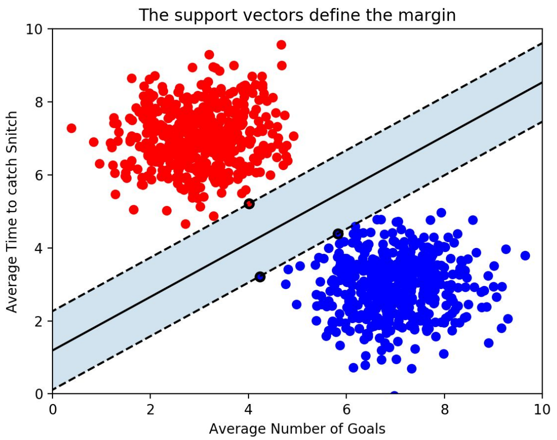
Factorization Machines

Jiho Kang
2022.04.23

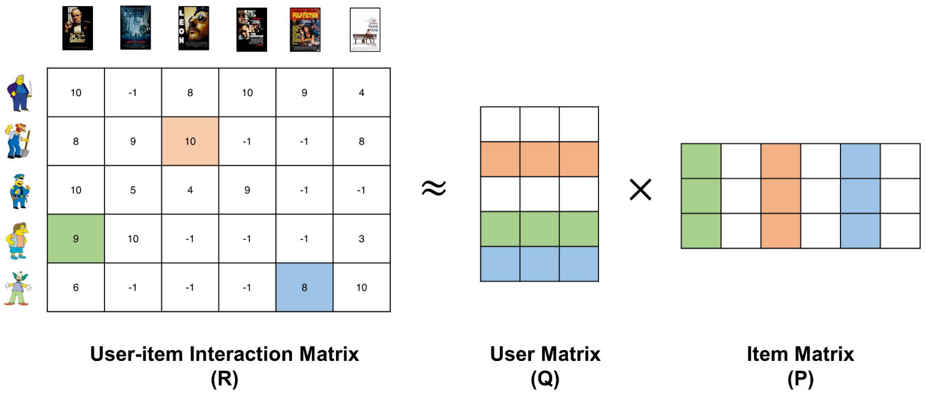
Factorization Machines

- Factorization Machines (FM) = SVM + factorization models (matrix factorization)

Support Vector Machine(SVM)



Matrix Factorization



Factorization Machines

Support Vector Machine(SVM)

- General model
- Dual form transformation
- More than linear time (n^2)
- Few interaction

Matrix Factorization

- Specialized model
- Expert knowledge
- Few interaction

FM

- General & Specialized model
- No dual form transformation
- Linear time
- No need expert knowledge
- Interactions

Factorization Machines

	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	...				
$x^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...			5	$y^{(1)}$
$x^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...			3	$y^{(2)}$
$x^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...			1	$y^{(2)}$
$x^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...			4	$y^{(3)}$
$x^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...			5	$y^{(4)}$
$x^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...			1	$y^{(5)}$
$x^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...			5	$y^{(6)}$

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

Factorization Machines

	Feature vector $\mathbf{x}$																						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	$y^{(6)}$

Fig. 1. Example for sparse real valued feature vectors $\mathbf{x}$ that are created from the transactions of example 1. Every row represents a feature vector $\mathbf{x}^{(i)}$ with its corresponding target $y^{(i)}$. The first 4 columns (blue) represent indicator variables for the active user; the next 5 (red) indicator variables for the active item. The next 5 columns (yellow) hold additional implicit indicators (i.e. other movies the user has rated). One feature (green) represents the time in months. The last 5 columns (brown) have indicators for the last movie the user has rated before the active one. The rightmost column is the target – here the rating.

