Factorization Machines (FM) and Field-aware Factorization Machines (FFM)

Hung-Hsuan Chen
https://www.ncu.edu.tw/~hhchen/

Linear model

- Formula

$$y = \sum_{i=0}^{d} \theta_i x_i$$

  - y: target
  - $x_1, \ldots, x_d$: features
  - x_0: bias
  - $\theta_0,\ \theta_1 \ldots,\ \theta_d$: parameters to learn
- The model can capture each feature's influence on the target

Degree-2 polynomial model (poly2)

- Formula

$$y = \sum_{i=0}^{d} \theta_i x_i + \sum_{(j,k) \in C_2} \theta_{j,k} x_j x_k$$

  - y: target
  - $x_1, \ldots, x_d$: features
  - x_0: bias
  - $\theta_0,\ \theta_1 \ldots,\ \theta_d, \theta_{1,2}, \theta_{1,3}, \ldots, \theta_{d-1,d}$: parameters to learn
  - C_2: 2-combination of elements in $[x_1, \ldots, x_d]$
- The model can capture
  - Each feature's influence on the target
  - Each feature-pair's influence on the target

Factorization machines (FM)

- Formula

$$y = \sum_{i=0}^{d} \theta_i x_i + \sum_{(j,k) \in C_2} \left\langle \boldsymbol{v}_j, \boldsymbol{v}_k \right\rangle x_j x_k$$

  - y: target
  - $x_1, \ldots, x_d$: features
  - x_0: bias
  - $\theta_0,\ \theta_1 \ldots,\ \theta_d, \boldsymbol{v}_1,\ \boldsymbol{v}_d$: parameters to learn, each $\boldsymbol{v}_j$ is a vector of length ℓ
  - C_2: 2-combination of elements in $[x_1, \ldots, x_d]$

Poly2 vs FM (# parameters)

- If we have *d* features
  - # parameters for poly-2:

$$(d+1)+\binom{d}{2}=d+1+\frac{d(d-1)}{2}\approx O\left(d^2\right)$$

  - # parameters for FM (assuming the length of the vector $\boldsymbol{v}_i$ is ℓ:

$$(d+1)+\ell d=1+\left(\ell+1\right)d\approx O\left(\ell d\right)$$

- If $d \gg \ell$, FM has fewer parameters to learn
  - FM is probably more appropriate when we have large but sparse features

Example: ad classification

Country	Day	Ad type	Clicked?
USA	Thanksgiving	Movie	1
China	Chinese New Year	Game	0
China	Thanksgiving	Game	1

- Task: given features, predict click or not

Example take from: https://www.slideshare.net/EvgeniyMarinov/factorization-machines-and-applications-in-recommender-systems

Example: ad classification (cont')

- Standard one-hot encoding

USA	China	Thanksgiving	Chinees new year	Movie	Game	Clicked?
1	0	1	0	1	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1

- Very large feature space
- Very sparse samples

Example: ad classification (cont')

- Features might be more important in "pairs"
  - Country == "USA" and Day == "Thanksgiving"
  - Country == "China" and Day == "Chinese new year"
- If we create features for every pair of features
  - Number of features goes from d to $\binom{d}{2}$
  - Samples: still sparse

Gradients

$$\hat{y} = \theta_0 + \sum_{i=1}^{d} \theta_i x_i + \sum_{(j,k)\in C_2} \left\langle \boldsymbol{v}_j, \boldsymbol{v}_k \right\rangle x_j x_k$$

$$L := \left(y - \hat{y}\right)^2$$

- $\frac{\partial L}{\partial \theta_0} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \theta_0} = -2(y - \hat{y})$
- $\frac{\partial L}{\partial \theta_i} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \theta_i} = -2(y - \hat{y})x_i \; (1 \le i \le d)$
- $\frac{\partial L}{\partial v_{jf}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial v_{jf}} = -2(y - \hat{y})v_{kf}x_j x_k$
- $\frac{\partial L}{\partial v_{kf}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial v_{kf}} = -2(y - \hat{y})v_{jf}x_j x_k$

FFM example

Publisher (P)	Advertiser (A)	Gender (G)	Clicked?
ESPN	Nike	Male	Yes

- On ESPN, a male clicked an ad about Nike
- For FM:

$$\hat{y} = \theta_0 + \sum_{i=1}^{d} \theta_i x_i + \boldsymbol{v}_{ESPN} \cdot \boldsymbol{v}_{Nike} + \boldsymbol{v}_{ESPN} \cdot \boldsymbol{v}_{Male} + \boldsymbol{v}_{Nike} \cdot \boldsymbol{v}_{Male}$$

  - Every feature has one corresponding latent vector to learn
  - E.g., $\boldsymbol{v}_{ESPN}$ is used to learn the effect with Nike ($\boldsymbol{v}_{ESPN} \cdot \boldsymbol{v}_{Nike}$) and Male ($\boldsymbol{v}_{ESPN} \cdot \boldsymbol{v}_{Male}$)
- However, Nike and Male belong to different fields, the effects of $\boldsymbol{v}_{ESPN}$ on ($\boldsymbol{v}_{ESPN} \cdot \boldsymbol{v}_{Nike}$) and ($\boldsymbol{v}_{ESPN} \cdot \boldsymbol{v}_{Male}$) could be different

FFM example (cont')

Publisher (P)	Advertiser (A)	Gender (G)	Clicked?
ESPN	Nike	Male	Yes
Disney	LEGO	Male	No

- For FFM:

$$\hat{y}_1 = \theta_0 + \sum_{i=1}^{d} \theta_i x_i + \boldsymbol{v}_{ESPN,A} \cdot \boldsymbol{v}_{Nike,P} + \boldsymbol{v}_{ESPN,G} \cdot \boldsymbol{v}_{Male,P} + \boldsymbol{v}_{Nike,G} \cdot \boldsymbol{v}_{Male,A}$$

$$\hat{y}_2 = \theta_0 + \sum_{i=1}^{d} \theta_i x_i + \boldsymbol{v}_{Disney,A} \cdot \boldsymbol{v}_{LEGO,P} + \boldsymbol{v}_{Disney,G} \cdot \boldsymbol{v}_{Male,P} + \boldsymbol{v}_{LEGO,G} \cdot \boldsymbol{v}_{Male,A}$$

  - Every feature has $K-1$ corresponding latent vector to learn (K: number of "fields")
  - E.g., to learn the effect of (ESPN, Nike), $\boldsymbol{v}_{ESPN,A}$ is used because Nike belongs to field Advertiser. However, to learn the effect of (ESPN, Male), $\boldsymbol{v}_{ESPN,G}$ is used because Male belongs to field Gender

Summary

- FM as an extension of linear model
- FM as a variation of poly2 model
- FFM as an extension of FM
- FM and FFM are especially useful when the features are large and sparse
  - E.g., advertisement click prediction

Quiz

- If we know that targets are influenced by each feature and each pair of features
  - Linear regression + original features can make good predictions (true or false)
  - Linear SVM + original features can make good predictions (true or false)
  - Polynomial kernel is helpful (true or false)
  - Linear regression + poly2 feature engineering is helpful (true or false)
  - Linear SVM + poly2 feature engineering is helpful (true or false)
  - Factorization machine is helpful (true or false)

create such feature
by ourselves