ECE 273 Convex Optimization and Applications — Instructor: Jun-Kun Wang

Scribe: Nikola Raicevic, Michael Tang, Ibrahim Kilinc — April 18, 2024

Editor/TA: Marialena Sfyraki

Lecture 6: Projected Gradient Descent and Frank-Wolfe Method

1 Preliminaries

Optimality Conditions of Constrained Convex Optimization

Theorem 1. *Assume f is a convex function, then (saying that)*

$$x_* \in \arg\min_{x \in C} f(x)$$

iff *there exists a subgradient g_{x_*} such that for any $y \in C$*

$$\langle g_{x_*}, y - x_* \rangle \geq 0$$

Corollary: When $C = \mathbb{R}^d$: the statement $\langle g_{x_*}, y - x_* \rangle \geq 0, \forall y \in \mathbb{R}^d$ is equivalent to $0 \in \partial f(x_*)$.

Theorem 2. *Assume f is a convex function and* ***differentiable****, then (saying that)*

$$x_* \in \arg\min_{x \in C} f(x) \tag{1}$$

iff *for any $y \in C$*

$$\langle \nabla f(x_*), y - x_* \rangle \geq 0 \tag{2}$$

Minimum v.s. Infimum
The minimum value of a function needs to be attained. However, the minimum does not necessarily exist, whereas, the infimum of a function is its largest lower bound, which always exists.

1. Example 1: $\min_{x \in \mathbb{R}} exp(-x) \quad$ vs. $\quad \inf_{x \in \mathbb{R}} exp(-x) = 0$
2. Example 2: $\min_{x \in \mathbb{R}} log(1 + exp(-x)) \quad$ vs. $\quad \inf_{x \in \mathbb{R}} log(1 + exp(-x)) = 0$

Definition 1. ***(Gradient Dominant or Polyak-Lojasiewicz (PL) Condition)****: We say a function $f : \mathbb{R}^d \to \mathbb{R}$ satisfies the "Gradient Dominance" condition if*

$$||\nabla f(\mathbf{x})||_2^2 \geq 2\mu \left(f(\mathbf{x}) - \min_{\mathbf{x}} f(\mathbf{x}) \right), \quad \textit{for some} \;\; \mu > 0.$$

Example: $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top A\mathbf{x}$, where $A \succeq 0$, is a convex function but not strongly convex.

Remark: f satisfies the μ-PL condition with the constant $\mu = \lambda_{i_*}$, the smallest positive eigenvalue of A.

Proof. Denote the eigen-decomposition of $A = \sum_{i=1}^d \lambda_i \mathbf{u}_i \mathbf{u}_i^\top$, where λ_i's and $\mathbf{u}_i$'s are eigenvalues and eigenvectors. As $0 = \min_{\mathbf{x}} f(\mathbf{x})$ and $\nabla f(\mathbf{x}) = A\mathbf{x}$, it suffices to establish the following inequality:

$$\mathbf{x}^\top A^\top A\mathbf{x} \geq \lambda_{i_*}\mathbf{x}^\top A\mathbf{x} \iff \sum_{i=1}^d \lambda_i^2(\mathbf{x}^\top \mathbf{u}_i)^2 \geq \lambda_{i_*} \sum_{i=1}^d \lambda_i(\mathbf{x}^\top \mathbf{u}_i)^2 \tag{3}$$

Denote $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{i_*} > \lambda_{i_*+1} = \cdots = 0$. Then, the above is equivalent to

$$\sum_{i=1}^{i_*} \lambda_i^2(\mathbf{x}^\top \mathbf{u}_i)^2 \geq \lambda_{i_*} \sum_{i=1}^{i_*} \lambda_i(\mathbf{x}^\top \mathbf{u}_i)^2, \tag{4}$$

which is true since $\lambda_i \geq \lambda_{i_*}$ for $i \in [i_*]$, i.e. $\lambda_i \geq \lambda_{i_*}$ for $i \leq i_*$ □

Constrained optimization: A constrained optimization problem is an optimization problem in which we aim to optimize a function f over a set $C \subset \mathbb{R}^d$. It can be represented in the following form:

$$\min_{\mathbf{x} \in C} f(\mathbf{x})$$

2 Projected Gradient Descent (PGD)

2.1 PGD: Algorithm

Algorithm 1 is a formal statement of the PGD algorithm. In addition to GD, it has a projection step after each GD calculation.

