MATRIX FACTORIZATION

MARIA CAMERON

Contents

1. Getting started: examples data science problems solved via matrix factorization

This chapter will be largely based on D. Bindel's lectures. My notes here are mostly complimentary to Bindel's.

Please read Lecture 5 "Latent Factor Models" for the introduction. The section "A gallery of examples" gives you an idea of what kind of problems we should think about in the context of data science and matrix factorization.

- *Document search and latent semantic analysis (LSA).* There is a nice demo in Wiki and an article Introduction to Latent Semantic Analysis by T. Landauer, P. Foltz, and D. Laham with a nice illustrative example. The idea is to use the truncated SVD.
- *K-means clustering* is the standard clustering algorithm. For a given data matrix A $n \times d$ whose rows represent data while columns represent attributes, it finds a

$k \times d$ matrix R consisting of k rows of A each of which represents a cluster, and an $n \times k$ matrix L with $L_{ij} = 1$ if row i of A belongs to cluster j, and 0 otherwise, such that $A \approx LR$. A nice demo for this algorithm is in Wiki.

- *Eigenfaces and Fisherfaces* are techniques for face recognition based on making face images low-resolution, and computing a covariance matrix and its eigenvalue decomposition – see Wiki.
- *Collaborative filtering and the Netflix challenge* is an instance of matrix completion problem. Imagine a matrix of users' ratings where rows correspond to movies and columns correspond to the users. Each user watched just a small subset of available movies. The task is to predict the missing entries, i.e., to find a matrix according to some model that fits the available data the best. Here is a helpful presentation for the Netfix challenge by M. Gromley.
- *Anchor words and interpretable topic models.* A shortcoming of SVD used in LSA is that the columns of U and V are hard-to-interpret. In particular, they may have negative entries. Instead, we would like to factor a matrix into factors with nonnegative entries. In order to make the factor have a probabilistic interpretation, i.e., this document is attributed to this topic, the entries corresponding to each topic should sum up to one. The resulting problem is called *Nonnegative Matrix Factorization or NMF.*

2. Some useful matrix decompositions

Please read D. Bindel's Lecture 6 "SVD and other low-rank decompositions".

2.1. **The symmetric eigenvalue problem.**

Real symmetric matrices have many uses in data analysis. Among them are:

- representation of weighted or unweighted undirected graphs;
- similarities between objects;
- covariances of vector random variables;
- counts of pairs of words that occur together across sets of documents.

The eigenvectors and eigenvalues of a symmetric matrix A are stationary points (a.k.a. critical points) and the corresponding values of the objective function for the constrained optimization problem

$$\phi(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x} \rightarrow \max \quad \text{subject to} \quad \|\mathbf{x}\|_2^2 = 1. \tag{1}$$

They are also the critical points for the *Rayleigh quotient*

$$\rho_A(\mathbf{x}) = \frac{\mathbf{x}^\top A\mathbf{x}}{\mathbf{x}^\top \mathbf{x}}. \tag{2}$$

Exercise Prove that stationary points of $\rho_A(\mathbf{x})$ are the eigenvectors of A.

Besides the symmetric eigenvalue problem above a *generalized eigenvalue problem* arises in applications, e.g. in Fisherfaces (see Bindel's lecture 5). If M is a symmetric positive definite matrix then one can define an associated inner product

$$\langle \mathbf{x}, \mathbf{y} \rangle_M = \mathbf{y}^\top M\mathbf{x}.$$

Then the problem is $A\mathbf{x} = \lambda M\mathbf{x}$.

There are two families of methods for solving symmetric eigenvalue problem.

- The first family aims at finding the full spectral decomposition. A is decomposed to $UTU^\top$ where U is orthogonal and T is tridiagonal. Then a very efficient algorithm for finding eigenvalues of the tridiagonal matrix is applied. This is what is done by the Matlab command `eig`.
- If matrix is very large and sparse, the full decomposition might require too much memory. Then a few eigenpairs corresponding to the largest eigenvalues are computed by iterative methods. The main workhorse is the Lanczos method. This is what is used in the Matlab command `eigs`.

Discussion of these methods belongs to the course Numerical Linear Algebra. I will not discuss them here.

2.2. **SVD.** Read Section 3.2.3 in [1]. Also, see Bindel's lecture 6.

We recall the definition of the full SVD decomposition for an $n \times d$ matrix A: $A = U\Sigma V = A$, where

- U is $n \times n$, $UU^\top = U^\top U = I$,
- Σ is $n \times d$ with the top left submatrix being $\mathsf{diag}\{\sigma_1, \ldots, \sigma_d\}$ and all entries in the rest of Σ are zeros, and
- V is $d \times d$, $VV^\top = V^\top V = I$.

