Factorization Machines

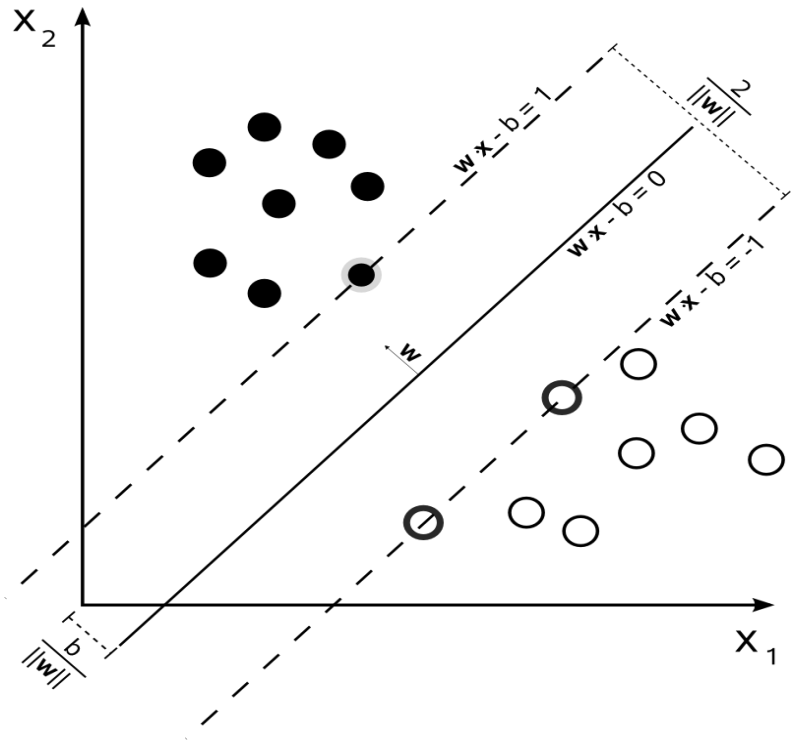
Steffen Rendle
Department of Reasoning for Intelligence
The Institute of Scientific and Industrial Research
Osaka University, Japan

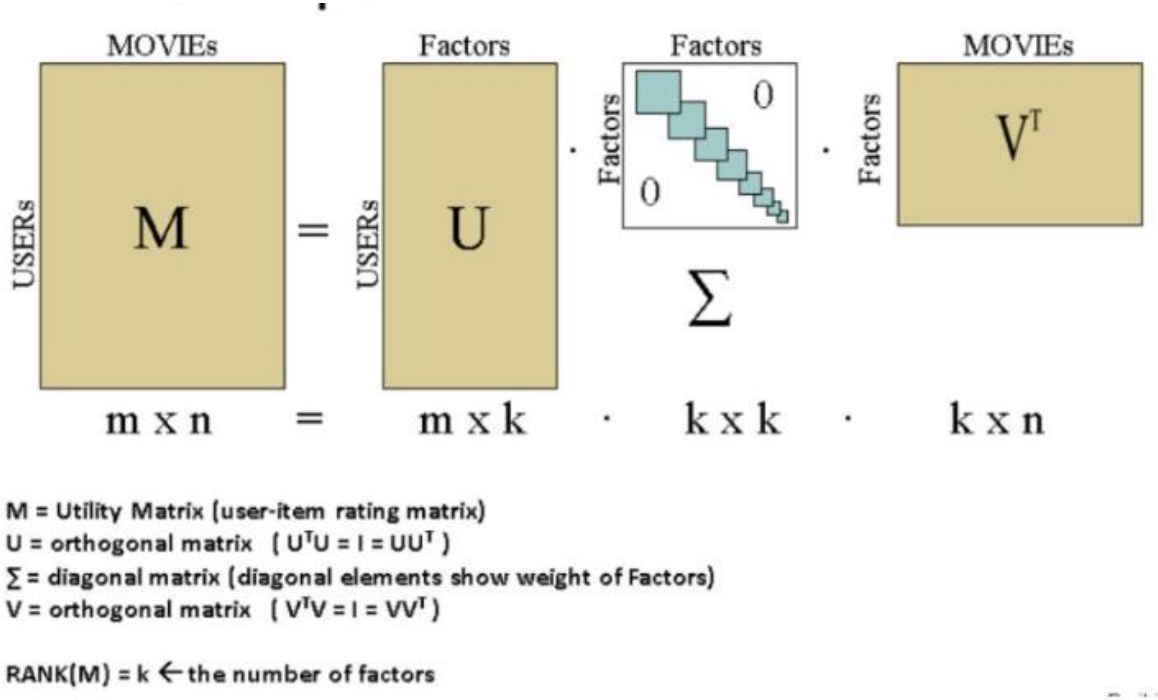
First. MAY. 2021.