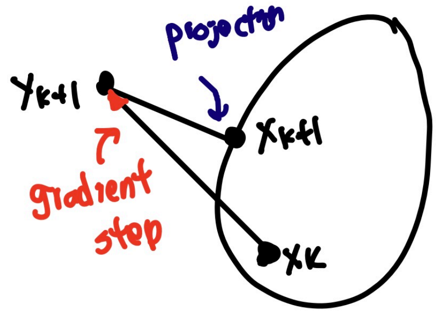


Figure 1: The illustration of the PGD algorithm

Algorithm 1 The steps of the PGD algorithm

1: **for** $k = 1, 2, \ldots$ **do**
2: $\quad \mathbf{x}_{k+1} = \mathrm{Proj}_C \left[\mathbf{x}_k - \eta \nabla f(\mathbf{x}_k)\right]$
3: **end for**

The projection for the projection step is defined as finding the point in C with the minimum Euclidean distance to a given point. The analytical expression for projection is expressed as:

$$\mathrm{Proj}_C(\mathbf{y}) := \arg\min_{\mathbf{x} \in C} \|\mathbf{y} - \mathbf{x}\|_2^2, \tag{5}$$

where $\mathrm{Proj}_C(\mathbf{y})$ means given $\mathbf{y}$ find the projection of $\mathbf{y}$ onto set C.

2.2 GD and PGD

In this subsection, we introduce the convergence rate of GD and PGD for L-smooth convex, and μ-strongly convex functions. The convergence rate of GD and PGD is the same as seen in Table 1 and 2. The convergence rate of the L-smooth convex functions is sublinear for the GD and PGD. The convergence rate of the L-smooth and μ-strongly convex functions is linear for the GD and PGD.

ϵ-optimality gap: $f(\mathbf{x}_k) - \min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}) \le \epsilon$	
L-smooth convex	$O\left(\frac{L}{k}\right)$
L-smooth and μ-strongly convex	$O\left(\exp\left(-\frac{\mu}{L}k\right)\right)$

Table 1: GD: $\min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x})$

ϵ-optimality gap: $f(\mathbf{x}_k) - \min_{\mathbf{x}\in C} f(\mathbf{x}) \leq \epsilon$	
L-smooth convex	$O\left(\frac{L}{k}\right)$
L-smooth and μ-strongly convex	$O\left(\exp\left(-\frac{\mu}{L}k\right)\right)$

Table 2: PGD for $\min_{\mathbf{x}\in C} f(\mathbf{x})$

2.3 When to choose PGD?

Finding the projection is another optimization problem, i.e,

$$\text{Proj}_C(\mathbf{y}) := \arg\min_{\mathbf{x}\in C} \|\mathbf{y} - \mathbf{x}\|_2^2$$

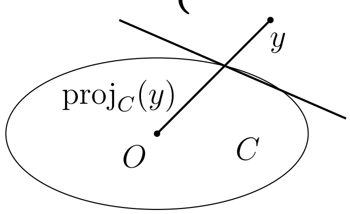
$$\min_{\mathbf{x}\in C} \|\mathbf{x} - \mathbf{y}\|_2^2$$

as shown in the PGD algorithm in Algorithm 1. Therefore, the PGD should be selected over GD when the projection step has a closed-form solution or there exists an efficient/specialized algorithm to solve projection.

