Agenda

- Netflix prize Solution Review
- Building a Time Sensitive RecSys
- MF for Feature Engineering
    - Implicit feature
- MF for Text Data
- MF for Images : Eigen Faces

Netflix prize Solution Review

Research Papers :

- Summary of Major ideas :

https://datajobs.com/data-science-repo/Recommender-Systems-%5BNetflix%5D.pdf

- Winning Solution Paper :

https://www2.seas.gwu.edu/~simhaweb/champalg/cf/papers/

Netflix → 2007 - 2009 → 1 million Dollar

Goal : To predict Rating of movie by user

Task : Reduce the existing RMSE by 10% or more

Final Solution: Tons of Smaller Models
+
Boosting
+
Blending

Let's Review :

Paper Notation

- $\rightarrow r_{ui}$ (rating user u will give to movie i : A_{ij})
- $\rightarrow p_u$ (User u) : u_i
- $\rightarrow q_i$ (movie i) : i_j

$$p_u = \begin{bmatrix} 0.9 \\ 0.8 \\ \vdots \\ d \end{bmatrix}_{d\times 1} \qquad q_i = \begin{bmatrix} 0.7 \\ 0.9 \\ \vdots \\ d \end{bmatrix}_{d\times 1}$$

$$r_{ui} \Rightarrow \text{Scalar} : 4 \quad 1\times 1$$

$$\min_{p_u, q_i} \sum_{u,k} \left(r_{ui} - \underbrace{\underset{1\times d}{q_i^T} \cdot \underset{d\times 1}{p_u}}_{\text{Predicted Rating}} \right)^2$$

* Regularization

$$\min_{p_u, q_i} \sum_{u,k} \left(r_{ui} - \underset{1\times d}{q_i^T} \cdot \underset{d\times 1}{p_u} \right)^2 + \lambda \left(\sum_i \|q_i\|^2 + \sum_u \|p_u\|^2 \right)$$

* Adding Bias

Lets say we want to estimate a user joe's rating of Titanic

① Avg rating of all movies $\mu = 3.7$ *

② Titanic Avg rating = $\mu + 0.5$

③ Joe is critical ⇒ $\mu + 0.5 - 1$

Hence the factor that determine a User's Rating of Movie

- (Global) μ : Platform's Avg Rating
- bias of Movie = b_i (Item avg)
- User's Bias = b_u

Hence, Net Rating :

$$\boxed{\hat{r}_{ui} = \mu + \underbrace{b_u + b_i}_{\text{biases}} + q_i^T \cdot p_u}$$

p_u, q_i, μ, b_u, b_i act as parameters now. Therefore the optimization problem becomes:-

$$min_{p_u,q_i,\mu,b_u,b_i}\Sigma_{u,i}(r_{ui} \underbrace{- \mu - b_u - b_i - q_i^T.p_u}_{\hat{r}_{ui}})^2 + \lambda * \Sigma(||q_i||^2 + ||p_u||^2 + \mu^2 + b_u^2 + b_i^2)$$

i.e.