- **Youngchun Kwon**

Overview

SVM (Support Vector Machine) + Factorization model





- SVM이 sparse하고 convex(비선형)한 kernel space에서는 효과적이지 않음.
- FM 모델은 일반적인 데이터에는 적용하기 어려움.
- 둘의 장점을 결합한 Factorization Machine이 선형적으로 계산이 가능하며 Sparse한 데이터에 유리.

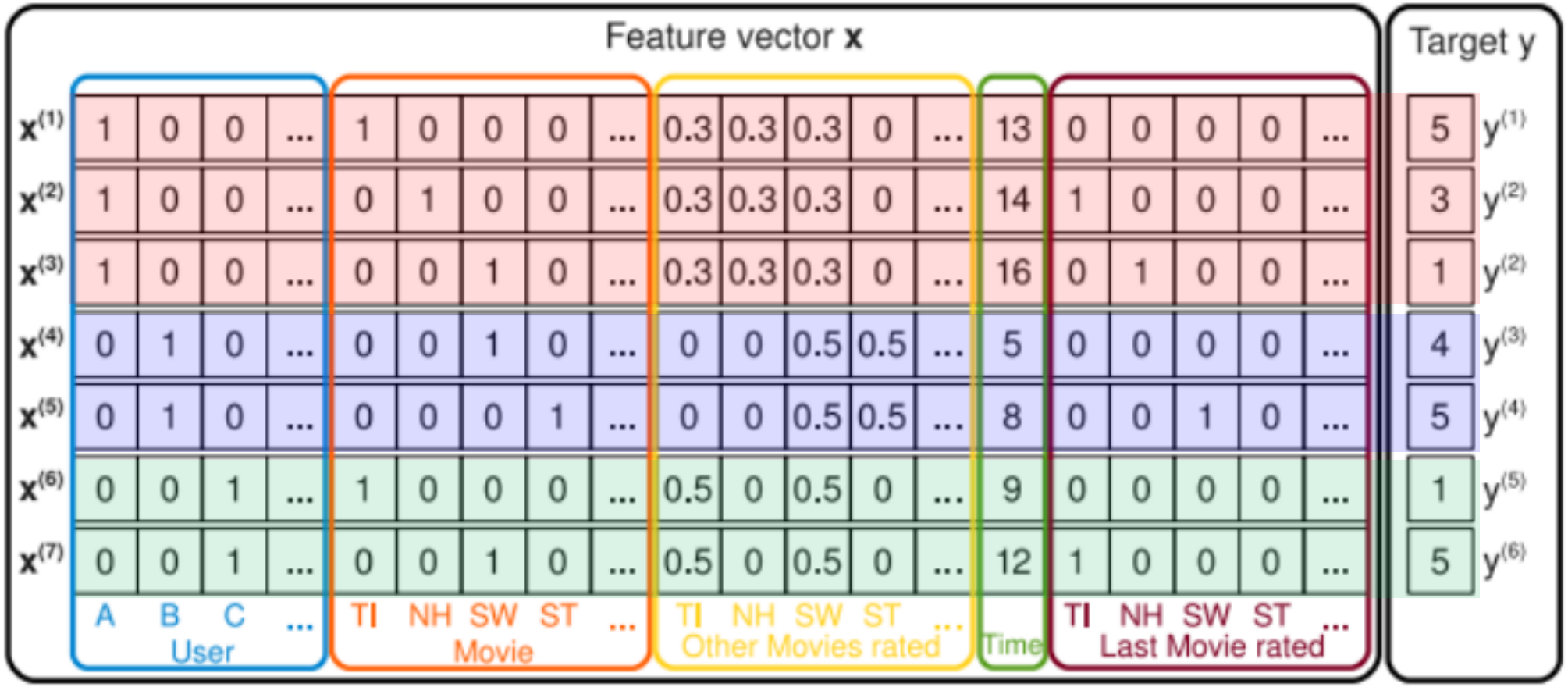
Example

