Factorization Machines

Authors: Steffen Rendle

Tiandi Ye

2020/09/09

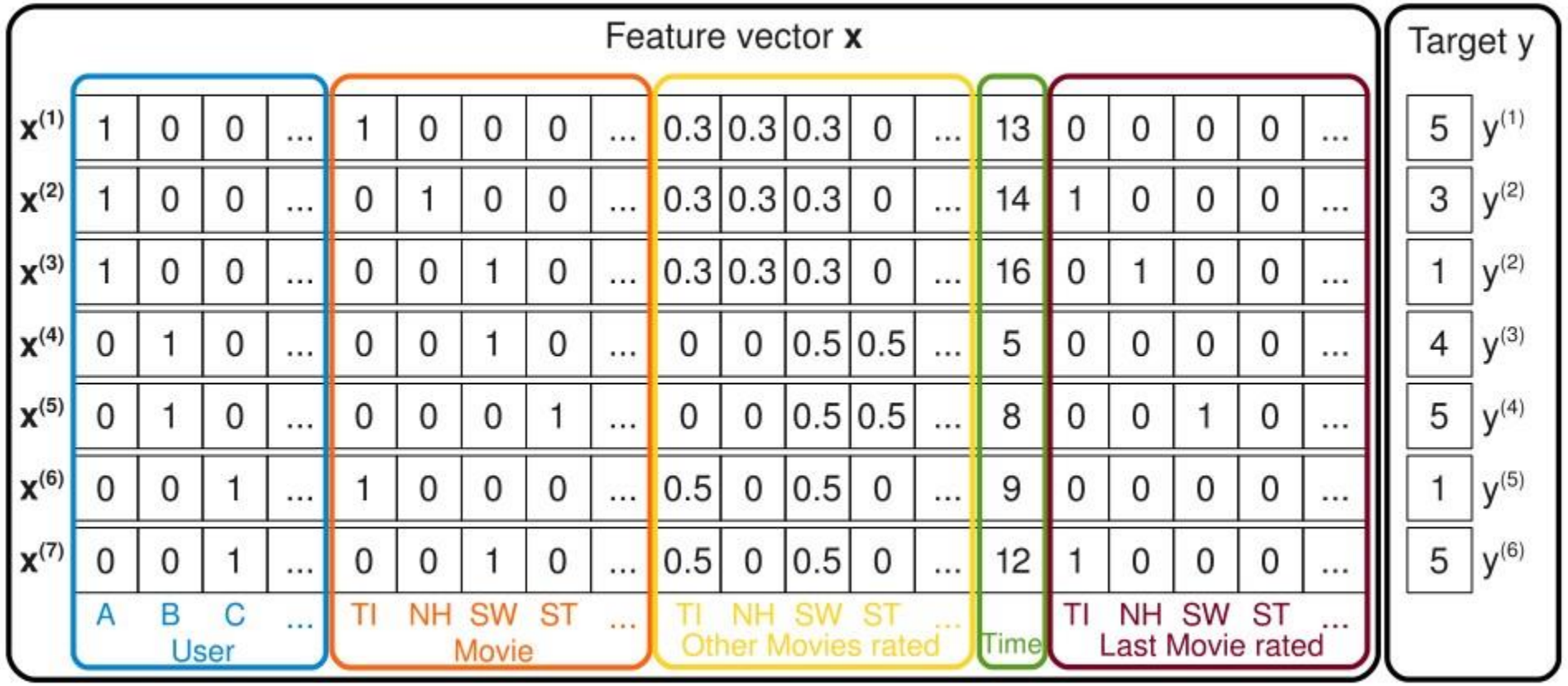
Movie Review System

✓ $user\ u \in U, movie(item)\ i \in I, time\ t \in \mathbb{R}, rating\ r \in \{1, 2, 3, 4, 5\}$

$$U = \{\ \text{Alice (A)},\ \text{Bob (B)},\ \text{Charlie (C)}, \ldots\}$$

$$I = \{\ \text{Titanic (TI)},\ \text{Notting Hill (NH)},\ \text{Star Wars (SW)},\ \text{Star Trek (ST)}, \ldots\}$$

