Matrix Factorization

Germain Bregeon - Emma Covili - Xavier Geffrier

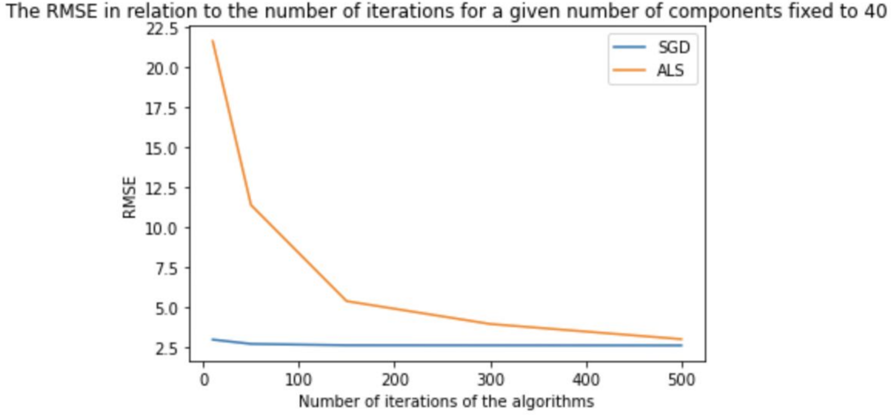
SGD & ALS for the 100k dataset

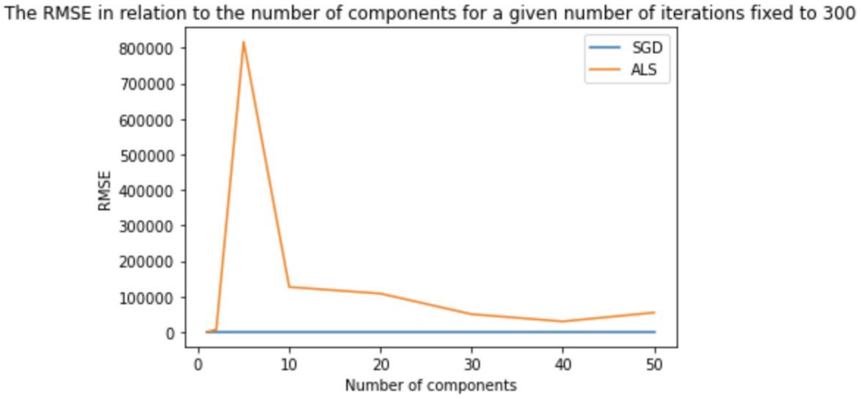
- SGD

$$I_{t+1} = I_t - \eta_t \frac{\partial C}{\partial I}(I_t, U_t)$$
$$U_{t+1} = U_t - \xi_t \frac{\partial C}{\partial U}(I_t, U_t)$$

- ALS

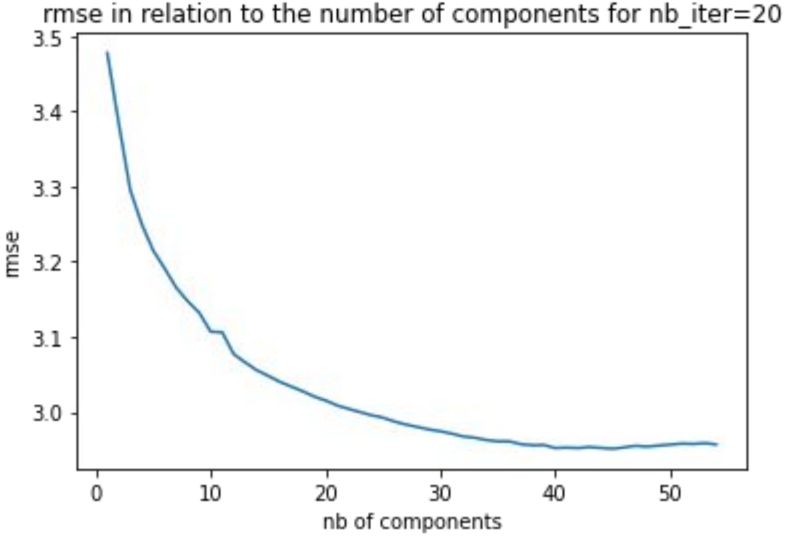
$$I_{t+1} = RU_t(U_t^\top U_t + \lambda \mathbb{I})^{-1}$$
$$U_{t+1} = R^\top I_t(I_t^\top I_t + \mu \mathbb{I})^{-1}$$

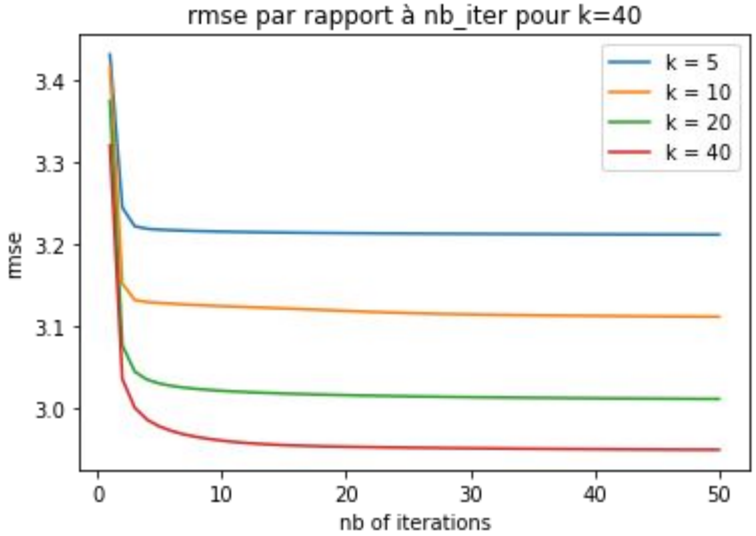




ALS with Sparse Matrices on the 25 M dataset

$$I_{t+1} = RU_t(U_t^\top U_t + \lambda \mathbb{I})^{-1}$$
$$U_{t+1} = R^\top I_t(I_t^\top I_t + \mu \mathbb{I})^{-1}$$





Constraint programming

Google OR-Tools

$$R_{m,n} = \begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m,1} & r_{m,2} & \cdots & r_{m,n} \end{pmatrix}$$

We search $IU^T \simeq R$

$$r_{i,j} = I_{i,*}^T \cdot U_{j,*} \pm \varepsilon = i_{i,1} * u_{i,1} + \cdots + i_{i,k} * u_{j,k} \pm \varepsilon$$

$$\begin{aligned} S &= \text{total number of possible solutions} \\ k &= \text{number of components} \\ m, n &= \text{dimensions of matrix R} \\ d &= \text{digits precision} \end{aligned}$$

$$S = k \times (m + n) \times 2 \times 10^d$$

Exemple : $S = 3 \times (10^5 + 10^3) \times 2 \times 10^5 = 60\,600\,000\,000$

What's next

➔ ALS not taking into account missing entries

$$W_{i,j} = \begin{cases} 1 \ if \ R_{i,j} \ is \ known \\ 0 \ if \ R_{i,j} \ is \ missing \end{cases} \qquad min_{U,Y} \parallel W \cdot (R - IU^T) \parallel_F^2 \qquad (W \cdot X)_{ij} = w_{ij}x_{ij}$$

➔ ALS by taking initialisation different than random

➔ DMF : Deep Matrix Factorization (by Xue et al, 2017)