2.3. **Ky-Fan norms.**

Definition 1. *We say that a function $f : \mathbb{R}^{n\times d} \to \mathbb{R}$ is orthogonally (or unitarily in the complex case) invariant if $f(Q_1AQ_2) = f(A)$ for any orthogonal (unitary) matrices Q_1 and Q_2.*

Any unitarily invariant function can be written in terms of singular values of A. Indeed, take $Q_1 = U^\top$ and $Q_2 = V$ and you

$$f(A) = f(\Sigma) = \tilde{f}(\sigma_1, \ldots, \sigma_d).$$

Among the most important unitarily invariant functions are the *Ky-Fan norms*, which are l^p norms of the vectors of singular values. The Ky-Fan norms we care about are:

- The l^∞ Ky-Fan norm is the 2-norm of A (the *operator 2-norm* or the *spectral norm*):

$$\|A\|_2 = \sigma_1 = \max_{1\le i\le d} \sigma_i. \tag{3}$$

- The l^2 Ky-Fan norm is the *Frobenius norm* of A:

$$\|A\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{d} |a_{ij}|^2}. \tag{4}$$

Exercise Prove that

$$\|A\|_F^2 = \sum_{i=1}^{d} \sigma_i^2. \tag{5}$$

Hint: use the full SVD of A and *the cyclic property of trace* (you can prove this property by direct checking):

$$\mathsf{trace}(ABC) = \mathsf{trace}(BCA) = \mathsf{trace}(CAB) \tag{6}$$

for all A, B , C such that their product is defined and is a square matrix.

- The *nuclear norm*

$$\|A\|_* = \sum_{i=1}^{d} \sigma_j \tag{7}$$

is the Ky-Fan l^1 norm.

The Eckart-Young-Mirsky theorem says that the truncated SVD of A is the best rank k approximation to A in the 2-norm and in the Frobenius norm. It also holds for any Ky-Fan norm, so, for the nuclear norm as well.

Theorem 1. (Eckart-Young-Mirsky) *Let $A = U\Sigma V^\top$ be the SVD of A. Then for any matrix M of rank $\le k$ we have*

$$\|A - M\|_2 \ge \left\|A - U_k\Sigma_k V_k^\top\right\|_2 = \sigma_{k+1}, \tag{8}$$

$$\|A - M\|_F \ge \left\|A - U_k\Sigma_k V_k^\top\right\|_F = \sqrt{\sum_{j=k+1}^{n} \sigma_j^2} \tag{9}$$

Proof. **Proof for the matrix 2-norm.** If $M = U_k\Sigma_k V_k^\top$ then

$$\|A - U_k\Sigma_k V_k^\top\|_2 = \left\|\sum_{j=k+1}^{d} \sigma_j u_j v_j^\top\right\|_2 = \sigma_{k+1}.$$

Now let M be an arbitrary rank $\le k$ matrix. If rank of A is $\le k$, then taking $M = A$ will zero out any norm of the difference $A - M$. So, assume $\mathsf{rank}(A) = r > k$. The null-space of M has dimension $\ge d - k$. The space spanned by $\{\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}\}$ has dimension $k + 1$. Hence

$$\mathsf{dim}\,\mathsf{null}(M) + \mathsf{dim}\,\mathsf{span}(V_{k+1}) \ge d + 1 > d,$$

which means that they have an intersection of dimension ≥ 1. Let

$$\mathbf{x} \in \mathsf{null}(M) \cap \mathsf{span}(V_{k+1}), \quad \|\mathbf{x}\|_2 = 1.$$

Then

$$\|A - M\|_2^2 \ge \|(A - M)\mathbf{x}\|_2^2 = \|U\Sigma V^\top\mathbf{x}\|_2^2 = \|\Sigma V^\top\mathbf{x}\|_2^2 \ge \sigma_{k+1}^2\|V^\top\mathbf{x}\|_2^2 = \sigma_{k+1}^2.$$

This completes the proof for the 2-norm. □

The proof for the Frobenius norm makes use of the following lemma.

Lemma 1. *Let a matrix $A \in \mathbb{R}^{n\times d}$ be decomposed into a sum $A = B + C$. Let $\sigma_j(M)$ denote the jth singular value of the matrix M, $M = A, B, C$. We also set $\sigma_j(M) = 0$ for all integer $j > \mathsf{rank}(M)$. Then*

$$\sigma_{i+j-1}(A) \le \sigma_i(B) + \sigma_j(C) \quad \forall i, j \ge 1, \tag{10}$$

Proof. Let i and j be arbitrary integers $1 \le i, j \le d$. Then, by the Eckart-Young-Mirsky theorem for the 2-norm

$$\sigma_i(B) + \sigma_j(C) \equiv \|B - B_{i-1}\|_2 + \|C - C_{j-1}\|_2 = \sigma_1(B - B_{i-1}) + \sigma_1(C - C_{j-1}).$$

By the triangle inequality for the 2-norm we have:

$$\|B - B_{i-1}\|_2 + \|C - C_{j-1}\|_2 \ge \|B - B_{i-1} + C - C_{j-1}\|_2 \equiv \|A - B_{i-1} - C_{j-1}\|_2.$$

Now we observe that the rank of a sum of two matrices cannot exceed the sum of their ranks. Therefore,

$$\mathsf{rank}(B_{i-1} + C_{j-1}) \le i + j - 2.$$

Then, by the Eckart-Young-Mirsky theorem for the 2-norm we have:

$$\|A - B_{i-1} - C_{j-1}\|_2 \ge \|A - A_{i+j-2}\|_2 = \sigma_{i+j-1}(A)$$

as desired. □

Now we prove the Eckart-Young-Mirsky theorem for the Frobenius norm.

