jt}, \\ 0, & \text{otherwise} \end{cases}$$

$$\frac{\partial}{\partial \Omega} \frac{1}{2} \sum_n \omega_n^T \Sigma_n^{-1} \omega_n = [\Sigma_1^{-1} \omega_1 \quad \cdots \quad \Sigma_N^{-1} \omega_N]$$

Learning the CPR

$$\frac{\partial}{\partial \Omega} \ln f(c, \hat{x}) = c(1 - \tanh(c\hat{x})) \frac{\partial}{\partial \Omega} \hat{x}$$

$$\frac{\partial}{\partial \Omega} \hat{x}_{ijm} = \begin{cases} (p_{it} - p_{jt}), & \omega_t = q_{tm}, \\ q_{tm}, & \omega_t = p_{it}, \\ -q_{tm}, & \omega_t = p_{jt}, \\ 0, & \text{otherwise} \end{cases}$$

$$\frac{\partial}{\partial \Omega} \frac{1}{2} \sum_n \omega_n^T \Sigma_n^{-1} \omega_n = [\Sigma_1^{-1} \omega_1 \quad \cdots \quad \Sigma_N^{-1} \omega_N]$$

Eventually

$$\Omega_{new} \leftarrow \Omega - \mu \frac{\partial}{\partial \Omega} \text{CPR}, \tag{12}$$

Numerical examples

Experimental setup

- Partial MovieLens dataset[4]
- 600 ratings given by 40 users judging 60 movies on a scale between 1 to 5
- We start by converting the rating matrix to comparison data and these data are stored in a memory-efficient way
- In order to handle large amount of data we resort to the stochastic gradient descent method and mini-batch learning

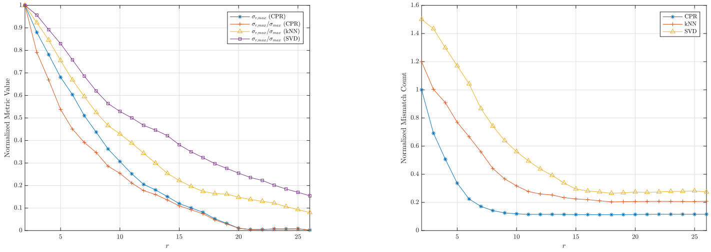
[4] F. M. Harper et al. The MovieLens datasets: History and context, 2015

Numerical examples

Nature of experiments

- The method relies on the data in an r-dimensional space. Also, as many users tend to show shared interest in only specific subsets of items, the rating matrix is low-rank
- A natural metric to determine the rank of the original rating matrix, r_X, can be to look at its r largest singular values
- When $r < r_X$, the method cannot allocate all the information in an r-dimensional space. And when $r > r_X$, the method puts most of the recovered information in an r_X-dimensional space and places little to no information in the remaining dimensions
- One can use the ratio of the r-th largest to the largest singular value of the recovered matrix as a metric to determine the true rank. This ratio should drop drastically as soon as r gets greater than r_X

Numerical examples



(a) (b)

Figure: The results for different algorithms: (a) normalized values of various metrics on the recovered rating matrices versus the expected rank r, (b) the normalized number of mismatches between the original comparison data and the comparisons made from the recovered data for *CPR*, *kNN* and *SVD*.

Summary

- We studied a new optimization framework based on one-bit preference comparison data to develop the Comprehensive Personalized Ranking (CPR) system.
- The algorithm relies on a Bayesian treatment of the data, and maximizes the posterior probability of the system parameters.
- A learning model w.r.t. the optimization problem using matrix factorization is provided.
- Initial numerical results were provided to show the effectiveness of the algorithm.
- The study of the impact of the rating matrix size on the projected rank would be an interesting future research avenue as the projected rank of a matrix significantly controls the storage and computational efficiency of the algorithm.

Thank you
and
Questions?