A Generic Coordinate Descent Framework for Learning from Implicit Feedback

Immanuel Bayer* (Swiss Re), **Xiangnan He*** (National University of Singapore), Bhargav Kanagal (Google), Steffen Rendle (Google)

Presented by **Xiangnan He** @ WWW 2017, April 07, 2017

*work done at Google

Overview

1. **Learning from Implicit Feedback**
2. Recommender Models
3. Generic Optimization Framework
   a. Implicit Regularization
   b. k-Separable Models
   c. Coordinate Descent Algorithm
4. Experiments

Google

Implicit vs. Explicit Feedback

Explicit Data

YT Videos

Users

2		3	
		4	
	5		2
3	1		4
		3	

Ratings

Implicit Data

YT Videos

Users

1	0	1	0
0	0	1	0
0	2	0	1
1	1	0	2
0	0	1	0

Watch Counts

Explicit Data

- Actual ratings by users
- Unknown ratings carry no signal

Implicit Data

- Convey preferences *implicitly*
- 0 - implicit dislike, carries signal

Challenge: Training over implicit data needs to account for the whole matrix, unlike explicit data.

Learning from Implicit Feedback

Scoring function over set of *context* C and *items* I

$$\hat{y} : C \times I \to \mathbb{R}$$

Training data S consists of tuples

$$(c, i, y, \alpha) \in S$$

Solve least squares problem

$$L(\Theta|S) = \sum_{(c,i,y,\alpha) \in S} \alpha \left(\hat{y}(c,i) - y \right)^2 + \sum_{\theta \in \Theta} \lambda_\theta \theta^2$$

$$\begin{array}{cc} & \text{Item I} \\ \text{Context C} & \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 1 & 0 & 2 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Implicit Data S_{impl}

Challenge: S_{impl} has |C x I| entries.
Applying generic learning algorithms
is infeasible.

Learning from Implicit Feedback

Hu et al. [ICDM 2008] have proposed a fast algorithm to learn a matrix factorization model over S_{impl}.

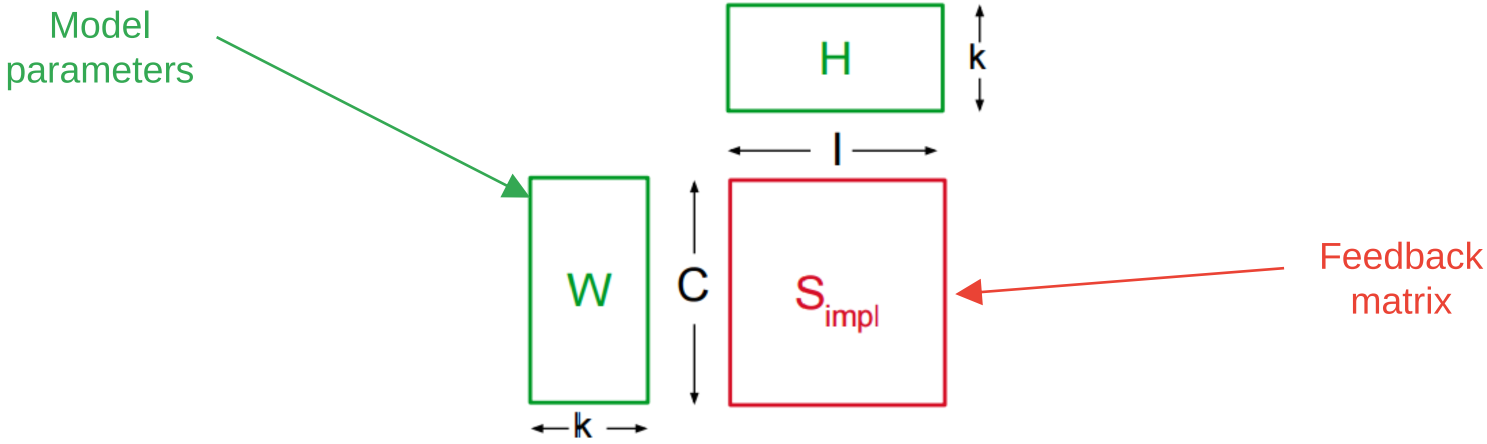
Our contributions

- A generic coordinate descent algorithm that can be applied to a variety of recommender models.
- We show this for models include: MF, MF with side information (SVDfeature), Factorization Machines, parallel factor analysis, Tucker Decomposition.

