ALS / SVD++ 개념

염은지

1. MF 모델 (ALS)

- **Collaborative Filtering**
  - 다른 유저의 정보까지 종합적으로 고려한 추천
  - A와 비슷한 과거, 선호를 지닌 B는 앞으로도 A와 비슷한 아이템을 좋아할 것이다
- **Matrix Factorization**
  - Rating (interaction) matrix를 Latent User / Item factor로 분해
  - User factor와 Item factor를 다시 곱하여 (Matrix completion) 누락된 Rating elements 추정
- **MF for Implicit Feedback Datasets**
  - **R (rating)을 P와 C의 표현으로 치환**
    - P (preference) : interaction 발생 유무
    - C (confidence) : interaction 횟수에 비례한 가중치 (alpha는 hyper param)

$$p_{ui} = \begin{cases} 1 \ if \ r_{ui} > 0 \\ 0 \ if \ r_{ui} = 0 \end{cases} \qquad c_{ui} = 1 + \alpha r_{ui}$$

  - **Loss Function**
    - 앞 수식은 prediction error
    - 뒤 수식은 regularization params

$$\min_{x^*,\, y^*} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda\left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2\right)$$

r_{ui} : original rating
$x_u^T y_i$: predicted rating
$(r_{ui} - x_u^T y_i)^2$: previous loss

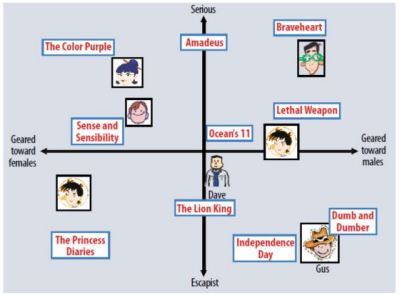
  - **Optimize**
    - Loss Function에서 X(User)를 학습할 때 Y(Item) 상수로, 반대의 경우도 동일하게 번갈아가며 진행
    - X와 Y에 대해 각각 미분하여, 해당 함수가 0이 되는 값으로 Latent Factor Update

$$x_u = (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p(u)$$

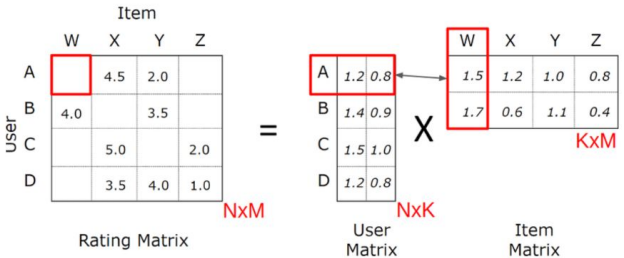
User Latent Factor Optimizer

$$y_i = (X^T C^i X + \lambda I)^{-1} X^T C^i p(i)$$

Item Latent Factor Optimizer



[Example] 이 표는 잠재 요인 모델 접근방법에 대한 설명을 나타낸 표로 남자 대 여자, Serious 대 Escapist-두 축을 사용하여 사용자들과 영화들 모두의 특성들을 표현했다.



$$R \approx P \times Q^T = \hat{R}$$

2. SVD++

- **SVD**
  - latent factor 모델 (MF)
  - R 정의

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u$$

  - Loss Function $\min_{b_*, q_*, p_*} \sum_{(u,i) \in \mathcal{K}} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda_4 (b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2)$
  - explicit rating만 고려
    - implicit rating을 고려하게 만들기 위해서는? -> P / C 두개의 표현으로 변경하여 표현 가능

- **SVD++**
  - 기존의 explicit rating만 고려하는 형식의 SVD를 implicit rating도 함께 이용하도록 확장
    - Pu를 explicit + implicit 표현으로 세밀화
  - R 정의

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T \left(p_u + |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{R}(u)} y_j \right) \qquad \hat{r}_{ui} = \mu + b_i + b_u + q_i^T \left(p_u + |\mathrm{N}^1(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{N}^1(u)} y_j^{(1)} + |\mathrm{N}^2(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{N}^2(u)} y_j^{(2)} \right)$$