Proof. **Proof for the Frobenius norm.** Lemma 1 implies that for any matrix M for rank $\le k$ and for all $i = 1, 2, \dots$ we have:

$$\sigma_{k+i}(A) \le \sigma_i(A - M) + \sigma_{k+1}(M) \equiv \sigma_i(A - M), \tag{11}$$

as $\sigma_{k+1}(M) = 0$ as $k + 1$ exceeds the rank of M. By (5):

$$\|A - M\|_F^2 = \sum_{i=1}^{d} \sigma_i^2(A - M) \ge \sum_{i=1}^{d-k} \sigma_i^2(A - M).$$

Then we apply (11) and obtain

$$\|A - M\|_F^2 \ge \sum_{i=1}^{d-k} \sigma_i^2(A - M) \ge \sum_{i=1}^{d-k} \sigma_{k+i}^2(A) = \sum_{j=k+1}^{d} \sigma_j^2(A)$$

as desired. □

Exercise Prove Eckart-Young-Mirsky theorem for an arbitrary Ky-Fan norm.

2.4. **Pivoted QR and pivoted Cholesky.** The pivoted QR is the QR decomposition returns a permutation matrix Π, an orthogonal matrix Q, and an upper-triangular matrix R such that

$$A\Pi = QR, \quad \text{such that} \quad r_{11} \geq r_{22} \geq \ldots \geq r_{dd}. \tag{12}$$

The matrix Π permutes columns of A. The pivoted QR can be computed in Matlab:

```
[Q,R,P] = qr(A) returns an upper triangular matrix R, a unitary matrix Q,
and a permutation matrix P, such that A*P = Q*R.
If all elements of A can be approximated by the floating-point numbers,
then this syntax chooses the column permutation P so that
abs(diag(R)) is decreasing.
Otherwise, it returns P = eye(n).
```

The permutation matrix is computed in a greedy fashion, i.e., one column in a time. The first column of Π selects the column of A with maximal Euclidean norm, i.e.,

$$A\Pi(:,1) = \mathbf{q}_1 r_{11}.$$

The second one is selected so that $A\Pi(:,2)$ has the larges component orthogonal to $\mathbf{q}_1$. And so on.

The pivoted Cholesky applied to a symmetric positive definite matrix A returns an upper-triangular matrix R and a permutation matrix P such that

$$\Pi^\top A\Pi = R^\top R, \quad \text{where} \quad r_{11} \geq r_{22} \geq \ldots \geq r_{dd} > 0. \tag{13}$$

3. Nonnegative matrix factorization (NMF)

Reference: D. Bindel's Lecture 7 "NMF". A common problem with low-rank factorizations is that they are hard-to-interpret. In this section, we switch to interpretable matrix factorizations.

Let A be an $n \times d$ matrix with nonnegative entries. We seek matrices $W \in \mathbb{R}_+^{n\times k}$ and $H \in \mathbb{R}_+^{k\times d}$ where the subscripts $+$ means that their entries must be nonnegative, such that

$$A \approx WH. \tag{14}$$

3.1. **Projected gradient descent.** Perhaps the simplest method to compute an NMF is using *projected gradient descent* (PGD). The projection used here is a simple nonnegativity constraint:

$$\mathcal{P}(\mathbf{x}) = [\mathbf{x}]_+, \quad \text{elementwise maximum of } \mathbf{x} \text{ and } 0.$$

Let ϕ be the objective function. The iteration is defined by

$$\mathbf{x}_{k+1} = \mathcal{P}(\mathbf{x}_k - \alpha_k \nabla\phi(\mathbf{x}_k)). \tag{15}$$

Its convergence properties are similar to those of the unprojected version. A convergence for convex functions and sufficiently small stepsizes can be proven. Ill-conditioning may make the convergence slow.

To work out the PGD iteration, it is handy to introduce the Frobenius inner product

$$\langle X, Y\rangle_F := \sum_{i,j} x_{ij}y_{ij} = \mathsf{trace}(X^\top Y) = \mathsf{trace}(Y^\top X). \tag{16}$$

For the problem (14), the objective function is

$$\phi(W,H) = \frac{1}{2}\|A - WH\|_F^2 = \frac{1}{2}\langle A - WH, A - WH\rangle_F. \tag{17}$$

Furthermore, to reduce the amount of writing, it is useful to use the notation $\delta\phi$, δW and δH for the variations of ϕ, W, and H, respectively. This is an analog of the differential of a function of several variables. Regular differentiation rules apply. Let $R := A - WH$. We have:

$$\begin{aligned}\delta\phi &= \frac{1}{2}\delta\langle R, R\rangle_F = \langle \delta R, R\rangle_F \\ &= -\langle(\delta W)H, R\rangle_F - \langle W(\delta H), R\rangle_F.\end{aligned} \tag{18}$$