Overview

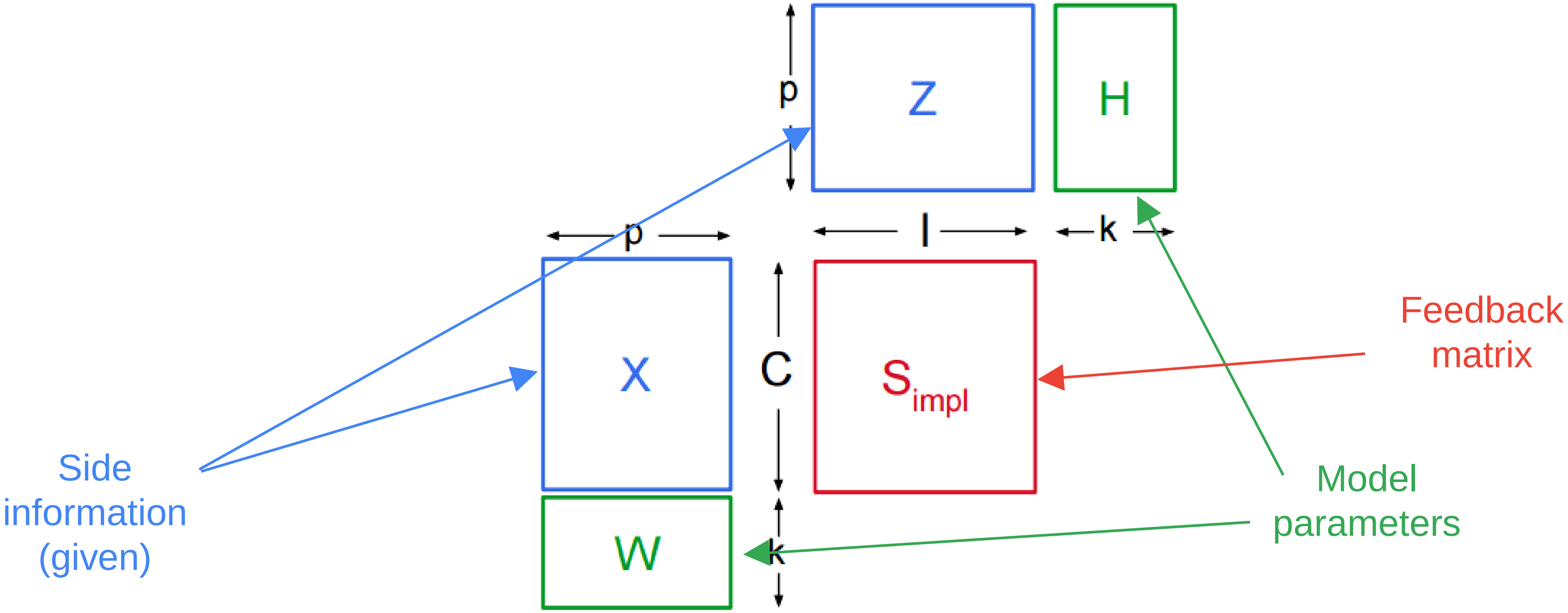
1. Learning from Implicit Feedback
2. **Recommender Models**
3. Generic Optimization Framework
    a. Implicit Regularization
    b. k-Separable Models
    c. Coordinate Descent Algorithm
4. Experiments

Matrix Factorization



$$\hat{y}(c,i) := \langle \mathbf{w}_c, \mathbf{h}_i \rangle = \sum_{f=1}^{k} w_{c,f}\, h_{i,f}$$

