Friday, April 5, 2024 9:02 PM

Matrix Factorization-5

→ Going through the 1 million Case study winner solution

→ Apply the learning.

→ How MF actually helps in Feature Engineering.

2008-2009

RMSE ~~RMSE~~ reduce by 10%.

RMSE = 40%
New RMSE = 36%

→ 10% reduction

Winner solution
~~tons~~ of model (30-40 models) along with ~~bagging~~ boosting

MF → outcome of the winner model

Sparse matrix

Sparse matrix

U → user matrix

I → Item matrix

$$\begin{array}{c|c} & I_1 \quad I_2 \quad \cdots \quad I_m \\ \hline \begin{array}{c} u_1 \\ u_2 \\ \vdots \\ u_n \end{array} & \begin{bmatrix} & & & \\ & & & \\ & & & \end{bmatrix} \end{array}$$

User

p_u : d-dimensional user matrix

q_i : d-dimension for movie matrix

r_{ui} : rating of movie

r

$$\begin{array}{cc} & q_1 \quad q_2 \quad q_3 \quad q_4 \quad \cdots \quad q_m \\ \begin{array}{c} p_1 \\ p_2 \\ p_3 \\ \vdots \\ p_n \end{array} & \begin{bmatrix} & & & \\ & & & \\ & & & \end{bmatrix} \end{array}$$

$r_2 \quad P \cdot Q^T$

$A_{n \times m} = B_{n \times d} \cdot C^T_{d \times m}$

Error: Actual − Predicted

$$\text{Error} \quad \min_{p_u q_i} \sum_{u,i} \Big| \underbrace{r_{ui}}_{\text{actual}} - \underbrace{q_i^T \cdot p}_{\text{predicted}} \Big)^2$$

$$\text{Error/Cost function} = \boxed{\min_{p_u q_i} \sum_{u,i} \big(\underbrace{r_{ui}}_{\text{actual}} - \underbrace{q_i^T}_{\text{movie}} \cdot \underbrace{p_u}_{\text{user matrix}}\big)^2}$$

$$+ \lambda \left(\sum \|q_i\|^2 + \sum \|p_u\|^2 \right)$$

Regularization controls overfit

$$\boxed{r_{n\times m} = P_{n\times d} \cdot q^T_{d\times m}}$$

$d \uparrow$

2nd change

bias

IMDB ROTTEN TOMATOES

Rating of Titanic by Lokesh

Rating of Titanic by Lokesh

→ The kind of audience

1> Avg. rating of all the movies in the platform, μ

2> General movie quality.

3> User

Predict whether a user will like a movie

$$\boxed{\text{rating} = \underbrace{p}_{\text{user}}\,\underbrace{q^T}_{\text{movie}} + \mu + b_i + b_u}$$

Error

$$\|r_{ui} - pq^T - \mu - b_i - b_u\|^2 + \lambda\left(\|p\|^2 + \|q\|^2 + \mu^2 + b_u^2 + b_i^2\right)$$

u

<10 — Animation →

15-20 — Super hero + thriller + Romantic +

>25, >30 → Action + Biography

Romantic
+
10-15 (18-22) comedy YU
Hindi → College Band → Beetles

$$\text{rating } r_{ui}(t) = \mu + b_u(t) + b_i(t) + q_i^T \cdot p_u(t)$$

μ ↓ avg. r

1990 rating × 2020

$$\text{Simp} = (r_{ui} - p \cdot q^T)$$

Summary

→ Netflix winner model use MF

→ Plain model → They broke sparse matrix into user and item

$$r = p \cdot q^T$$

→ bias:

$$r = p \cdot q^T + \mu + b_u + b_i$$

→ bias Temporal

$$r = p \cdot q^T + b_u(t) + b_i(t) + \mu$$

→ bias Temporal

$r(t) = P \cdot q_i^T(t) + b_u(t) + b_i(t) + \mu$

$r = \mu + b_u + b_i$

μ ↓ average rating of all the movies

b_u ↓ bias + avg rating by a user

b_i ↓ average rating of movie

building a time sensitive recommendation

YouTube

Idea-1

$$\boxed{r = \mu + b_u + b_i + P \cdot q^T}$$

↓ same treatment To all data

Idea-2

More weightage to recent data.

Error

$\min_{P, q, \mu, b_u, b_i} \sum_{u,i} w_{ui} (r - \mu - b_u - b_i - P \cdot q_i^T) + \lambda (\text{regularizer})$

weight

for a user u, I am trying predict rating for item i Rec sys

sys

pure recency

k-nearest neigh

Hybrid etc.