In the calculation below, we will use the cyclic property of the trace (6) to isolate the variations of W and H in the Frobenius inner products in (18):

$$\langle(\delta W)H, R\rangle_F = \mathsf{trace}\left(H^\top(\delta W)^\top R\right) = \mathsf{trace}\left((\delta W)^\top RH^\top\right) = \langle(\delta W), RH^\top\rangle_F, \tag{19}$$

$$\langle W(\delta H), R\rangle_F = \mathsf{trace}\left((\delta H)^\top W^\top R\right) = \langle(\delta H), W^\top R\rangle_F. \tag{20}$$

These equations mean that

$$\frac{\partial\phi}{\partial W_{ij}} = -\left(RH^\top\right)_{ij}, \quad \frac{\partial\phi}{\partial H_{ij}} = -\left(W^\top R\right)_{ij}, \quad 1 \le i \le n, \quad 1 \le j \le d. \tag{21}$$

Therefore, the PGD iteration for minimizing (17) among $W \in \mathbb{R}_+^{n\times k}$ and $H \in \mathbb{R}_+^{k\times d}$ is:

$$W_{new} = \left[W + \alpha RH^\top\right]_+, \quad H_{new} = \left[H + \alpha W^\top R\right]_+. \tag{22}$$

3.2. **Multiplicative update scheme by Lee and Seung.** One of the earliest and most popular algorithms for NMF is the multiplicative update by D. D. Lee and H. S. Seung (2001) [2]. A derivation of their iteration can also be found here. In this algorithm, the entries of the matrices W and H are all updated with individually selected stepsizes. Let S_W be the matrix of stepsizes for the entries of W, and S_H be the same for H. The iteration is the projected gradient descend modified accordingly:

$$W_{new} = \left[W + S_W \odot RH^\top\right]_+, \quad H_{new} = \left[H + S_H \odot W^\top R\right]_+. \tag{23}$$

The projection in (23) zeroes out negative values. They may appear due to the subtractions hidden in $R = A - WH$ provided that all current entries in all matrices involved are nonnegative. The trick proposed by Lee and Seung allows us to avoid the need for the projection: the stepsizes are chosen so that the subtraction is eliminated! Let us rewrite (23) decoding R and removing the projection:

$$W_{new} = W + S_W \odot \left[AH^\top - WHH^\top\right], \quad H_{new} = H + S_H \odot \left[W^\top A - W^\top WH\right]. \tag{24}$$

The Lee and Seung stepsizes are

$$S_W = W \oslash \left[WHH^\top\right], \quad S_H = H \oslash \left[W^\top WH\right], \tag{25}$$

where $\oslash$ denotes entrywise division. It is easy to see that with this choice of stepsizes, the subtractions in (24) are completely eliminated as the subtracted term is canceled with the W and H, respectively. Therefore, the Lee-Seung iteration is:

$$W_{new} = \left[W \odot AH^\top\right] \oslash \left[WHH^\top\right], \quad H_{new} = \left[H \odot W^\top A\right] \oslash \left[W^\top WH\right]. \tag{26}$$

Monotone convergence of this algorithm is proven in [2]. A shortcoming of this algorithm is that it takes very conservative stepsizes, and it may take very many steps to achieve the desired convergence.

3.3. **Coordinate descent (CD).**

Coordinate descent embraces the class of methods where the update directions are chosen along particular coordinates or their blocks.

3.3.1. *One entry at-a-time.* The simplest version of CD updates one entry of W or H at-a-time. Let s be step length and $\mathbf{e}_i$ be a column vector with entry 1 at position i and the rest of its entries being zeros. To determine s for updating entry (i,j) of W, we solve the following constrained least squares problem:

$$\begin{aligned} \frac{1}{2}\left\|A-(W+s\mathbf{e}_i\mathbf{e}_j^\top)H\right\|_F^2 &= \frac{1}{2}\left\|R-s\mathbf{e}_i\mathbf{e}_j^\top H\right\|_F^2 \\ &= \frac{1}{2}\|R\|_F^2 - s\langle(\mathbf{e}_i\mathbf{e}_j^\top), RH^\top\rangle_F + \frac{s^2}{2}\|\mathbf{e}_i\mathbf{e}_j^\top H\|_F^2 \qquad (27) \\ &\text{subject to} \quad s \ge -w_{ij}. \qquad (28) \end{aligned}$$

Here, we used the cyclic property of the trace (6) and the rule:

$$\|A+B\|_F^2 = \|A\|_F^2 + \|B\|_F^2 + 2\langle A, B\rangle_F. \tag{29}$$

Exercise Prove (29).

The matrix $\mathbf{e}_i\mathbf{e}_j^\top$ has only one nonzero entry at the position (i,j) equal to one, and the matrix $\mathbf{e}_i\mathbf{e}_j^\top H$ has a single nonzero row i equal to the row j of H. Therefore,

$$\langle(\mathbf{e}_i\mathbf{e}_j^\top), RH^\top\rangle_F = (RH^\top)_{ij}, \quad ||\mathbf{e}_i\mathbf{e}_j^\top H\|_F^2 = (HH^\top)_{jj}.$$

Hence s minimizing the quadratic function

$$\frac{s^2}{2}(HH^\top)_{jj} - s(RH^\top)_{ij} + \frac{1}{2}\|R\|_F^2 \quad \text{is} \quad s = \frac{(RH^\top)_{ij}}{(HH^\top)_{jj}}.$$