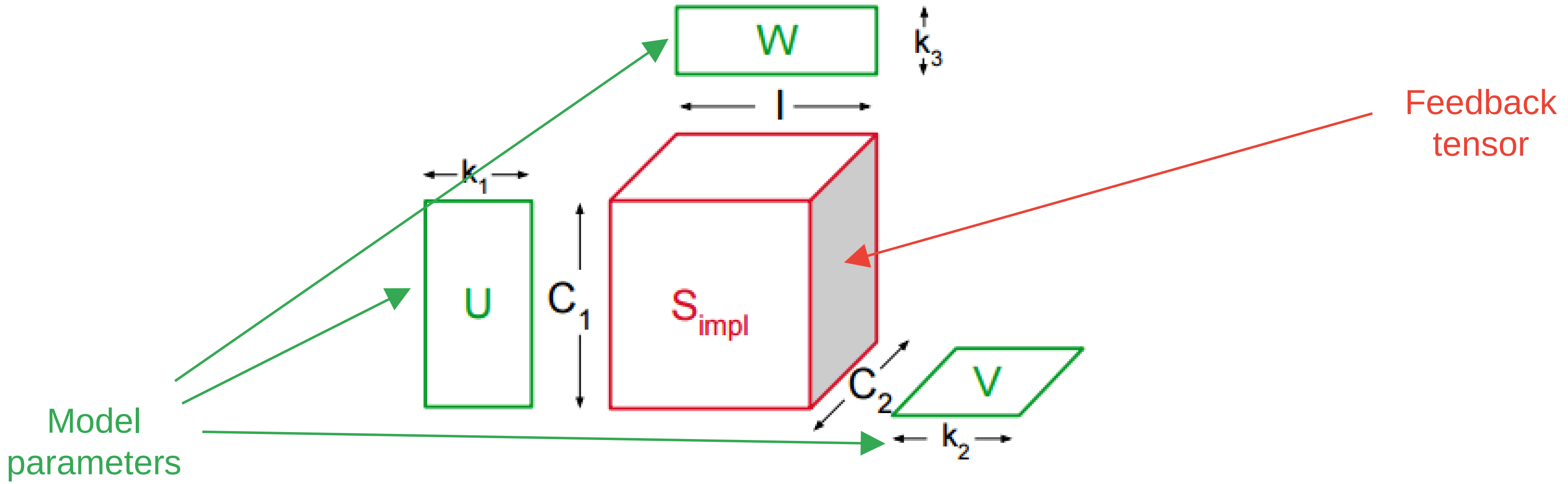
Matrix Factorization with Side Information



$$\hat{y}(c,i) = \boldsymbol{x}_c\, W\, (\boldsymbol{z}_i\, H)^t = \sum_{f=1}^{k} \left(\sum_{l=1}^{p} x_{c,l}\, w_{l,f} \right) \left(\sum_{l=1}^{p} z_{i,l}\, h_{l,f} \right)$$

Google

Tensor Factorization (PARAFAC)



$$\hat{y}(c_1, c_2, i) := \sum_{f=1}^{k} u_{c_1,f}\, v_{c_2,f}\, w_{i,f}$$

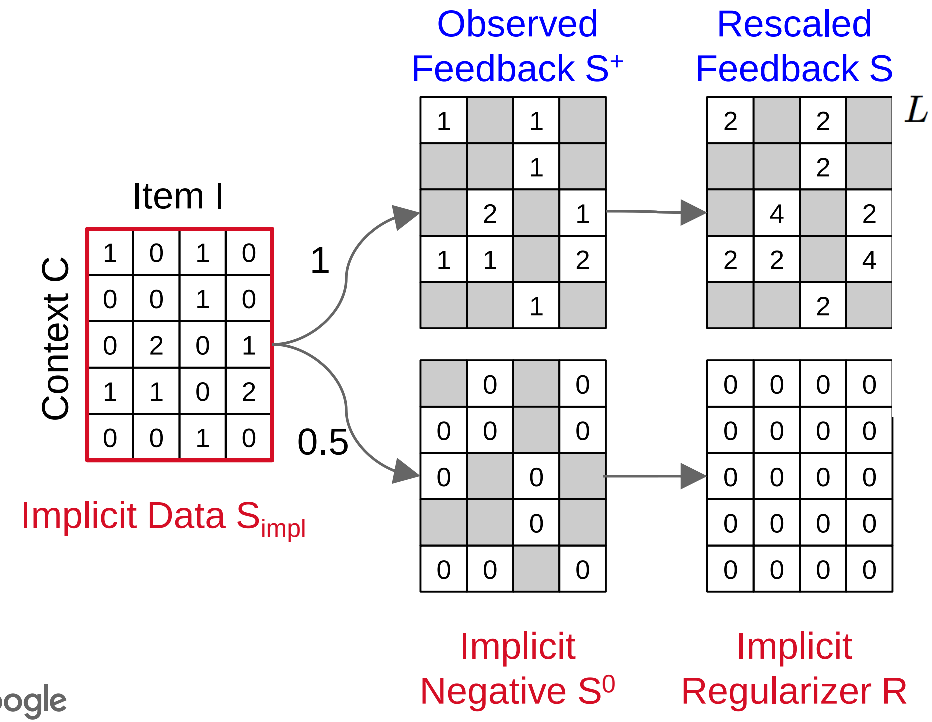
Google

Overview

1. Learning from Implicit Feedback
2. Recommender Models
3. **Generic Optimization Framework**
    a. Implicit Regularization
    b. k-Separable Models
    c. Coordinate Descent Algorithm
4. Experiments

