Factorization Machines

20200045 김건우

Contents

- Introduction
- Prediction Under Sparsity
- Factorization Machines
- Performance of FM
- Code implement

1.Introduction

- Factorization machine = SVM + Factorization model
- Real Value Feature Vector를 이용한 General Predictor
- Calculated in Linear time
- Sparse 한 상황에서도 적용가능

2.Prediction Under Sparsity

(User, Item, Date, rating)으로 이루어진 데이터셋

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

2.Prediction Under Sparsity

Item

User	W	X	Y	Z
A		4.5	2.0	
B	4.0		3.5	
C		5.0		2.0
D		3.5	4.0	1.0

Rating Matrix

=

A	1.2	0.8
B	1.4	0.9
C	1.5	1.0
D	1.2	0.8

User Matrix

X

W	X	Y	Z
1.5	1.2	1.0	0.8
1.7	0.6	1.1	0.4

Item Matrix

2.Prediction Under Sparsity

Feature vector **x** / Target y

	User				Movie					Other Movies rated					Time	Last Movie rated					Target y	
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...		
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y^{(2)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y^{(3)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y^{(4)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y^{(5)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y^{(6)}$

“One-hot encoding”

3.Factorization Machines(FM)

Model equation

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

Global bias   Linear term   Interaction term

Model parameters

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^n, \quad \mathbf{V} \in \mathbb{R}^{n \times k} \qquad (2)$$

Weight   Latent vector

3.Factorization Machines(FM)-Expressiveness

- 어떤 positive define matrix W에 대해 $W = V*V^t$ 을 만족하는 V 가 존재함 (k가 sufficiently large할 때)

⇒이는 k가 충분히 크다면 FM model이 어떠한 interaction matrix라도 표현할 수 있음을 의미

⇒그러나 sparse한 환경에서는 데이터가 충분하지 않다.

⇒K 제한>FM의 표현력⬆ >sparse한 상황에서 일반화 성능⬆

3.Factorization Machines(FM)
-Parameter Estimation Under Sparsity

Feature vector **x** / Target y

	User				Movie					Other Movies rated					Time	Last Movie rated					Target y	
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...		
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y^{(2)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y^{(3)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y^{(4)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y^{(5)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y^{(6)}$

A(Alice)와 ST 사이의 interaction?
=>데이터가 없음

FM은 interaction parameter를 factorizing함으로써
독립성을 깨뜨림...추정 가능!

3.Factorization Machines(FM)

-Parameter Estimation Under Sparsity

Feature vector **x**

	User				Movie					Other Movies rated					Time	Last Movie rated					Target y	
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...		
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y^{(2)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y^{(3)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y^{(4)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y^{(5)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y^{(6)}$

$$\langle V_B, V_{SW} \rangle \approx \langle V_C, V_{SW} \rangle$$

$$V_B \approx V_C$$

3.Factorization Machines(FM)

-Parameter Estimation Under Sparsity

	Feature vector **x**																						Target y	
	User				Movie					Other Movies rated					Time	Last Movie rated								
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...				
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...			5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...			3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...			1	$y^{(2)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...			4	$y^{(3)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...			5	$y^{(4)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...			1	$y^{(5)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...			5	$y^{(6)}$

$$\langle V_A, V_{TI} \rangle \not\approx \langle V_C, V_{TI} \rangle$$

$$\langle V_A, V_{SW} \rangle \not\approx \langle V_C, V_{SW} \rangle$$

$$V_A \not\approx V_C$$

3.Factorization Machines(FM)
-Parameter Estimation Under Sparsity

Feature vector **x**

	User				Movie					Other Movies rated					Time	Last Movie rated					Target y	
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...		
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y^{(2)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y^{(3)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y^{(4)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y^{(5)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y^{(6)}$

$$<V_B, V_{SW}> \approx <V_B, V_{ST}>$$

$$V_{SW} \approx V_{ST}$$

3.Factorization Machines(FM)
-Parameter Estimation Under Sparsity

	Feature vector x																					Target y	
	User				Movie					Other Movies rated					Time	Last Movie rated							
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...			
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...		5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...		3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...		1	$y^{(2)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...		4	$y^{(3)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...		5	$y^{(4)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...		1	$y^{(5)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...		5	$y^{(6)}$