User – Movie Collaborative filtering

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

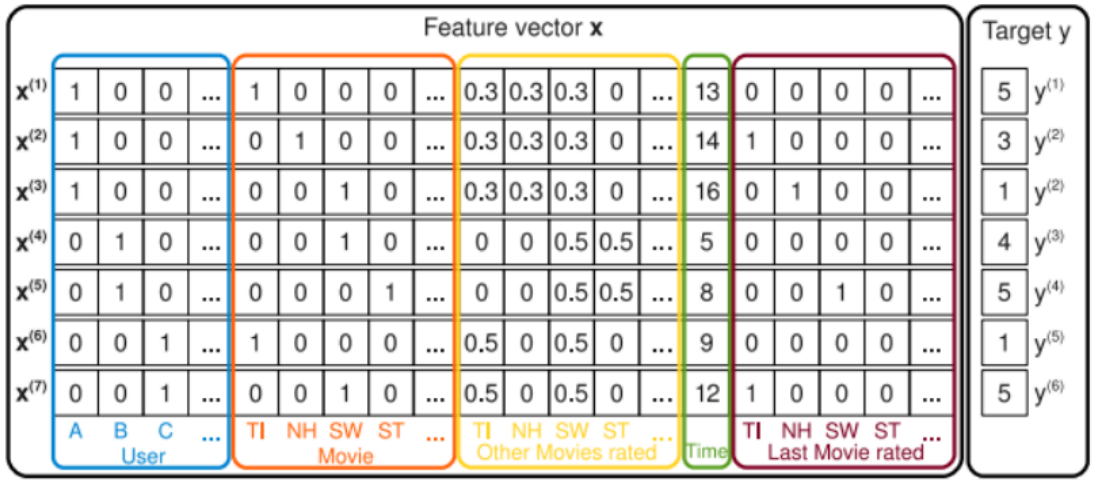
$I = \{\text{Titanic (TI)}, \text{Notting Hill (NH)}, \text{Star Wars (SW)}, \text{Star Trek (ST)}, \ldots\}$

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



FM (Factorization Model)