Implicit Regularization

Idea 1: Decompose expensive loss into a cheap loss and an expensive regularizer



$$L = \sum_{(c,i)\in S^+} (y_{c,i} - \hat{y}(c,i))^2 + \alpha_0 \sum_{(c,i)\in S^-} (0 - \hat{y}(c,i))^2$$

$$= \sum_{(c,i)\in S^+} [(y_{c,i} - \hat{y}(c,i))^2 - \alpha_0 \hat{y}^2_{c,i}] + \alpha_0 \sum_c \sum_i \hat{y}(c,i)^2$$

$$\propto \sum_{(c,i)\in S^+} \left(\hat{y}(c,i) - \frac{1}{1-\alpha_0} y_{c,i}\right)^2 + \alpha_0 \sum_c \sum_i \hat{y}(c,i)^2$$

L(θ|S): Computing loss over Observed Feedback is cheap

R(θ): Computing Implicit Regularizer R is expensive!

k-Separable Models

Idea 2: The regularizer R of any *k-separable* model ŷ can be calculated efficiently.

A model ŷ is *k-separable* if it can be written as:

$$\hat{y}(c,i) = \langle \boldsymbol{\phi}(c), \boldsymbol{\psi}(i) \rangle = \sum_{f=1}^{k} \phi_f(c)\, \psi_f(i)$$

with functions

$$\boldsymbol{\phi} : C \to \mathbb{R}^k, \quad \boldsymbol{\psi} : I \to \mathbb{R}^k$$

This allows to rewrite the regularizer R as:

$$R(\Theta) = \sum_{f=1}^{k} \sum_{f'=1}^{k} \sum_{c \in C} \phi_f(c)\, \phi_{f'}(c) \sum_{i \in I} \psi_f(i)\, \psi_{f'}(i)$$

Computing R or its derivatives R' and R'' is in O((|C| + |I|)*k^2) instead of O(k*|C|*|I|).

Google

k-Separable Models: Examples

Definition: k-separable

$$\hat{y}(c,i) = \langle \boldsymbol{\phi}(c), \boldsymbol{\psi}(i) \rangle$$

$$\boldsymbol{\phi} : C \to \mathbb{R}^k, \quad \boldsymbol{\psi} : I \to \mathbb{R}^k$$

Matrix Factorization

$$\phi_f(c) = w_{c,f}, \quad \psi_f(i) = h_{i,f}$$

Matrix Factorization with side information (SVDfeature)

$$\phi_f(c) = \sum_{l=1}^{p} x_{c,l}\, w_{l,f}, \quad \psi_f(i) = \sum_{l=1}^{p} z_{i,l}\, h_{l,f}$$

Candecomp / Parafac

$$\phi_f(c_1, c_2) = u_{c_1,f}\, v_{c_2,f}, \quad \psi_f(i) = w_{i,f}$$

Factorization Machines, Tucker Decomposition, ..

Efficient Implicit Coordinate Descent (iCD)

Use idea 1 and 2 to speed up coordinate descent:

Update rule for one coordinate $\theta \leftarrow \theta - L'(\theta|S_{impl})/L''(\theta|S_{impl})$

$L(\theta|S_{impl})$ can be decomposed into $L(\theta|S) + R(\theta)$

The derivatives for the expensive part $R(\theta)$ can be computed efficiently:

$$R'(\theta) = 2 \sum_{f=1}^{k} \sum_{f'=1}^{k} J_I(f, f') \sum_{c \in C} \phi_f(c)\, \phi'_{f'}(c)$$

$$R''(\theta) = 2 \sum_{f=1}^{k} \sum_{f'=1}^{k} J_I(f, f') \sum_{c \in C} \left[\phi_f(c)\, \phi''_{f'}(c) + \phi'_f(c)\, \phi'_{f'}(c)\right]$$

Overview

1. Learning from Implicit Feedback
2. Recommender Models
3. Generic Optimization Framework
    a. Implicit Regularization
    b. k-Separable Models
    c. Coordinate Descent Algorithm
4. **Experiments**

