annotated version

Machine Learning Course - CS-433

Matrix Factorizations

Dec 5, 2023

Martin Jaggi
Last updated on: December 3, 2023

credits to Mohammad Emtiyaz Khan

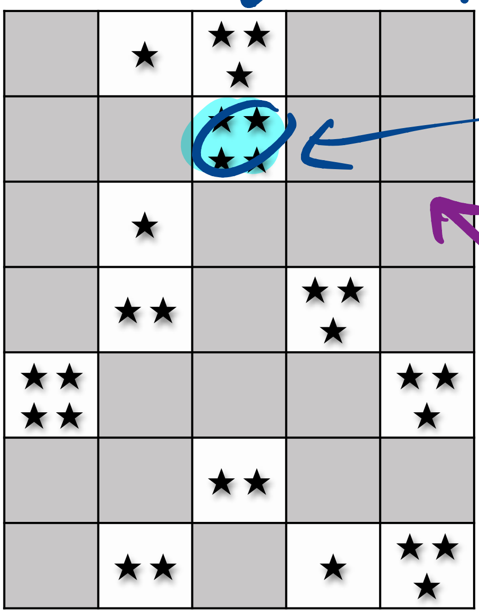
EPFL

Motivation

D = 20k
N = 500k

In the Netflix prize, the goal was to predict ratings of users for movies, given the existing ratings of those users for other movies. We are going to study the method that achieved the best error (for a single method).

users
n
1 N
((2,3), 4)
(d,n), rating
d →
movies



unknown
99.99 %
to predict
D

The Movie Ratings Data

Given items (movies) $d = 1, 2, \ldots, D$ and users $n = 1, 2, \ldots, N$, we define $\mathbf{X}$ to be the $D \times N$ matrix containing all rating entries. That is, x_{dn} is the rating of n-th user for d-th item.

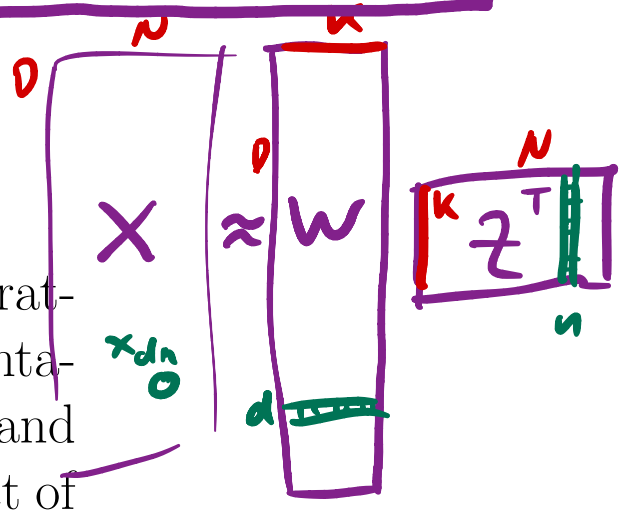
Note that most ratings x_{dn} are missing and our task is to predict those missing ratings accurately.

$\min_{w,z} \left(\mathcal{L}(w,z) = f(wz^\top) \right)$

Prediction Using a Matrix Factorization

We will aim to find $\mathbf{W}, \mathbf{Z}$ s.t.

$$\mathbf{X} \approx \mathbf{W}\mathbf{Z}^\top .$$



$x_{dn} \approx (w z^\top)_{dn}$

So we hope to 'explain' each rating x_{dn} by a numerical representation of the corresponding item and user - in fact by the inner product of an item feature vector with the user feature vector.

true rating

$$\min_{\mathbf{W},\mathbf{Z}} \mathcal{L}(\mathbf{W}, \mathbf{Z}) := \frac{1}{2} \sum_{(d,n)\in\Omega} \left[x_{dn} - (\mathbf{W}\mathbf{Z}^\top)_{dn}\right]^2$$

← loss function

model

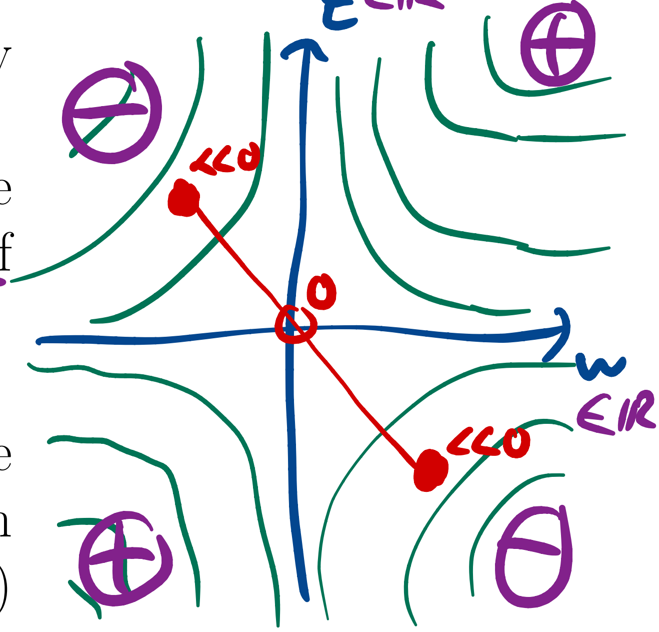
observed entry

where $\mathbf{W} \in \mathbb{R}^{D\times K}$ and $\mathbf{Z} \in \mathbb{R}^{N\times K}$ are tall matrices, having only $K \ll D, N$ columns.

