TADA : Talk About Data Analytics | Week 14

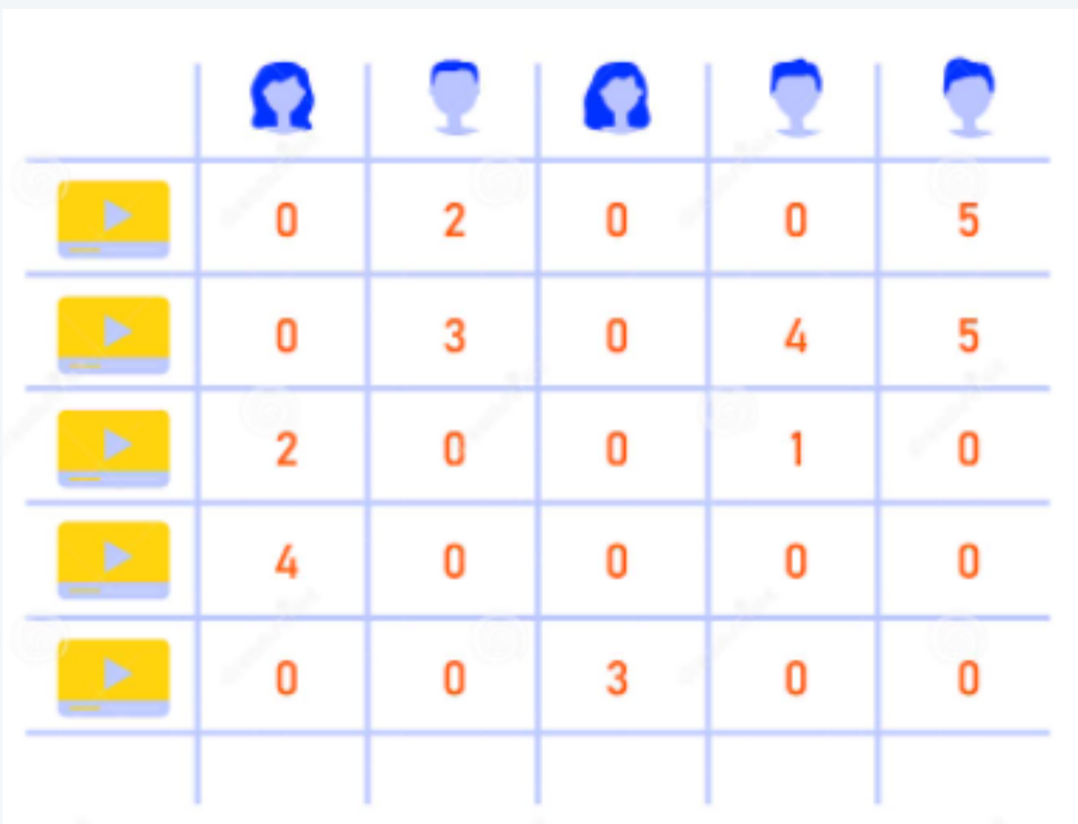
Alternating Least Square (ALS)

Parallel matrix factorization algorithm

Hannah Do | March 19th, 2022

Introduction

ALS is implemented in Apache Spark ML, and built for LARGE-scale Collaborative Filtering problems.



- Has benefits of solving scalability and sparseness of the Ratings data
- Collects Implicit feedback - instead of getting surveys of how people think of the product, user behavior is monitored and collected as data
  - User bought products
  - Page transition & Views
  - User Cart products

https://medium.com/radon-dev/als-implicit-collaborative-filtering-5ed653ba39fe

Algorithm

The algorithm takes a matrix and factors it into some smaller representations (U, P) of the original matrix. You can think of it in the same way as we would take a large number and factor it into smaller primes.

$$6 \rightarrow 2 \times 3$$

Original Matrix → U x P → Predicted Matrix

The idea is that these smaller representations - the Latent factors or features - tie the groups of the people with the similar item preferences.

https://medium.com/radon-dev/als-implicit-collaborative-filtering-5ed653ba39fe

Algorithm

Initially, the latent matrices are given random values, then updated as the dot product of the two matrices are compared with the actual Ratings Matrix.