Applying the constraint (28) we get the stepsize for the entry (i,j):

$$s = \max\left\{-w_{ij}, \frac{(RH^\top)_{ij}}{(HH^\top)_{jj}}\right\}. \tag{30}$$

This leads to the update for w_{ij} and for row i of R:

$$w_{ij} = w_{ij} + s, \quad R_{i,:} = R_{i,:} - sH_{j,:}. \tag{31}$$

To update entry (i,j) of H, we are solving

$$\frac{1}{2}\left\|A - W(H + s\mathbf{e}_i\mathbf{e}_j^\top)\right\|_F^2 = \frac{1}{2}\left\|R - sW\mathbf{e}_i\mathbf{e}_j^\top\right\|_F^2$$

$$= \frac{1}{2}\|R\|_F^2 - s\langle(\mathbf{e}_i\mathbf{e}_j^\top), W^\top R\rangle_F + \frac{s^2}{2}\|W\mathbf{e}_i\mathbf{e}_j^\top\|_F^2 \tag{32}$$

$$\text{subject to} \quad s \ge -h_{ij}. \tag{33}$$

The stepsize and the update is obtained by a similar calculation:

$$s = \max\left\{-h_{ij}, \frac{(W^\top R)_{ij}}{(W^\top W)_{ii}}\right\}, \quad h_{ij} = h_{ij} + s, \quad R_{:,j} = R_{:,j} - sW_{:,i}. \tag{34}$$

3.3.2. *Hierarchical alternating least squares (HALS) or rank-one residual iteration (RRI).* The formulas developed in Section 3.3.1 are readily adapted for updating one column of W at-a-time and one row of H at-a-time. The corresponding constrained least squares problems

$$\frac{1}{2}\|R - uH_{j,:}\|_F^2 \to \min \quad \text{subject to} \quad u \ge -W_{:,j}, \tag{35}$$

$$\frac{1}{2}\|R - W_{:,i}v\|_F^2 \to \min \quad \text{subject to} \quad v \ge -H_{i,:}. \tag{36}$$

These problems are equivalent to

$$\frac{1}{2}\|R - u_iH_{j,:}\|_F^2 \to \min \quad \text{subject to} \quad u_i \ge -w_{ij}, \quad 1 \le i \le n, \tag{37}$$

$$\frac{1}{2}\|R - W_{:,i}v_j\|_F^2 \to \min \quad \text{subject to} \quad v_j \ge -h_{ij}, \quad 1 \le j \le d. \tag{38}$$

Therefore, the formulas for stepsizes (30) and (34) are suitable for computing u and v. The update formulas for column j of W and the matrix R, and then row i or H and R are:

$$W_{:,j} = W_{:,j} + u, \quad R = R - uH_{j,:}; \quad H_{i,:} = H_{i,:} + v, \quad R = R - W_{:,i}v.$$

3.3.3. *Alternating non-negative least squares (ANLS).* The ANLS updates all entries of W then all entries of H by solving the following constrained convex optimization problems:

$$\phi_1(W) = \frac{1}{2}\|A - WH\|_F^2 \to \min \quad \text{subject to} \quad W \ge 0, \tag{39}$$

$$\phi_2(H) = \frac{1}{2}\|A - WH\|_F^2 \to \min \quad \text{subject to} \quad H \ge 0. \tag{40}$$

Unfortunately, contrary to HALS, these problems cannot be solved in simple closed form. One approach is to solve them using the active set method that we have studied in this course. Its shortcoming is that we add or remove only one free variable from the active set at-a-time. As a result, the active set method can take many iterations to converge in this high-dimensional problem.

4. Collaborative filtering and matrix completion

Reference: D. Bindel's Lecture 8 "Matrix Completion". Imagine a spreadsheet with columns corresponding to movies and rows corresponding to users. Each user has watched some subset of movies. The problem posed by the Netflix company is to make intelligent guesses based on the available data how much each user would like the movies that she/he hasn't watched yet and make appropriate recommendations. Similar problems are also important for other online sellers. When you are shopping for some product, you often see that the website recommends you to look at some other products by saying something like: "The customers who looked at this products also looked at these products". I see this often and find that the system predicts my tastes and my needs quite well.

In this section, we will explore some methods for making such intelligent predictions. Let A be an incomplete $n \times d$ matrix in which only a_{ij} for

$$(i,j) \in \Omega \subset \{(l_1,l_2) \mid 1 \le l_1 \le n,\ 1 \le l_2 \le d\}$$

are known. We will denote by $P_\Omega(A)$ the projection of A onto Ω:

$$P_\Omega(A) = \begin{cases} a_{ij}, & (i,j) \in \Omega, \\ 0, & \text{otherwise.} \end{cases}$$

The idea is to pick a model M for some parametric family in order to minimize some loss function. For now, we pick the squared errors loss:

$$\phi(M) = \frac{1}{2}\|P_\Omega(A - M)\|_F^2 = \frac{1}{2}\sum_{(i,j)\in\Omega}(a_{ij} - m_{ij})^2. \tag{41}$$

Let us consider some models for M to build up some intuition.