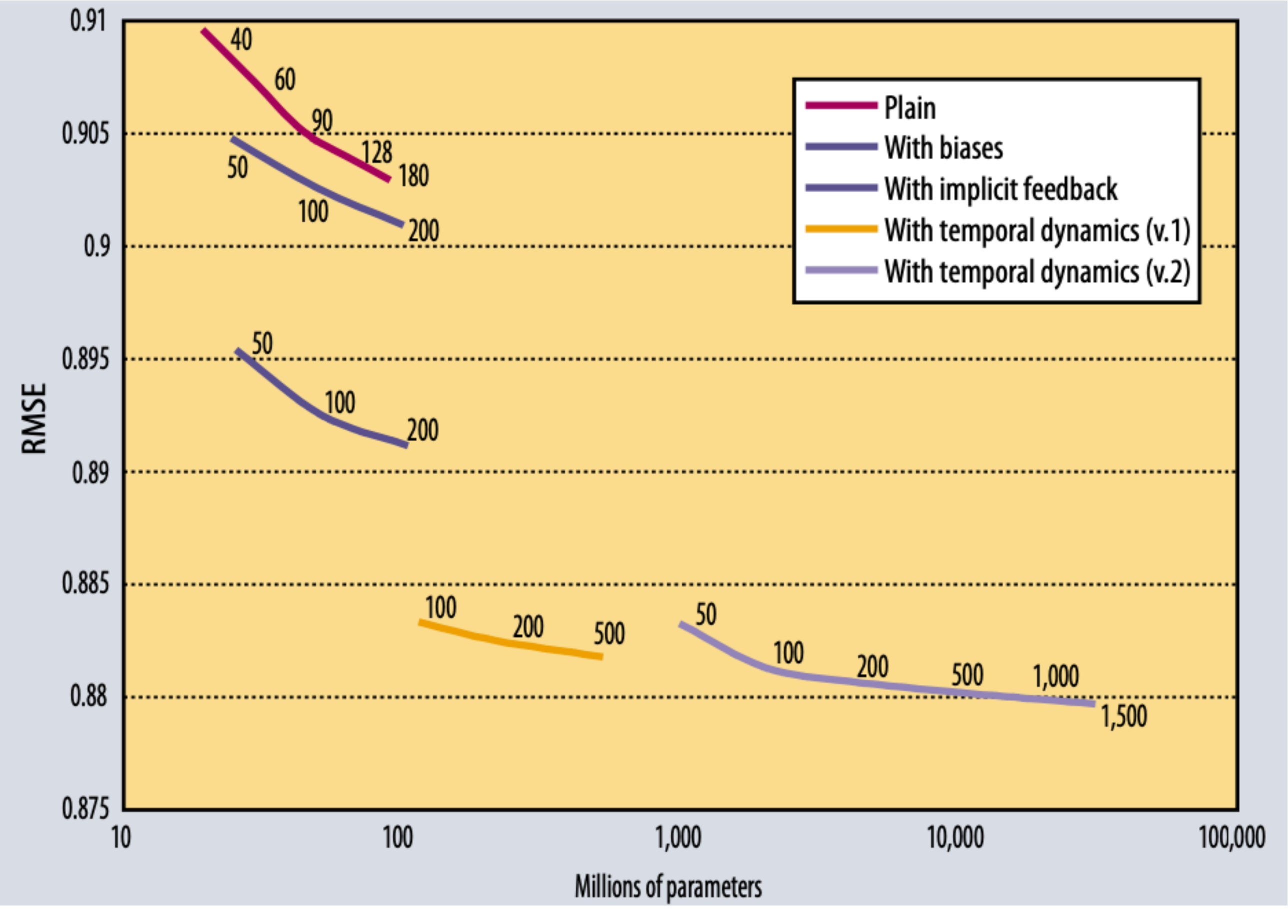
$$\boxed{\min_{\substack{p_u, q_i \\ \mu, b_u, b_i}} \sum_{u,i} (r_{ui} - \mu - b_u - b_i - p_u^T q_i) + \lambda * \sum(||q_i||^2 + ||p_u||^2 + \mu^2 + b_u^2 + b_i^2)}$$

* Implicit Feedback

* Temporal Dynamics:

(User's preference changes)

$$\boxed{\hat{r}_{ui}(t) = \mu + b_u(t) + b_i(t) + q_i^T \cdot p_u(t)}$$



Building Time Sensitive RecSys

idea ① : Train and update Model periodically

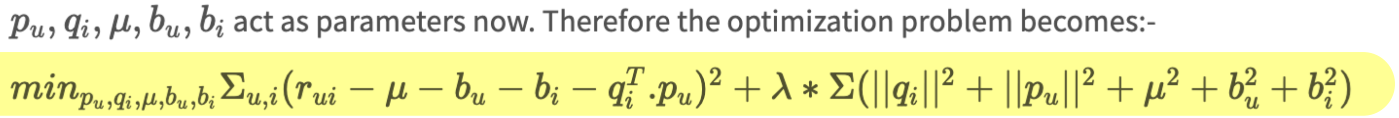
$2020 \rightarrow \overset{u,\ q_i}{\mu,\ b_u,\ b_i}$

$2021 \rightarrow \overset{u,\ q_i}{\mu,\ b_u,\ b_i}$ (Update by retraining)

↓

n-time-periods

idea 2 : Add weightage to give preference to more recent data



p_u, q_i, μ, b_u, b_i act as parameters now. Therefore the optimization problem becomes:-

$$min_{p_u,q_i,\mu,b_u,b_i} \Sigma_{u,i}(r_{ui} - \mu - b_u - b_i - q_i^T.p_u)^2 + \lambda * \Sigma(||q_i||^2 + ||p_u||^2 + \mu^2 + b_u^2 + b_i^2)$$

$$\min_{\substack{p_u,\ q_i \\ \mu,\ b_u,\ b_i}} \sum_{u,i} \omega_{ui} (r_{ui} - \mu - b_u - b_i - p_u^T q_i) + \lambda * \sum(||q_i||^2 + ||p_u||^2 + \mu^2 + b_u^2 + b_i^2)$$

↓

more recent item, more weigh

$$\boxed{\begin{array}{l} 2020 \to 0.5 \\ 2021 \to 0.8 \\ 2023 \to 0.95 \\ 2024 \to 1.2 \end{array}} \longrightarrow \begin{bmatrix} 0.5 \\ 0.8 \\ 0.95 \\ 1.2 \end{bmatrix}$$

So, How does final Solution Works:

1. Purely recency based

   Knn → Content based (very fast)

2. Historical behaviour

   1) MF/Deeplearning Meth
   2) Batch Mode (once in a week) (once every Night)