Experiments

Dataset

Implicit feedback: short-term Youtube watch history of 200k users on 68k popular videos

Side information: age, country, gender, device

Techniques

Popularity (baseline)

Coview (baseline)

iCD Matrix Factorization (iCD MF)

iCD Factorization Machines (iCD FM)

Offline Recommendation

Protocol

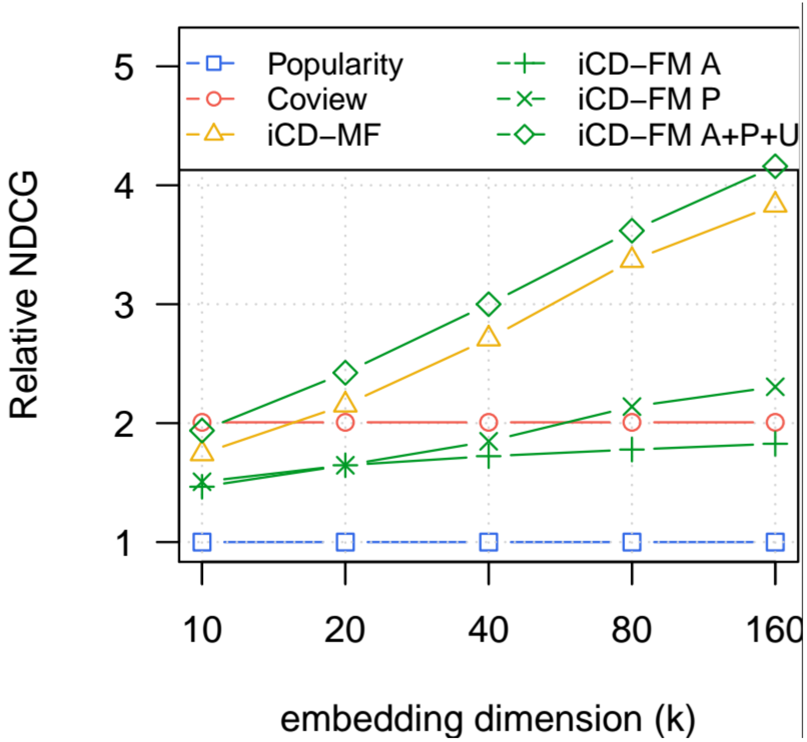
Holdout last watch for evaluation

Features

U: user id

A: user attributes

P: the previously watched video



Google

Cold-Start

Protocol

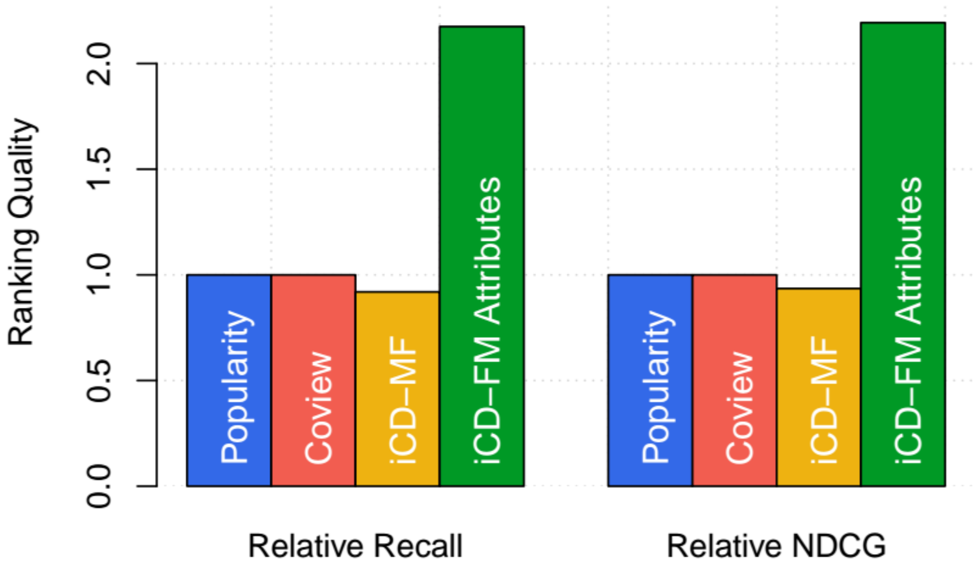
Evaluation users don’t have any training data