이해를 위해 Linear Regression에서 시작,
모든 변수를 X 예측하고자 하는 값을 y로 보면

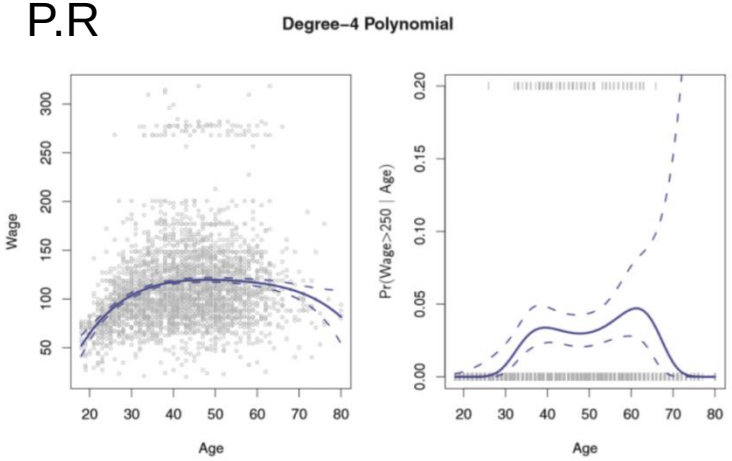


$$\hat{y} = w_0 + \sum_{i=1}^{n} w_i x_i \qquad d=1$$

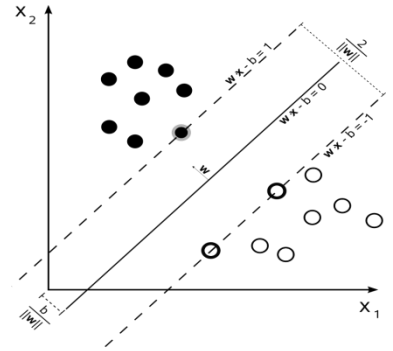
하지만 높은 차원의 데이터, 높은 sparsity, 무시된 변수 간 interaction 등 LM은 여러 한계가 있음.
이를 보완하기 위해 polynomial regression을 고려하거나, 비선형 모델인 SVM, NN 등을 사용할 수 있음.
각 변수 간 interaction을 고려한 polynomial model(2)의 수식은 다음과 같다.

$$\hat{y} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{ij} x_i x_j \qquad d=2$$

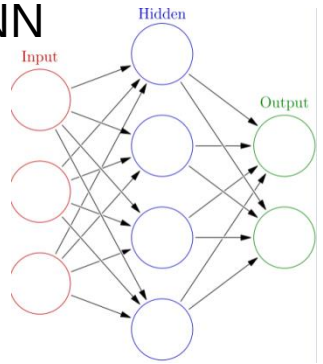
P.R



SVM



NN



FM (Factorization Model)

이렇게 모델링을 하면, LM의 여러 한계를 극복할 수 있지만, paremeter 수의 증가로 연산 복잡도가 증가하게 된다. Factorization machines은 feature interaction vector를 저차원으로 factorize해서 이를 해결.

$$\hat{y} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} (\vec{v}_i \cdot \vec{v}_j) x_i x_j$$

FM 수식은 위와 같음. Interaction weight w_{ij} 는 (V_i, V_j) 의 내적으로 Factorize 됨.

→ 두 변수(x)의 interaction을 K 차원으로 factorizing 된 두 vector의 내적으로 표현
→ 두 (V_i, V_j) 는 Interaction의 latent vector 이고 기존 변수 정보를 k 차원으로 확장해 표현.

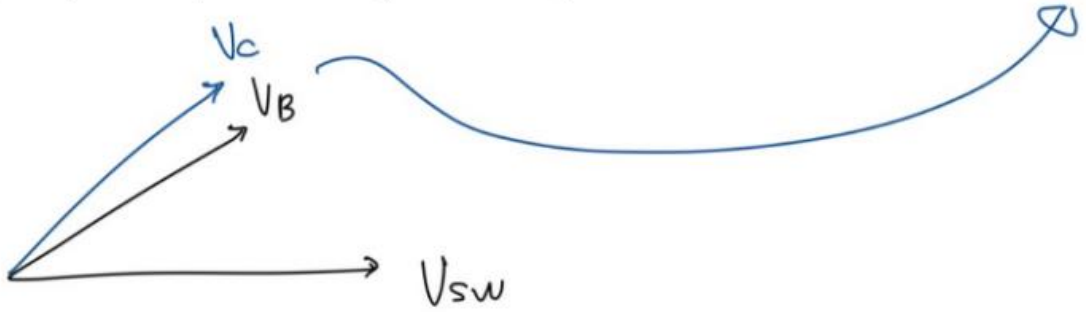
→ Spase 한 상황의 변수들은 서로 독립성이 강하기 때문에 interaction을 포착하기 어렵지만 각 변수 정보를 latent vector로 표현해 latent space 내 독립성을 상쇄하며 interaction 보다 잘 표현할 수 있음.

