스탠포드 Distributed Algorithms and Optimization

http://stanford.edu/~rezab/classes/cme323/S15/notes/lec14.pdf

Matrix Factorization : Objective and ALS Algorithm on a single Machine

Matrix factorization은 상대적으로 작은 수(k ~ 10)로 차원이 정해지며
각각 user u의 k차원을 Xu, item i의 k차원을 yi라고 정의한다. 그 후 user u의 item i에 대한 평점 (Rui)을 예측한다.

$$X = \begin{bmatrix} | & & | \\ x_1 & \cdots & x_n \\ | & & | \end{bmatrix}, Y = \begin{bmatrix} | & & | \\ y_1 & \cdots & y_m \\ | & & | \end{bmatrix}$$

$R \approx X^T Y$. ratings matrix R을 예측하기 위해서 최소제곱법을 이용하여 X, Y의 최적화 값을 구해준다.

$$\min_{X,Y} \sum_{r_{ui}\ observed} (r_{ui} - x_u^\intercal y_i)^2 + \lambda(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2)$$

참고로 이 objective는 non-convex함수이다. Gradient descent로 최적화 값을 예측할 수 있지만 느리고, 많은 비용이 발생한다.

이때 ALS(Alternating Least Squares)방법을 사용한다. X변수를 고정하고 상수항으로 취급하면 objective는 Y에 대한 Convex함수가 된다. 반대로 Y변수를 고정하고 상수항으로 취급하면 x에 대한 Convex함수가 된다.

Algorithm 1 ALS for Matrix Completion

Initialize X, Y
repeat
  for $u = 1 \dots n$ **do**

$$x_u = (\sum_{r_{ui} \in r_{u*}} y_i y_i^\intercal + \lambda I_k)^{-1} \sum_{r_{ui} \in r_{u*}} r_{ui} y_i$$

  end for
  for $i = 1 \dots m$ **do**

$$y_i = (\sum_{r_{ui} \in r_{*i}} x_u x_u^\intercal + \lambda I_k)^{-1} \sum_{r_{ui} \in r_{*i}} r_{ui} x_u$$

  end for
until convergence