1 Outline

In this lecture, we study

- matrix completion via the singular value decomposition,
- matrix completion via the power method,
- matrix completion via projected gradient descent,
- the Frank-Wolfe algorithm,
- matrix completion via the Frank-Wolfe algorithm.

2 Matrix Completion via the Singular Value Decomposition

Let us consider

$$\min_{X\in\mathbb{R}^{n\times p}} \quad \|D-X\|_F \quad \text{subject to} \quad \operatorname{rank}(X)\le k$$

where

- D is an $n\times p$ matrix,
- $\|A\|_F$ denotes the Frobenius norm, i.e., $\|A\|_F=\sqrt{\sum_{i=1}^n\sum_{j=1}^p A_{ij}^2}$.

By the definition of the Frobenius norm, the problem is equivalent to

$$\min_{X\in\mathbb{R}^{n\times p}} \quad \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^p (D_{ij}-X_{ij})^2 \quad \text{subject to} \quad \operatorname{rank}(X)\le k.$$

2.1 Applications

The most common application is matrix compression, where the goal is to provide a lossy compressed version of a given matrix. There are other practical applications, which we elaborate on below.

Movie Recommendation We discussed movie recommendation as an application where D is a user-rating matrix for movies. Typically, D is a sparse matrix with a huge number of rows and a large number of columns. The missing entries are filled with 0. The goal is to infer the values of the missing entries, thereby predicting the unseen user ratings for movies. As explained before, the hypothesis is that the true user-rating matrix $X\in\mathbb{R}^{n\times p}$ is generated by the product of an

user-feature matrix $U \in \mathbb{R}^{n\times k}$ and a movie-feature matrix $V \in \mathbb{R}^{p\times k}$ over k features as follows.

$$\begin{bmatrix} & & \\ & X & \\ & & \end{bmatrix} = \begin{bmatrix} & & \\ & U & \\ & & \end{bmatrix} \begin{bmatrix} & V^\top & \end{bmatrix}.$$

Then matrix X has rank at most k.

Localization in Sensor Networks Suppose that there are n sensors whose physical locations are given by coordinate vectors $x_1, \ldots, x_n \in \mathbb{R}^3$. Here, these vectors are not given while we have access to the distance information about some pairs of distinct sensors. Note that the distance between x_i and x_j is given by

$$D_{ij} = \|x_i - x_j\|_2^2 = \|x_i\|_2^2 + \|x_j\|_2^2 - 2x_i^\top x_j.$$

Let X be the $n \times 3$ matrix whose rows are $x_1^\top, \ldots, x_n^\top$, $y \in \mathbb{R}^n$ be the vector whose components are $\|x_1\|_2^2, \ldots, \|x_n\|_2^2$, and let $\mathbf{1} \in \mathbb{R}^n$ be the vector of all ones. Then it follows that

$$D = y\mathbf{1}^\top + \mathbf{1}^\top y - 2XX^\top.$$

As the rank of X is at most 3, the rank of matrix D is at most 5. Basically, we are given only a subset of the entries of D, from which we want to predict matrix the sensor location matrix X.

2.2 Low-Rank Approximation by the Singular Value Decomposition

Recall that the singular value decomposition (SVD) of a matrix $D \in \mathbb{R}^{n\times p}$ is given by

$$D = U\Sigma V^\top$$

where

- $r = \min\{n, p\}$,
- $U \in \mathbb{R}^{n\times r}$ is an $n \times r$ matrix with orthonormal columns,
- $\Sigma \in \mathbb{R}^{r\times r}$ is a $r \times r$ diagonal matrix whose entries are the singular values of D: $\sigma_1 \geq \cdots \geq \sigma_r \geq 0$,
- $V \in \mathbb{R}^{p\times r}$ is a $p \times r$ matrix with orthonormal columns.

Here, for any $k \leq r$, let U_k denote the column submatrix of U that consists of the first k columns of U, and let V_k denote the column submatrix of V that consists of the first k columns of V. Moreover, let Σ_k be the $k \times k$ diagonal matrix whose entries are the top k singular values $\sigma_1, \ldots, \sigma_k$ of D. Then we may consider

$$D_k = U_k\Sigma_k V_k^\top. \tag{11.1}$$

Here, the matrix D_k is of rank k. We can argue is that D_k is the best rank-k approximation of D. To be more precise, we have the following result.