FM (Factorization Model)

$\langle X_B, X_C \rangle = 0$

$\langle V_B, V_C \rangle = ?$

$\langle V_B, V_{SW} \rangle \approx \langle V_C, V_{SW} \rangle \rightarrow \langle V_B, V_C \rangle$ = high similarity.



- User B, C 는 겹치는 값이 하나도 없으므로 interaction이 0.
- But 해당 두 변수의 latent vector를 이용하면 숨겨진 interaction을 구할 수 있다.
- $V_{sw}와 두 user의 latent vector와의 내적은 유사하다.
- 이를 이용해 두 유저 latent vector가 유사도(interaction)가 높음을 알 수 있다.
- X_A, X_C 는$V_{TI}의 점수차이가 크기 때문에 유사도(취향)이 낮음(다름)

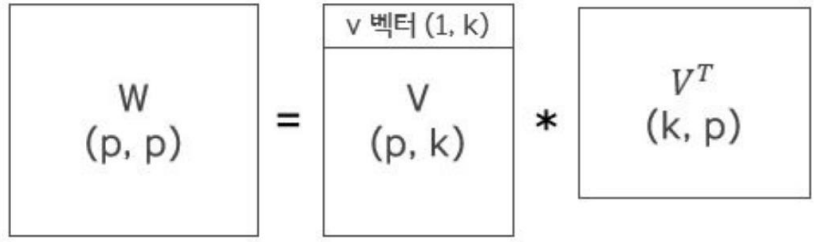
실수 $y\hat{}(x)$ = 실수 w_0 + [x_1 w_1 + ⋯ + x_n w_n] + (< v_1 v_2> x_1 x_2
+ < v_2 v_3> x_2 x_3
+ ⋯
+ < v_{n-1} v_n> x_{n-1} x_n)

FM (Factorization Model)

다음과 같은 전개를 통해 연산 복잡도를 줄일 수 있다.

$O(k\,n^2)$

$$\hat{y} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} (\vec{v}_i \cdot \vec{v}_j) x_i x_j$$



$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} (\vec{v}_i \cdot \vec{v}_j) x_i x_j = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\vec{v}_i \cdot \vec{v}_j) x_i x_j - \frac{1}{2} \sum_{i=1}^{n} (\vec{v}_i \cdot \vec{v}_i) x_i x_i$$

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

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

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

$O(k\,n)$

- x_i: X 데이터 셋의 하나의 행 벡터(feature vector)
- w_0: global bias
- w_i: i번째 변수의 영향력을 모델화 함
- $\hat{w}_{i,j} = < v_i, v_j >$: i, j번째 변수간의 상호작용을 모델화 함
- v 벡터: factor vector

FM (Factorization Model)

B. Factorization Machines as Predictors

Overfitting 문제를 해결하기 위해 L2 Normalization 진행.