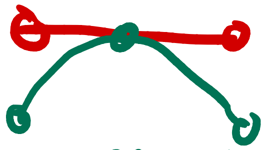
The set $\Omega \subseteq [D] \times [N]$ collects the indices of the observed ratings of the input matrix $\mathbf{X}$.

$\mathcal{L}(w,z) = f(w \cdot z) = w \cdot z$

$z \in \mathbb{R}$



≠ non-convex



$f((w,z)) = (w,z)^2$

Each row of those matrices is the feature representation of an item (row of $\mathbf{W}$) or a user (row of $\mathbf{Z}$) respectively.

(1) no!

Is this cost jointly convex w.r.t. $\mathbf{W}$ and $\mathbf{Z}$? Is the model identifiable? (2) no!

$(w^*, z^*) \to (\beta w^*, \frac{1}{\beta} z^*)$

Can we incorporate side-information on users or items?

Choosing K

movie d $\boxed{W} \cdot Z^T$

K is the number of *latent* features.

hardcoded W learn W

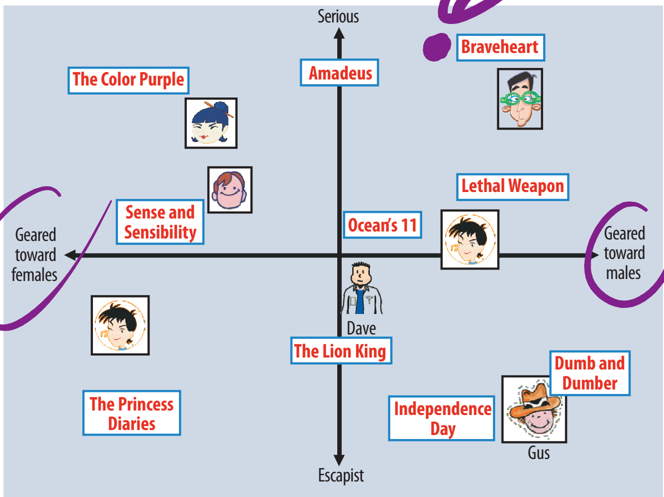


Figure 2. A simplified illustration of the latent factor approach, which characterizes both users and movies using two axes—male versus female and serious versus escapist.

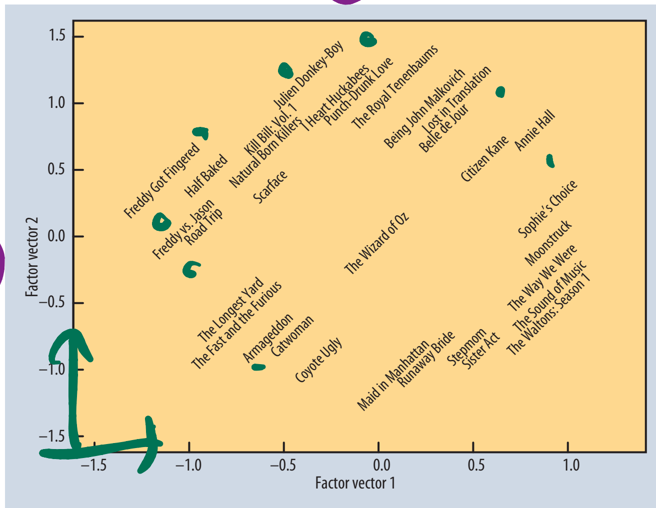


Figure 3. The first two vectors from a matrix decomposition of the Netflix Prize data. Selected movies are placed at the appropriate spot based on their factor vectors in two dimensions. The plot reveals distinct genres, including clusters of movies with strong female leads, fraternity humor, and quirky independent films.

Recall that for K-means, K was the number of clusters. (Similarly for GMMs, K was the number of latent variable dimensions).

example for $K \gg N, D$

X: $W \cdot Z^T$

W	Z^T
X	I
I	X

Large K facilitates overfitting.

small K → underfitting

Regularization

We can add a regularizer and minimize the following cost:

Regularization

$$\overbrace{\frac{1}{2}\sum_{(d,n)\in\Omega}\left[x_{dn} - (\mathbf{W}\mathbf{Z}^\top)_{dn}\right]^2}^{f(W\cdot Z^T)} + \frac{\lambda_w}{2}\|\mathbf{W}\|^2_{\text{Frob}} + \frac{\lambda_z}{2}\|\mathbf{Z}\|^2_{\text{Frob}}$$

where $\lambda_w, \lambda_z > 0$ are scalars.