MF for Feature Engineering

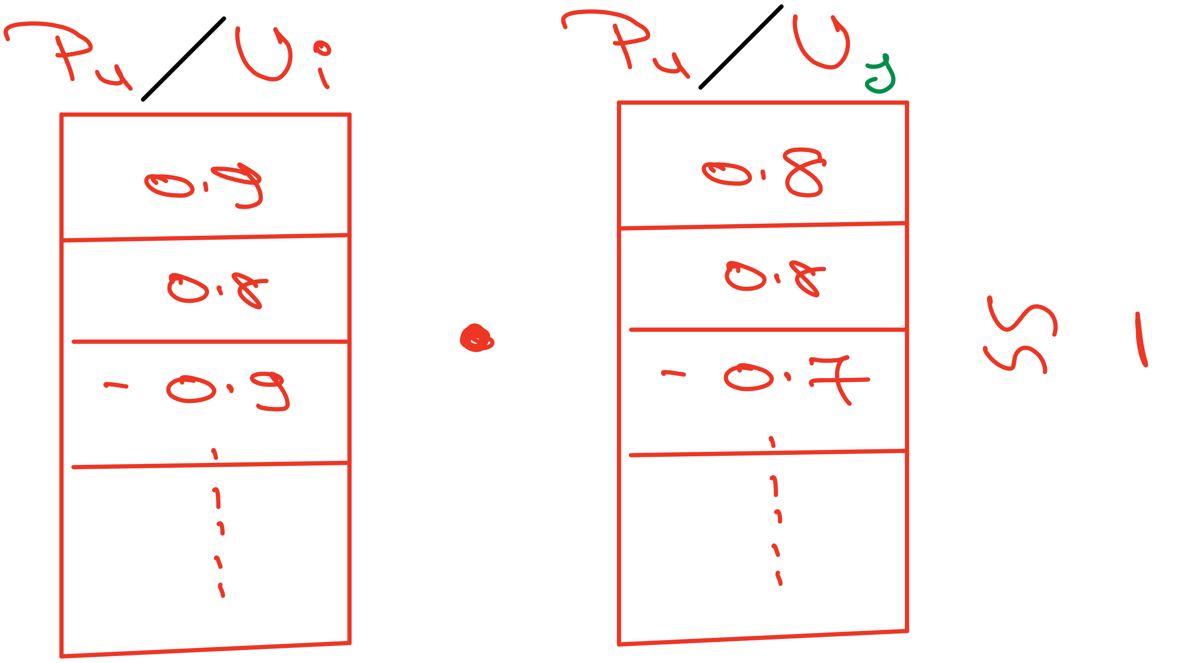
1. Implicit feature $P_u \ni 2 \times 1$

(meaningless but useful)

2. Explicit features

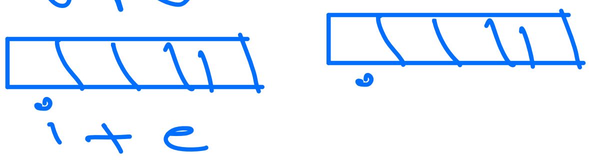
Country	Age	Salary	Occ-	...... --

Metadata



Similarity

* ads ⇒ $\underbrace{P(\text{click})}^{y_i}$ / $\underbrace{\text{user}}$, ads



* Search : item ⟶ (implicit) ⟵ i_1, i_2, i_3

MF for Text Data

Articles
www

$A_1 \longrightarrow w_1, w_2, w_4, w_8 - - - w'$

$A_2 \longrightarrow w_2, w_4, w_7 - - - w'$

A_3

⋮

A_n

$$W \longrightarrow (m, d)$$

$$A \longrightarrow (n, d)$$

word

A

	1	2	3	– – – – – – –	m
1					
2			cell23		
⋮ n					

word3 (0,1) freq (0 − ∞)

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

Article/Doc

Words

① doc matrix → doc_i ⇒ features of doc_i

Similar Docs
- → Google News
- → Kindle
- → websites/Search
- → Medium/Blogs

② Word matrix → w_i → features of word i

Similar Words
- → Search
- → Next Word Suggestio
- → Grammarly

Co-occurence Matrix

	w_1	w_2	w_3	- - - - - - -	w_m
w_1					
w_2			x_{ij}		
w_3 ⋮ w_m					

$x_{ij} = 1$ if w_i and w_j occur in a fixed vicinity together

w_2

The blue brown fox jumped over fence

← 3   3 →

$T=3$

$w_2 \Rightarrow$ blue

$w_3 \Rightarrow$ over

$w_4 \Rightarrow$ fox

$\left.\begin{array}{l} w_2 \Rightarrow \text{blue} \\ w_3 \Rightarrow \text{over} \end{array}\right\} x_{23} \Rightarrow 0$

$x_{24} \Rightarrow 1$

$$X_{n \times m} \Rightarrow B_{n \times d} \cdot C_{d \times m}$$

⇒ two vectors per word

$$X_{n \times m} \Rightarrow U_{n \times m} \Sigma_{m \times m} V^{T}_{m \times m}$$

Todo:

MF for Images: Eigen Faces