4.1. Two simple trial models.

4.1.1. *Baseline model.* Let $M = \mu 1_{n\times d}$ where μ is to be determined. Plugging M into (41) and finding the minimizer of the resulting 1D quadratic function, we find

$$\mu = \frac{1}{|\Omega|}\sum_{(i,j)\in\Omega} a_{ij}. \tag{42}$$

While this model is exactly solvable, it is useless for making predictions. It gives the same rating for all movies and all users.

4.1.2. *Baseline plus uniform adjustments for each user and each movie.* This model is given by

$$M = \mu 1_{n\times d} + \mathbf{b}1_{1\times d} + 1_{n\times 1}\mathbf{c}^\top, \quad \mathbf{b} \in \mathbb{R}^n, \quad \mathbf{c} \in \mathbb{R}^d. \tag{43}$$

The first term in (43) gives some uniform base rating to all movies for each user. The second term uniformly adjusts ratings for all movies, but these adjustments are chosen individually for each user. The third term adjusts movie rating uniformly for all users, but these adjustments are individual for each movie.

The solution to this problem is the solution to the linear system obtained by taking the gradient of

$$\phi(\mu, \mathbf{b}, \mathbf{c}) = \frac{1}{2} \sum_{(i,j)\in\Omega} (a_{ij} - \mu - b_i - c_j)^2 : \tag{44}$$

$$\frac{\partial \phi}{\partial \mu} = \sum_{(i,j)\in\Omega} (a_{ij} - \mu - b_i - c_j) = 0, \tag{45}$$

$$\frac{\partial \phi}{\partial b_i} = \sum_{j\in\Omega_i} (a_{ij} - \mu - b_i - c_j) = 0, \quad \Omega_i := \{j \mid (i,j) \in \Omega\}, \tag{46}$$

$$\frac{\partial \phi}{\partial c_j} = \sum_{i\in\Omega_j} (a_{ij} - \mu - b_i - c_j) = 0, \quad \Omega_j := \{i \mid (i,j) \in \Omega\}. \tag{47}$$

- Note that the set of solutions to (45)–(47) is at least two-dimensional:

$$\text{if } (\mu, \mathbf{b}, \mathbf{c}) \text{ is a solution, then so is } (\mu + \alpha, \mathbf{b} + \beta 1_{n\times 1}, \mathbf{c} - (\alpha+\beta)1_{d\times 1}), \quad \forall \alpha, \beta \in \mathbb{R}.$$

  However, this is not really a problem since each of these solutions give the same matrix M, hence the same recommendations. One way to get rid of this nonuniqueness is to impose the condition that $\mathbf{b}$ and $\mathbf{c}$ both sum up to zero. Them μ is given by (42).
- Also note that the resulting linear system is large: $|\Omega| \times (1 + n + d)$. This only will be a problem if we try to use factorization-based direct methods. But certain iterative methods will work just fine.
- Finally, and most importrantly, this model is still not useful as it predicts the same relative rankings for each user.

4.2. **Low-rank factorization.** An actually useful model is the low-rank model:

$$M = XY^\top, \quad X \in \mathbb{R}^{n\times k}, \quad Y \in \mathbb{R}^{d\times k}. \tag{48}$$

As the second model considered in the previous section, the factorization $M = XY^\top$ is not unique. To get rid of this nonuniqueness and to penalize crazy choices of X and Y, we regularize the problem by introducing penalty for large Frobenius norms of X and Y:

$$F(X,Y) = \frac{1}{2}\|P_\Omega(A - XY^\top)\|_F^2 + \frac{\lambda}{2}\left(\|X\|_F^2 + \|Y\|_F^2\right). \tag{49}$$

$$\text{Let } R = P_\Omega(A - XY^\top). \text{ Then } F(X,Y) = \frac{1}{2}\langle R, R\rangle_F + \frac{\lambda}{2}\left(\|X\|_F^2 + \|Y\|_F^2\right).$$

To take the variation of F, we first calculate the variation of its first term:

$$\begin{aligned}\langle \delta R, R\rangle_F &= \langle (-\delta X)Y^\top - X(\delta Y)^\top, R\rangle_F \\ &= -\langle (\delta X)Y^\top, R\rangle_F - \langle X(\delta Y)^\top, R\rangle_F \\ &= -\mathsf{trace}(Y(\delta X)^\top R) - \mathsf{trace}((\delta Y)X^\top R) \\ &= -\mathsf{trace}((\delta X)^\top RY) - \mathsf{trace}((R^\top X)^\top(\delta Y)) \\ &= -\langle \delta X, RY\rangle_F - \langle \delta Y, R^\top X\rangle_F. \qquad (50)\end{aligned}$$

Note that in the calculation above we did not need to project the first argument of the Frobenius inner product because the terms of $(-\delta X)Y^\top - X(\delta Y)^\top)$ corresponding to unknown entries of A will be zeroes out due to the second term R.