Theorem 11.1. *For any $k \leq \min\{n, p\}$, we have*

$$D_k \in \operatorname{argmin}_{X \in \mathbb{R}^{n \times p}} \{\|D - X\|_F : \ \operatorname{rank}(X) \leq k\}$$

where D_k is defined as in (11.1).

Hence, one can compute an optimal low-rank approximation by computing the singular value decomposition. It is known that the complexity of deriving the singular value decomposition is

$$O(npr).$$

When D is an $n = p = r$, the complexity is of order $O(n^3)$, which is typically inefficient in practice.

2.3 Low-Rank Approximation by the Power Method

Instead of computing the full singular value decomposition of a given matrix D, we may use the power method to compute the optimal low-rank approximation D_k of rank k. Recall that we can compute the top left singular vector u_1 and the top right singular vector v_1 as well as the largest singular value σ_1 by the power method as follows.

Algorithm 1 Power Method for the Top Left Singular Vector

Initialize $\bar{u}_0 \in \mathbb{R}^n \setminus \{0\}$ and $u_0 = \bar{u}_0 / \|\bar{u}_0\|_2$
for $t = 1, \ldots, T$ **do**
  Update $\bar{u}_t = (DD^\top)^t x_0$
  Obtain $u_t = \bar{u}_t / \|\bar{u}_t\|_2$
end for
Return u_T.

Algorithm 2 Power Method for the Top Right Singular Vector

Initialize $\bar{v}_0 \in \mathbb{R}^p \setminus \{0\}$ and $v_0 = \bar{v}_0 / \|\bar{v}_0\|_2$
for $t = 1, \ldots, T$ **do**
  Update $\bar{v}_t = (D^\top D)^t v_0$
  Obtain $v_t = \bar{v}_t / \|\bar{v}_t\|_2$
end for
Return v_T.

Then it follows that

$$D - u_1\sigma_1 v_1^\top = U\Sigma V^\top - u_1\sigma_1 v_1^\top = \begin{bmatrix} u_2 & \cdots & u_r \end{bmatrix} \begin{bmatrix} \sigma_2 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix} \begin{bmatrix} v_2^\top \\ \vdots \\ v_r^\top \end{bmatrix}.$$

Then we may apply the power method to the matrix $D - u_1\sigma_1 v_1^\top$ to compute u_2, v_2, and σ_2, which are the top left singular vector, the top right singular vector, and the largest singular value of $D - u_1\sigma_1 v_1^\top$.

2.4 Low-Rank Approximation via the Projected Gradient Descent

Note that

$$f(X) = \frac{1}{2}\|D - X\|_F^2 = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}(D_{ij} - X_{ij})^2$$

is convex in X. However, the constraint $\text{rank}(X) \leq k$ induces a nonconvex feasible set. Then we may take the following relaxation using the **nuclear norm**.

$$\min_{X \in \mathbb{R}^{n\times p}} \quad \frac{1}{2}\|D - X\|_F^2 \quad \text{subject to} \quad \|X\|_* \leq k \tag{11.2}$$

where

$$\|X\|_* = \text{Trace}(\sqrt{X^\top X}) = \sum_{i=1}^{\min\{n,p\}} \sigma_i(X) \leq k$$

and $\sigma_1(X), \ldots, \sigma_{\min\{n,p\}}(X)$ are the singular values of X. It is known that the nuclear norm is a convex function in X.

In problem (11.2), the constraint set

$$B = \left\{X \in \mathbb{R}^{n\times p} : \ \|X\|_* \leq k\right\}$$

is what is obtained from scaling up the unit nuclear norm ball $\{X \in \mathbb{R}^{n\times p} : \ \|X\|_* \leq 1\}$. To solve (11.2), we may apply **projected gradient descent** over the constraint set B. It is known that projection onto the constraint set B amounts to computing the singular value decomposition of matrix D [DSSSC08].