2.4 How to implement the projection: $\arg\min_{\mathbf{x}\in C} \|\mathbf{y} - \mathbf{x}\|_2^2$

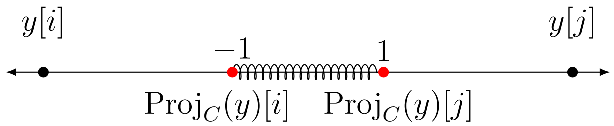
Example 1: (with closed-form solution)
Let $C := \{\mathbf{x} \in \mathbb{R}^d : ||\mathbf{x}||_2 \leq 1\}$. Then,

$$\text{Proj}_C(y) = \begin{cases} \frac{y}{\|y\|_2}, & \text{if } y \notin C \\ y, & \text{otherwise} \end{cases}$$



Example 2: (with closed-form solution)
Let $C := \{\mathbf{x} \in \mathbb{R}^d : ||\mathbf{x}||_\infty \leq 1\}$, with $||\mathbf{x}||_\infty := \max_i |\mathbf{x}[i]|$.
Then, $\forall i \in [d], -1 \leq x[i] \leq 1$

$$\text{Proj}_C(y)[i] = \begin{cases} 1, & \text{if } y[i] > 1 \\ -1, & \text{if } y[i] < -1 \\ y[i], & \text{otherwise} \end{cases}$$



Example 3: (without closed-form solution)
Let $C := \{\mathbf{x} \in \mathbb{R}^d : ||\mathbf{x}||_1 \leq 1\}$.
Denote $(\mathbf{z})_+ \triangleq \max\{0, \mathbf{z}\}$.
Then, we have the following *Characterization of* $\text{Proj}_C(\mathbf{y})$ *when* $\mathbf{y} \notin C$

$$\text{Proj}_C(y)[i] \triangleq \hat{\mathbf{x}}[i] = \text{sign}(\mathbf{y}[i])\,(|\mathbf{y}[i]| - \lambda)_+,$$

where λ is the solution to $\sum_{i=1}^d (|\mathbf{y}[i]| - \lambda)_+ = 1$.

2.5 Optimality Gap of PGD

Recall the update step of PGD: $\mathbf{x}_{k+1} = \text{Proj}_C\left[\mathbf{x}_k - \eta \nabla f(\mathbf{x}_k)\right]$.

Theorem 3. *Let $f(\cdot)$ be L-smooth and μ-strongly convex. Denote* $\mathbf{x}_* := \arg\min_{\mathbf{x} \in C} f(x)$. *With step size $\eta = \frac{1}{L}$, PGD has*

$$\|\mathbf{x}_{K+1} - \mathbf{x}_*\|_2^2 \leq \left(1 - \frac{\mu}{L}\right)^K \|\mathbf{x}_1 - \mathbf{x}_*\|_2^2. \tag{6}$$

Proof. By L-smoothness:

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) \leq \langle \nabla f(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k \rangle + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2^2 \tag{7}$$

and by μ-strong convexity:

$$f(\mathbf{x}_k) - f(\mathbf{x}_*) \leq \langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{x}_* \rangle - \frac{\mu}{2}\|\mathbf{x}_k - \mathbf{x}_*\|_2^2 \tag{8}$$

We introduce the following lemma, which can be proven by adding (7) and (8):

Lemma 1. *If $f(\cdot)$ is L-smooth and $\mu-$strongly convex, the following holds:*

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}_*) \leq \langle \nabla f(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_* \rangle + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2^2 - \frac{\mu}{2}\|\mathbf{x}_k - \mathbf{x}_*\|_2^2 \tag{9}$$

We define

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x} \in C} \|\mathbf{x} - \big(\mathbf{x}_k - \eta \nabla f(\mathbf{x}_k)\big)\|_2^2$$

By the optimality condition of $\mathbf{x}_{k+1}$, we know

$$\langle \mathbf{x}_{k+1} - \left(\mathbf{x}_k - \eta \nabla f(\mathbf{x}_k)\right), \mathbf{z} - \mathbf{x}_{k+1} \rangle \geq 0, \quad \forall \mathbf{z} \in C. \tag{10}$$