$$<V_A, V_{ST}> \approx <V_A, V_{SW}>$$

3.Factorization Machines(FM)
-computation

O(kn^2)

$$
\begin{aligned}
&\sum_{i=1}^{n}\sum_{j=i+1}^{n}\langle \mathbf{v}_i, \mathbf{v}_j\rangle\, x_i\, x_j \\
=&\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\langle \mathbf{v}_i, \mathbf{v}_j\rangle\, x_i\, x_j - \frac{1}{2}\sum_{i=1}^{n}\langle \mathbf{v}_i, \mathbf{v}_i\rangle\, x_i\, x_i \\
=&\frac{1}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{f=1}^{k} v_{i,f}\, v_{j,f}\, x_i\, x_j - \sum_{i=1}^{n}\sum_{f=1}^{k} v_{i,f}\, v_{i,f}\, x_i\, x_i\right) \\
=&\frac{1}{2}\sum_{f=1}^{k}\left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i\right)\left(\sum_{j=1}^{n} v_{j,f}\, x_j\right) - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2\right) \\
=&\frac{1}{2}\sum_{f=1}^{k}\left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i\right)^2 - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2\right)
\end{aligned}
$$

O(kn)

3.Factorization Machines(FM)

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

- **Regression**: $\hat{y}(\mathbf{x})$ can be used directly as the predictor and the optimization criterion is e.g. the minimal least square error on D.
- **Binary classification**: the sign of $\hat{y}(\mathbf{x})$ is used and the parameters are optimized for hinge loss or logit loss.
- **Ranking**: the vectors $\mathbf{x}$ are ordered by the score of $\hat{y}(\mathbf{x})$ and optimization is done over pairs of instance vectors $(\mathbf{x}^{(\mathbf{a})}, \mathbf{x}^{(\mathbf{b})}) \in D$ with a pairwise classification loss (e.g. like in [5]).

How to learn?

Use SGD!

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

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i \, x_i + \frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{i,f} \, x_i \right)^2 - \sum_{i=1}^{n} v_{i,f}^2 \, x_i^2 \right)$$

d-way Factorization Machine

$$\hat{y}(x) := w_0 + \sum_{i=1}^{n} w_i \, x_i + \sum_{l=2}^{d} \sum_{i_1=1}^{n} \cdots \sum_{i_l=i_{l-1}+1}^{n} \left(\prod_{j=1}^{l} x_{i_j} \right) \left(\sum_{f=1}^{k_l} \prod_{j=1}^{l} v_{i_j,f}^{(l)} \right) \quad (5)$$

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

3.Factorization Machine-Summary

- FM : using factorized interactions
- Advantage

    1. 높은 sparsity 상에서도 모든 Interaction을 추정할 수 있음.
    2. prediction과 learning의 시간과 파라미터의 수가 선형적.

    (SGD를 이용한 학습이 가능, 다양한 손실 함수에 적용가능)

4. Performance of FM
FM vs SVM

1. SVM과 달리 FM은 sparse한 상황에서도 잘 작동함.
2. SVM은 dual상에서 학습해야 하지만 FM은 바로 학습이 가능하다.
3. FM의 model equation은 training data와 무관하다.

