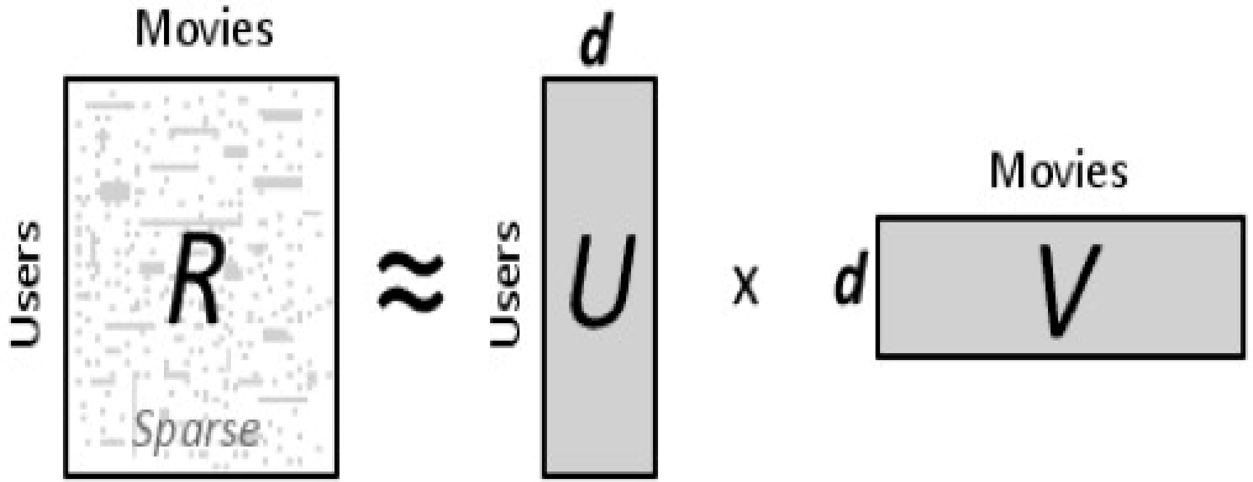
Matrix Factorization

Radek Bartyzal

20. 4. 2017

Let’s Talk ML Prague

Matrix factorization

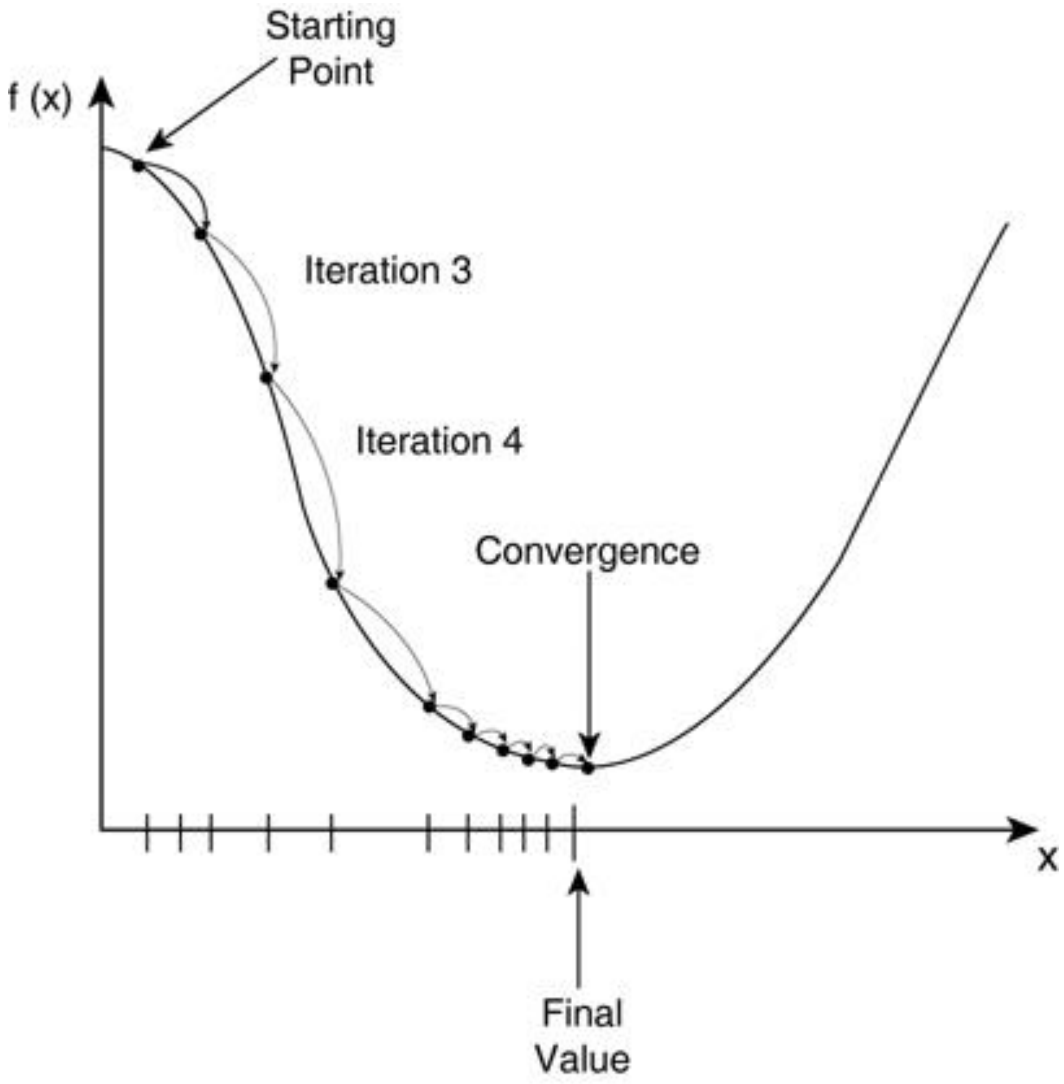


$$\operatorname{argmin} \|R - U^T V\|^2 + \lambda(\|U\|_2^2 + \|V\|_2^2),$$

where R is partially observed the user-item preference matrix, $U \in \mathbb{R}^{k\times m}$,and $V \in \mathbb{R}^{k\times n}$

Gradient descent

- typical gradient descent
- (n*k + m*k) parameters to optimize



Alternating least squares

- exploits the structure of optimization problem
  = fixing W or H leads to a simple linear regression

Algorithm:

Repeat:

Solve U while V is fixed

Solve V while U is fixed

Linear regression

Optimization function:

$$\operatorname{argmin} ||y - Xw||^2 + \lambda||w||_2^2$$

Solution = Ordinary Least Squares (OLS):

$$W = (X^T X + \lambda I)^{-1} X^T Y$$

Problem:

inversion of large matrix is too expensive

Alternating least squares

MF goal:

$$\operatorname{argmin} \left\| R - U^T V \right\|^2 + \lambda(\|U\|_2^2 + \|V\|_2^2)$$

Solve U according to OLS while V is fixed:

$$U = (VV^T + \lambda I)^{-1} V R^T$$

Solve V according to OLS while U is fixed:

$$V = (UU^T + \lambda I)^{-1} U R$$

Alternating least squares

$$U = (VV^T + \lambda I)^{-1} V R^T$$

- inverse of (k x k) matrix, which is ok
- can be independently calculated per row OR per column

  = each row/column of U can be calculated in parallel