  - Optimization

$$\begin{aligned}
&b_u \leftarrow b_u + \gamma \cdot (e_{ui} - \lambda_5 \cdot b_u) \\
&b_i \leftarrow b_i + \gamma \cdot (e_{ui} - \lambda_5 \cdot b_i) \\
&q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot (p_u + |\mathrm{N}(u)|^{-\frac{1}{2}} \textstyle\sum_{j \in \mathrm{R}(u)} y_j) - \lambda_6 \cdot q_i) \\
&p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda_6 \cdot p_u) \\
&\forall j \in \mathrm{R}(u) : \\
&y_j \leftarrow y_j + \gamma \cdot (e_{ui} \cdot |\mathrm{R}(u)|^{-\frac{1}{2}} \cdot q_i - \lambda_6 \cdot y_j)
\end{aligned}$$

2. SVD++

- **Spark 환경에서 ALS 방식 활용**
  - ALS (Alternating least squares)
    - 사용자와 아이템의 Latent Factor를 한번 씩 번갈아가며 학습
    - 두 행렬을 한꺼번에 최적화시키는 것은 어렵기 때문에 둘 중 하나를 상수로 고정
    - Error 계산 및 최적화 과정은 Worker node(RDD 연산)에서 진행
- **최적화 구현**
  - 논문 수식 직접 구현
  - Least square method 최적화 진행 (sklearn 모델 사용)

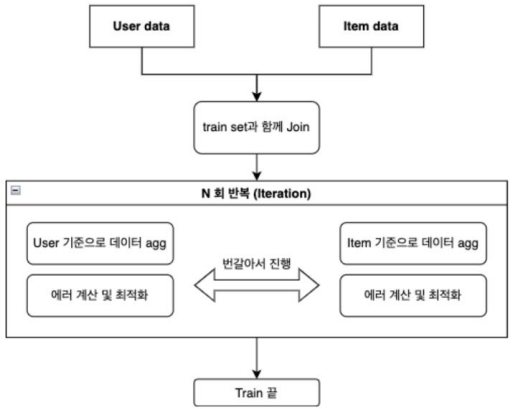
[User data / Item data]

```
+-------------+---+--------------------+--------+
|shop_prod_id_a| bu|                  pu|  sqrt_Iu|
+-------------+---+--------------------+--------+
|   A1659552181|0.0|[-0.12108875, 0.0...|2.6457512|
|   A2119235940|0.0|[0.09754314, -0.0...|     1.0|
|   A2187867003|0.0|[0.13873197, -0.0...| 4.582576|
|   A2196357064|0.0|[0.110774204, -0....|     1.0|
|   A2203924988|0.0|[-0.015826616, 0....|     1.0|
+-------------+---+--------------------+--------+
only showing top 5 rows

+-------------+---+--------------------+--------------------+
|shop_prod_id_b| bi|                  qi|                  yj|
+-------------+---+--------------------+--------------------+
|   A5096606133|0.0|[0.10818558, -0.0...|[0.09697321, 0.05...|
|   J6696061506|0.0|[-0.15400335, 0.1...|[0.02153944, 0.22...|
|   B5105570570|0.0|[0.06425026, -0.0...|[-0.08500709, 0.0...|
|   D5086237873|0.0|[-0.05927827, 0.0...|[0.111369066, -0....|
|  R13307021390|0.0|[0.024292057, -0....|[-0.037517827, -0...|
+-------------+---+--------------------+--------------------+
only showing top 5 rows
```

[Trainset]

```
+-------------+-------------+------+
|shop_prod_id_a|shop_prod_id_b|rating|
+-------------+-------------+------+
|   A1001968973|  H10443559447|     3|
|   A1058977100|  U15349028090|     3|
|   A1105753487|   B5095005476|     3|
|    A111068111|   C5107181031|    12|
|    A115957563|   G9215291458|    14|
|    A115957563|   W8438536120|     3|
|    A115957563|   Y9447238366|     3|
|   A1245994986|   A5086060334|     3|
|    A126031447|   E3215151476|     4|
|    A126031447|   E5102725324|    11|
+-------------+-------------+------+
only showing top 10 rows
```



[Process]

끝!