The variation of the second term of F is

$$\langle \delta X, \lambda X\rangle_F + \langle \delta Y, \lambda Y\rangle_F.$$

Hence

$$\delta F = \langle \delta X, \lambda X - RY\rangle_F + \langle \delta Y, \lambda Y - R^\top X\rangle_F. \qquad (51)$$

First, consider the case where all entries of A are available. Let us show that in this case, the solution to $F \to \min$ would be

$$X = U_k\sqrt{s_\lambda(\Sigma_k)}, \quad Y = V_k\sqrt{s_\lambda(\Sigma_k)}, \quad \text{where} \quad s_\lambda(\sigma) = [\sigma - \lambda]_+, \qquad (52)$$

and $U_k\Sigma_kV_k^\top$ is the truncated SVD of A. Note that if $\lambda = 0$ then $XY^\top = U_k\Sigma_kV_k^\top$ would be the optimal rank k approximation of k as we know from the Eckart-Young-Mirsky theorem.

We will show that, if X and Y are given by (52), then the variation of F is zero, i.e.

$$RY = \lambda X, \quad R^\top X = \lambda Y. \qquad (53)$$

Indeed,

$$\begin{aligned}RY &= U\Sigma V^\top V_k\sqrt{s_\lambda(\Sigma_k)} - U_ks_\lambda(\Sigma_k)V_k^\top V_k\sqrt{s_\lambda(\Sigma_k)} \\ &= U_k\Sigma_k\sqrt{s_\lambda(\Sigma_k)} - U_ks_\lambda(\Sigma_k)\sqrt{s_\lambda(\Sigma_k)} \\ &= U_k\left[\Sigma_k - s_\lambda(\Sigma_k)\right]\sqrt{s_\lambda(\Sigma_k)} = \lambda U_k\sqrt{s_\lambda(\Sigma_k)} = \lambda X, \\ R^\top X &= V\Sigma U^\top U_k\sqrt{s_\lambda(\Sigma_k)} - V_ks_\lambda(\Sigma_k)U_k^\top U_k\sqrt{s_\lambda(\Sigma_k)} \\ &= V_k\Sigma_k\sqrt{s_\lambda(\Sigma_k)} - V_ks_\lambda(\Sigma_k)\sqrt{s_\lambda(\Sigma_k)} \\ &= V_k\left[\Sigma_k - s_\lambda(\Sigma_k)\right]\sqrt{s_\lambda(\Sigma_k)} = \lambda V_k\sqrt{s_\lambda(\Sigma_k)} = \lambda Y.\end{aligned}$$

Here we have used the fact that for $1 \le j \le k$,

$$\sigma_j - [\sigma_j - \lambda]_+ = \begin{cases}\lambda, & \lambda \le \sigma_j, \\ \sigma_j, & \lambda > \sigma_j,\end{cases}$$

and, if $\lambda > \sigma_j$, then $s_\lambda(\sigma_j) = 0$.

The situation is trickier if only partial data are available. In this case, we need to minimize F numerically. For example, we can use stochastic gradient descent. Another option is to do the alternating iteration:

$$X^{k+1} = \arg\min_X F(X, Y^k), \tag{54}$$

$$Y^{k+1} = \arg\min_Y F(X^{k+1}, Y). \tag{55}$$

Each of these steps can be further decomposed into a collection of small linear least squares problems. For example, at each substep of (54), we solve the linear least squares problem to compute the row i of X:

$$\mathbf{x}_i^\top = \arg\min_{\mathbf{x}} \frac{1}{2}\left\|\mathbf{x}^\top Y_{\Omega_i} - a_{\Omega_i}\right\|_2^2 + \frac{\lambda}{2}\|\mathbf{x}\|^2, \tag{56}$$

where $\Omega_i := \{j \mid (i,j) \in \Omega\}$.

4.3. **Penalizing nuclear norm.** Reference: Bindel's lecture 8, Section 3 "Nuclear norm trick".

An alternative approach to optimizing factors of the model matrix M is to optimize the matrix M itself. Note that rank is not a continuous function of matrix entries, hence, imposing rank constraints is not a promising approach. Instead, we are going to penalize the nuclear norm of M, i.e.,

$$\phi(M) = \frac{1}{2}\|P_\Omega(A) - P_\Omega(M)\|_F^2 + \lambda\|M\|_*, \quad \|M\|_* = \sum_i \sigma_i(M). \tag{57}$$

The nuclear norm constraint is low-rank promoting for the same reason as the lasso regularizer is sparsity promoting. Below I offer an explanation for it.

Observe that $\phi(M)$ can be viewed as a Lagrangian function minus $t\lambda$ for the following constrained optimization problem

$$f(M) := \frac{1}{2}\|P_\Omega(A) - P_\Omega(M)\|_F^2 \ \to \ \min \quad \text{subject to} \quad t - \|M\|_* \geq 0 \tag{58}$$

for some positive constant t. If there exists a matrix M with $\|M\|_* < t$ such that $P_\Omega(A - M) = 0$, then the KKT optimality conditions require that $\lambda = 0$ as $\lambda(t - \|M\|_*) = 0$ while $t - \|M\|_* > 0$. Since we set $\lambda > 0$ in (57), this is not the case. This means that $\|M\|_* = t$.