3 Matrix Completion via the Frank-Wolfe Algorithm

Given a matrix $D \in \mathbb{R}^{n\times p}$, let us consider

$$\min_{X \in \mathbb{R}^{n\times p}} \quad \|D - O \odot X\|_F \quad \text{subject to} \quad \text{rank}(X) \leq k$$

where

- O is the binary matrix with
$$O_{ij} = \begin{cases} 1, & \text{if } D_{ij} \neq 0, \\ 0, & \text{if } D_{ij} = 0, \end{cases}$$
- $O \odot X$ is the **Hadamard product** of O and X given by
$$(O \odot X)_{ij} = O_{ij}X_{ij}.$$

Basically, $O \odot X$ is what is obtained from X after keeping only the entries that correspond to the observable entries of D. For this version of the matrix completion problem, we need a different method. As before, we take a relaxation using the nuclear norm as follows.

$$\min_{X \in \mathbb{R}^{n\times p}} \quad \frac{1}{2}\|D - O \odot X\|_F^2 \quad \text{subject to} \quad \|X\|_* \leq k.$$

Moreover, note that

$$f(X) = \frac{1}{2}\|D - O \odot X\|_F^2 = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}(D_{ij} - O_{ij}X_{ij})^2$$

is convex in X as D and O are fixed matrices. Note that

$$\nabla f(X) = O \odot X - D.$$

In general, we may consider any convex function $f(X)$ on X and

$$\min_{X \in \mathbb{R}^{n \times p}} \quad f(X) \quad \text{subject to} \quad \|X\|_* \leq k \tag{11.3}$$

with the gradient $\nabla f(X)$ of the function $f(X)$. To solve (11.3), we apply another interative algorithm, the **Frank-Wolfe method**.

3.1 Frank-Wolfe Method

In this section, we introduce the **conditional gradient method**, introduced by Frank and Wolfe in 1956 [FW56]. Named after the author, the conditional gradient method is often referred to as the **Frank-Wolfe algorithm**. We consider the following convex optimization problem

$$\min_{x \in \mathbb{R}^d} \quad f(x) \quad \text{subject to} \quad x \in C$$

where f is β-smooth in a norm $\|\cdot\|$ for some $\beta > 0$ and C is a convex set. A pseudo-code of the method is given in Algorithm 3.

Algorithm 3 Frank-Wolfe Algorithm

Initialize $x_1 \in C$.
for $t = 1, \ldots, T-1$ **do**
  Take $v_t \in \operatorname{argmin}_{v \in C} \nabla f(x_t)^\top v$.
  Update $x_{t+1} = (1 - \lambda_t)x_t + \lambda_t v_t$ for some $0 < \lambda_t < 1$.
end for
Return x_T.

The main component of the conditional gradient method is to compute the direction v_t by solving

$$\min_{v \in C} \nabla f(x_t)^\top v$$

whose objective is a linear function. In particular, when C is a polyhedron, it is just a linear program. Figure 11.1 provides a pictorial description of the update rule of the Frank-Wolfe algorithm. v_t is a point up to which we can move as far as we can in the direction of $-\nabla f(x_t)$ within C. Then we take a convex combination of the current point x_t and v_t to obtain the new iterate v_{t+1}.

The next theorem shows that conditional gradient descent converges with rate $O(1/T)$ for any smooth function with repsect to an arbitrary norm.

Theorem 11.2. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex function that is β-smooth in a norm $\|\cdot\|$ for some $\beta > 0$. Let $\{x_t : t = 1, \ldots, T\}$ be the sequence of iterates generated by the Frank-Wolfe algorithm with*

$$\lambda_t = \frac{2}{t+1}$$

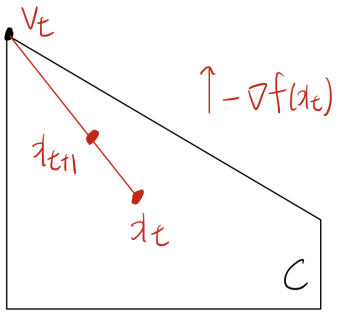

Figure 11.1: Illustration of an update from conditional gradient descent

for each t. Then for any $t \geq 2$,

$$f(x_t) - f(x^*) \leq \frac{2\beta R^2}{t+1}$$

where x^* is an optimal solution to $\min_{x \in C} f(x)$ and $R = \sup_{x,y \in C} \|x - y\|$.