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

where the model parameters that have to be estimated are:

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

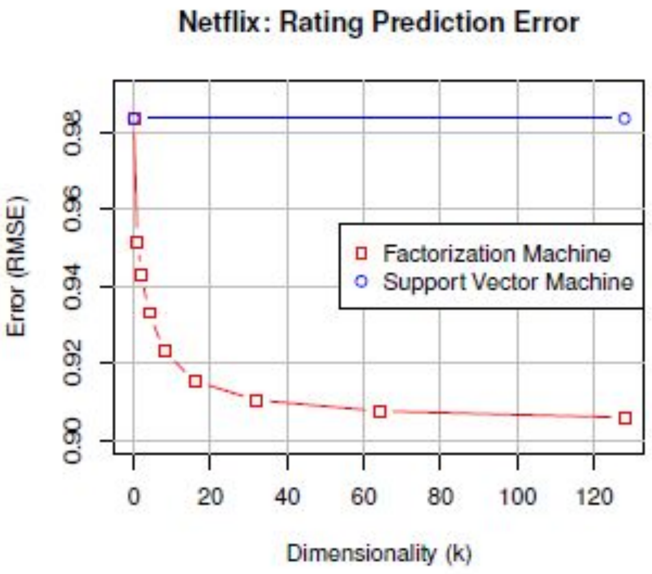
And $\langle \cdot, \cdot \rangle$ is the dot product of two vectors of size k:

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle := \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f} \qquad (3)$$

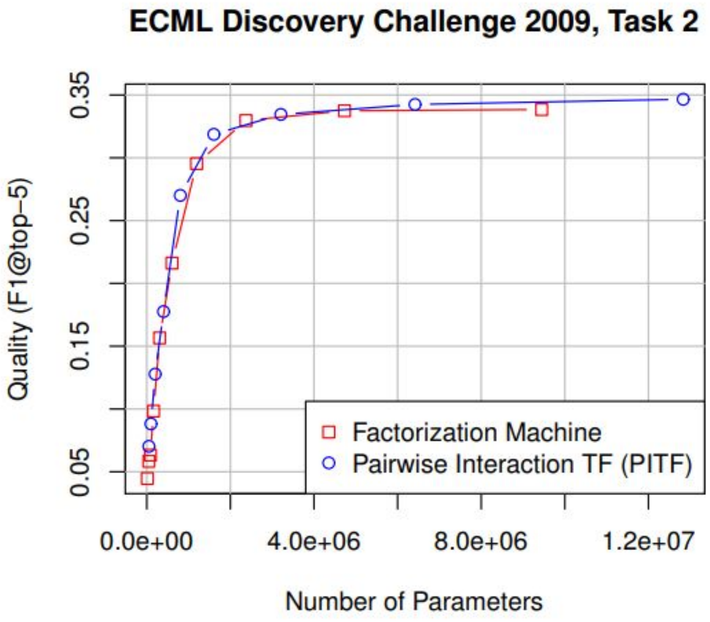
$$\begin{aligned}
&\sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle\, x_i\, x_j \\
&= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle\, x_i\, x_j - \frac{1}{2} \sum_{i=1}^{n} \langle \mathbf{v}_i, \mathbf{v}_i \rangle\, x_i\, x_i \\
&= \frac{1}{2}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

Factorization Machines

FMs vs SVMs



FMs vs Other models



Factorization Machines

- FMs allow parameter estimation under very **sparse data** where SVMs fail.
- FMs have **linear** complexity, can be optimized in the **primal** and do not rely on support vectors like SVMs. It shows that FMs scale to large datasets like Netflix with 100 millions of training instances.
- FMs are a **general predictor** that can work with any real valued feature vector. In contrast to this, other SOTA factorization models work only on very restricted input data.
- The **interactions between values** can be estimated even under high sparsity. Especially, it is possible to **generalize to unobserved interactions.**

References

https://greeksharifa.github.io/machine_learning/2019/12/21/FM/

https://hongl.tistory.com/242

https://jae-eun-ai.tistory.com/21

https://hyunlee103.tistory.com/69

https://cleancode-ws.tistory.com/160