As part of the training process - Gradient Descent is used with L2 regularization to update the value the latent matrix.

P : Product Matrix (product row = j)

	M1	M2	M3	M4	M5
F1	1.2	3.1	0.3	2.5	0.2
F2	2.4	1.5	4.4	0.4	1.1

U : User Matrix (user row = k)

	F1	F2
A	0.2	0.5
B	0.3	0.4
C	0.7	0.8
D	0.4	0.5

	M1	M2	M3	M4	M5
	1.44	1.37	2.26	0.7	0.59
	1.32	1.53	1.85	0.91	0.5
	2.76	3.37	3.73	2.07	1.02
	1.68	1.99	2.32	1.2	0.63

R : Ratings Matrix

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

https://www.youtube.com/watch?v=ZspR5PZemcs&ab_channel=Serrano.Academy

Algorithm

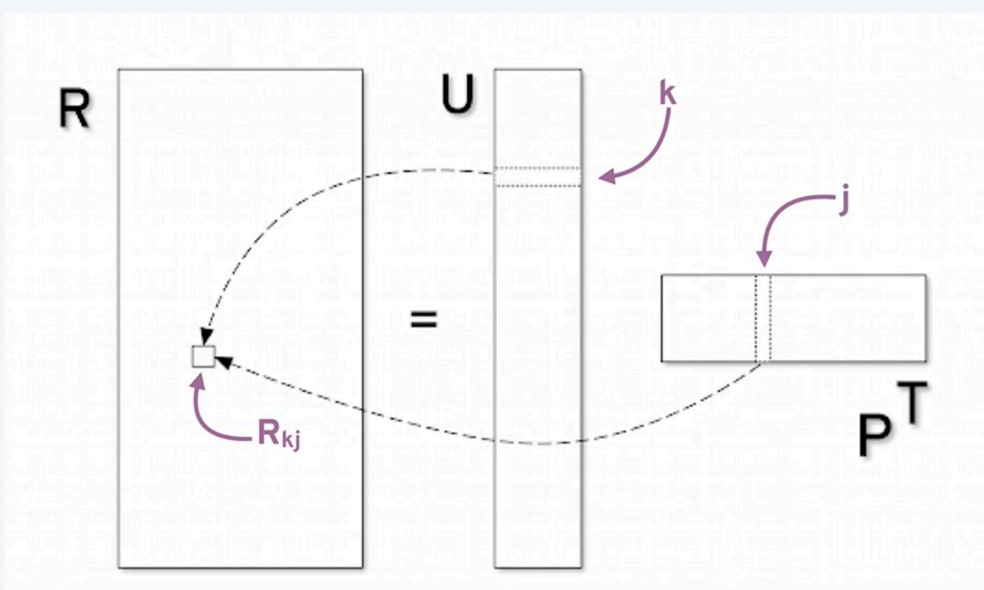
Why “Alternating” Least Squares?

U and P are optimized in an alternating sequence. Optimizing as in minimizing the least squares error.

The differences between the actual ratings matrix and predicted matrix is squared, added up together, and taken a derivative to update the values in the latent features.

$$\underset{H,W}{\arg\min}\, ||R - \tilde{R}||_{\mathrm{F}} + \alpha||H|| + \beta||W||$$

where H is user matrix, W is item matrix



R : Ratings Matrix

U : User Matrix (user row = k)
P : Product Matrix (product row = j)

R (k x j) = U (k x q) x P (q x j)
q : # of latent features / rank

https://www.youtube.com/watch?v=58OjaDH2FI0&ab_channel=Databricks

References

- ALS Implicit Collaborative Filtering, Victor, Medium (Aug 24, 2017) https://medium.com/radon-dev/als-implicit-collaborative-filtering-5ed653ba39fe
- How does Netflix recommend movies? Matrix Factorization, Serrano.Academy, Youtube (Sep 8, 2018) https://www.youtube.com/watch?v=ZspR5PZemcs&ab_channel=Serrano.Academy
- Building an Implicit Recommendation Engine with Spark with Sophie Watson, Red Hat (Oct 10, 2018) https://www.youtube.com/watch?v=58OjaDH2FI0&ab_channel=Databricks