Item matrix

Tonight

MF

$A_{n \times m}$ → Sparse matrix where most of the cells are empty.

$$\begin{array}{c|c} & \begin{matrix} m_1 & m_2 & m_3 \end{matrix} \\ \hline \begin{matrix} u_1 \\ u_2 \\ u_3 \end{matrix} & \begin{bmatrix} ! & x & x \\ x & ! & x \\ ! & x & ! \end{bmatrix} \end{array}$$

$$A_{n \times m} = \underline{B}_{n \times d} \cdot \underline{C^T_{d \times m}}$$

$\underline{B}$ → user, $\underline{C}$ → Item

Error = Actual − Predicted.

Netflix solution

MF

1) Plain MF $\Rightarrow r_{ui} = P \cdot q^T$

2) Regulariz $\quad ||(r_{ui} - P \cdot q^T)||^2 + \lambda \left[||P||^2 + ||q||^2 \right]$

3) bias $\rightarrow \mu$, $\rightarrow b_u$, $\rightarrow b_i$

4) Temporal $\rightarrow \mu$, $\rightarrow b_u(t)$, $\rightarrow b_i(t)$

How real time model work

1) MF + bias + Temporal

weight ~~vector~~ matrix

$$
\begin{array}{c|cccc}
R & I_1 & I_2 & I_3 & I_n \\
\hline
u_1 & & & & \\
u_2 & & & & \\
u_3 & & & & \\
u_4 & & & &
\end{array}
$$

r_{33}

$r_{33} \approx \mu + b_u(t) + b_i(t) + P \cdot q^T$

MF as FE

movies

Netflix — movies
show
movie

user

Sparse matrix
$\sum r_{ui}$

$$r_{ui} = p_u q_i^T$$

$r \to n \times m$
$P = n \times d$ → d-dimension
$q = n \times d$

P_u → country, language, Age
movie → lang, genre, target age

user p_u → d feature
movies, q_i → d feature

Example

search ... about

Example

Amazon → Eshan $\xrightarrow{\text{search}}$ learn about eshan

User: Eshan : { India, Bangalore, electronics, low brand, discounts }

User

Eshan : { 0.11, 1.78, , }

lokesh :

Ashish :

Raeesa :

feature store

Raeesa ← for Recommendation

Today :

$r = p \cdot q^T$

q – item vector → Item Item similarity

Raeesa → airpods

airpods case

airpod charger

airpod charger

user → feature store to make recommendat.

user → search → Item Item similarity.

data factor

Arjun	21, no money, late night, cat video, cricket
Rae	19, lot of money, food item

Facebook / IG

Arjun → IG

Raesa → food ads.

MF - User × Item

MF Feature Engineering for text

Times of India
Chetan Bhagat
Amish Tripathi
The Hindu blogs
wiki

→ document

w2 = Poop bagati

list of all the words

w_1 w_2 w_3 — — — — — w_m

5

Total document d_1 d_2

document d_1
d_2
$\vdots$
d_n

$$W_{n \times m} = D_{n \times d} \times A^T_{d \times m}$$

D = list of documents with d feature.
= document similarity.

Type something → autosuggestion.

Word vector $\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$

Together

Cooccurrence matrix

$W_m \times W_m$

$$\begin{array}{c|cccccc} & w_1 & w_2 & \ldots & \text{to} & \text{say} & w_n \\ \hline w_1 & & & & & & \\ \text{talk } w_2 & & & & & & \end{array}$$

Vizualize what coords could be grouped together

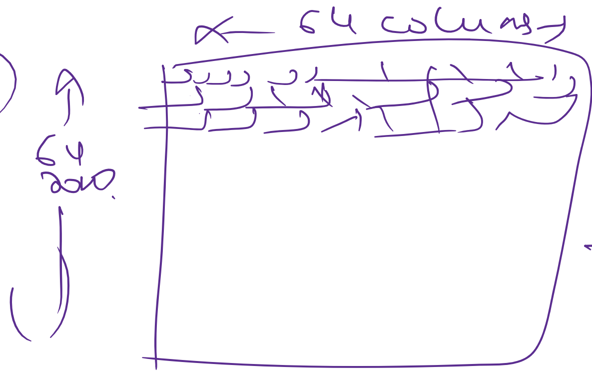
— MF face recognition —

Images are represented

2D-matrix r rows and c-column.

r = 64 , c = 64

rbg



→ $r \times c$ → for k.

400 imge..

matric

← 4096 →

I_1

I_2

I_3

100

I_{100}

SVD, PCA → special case of matrix

PCA → Theory of MF

→ How Netflix Competition led to a great research → MF

→ MF → Feature Engineering