To get a sense of what is the set $\|M\|_* = t$, let us consider a very simple example that we can visualize. Consider the set of 2×2 matrices

$$M(w,x,y,z) := \begin{bmatrix} w & x \\ y & z \end{bmatrix}.$$

Let us find a subset S_1 of $(w,x,y,z) \in \mathbb{R}^4$ such that the nuclear norm of the matrix is 1, i.e.

$$S_1 := \left\{(w,x,y,z) \in \mathbb{R}^4 \mid \sigma_1(M(w,x,y,z)) + \sigma_2(M(w,x,y,z)) = 1\right\}. \tag{59}$$

The set S_1 is a 3D surface in a 4D space. We are particularly interested in two aspects of S_1:

- Does the surface have singularities (2D edges)? This is because the level sets of $f(M)$ in (58) are smooth ellipsoids, and the minimal ellipsoid having a nonempty intersection with the surface $\|M\|_*$ tends to have this intersection on a singular edge.
- Do these singularities of the surface correspond to a low rank of $M(w,x,y,z)$?

We cannot visualize a 3D surface in 4D, but we can visualize a family of its 2D slices each of which corresponds to a fixed value of w. Three of these slices are displayed in Fig. 1. We color the slices according to the value of the determinant of $M(w,x,y,z)$. If $\det(M(w,x,y,z)) = 0$ then $\mathsf{rank}(M(w,x,y,z)) < 2$. We see that each slice has singular a singular edge, and the surface color near the edge is green which corresponds to $\det(M(w,x,y,z)) = 0$. The full set of slices is shown in the Youtube video.

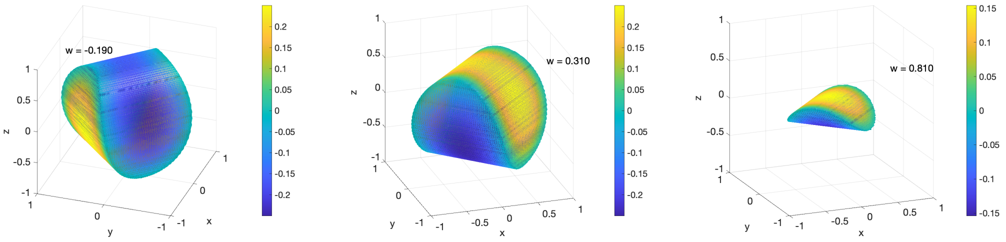

FIGURE 1. Slices of the set S_1 (see (59)) corresponding to $w = -0.19$ (left), $w = 0.31$ (middle), and $w = 0.81$ (right). The coloring of the surfaces corresponds to the values of $\det(M(w,x,y,z))$. Note that the edge of these surfaces is green which corresponds to $\det(M(w,x,y,z)) = 0$.

Now let us discuss methods for solving the minimization problem (57). It is helpful to see what is the solution to it in the case if all entries of A are known, the so-called *proximal problem*. Then (57) becomes

$$\phi(M) = \frac{1}{2}\|A - M\|_F^2 + \lambda\|M\|_*. \tag{60}$$

Its minimizer is given by

$$S_\lambda(A) := U s_\lambda(\Sigma) V^\top, \tag{61}$$

where $A = U\Sigma V^\top$ is the SVD of A and $s_\lambda(\sigma_j) = \max\{\sigma_j - \lambda, 0\}$ as before. Note that if there are exactly k singular values of A greater than λ then (61) is also the minimizer for the problem considered in Section (4.2):

$$F(X,Y) = \frac{1}{2}\left\|A - XY^\top\right\|_F^2 + \frac{\lambda}{2}\left(\|X\|_F^2 + \|Y\|_F^2\right) \to \min, \quad X \in \mathbb{R}^{n\times k},\ Y \in \mathbb{R}^{d\times k}. \tag{62}$$

A similar result holds when only part of the data matrix A is available: the nuclear norm regularization and the optimization of the factored form with Frobenius norm regularization on the factors yield the same model predictions when the factor size k in the latter problem is at least as large as the rank observed in the nuclear norm problem.

The SVD solution (61) to the proximal problem suggests the following iteration:

$$M^{j+1} = S_\lambda \left(M^j + P_\Omega(A - M^j) \right). \tag{63}$$

Since there are only a few singular values greater than λ at each step, the necessary components of the SVD at each step can be computed very efficiently using a Lanczos-type algorithm (see e.g. Trefenthen and Bau "Numerical Linear Algebra").

5. CUR matrix decomposition

Please read the PNAS article by M. Mahoney and P. Drineas of 2009 [3]. The CUR algorithm in it is the one I would like you to implement. The preceding article by the same authors plus S. Muthukrishnan [4] contains detailed proofs and more complicated algorithms with better worst-case scenario guarantees. A background material on leverage scores can be found e.g. here. In addition, here is a nice lecture on CUR by Jeff M. Philips, University of Utah.

References

[1] J. W. Demmel, *Applied Numerical Linear Algebra*. SIAM, 1997.

[2] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Proc. Neural Information Processing Systems*, 2001.

[3] M. W. Mahoney and P. Drineas, "Cur matrix decompositions for improved data analysis," *PNAS*, vol. 106, no. 3, pp. 697–702, 2009.

[4] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, "Relative-error cur matrix decompositions," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 2, pp. 844–881, 2008.