Proof. Note that

$$\begin{aligned} f(x_{t+1}) - f(x_t) &\leq \nabla f(x_t)^\top (x_{t+1} - x_t) + \frac{\beta}{2}\|x_{t+1} - x_t\|^2 \\ &= \lambda_t \nabla f(x_t)^\top (v_t - x_t) + \frac{\beta}{2}\|x_{t+1} - x_t\|^2 \\ &\leq \lambda_t \nabla f(x_t)^\top (x^* - x_t) + \frac{\beta}{2}\|x_{t+1} - x_t\|^2 \\ &\leq \lambda_t (f(x^*) - f(x_t)) + \frac{\beta}{2}\|x_{t+1} - x_t\|^2 \end{aligned}$$

where the first inequality is from the β-smoothness of f, the first equality follows from $x_{t+1} = (1 - \lambda_t)x_t + \lambda_t v_t$, the second inequality is due to the definition of $v_t = \operatorname{argmin}_{v \in C} \nabla f(x_t)^\top v$, and the last inequality is by the convexity of f. Since

$$\|x_{t+1} - x_t\| = \lambda_t \|v_t - x_t\| \leq \lambda_t R,$$

it follows that

$$\begin{aligned} f(x_{t+1}) - f(x^*) &\leq (1 - \lambda_t)(f(x_t) - f(x^*)) + \frac{\beta \lambda_t^2 R^2}{2} \\ &= \frac{t-1}{t+1}(f(x_t) - f(x^*)) + \frac{2\beta R^2}{(t+1)^2}. \end{aligned}$$

By this inequality, it follows that

$$f(x_2) - f(x^*) \leq \frac{\beta R^2}{2} \leq \frac{2\beta R^2}{3}.$$

Then by the induction hypothesis,

$$f(x_{t+1}) - f(x^*) \leq \frac{2(t-1)+2}{(t+1)^2}\beta R^2 = \frac{t}{(t+1)^2} 2\beta R^2 \leq \frac{1}{t+2}\beta R^2,$$

as required. □

3.2 Applying the Frank-Wolfe method

The Frank-Wolfe algorithm applied to (11.3) has the following step of computing the direction V_t. Given $X_t \in \mathbb{R}^{n\times p}$,

$$V_t \in \operatorname{argmin}_{V\in B} \nabla f(X_t)^\top V$$

where

$$B = \left\{X \in \mathbb{R}^{n\times p} : \ \|X\|_* \le k\right\}.$$

We will show that V_t can be computed by the power method! To argue this, we need the following lemmas.

Lemma 11.3. *The unit nuclear norm ball is equivalent to the convex hull of rank 1 matrices, i.e.,*

$$\left\{X \in \mathbb{R}^{n\times p} : \ \|X\|_* \le 1\right\} = \operatorname{conv}\left\{uv^\top : \ \|u\|_2 = \|v\|_2 = 1, \ u \in \mathbb{R}^n, \ v \in \mathbb{R}^p\right\}.$$

Based on Lemma 11.3, we prove the second lemma.

Lemma 11.4. *Let $A \in \mathbb{R}^{n\times p}$ be an $n \times p$ matrix. Let u and v be the top left and right singular vectors of $-A$, respectively. Then*

$$k \cdot uv^\top \in \operatorname{argmin}\left\{A^\top X : \ \|X\|_* \le k\right\}.$$

By Lemma 11.4, it follows that we can set V_t as

$$V_t = k \cdot u_t v_t^\top$$

where u_t and v_t are the top left and right singular vectors of

$$-\nabla f(X_t) = D - O \odot X_t,$$

respectively. Here, u_t and v_t can be computed by the power method. To summarize, we get the following pseudo-code.

Algorithm 4 Matrix Completion by the Frank-Wolfe Algorithm

Initialize $X_1 \in B$.
for $t = 1, \ldots, T-1$ **do**
  Compute the top left and right singular vectors u_t and v_t of $D - O \odot X_t$ by the power method
  Update $X_{t+1} = (1-\lambda_t)X_t + \lambda_t V_t$ for some $0 < \lambda_t < 1$.
end for
Return X_T.

References

[DSSSC08] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 272–279, New York, NY, USA, 2008. Association for Computing Machinery. 2.4

[FW56] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956. 3.1