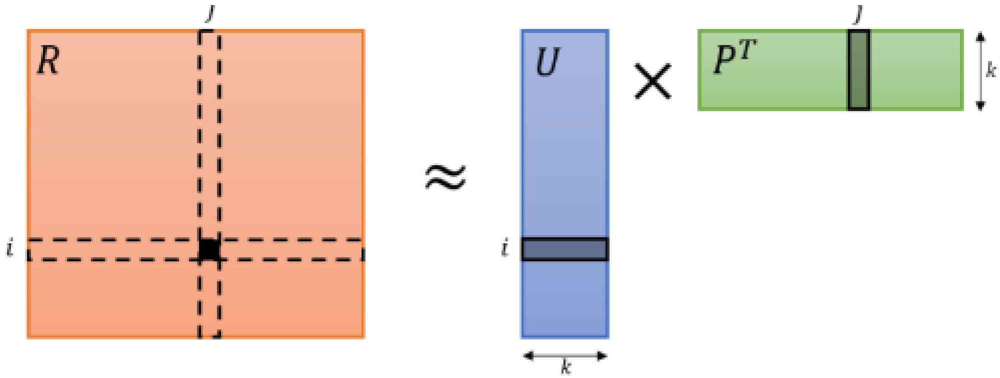
How do you build a “People who bought this also bought that” - style recommendation engine

https://datasciencemadesimpler.wordpress.com/2015/12/16/understanding-collaborative-filtering-approach-to-recommendations/

Matrix Factorization

Model Formulation

Collaborative recommender systems에서와 Co-clustering problems을 해결하기 위해서 사용되는 가장 유명한 알고리즘이다. MF(Matrix Factorization)이라고 부른다.



Rij는 Ui와 Pj의 dot product으로 factorized 된다. 이때 k는 U와 p의 차원이고 모든 R은 k에 영향을 받는다.

R은 평점을 준 USER의 matrix라고 가정하자. 사용자가 영화에 대해서 주는 평점은 그의 성향과 영화의 특성에 따라 달려있다.
어떻게 카테고리들을 효과적으로 구성 할 것인지가 문제이다. 어떤 영화배우, 감독, 언어, 촬영장소, 여러 특징들에 의해 특정한 관련성을 가질 수 있다.

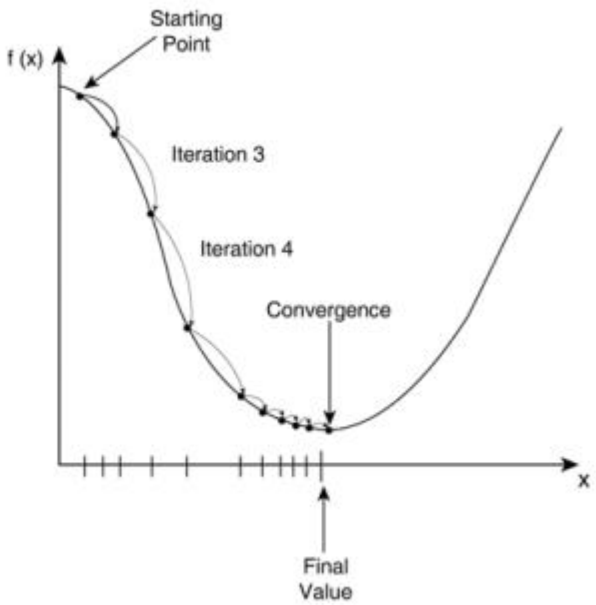
Co-clustering problem는 user의 평점을 주는 부분이 아주 적은 부분 존재하고 다양한 많은 영화들이 존재하기 때문에 user와 영화 간의 밀도가 아주 낮은 rating matrix을 만들어내는 것으로 알고 있다. 따라서 각 요소들이 rating에 얼마나 영향을 미치는 계량화하기 쉽지 않다.

MF은 비용함수를 최소화하여 R행렬을 목표에 가깝게 하는 최적화 프로세스이다.

$$J = ||R - U \times P^T||_2 + \lambda\left(||U||_2 + ||P||_2\right)$$ (cost function)

1번째 항은 평균 제곱합(Mean Square Error, Mean)이며 R행렬과 U * P 행렬의 근사치의 거리를 계산한 것이다. 2번째 항은 정규항이며 과적합을 방지하기 위한 일반적인 해결책이다.

Gradient Descent



Gradient Descent는 머신러닝 분야에서 가장 폭넓게 사용되는 첫 번째 최적화 알고리즘이다
비용함수의 최적화 변수에 대해서 반복해서 계산한다. 비용함수가 최솟값으로 수렴할 때까지 비용함수의 Gradient를 음의 값으로 줄여나간다. 그러나 Gradient Descent는 local minization에 대해서는 수렴하는 문제점이 있다.

Gradient descent는 MF모델을 Powerful하게 최적화 시킬 수 있다. 그러나 실제로 항목의 수가 많아진다면 parallelization mechanism(평행화 메커니즘)과 MF의 비용함수를 개선 시켜야 한다.

Alternating Least Squares

Two-step으로 반복되는 최적화 프로세스이다. U를 해결하기 위해선 P를 고정하고, P를 해결하기 위해선 U를 고정한다.

$$\forall u_i : J(u_i) = ||R_i - u_i \times P^T||_2 + \lambda \cdot ||u_i||_2$$

$$\forall p_j : J(p_j) = ||R_i - U \times p_j^T||_2 + \lambda \cdot ||p_j||_2$$

$$u_i = \left(P^T \times P + \lambda I\right)^{-1} \times P^T \times R_i$$

$$p_j = \left(U^T \times U + \lambda I\right)^{-1} \times U^T \times R_j$$

두 가지 방법을 번갈아 사용하면 비용함수의 값이 수렴할 때까지 감소한다.
Gradient descent optimization와 유사하게 local minima에만 수렴하고 U,P의 초기값에 의존한다.

Missing Values

행렬R이 Missing values를 가질 때 비용함수

$$J = \sum_{i,j} w_{i,j} \cdot \left(R_{i,j} - u_i \times p_j^T\right)^2 + \lambda\left(||U||_2 + ||P||_2\right)$$

where

$$w_{i,j} = \begin{cases} 1 & R_{i,j} \; is \; known \\ 0 & R_{i,j} \; is \; unknown \end{cases}$$

$$u_i = \left(P^T \times w_i \times P + \lambda I\right)^{-1} \times P^T \times w_i \times r_i$$

$$p_j = \left(U^T \times w_j \times U + \lambda I\right)^{-1} \times U^T \times w_j \times r_j$$