$$\begin{aligned} S = \{&(\text{A}, \text{TI}, 2010-1, 5), (\text{A}, \text{NH}, 2010-2, 3), (\text{A}, \text{SW}, 2010-4, 1) \\ &(\text{B}, \text{SW}, 2009-5, 4), (\text{B}, \text{ST}, 2009-8, 5) \\ &(\text{C}, \text{TI}, 2009-9, 1), (\text{C}, \text{SW}, 2009-12, 5)\} \end{aligned}$$



- 第一部分：表征当前评分用户信息，维度为$|U|$
- 第二部分：当前被评分电影信息，维度为$|I|$
- 第三部分：当前评分用户评分过的所有电影信息（归一化），维度为$|I|$
- 第四部分：评分日期信息，维度为1
- 第五部分：当前评分用户最近评分过的一部电影的信息，维度为$|I|$

➢ 特征向量总的维度：$|U| + |I| + |I| + 1 + |I| = |U| + 3|I| + 1$远大于每个用户参与评分的电影的数目

- 线性回归建模（没有考虑特征分量之间的交互关系）

$$\widehat{y}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

$$= w_0 + \sum_{i=1}^{n} w_i x_i$$

- 考虑互异特征分量之间的交互关系

$$\hat{y}(\mathbf{x}) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} x_i x_j$$

- 对于数据集中未出现过交互的特征分量，不能对相应的参数进行估计，在高度稀疏数据场景中，由于数据量不足，样本中出现未交互的特征分量是很普遍的

- 针对每个维度的特征分量x_i，引入辅助向量v_i（k超参数）

$$\mathbf{v}_i = \left(v_{i1}, v_{i2}, \cdots, v_{ik}\right)^\top \in \mathbb{R}^k, i = 1, 2, \cdots, n$$

$$\hat{\omega}_{i,j} = \langle \mathrm{v_i}, \mathrm{v_j} \rangle$$

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle := \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f}$$

rating	TI	NH	SW	ST
A	5	3	1	
B			4	5
C	1		5	

B对SW和ST的评分相近，意味着v_{SW}与v_{ST}相似，
那么可以用$\hat{\omega}_{A,SW} = \langle \mathrm{v}_A, \mathrm{v}_{SW} \rangle$近似$\hat{\omega}_{A,ST} = \langle \mathrm{v}_A, \mathrm{v}_{ST} \rangle$

Model Equation

- 2-way二阶FM方程

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

$$\widehat{W} = VV^\top = \begin{pmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \vdots \\ \mathbf{v}_n^\top \end{pmatrix} (\mathbf{v}_1 \mathbf{v}_2 \cdots \mathbf{v}_n) = \begin{pmatrix} \mathbf{v}_1^\top \mathbf{v}_1 & \mathbf{v}_1^\top \mathbf{v}_2 & \cdots & \mathbf{v}_1^\top \mathbf{v}_n \\ \mathbf{v}_2^\top \mathbf{v}_1 & \mathbf{v}_2^\top \mathbf{v}_2 & \cdots & \mathbf{v}_2^\top \mathbf{v}_n \\ & & \ddots & \\ \mathbf{v}_n^\top \mathbf{v}_1 & \mathbf{v}_n^\top \mathbf{v}_2 & \cdots & \mathbf{v}_n^\top \mathbf{v}_n \end{pmatrix}_{n \times n}$$

$$= \begin{pmatrix} \mathbf{v}_1^\top \mathbf{v}_1 & \widehat{w}_{12} & \cdots & \widehat{w}_{1n} \\ \widehat{w}_{21} & \mathbf{v}_2^\top \mathbf{v}_2 & \cdots & \widehat{w}_{2n} \\ & & \ddots & \\ \widehat{w}_{n1} & \widehat{w}_{n2} & \cdots & \mathbf{v}_n^\top \mathbf{v}_n \end{pmatrix}_{n \times n}$$

Expressiveness

$$\mathbf{v}_i = (v_{i1}, v_{i2}, \cdots, v_{ik})^\top \in \mathbb{R}^k, i = 1, 2, \cdots, n$$

当k足够大时，对于任意正定矩阵W，均存在矩阵V，使得$W = V \cdot V^T$

This shows that a FM can express any interaction matrix W if k is chosen large enough.