Stochastic Gradient Descent (SGD)

The training objective is a sum over $|\Omega|$ terms (one per rating):

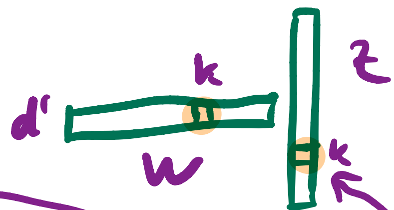
$$\mathcal{L} = \frac{1}{|\Omega|}\sum_{(d,n)\in\Omega} \underbrace{\tfrac{1}{2}\left[x_{dn} - (\mathbf{W}\mathbf{Z}^\top)_{dn}\right]^2}_{f_{d,n}(\mathbf{W},\mathbf{Z})}$$

Derive the stochastic gradient for $\mathbf{W}, \mathbf{Z}$, given one observed rating $(d,n) \in \Omega$.

$$\nabla_{\mathbf{W}} f_{dn}(\mathbf{W},\mathbf{Z}) \in \mathbb{R}^{D\times K}$$

$$\nabla_{\mathbf{Z}} f_{dn}(\mathbf{W},\mathbf{Z}) \in \mathbb{R}^{N\times K}$$

For one fixed element (d,n) of the sum, we derive the gradient entry (d',k) for $\mathbf{W}$, that is $\frac{\partial}{\partial w_{d',k}} f_{d,n}(\mathbf{W},\mathbf{Z})$, and analogously entry (n',k) of the $\mathbf{Z}$ part:



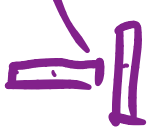
$$\frac{\partial}{\partial w_{d',k}} f_{d,n}(\mathbf{W},\mathbf{Z}) = \begin{cases} -\overbrace{\left[x_{dn} - (\mathbf{W}\mathbf{Z}^\top)_{dn}\right]}^{\text{truth} - \text{prediction}} z_{n,k} & \text{if } d' = d \\ 0 & \text{otherwise} \end{cases}$$

inner derivative

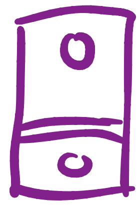
because f_{dn} independent of $w_{d'}$

$$\frac{\partial}{\partial z_{n',k}} f_{d,n}(\mathbf{W},\mathbf{Z}) = \begin{cases} -\left[x_{dn} - (\mathbf{W}\mathbf{Z}^\top)_{dn}\right] w_{d,k} & \text{if } n' = n \\ 0 & \text{otherwise} \end{cases}$$

row of W matrix



$\nabla_W =$



Cost $O(K)$

Alternating Least-Squares (ALS)

For simplicity, let us first assume that there are no missing ratings, that is $\Omega = [D] \times [N]$. Then

min W

(fix Z)

$$\frac{1}{2}\sum_{d=1}^{D}\sum_{n=1}^{N}\left[x_{dn} - (\mathbf{W}\mathbf{Z}^\top)_{dn}\right]^2$$

$$= \frac{1}{2}\|\mathbf{X} - \mathbf{W}\mathbf{Z}^\top\|^2_{\mathsf{Frob}}\ .$$

$$= \|\vec{x} - \vec{w}^\top \vec{z}\|^2_{vec}$$

min z

(fix w)

We can use coordinate descent to minimize the cost plus regularizer: We first minimize w.r.t. $\mathbf{Z}$ for fixed $\mathbf{W}$, and then minimize $\mathbf{W}$ given $\mathbf{Z}$.

$\nabla_z \mathcal{L}(w,z) \overset{!}{=} 0$

$\Downarrow$

$$\mathbf{Z}^\top := (\underbrace{\mathbf{W}^\top\mathbf{W} + \lambda_z\mathbf{I}_K}_{K\times K})^{-1}\mathbf{W}^\top\mathbf{X} \leftarrow \text{see least squares}$$

$$\mathbf{W}^\top := (\underbrace{\mathbf{Z}^\top\mathbf{Z} + \lambda_w\mathbf{I}_K}_{K\times K})^{-1}\mathbf{Z}^\top\mathbf{X}^\top \leftarrow \text{"}$$

What is the computational complexity? How can you decrease the cost when N and D are large?

ALS with Missing Entries

Can you derive the ALS updates for the more general setting, when only the ratings $(d, n) \in \Omega$ contribute to the cost, i.e.

$$\mathcal{L} = \frac{1}{2} \sum_{(d,n)\in\Omega} \left[x_{dn} - (\mathbf{W}\mathbf{Z}^\top)_{dn}\right]^2$$

Hint: Compute the gradient with respect to each group of variables, and set to zero.

- $\nabla_W \mathcal{L}(W, Z) \overset{!}{=} 0$

  solve for W

- $\nabla_Z \mathcal{L}(W, Z) \overset{!}{=} 0$

  solve for Z

cost per step $\gg N, D$