C. Learning Factorizatino Machines

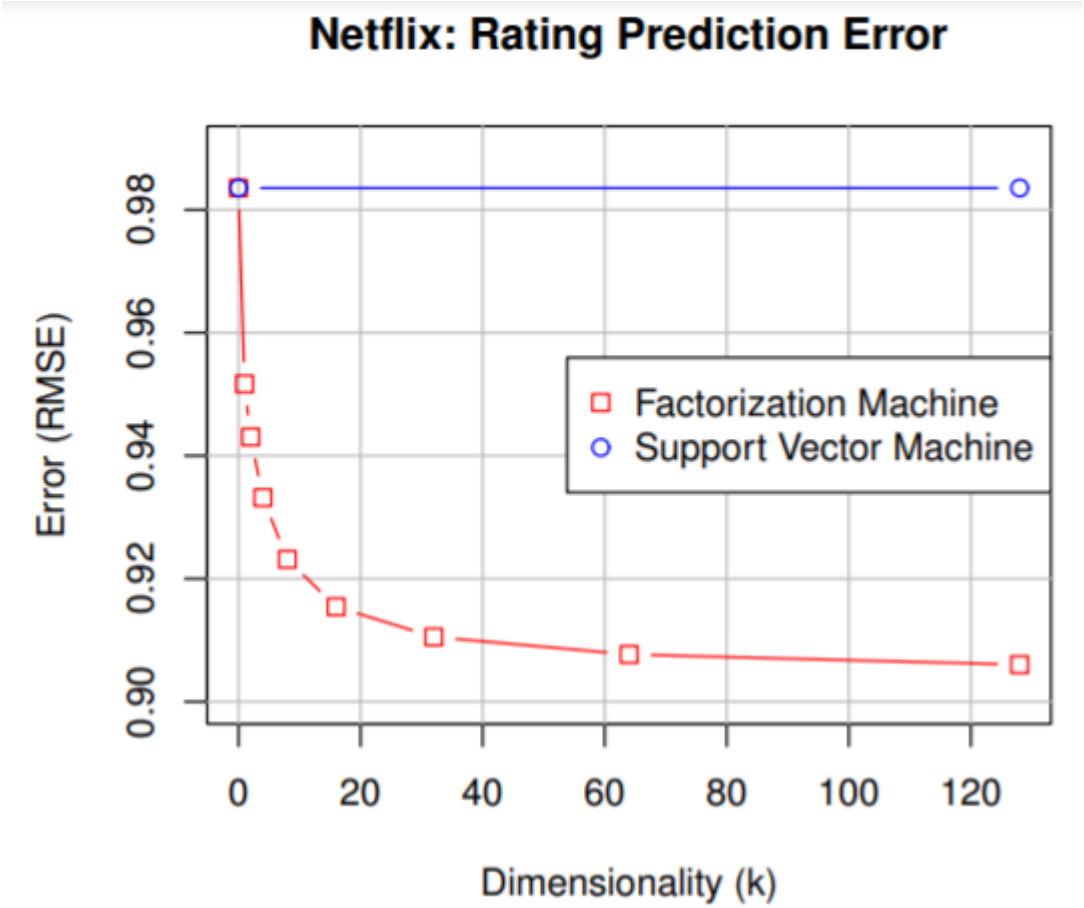
FM은 선형적으로 계산되는 모델 방정식이므로 w_0, w, V 같은 모델 parameter들은 Gradient descent 로 효과적으로 학습 가능.

$$\frac{\partial}{\partial \theta}\hat{y}(\mathbf{x}) = \begin{cases} 1, & \text{if } \theta \text{ is } w_0 \\ x_i, & \text{if } \theta \text{ is } w_i \\ x_i \sum_{j=1}^{n} v_{j,f} x_j - v_{i,f} x_i^2, & \text{if } \theta \text{ is } v_{i,f} \end{cases} \quad (4)$$

E. Summerize

1) 희소한 환경에서도 값들의 상호작용을 추정, 관측되지 않는 상호작용의 일반화 가능.
2) 학습 및 예측에 소요되는 시간 및 parameter 수는 선형적, 따라서 SGD를 이용해 다양한 Loss Function들을 최적화하는 것을 가능.