Restricting k – and thus the expressiveness of the FM – leads to better generalization and thus improved interaction matrices under sparsity.

Computation

- 复杂度：$O(kn^2)$

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

$$= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j - \frac{1}{2} \sum_{i=1}^{n} \langle \mathbf{v}_i, \mathbf{v}_i \rangle x_i x_i$$

$$= \frac{1}{2} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{f=1}^{n} v_{i,f} v_{j,f} x_i x_j - \sum_{i=1}^{n} \sum_{f=1}^{n} v_{i,f} v_{i,f} x_i x_i \right)$$

$$= \frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{i,f} x_i \right) \left(\sum_{j=1}^{n} v_{j,f} x_j \right) - \sum_{i=1}^{n} v_{i,f}^2 x_i^2 \right)$$

$$= \frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{i,f} x_i \right)^2 - \sum_{i=1}^{n} v_{i,f}^2 x_i^2 \right)$$

$$\begin{pmatrix} \mathbf{v}_1^\top \mathbf{v}_1 & \widehat{w}_{12} & \cdots & \widehat{w}_{1n} \\ \widehat{w}_{21} & \mathbf{v}_2^\top \mathbf{v}_2 & \cdots & \widehat{w}_{2n} \\ & & \ddots & \\ \widehat{w}_{n1} & \widehat{w}_{n2} & \cdots & \mathbf{v}_n^\top \mathbf{v}_n \end{pmatrix}_{n \times n}$$

- 复杂度：$O(kn)$

Factorization Machines as Predictors

FM can be applied to a variety of prediction tasks. Among them are:

- Regression
- Binary classification
- Ranking

Learning Factorization Machines

$$\frac{\partial}{\partial \theta} \hat{y}(\mathbf{x}) = \begin{cases} 1, & \text{if } \theta \text{ is } w_0 \\ x_i, & \text{if } \theta \text{ is } w_i \\ x_i \sum_{j=1}^{n} v_{j,f} x_j - v_{i,f} x_i^2, & \text{if } \theta \text{ is } v_{i,f} \end{cases}$$

Each gradient can be computed in constant time O(1)

All parameter updates for a case (x, y) can be done in O(kn)

d-way Factorization Machine

$$\hat{y}(x) := w_0 + \sum_{i=1}^{n} w_i x_i$$
$$+ \sum_{l=2}^{d} \sum_{i_1=1}^{n} \cdots \sum_{i_l=i_{l-1}+1}^{n} \left(\prod_{j=1}^{l} x_{i_j} \right) \left(\sum_{f=1}^{k_l} \prod_{j=1}^{l} v_{i_j,f}^{(l)} \right)$$

- 计算复杂度：$O(k_d n^d)$

Summary

- Advantages
  - The interactions between values can be estimated even under high sparsity.
  - The number of parameters as well as the time for prediction and learning is linear.

FMs VS. SVMs

A. SVM model

1) Linear Kernel(identical to a FM of degree d = 1)

$$K_l(\mathbf{x}, \mathbf{z}) := 1 + \langle \mathbf{x}, \mathbf{z} \rangle$$

$$\phi(\mathbf{x}) := (1, x_1, \ldots, x_n)$$

$$\hat{y}(\mathbf{x}) = w_0 + \sum_{i=1}^{n} w_i x_i, \quad w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^n$$