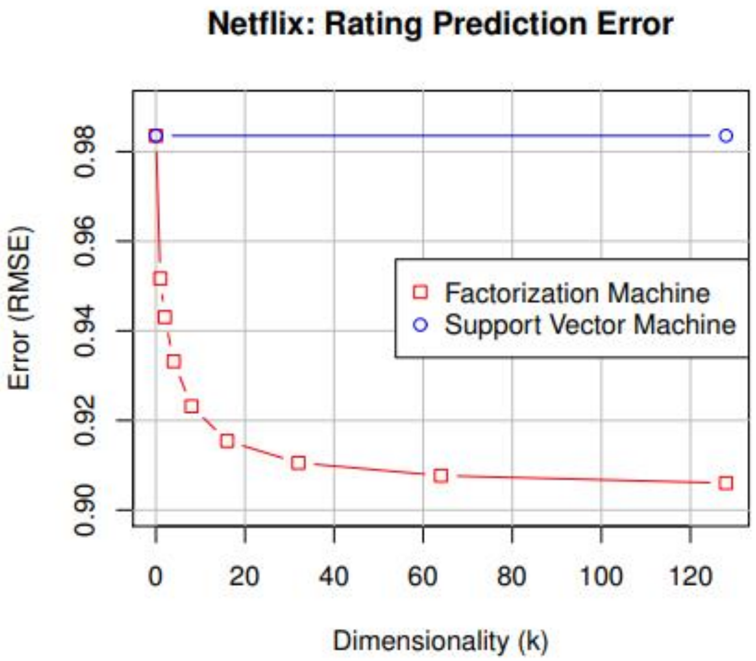

Fig. 2. FMs succeed in estimating 2-way variable interactions in very sparse problems where SVMs fail (see section III-A3 and IV-B for details.)

4.Performance of FM
FM vs OTHER FACTORIZATION MODEL

1. PARAFAC or MF와 같은 standard factorization model은 general prediction model이 아님.
2. Single task를 위해 많은 proposals(제안)이 필요...FM은 더 쉽게 적용 가능하고 성능은 비슷하다!

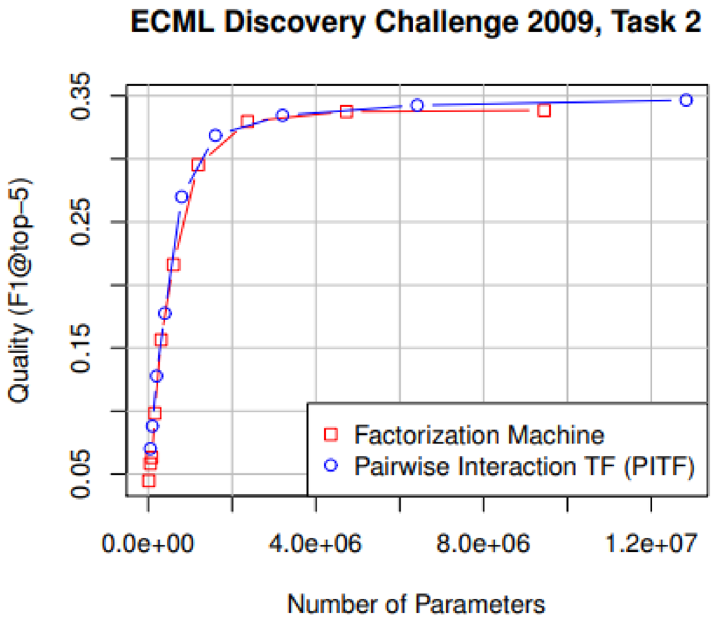

Fig. 3. Recommendation quality of a FM compared to the winning PITF model [3] of the ECML/PKDD Discovery Challenge 2009. The quality is plotted against the number of model parameters.

5. Code implement

```python
class  MyFM:

    def __init__(self, k=2):
        self.k = k


    def fit(self,X, y, learning_rate=0.01):
        # initialize
        n_samples, coef_size = X_train.shape
        self.w0 = np.random.randn(1)
        self.w = np.random.randn(coef_size)
        self.v = np.random.randn(coef_size, self.k)



        for i in range(n_samples):
            diff = y.iloc[i]- self._predict(X.iloc[i:i+1,:])
            loss = (diff**2)/2
            loss_gradient = diff


            self.w0 -= learning_rate * loss_gradient
            for j in range(coef_size):
                self.w[j] -= learning_rate * (loss_gradient * X.iloc[i:i+1,j:j+1].values)

            for k in range(self.k):
                for j in range(coef_size):
                    term = np.sum(np.dot(X.iloc[i:i+1,:],self.v[:,k:k+1]), axis=1) - np.dot(X.iloc[i:i+1,j:j+1], self.v[j:j+1,k:k+1])
                    self.v[j][k] -= learning_rate * ( loss_gradient * X.iloc[i:i+1,j:j+1].values* term)



    def _predict(self,X):
        linear = self.w0 + np.dot(X, self.w)
        first_term = (np.dot(X,self.v))**2 - np.dot(X**2,self.v**2)
        interaction = 0.5 * np.sum(first_term, axis=1)
        score = linear + interaction
        return score
```