Rely on user attributes

Features

iCD-MF: user id x item id

iCD-FM: user attributes x item id



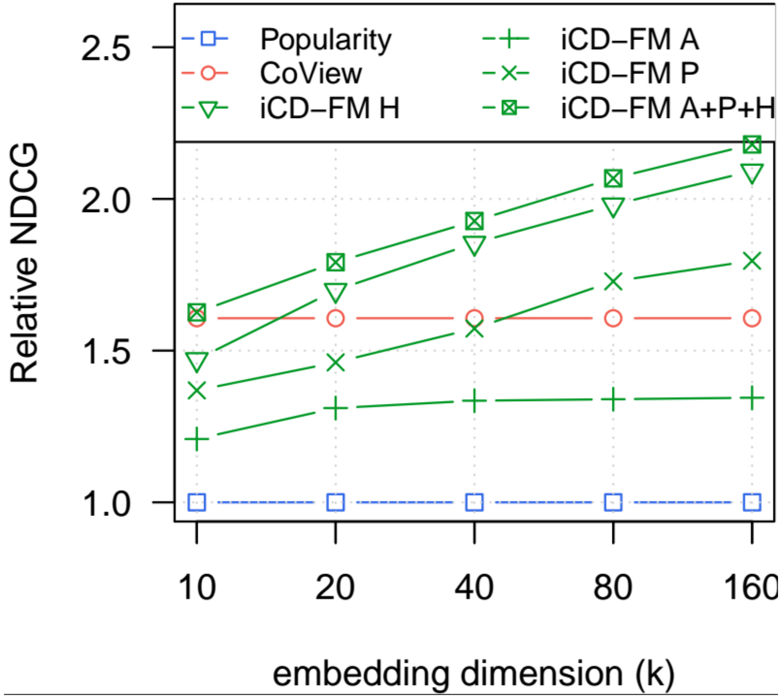
Instant Recommendation

Protocol

- Model is trained offline up to a cutoff date
- Evaluation: all watches up to the query time can be taken into account as side information but model cannot be retrained

Features

- H: all previously watched videos by the user (as bag of words)
- A: user attributes
- P: the previously watched video



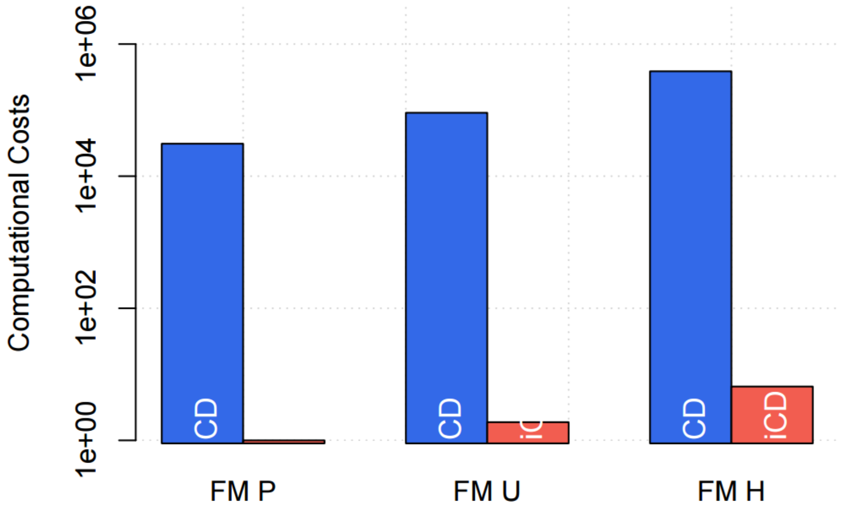
Computational Costs

Protocol

- Costs of training a FM model with expensive CD to proposed iCD
- Costs relative to CD-FM P

Features

- P: the previously watched video
- U: user ID
- H: all previously watched videos by the user (as bag of words)



Summary

Proposed a **generic, efficient framework** to learn recommender systems from **implicit feedback**.

Main ideas:

- **Implicit Regularization**: Reformulate expensive implicit loss as cheap explicit loss and expensive regularizer.
- **k-Separable Models**: Implicit regularizer can be computed cheaply for models that decompose into a dot-product of a context and item part.

Many popular recommender models can be learned by our framework.

- E.g., Matrix factorization, Matrix factorization with side information, Factorization Machines, Parallel factor analysis, Tucker decomposition