2) Polynomial Kernel

$$K(\mathbf{x}, \mathbf{z}) := (\langle \mathbf{x}, \mathbf{z} \rangle + 1)^d$$

$$\phi(\mathbf{x}) := \Big(1, \sqrt{2}x_1, \ldots, \sqrt{2}x_n, x_1^2, \ldots, x_n^2$$

$$\sqrt{2}x_1x_2, \ldots, \sqrt{2}x_1x_n, \sqrt{2}x_2x_3, \ldots, \sqrt{2}x_{n-1}x_n\Big)$$

$$\hat{y}(\mathbf{x}) = w_0 + \sqrt{2}\sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} w_{i,i}^{(2)} x_i^2$$

$$+ \sqrt{2}\sum_{i=1}^{n}\sum_{j=i+1}^{n} w_{i,j}^{(2)} x_i x_j \quad w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^n, \quad \mathbf{W}^{(2)} \in \mathbb{R}^{n \times n} \text{ (symmetric matrix)}$$

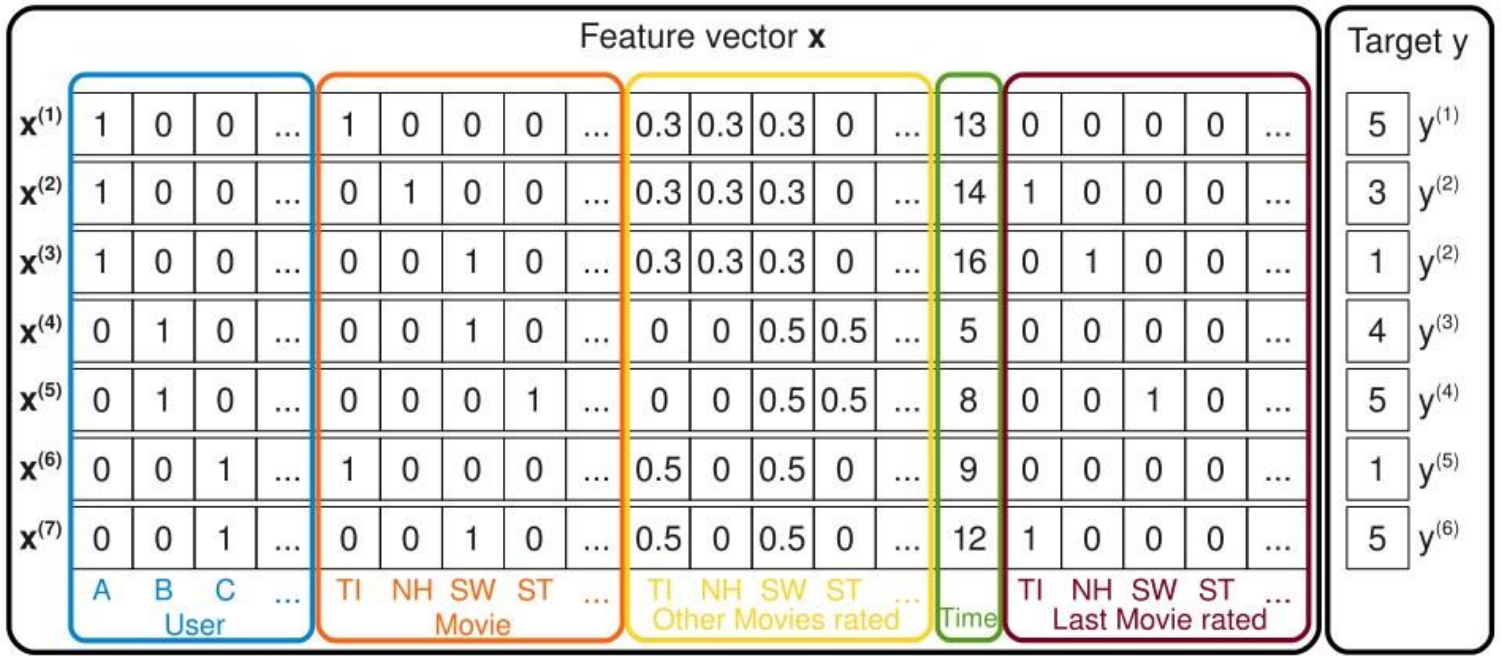
A. SVM model

- **Difference**:

all interaction parameters $\omega_{i,j}$ of SVMs are completely independent, e.g. $\omega_{i,j}$ and $\omega_{i,l}$

the interaction parameters of FMs are factorized and thus $\langle \mathbf{v}_i, \mathbf{v}_j \rangle$ and $\langle \mathbf{v}_i, \mathbf{v}_l \rangle$ depend on each other as they overlap and share parameters (here $\mathbf{v}_i$).