Validation (FMs VS. SVMs)

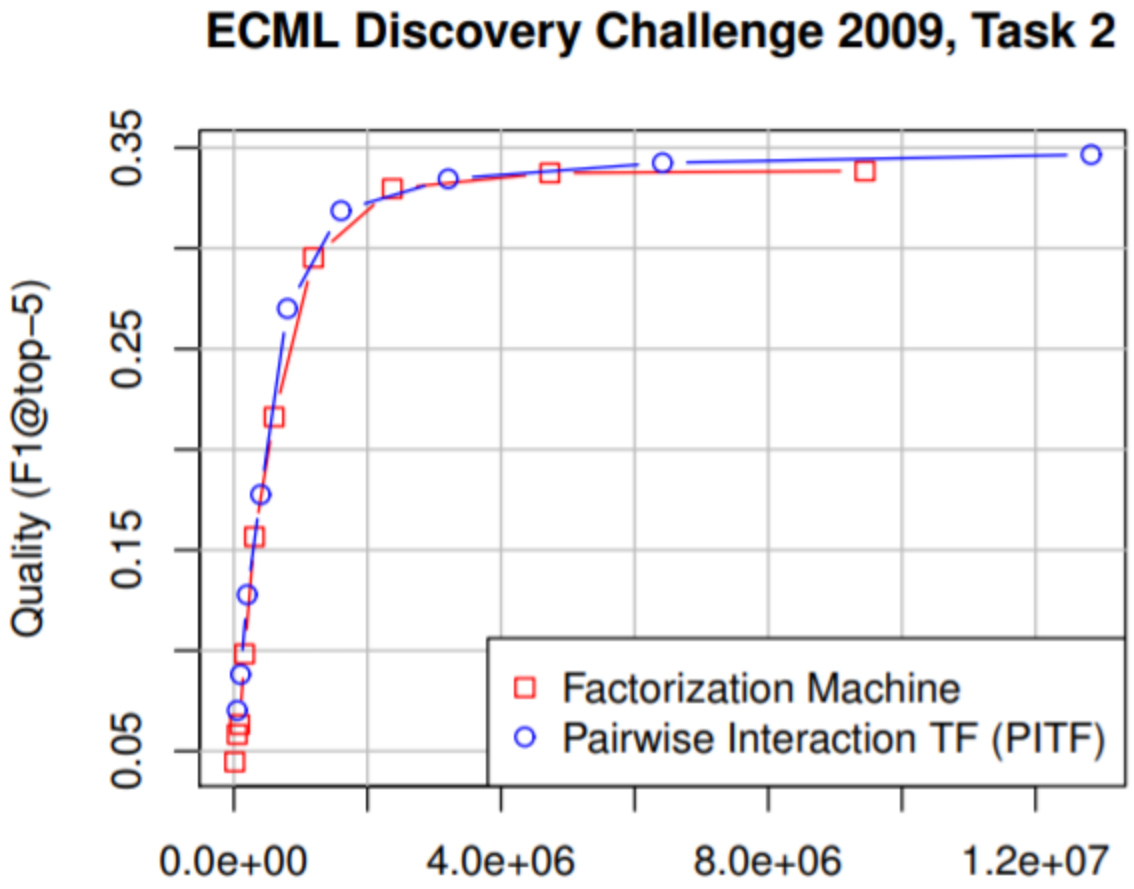


1) The dense parametrization of SVMs requires direct observations for the interactions which is often not given in sparse settings. Parameters of FMs can be estimated well even under sparsity (see section III-A3).

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).

Validation (FMs VS. PITF for Tag Recommendation)



1) Standard factorization models like PARAFAC or MF are not general prediction models like factorization machines. Instead they require that the feature vector is partitioned in m parts and that in each part exactly one element is 1 and the rest 0.

2) There are many proposals for specialized factorization models designed for a single task. We have shown that factorization machines can mimic many of the most successful factorization models (including MF, PARAFAC, SVD++, PITF, FPMC) just by feature extraction which makes FM easily applicable in practice.

Conclusion

(1) FMs are able to estimate parameters under huge sparsity.

(2) The model equation is linear and depends only on the model parameters.

(3) They can be optimized directly in the primal. The expressiveness of FMs is comparable to the one of polynomial SVMs.

Q & A