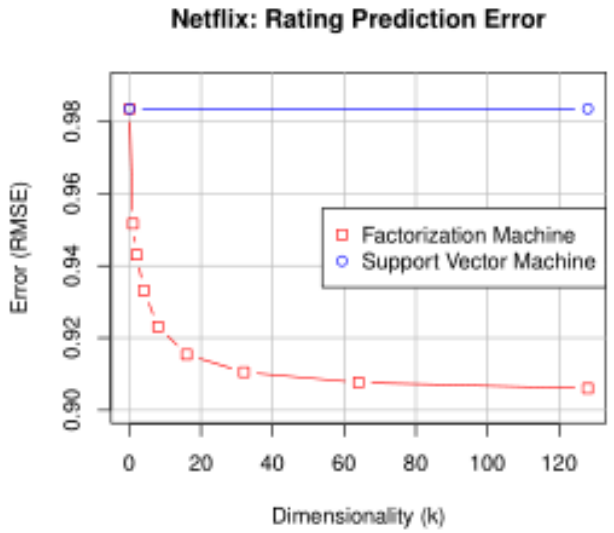
B. Parameter Estimation Under Sparsity



B. Parameter Estimation Under Sparsity

1) Linear SVM

$$\hat{y}(\mathbf{x}) = w_0 + w_u + w_i$$



2) Polynomial SVM

$$\hat{y}(\mathbf{x}) = w_0 + \sqrt{2}(w_u + w_i) + w_{u,u}^{(2)} + w_{i,i}^{(2)} + \sqrt{2}w_{u,i}^{(2)}$$

C. Summary

- 1) Parameters of FMs can be estimated well even under sparsity while SVMs fails.
- 2) FMs can be directly learned in the primal. Non-linear SVMs are usually learned in the dual.
- 3) The model equation of FMs is independent of the training data. Prediction with SVMs depends on parts of the training data (the support vectors).

FMS VS. OTHER FACTORIZATION MODELS

FM VS MF（Matrix Factorization）

- 分解机的思想是从线性模型中通过增加二阶交叉项来拟合特征之间的交互，为了拓展到数据稀疏场景并便于计算，吸收了矩阵分解的思想。FM是MF的全能版本，MF是FM的一种简单存在形式。

$\hat{r}_{ui} = p_u^T q_i$ $\qquad\qquad$ $\hat{r}_{ui} = \alpha + \beta_u + \beta_i + p_u^T q_i$

- $x^{(1)} = [1, 0, 0, 1, 0, 0, 0]$

$$\hat{y}(\mathbf{x}) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j > x_i x_j$$

$$= w_0 + \sum_{i=1}^{7} w_i x_i^{(1)} + \sum_{i=1}^{7} \sum_{j=i+1}^{7} < \mathbf{v}_i, \mathbf{v}_j > x_i^{(1)} x_j^{(1)}$$

$$= w_0 + w_1 + w_4 + < \mathbf{v}_1, \mathbf{v}_4 >$$

FM与MF的不同点

- 输入数据的形式不同
  - FM：实值特征向量
  - MF：二元组(u,i)
- 参数矩阵不同
  - FM：参数矩阵V
  - MF：用户矩阵P和商品（电影）矩阵Q

FM VS SVD++

$$\hat{y}(\mathbf{x}) = \overbrace{w_0 + w_u + w_i + \langle \mathbf{v}_u, \mathbf{v}_i \rangle + \frac{1}{\sqrt{|N_u|}} \sum_{l \in N_u} \langle \mathbf{v}_i, \mathbf{v}_l \rangle}^{\text{SVD++}} + \frac{1}{\sqrt{|N_u|}} \sum_{l \in N_u} \left(w_l + \langle \mathbf{v}_u, \mathbf{v}_l \rangle + \frac{1}{\sqrt{|N_u|}} \sum_{l' \in N_u, l' > l} \langle \mathbf{v}_l, \mathbf{v}'_l \rangle \right)$$

- $N_{\mathbf{u}}$： the set of all movies the user has ever rated
- FM contains also some additional interactions between users and movies $N_{\mathbf{u}}$ as well as basic effects for the movies $N_{\mathbf{u}}$ and interactions between pairs of movies in $N_{\mathbf{u}}$.

FM VS PITF

The problem of tag prediction is defined as ranking tags for a given user and item combination.

$$\hat{y}(\mathbf{x}) = w_0 + w_u + w_i + w_t + \langle \mathbf{v}_u, \mathbf{v}_i \rangle + \langle \mathbf{v}_u, \mathbf{v}_t \rangle + \langle \mathbf{v}_i, \mathbf{v}_t \rangle$$

$$\hat{y}(\mathbf{x}) := w_t + \langle \mathbf{v}_u, \mathbf{v}_t \rangle + \langle \mathbf{v}_i, \mathbf{v}_t \rangle$$

Difference:

- the FM model has a bias term w_t for tag t
- the factorization parameters for the tags (v_t)

between the (u, t)- and (i, t)-interaction are shared for the FM model but individual for the original PITF model.

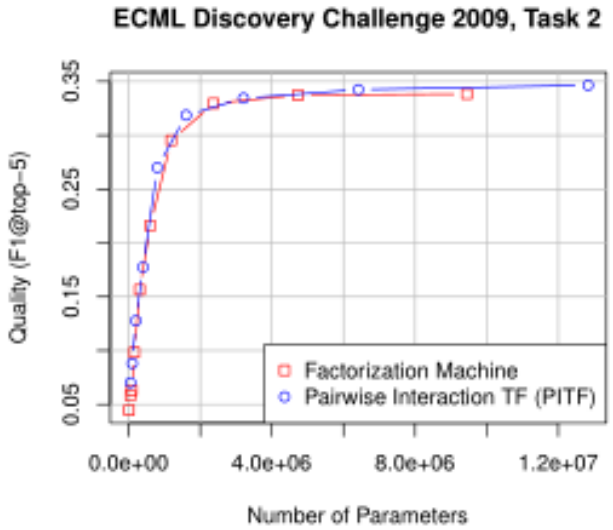


Fig. 3. Recommendation quality of a FM compared to the winning PITF model [3] of the ECML/PKDD Discovery Challenge 2009. The quality is plotted against the number of model parameters.

FM VS Factorized Personalized Markov Chains (FPMC)

The FPMC model tries to rank products in an online shop based on the last purchases (at time t−1) of the user u.

$$\hat{y}(\mathbf{x}) = w_0 + w_u + w_i + \langle \mathbf{v}_u, \mathbf{v}_i \rangle + \frac{1}{|B^u_{t-1}|} \sum_{l \in B^u_{t-1}} \langle \mathbf{v}_i, \mathbf{v}_l \rangle$$
$$+ \frac{1}{|B^u_{t-1}|} \sum_{l \in B^u_{t-1}} \left(w_l + \langle \mathbf{v}_u, \mathbf{v}_l \rangle + \frac{1}{|B^u_{t-1}|} \sum_{l' \in B^u_t, l' > l} \langle \mathbf{v}_l, \mathbf{v}'_l \rangle \right)$$

$$\hat{y}(\mathbf{x}) = w_i + \langle \mathbf{v}_u, \mathbf{v}_i \rangle + \frac{1}{|B^u_{t-1}|} \sum_{l \in B^u_{t-1}} \langle \mathbf{v}_i, \mathbf{v}_l \rangle$$

Difference:

- ω_i
- 辅助向量

Thanks!