By setting $\mathbf{z} = \mathbf{x}^*$, we can rearrange (10) into

$$\langle \nabla f(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_* \rangle \leq \frac{1}{\eta} \langle \mathbf{x}_{k+1} - \mathbf{x}_k, \mathbf{x}_* - \mathbf{x}_{k+1} \rangle. \tag{11}$$

We can plug the estimate (11) into the Lemma 1 to obtain

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}_*) \leq \frac{1}{\eta} \langle \mathbf{x}_{k+1} - \mathbf{x}_k, \mathbf{x}_* - \mathbf{x}_{k+1} \rangle + \frac{L}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2^2 - \frac{\mu}{2} \|\mathbf{x}_k - \mathbf{x}_*\|_2^2,$$

which can be rearranged into

$$\begin{aligned} -\frac{1}{\eta} \langle \mathbf{x}_{k+1} - \mathbf{x}_k, \mathbf{x}_* - \mathbf{x}_{k+1} \rangle &\leq f(\mathbf{x}_*) - f(\mathbf{x}_{k+1}) + \frac{L}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2^2 - \frac{\mu}{2} \|\mathbf{x}_k - \mathbf{x}_*\|_2^2 \\ &\leq \frac{L}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2^2 - \frac{\mu}{2} \|\mathbf{x}_k - \mathbf{x}_*\|_2^2, \end{aligned} \tag{12}$$

where the bottom inequality follows from the fact that $f(\mathbf{x}_*) - f(\mathbf{x}_{k+1}) \leq 0$. Then we have

$$\begin{aligned} \|\mathbf{x}_{k+1} - \mathbf{x}_*\|_2^2 &= \|\mathbf{x}_k - (\mathbf{x}_k - \mathbf{x}_{k+1}) - \mathbf{x}_*\|_2^2 \\ &= \|\mathbf{x}_k - \mathbf{x}_*\|_2^2 - 2\langle \mathbf{x}_{k+1} - \mathbf{x}_k, \mathbf{x}_* - \mathbf{x}_k \rangle + \|\mathbf{x}_k - \mathbf{x}_{k+1}\|_2^2 \\ &= \|\mathbf{x}_k - \mathbf{x}_*\|_2^2 - 2\langle \mathbf{x}_{k+1} - \mathbf{x}_k, \mathbf{x}_* - \mathbf{x}_{k+1} + \mathbf{x}_{k+1} - \mathbf{x}_k \rangle + \|\mathbf{x}_k - \mathbf{x}_{k+1}\|_2^2 \\ &\leq \|\mathbf{x}_k - \mathbf{x}_*\|_2^2 + (L\eta - 1)\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 - \eta\mu \|\mathbf{x}_k - \mathbf{x}_*\|_2^2 \qquad (13) \\ &\leq (1 - \eta\mu) \|\mathbf{x}_k - \mathbf{x}_*\|_2^2. \qquad (14) \end{aligned}$$

where (13) follows from (12) and (14) follows from the fact that $\eta \leq \frac{1}{L}$. By recursively applying this estimate from $k = K$ to $k = 1$, we complete the proof. □

3 Frank-Wolfe Method

The Frank-Wolfe algorithm is an iterative method to solve constrained optimization problems. More formally, it can be stated as follows:

Algorithm 2 The steps of Frank-Wolfe method

1: Initialize $\mathbf{x}_1 \in C$ (convex set)
2: **for** $k = 1, 2, \ldots$ **do**
3: $\quad \mathbf{v}_k = \arg\min_{\mathbf{v} \in C} \langle \mathbf{v}, \nabla f(\mathbf{x}_k) \rangle$ (linear optimization)
4: $\quad \mathbf{x}_{k+1} = (1 - \eta_k)\mathbf{x}_k + \eta_k \mathbf{v}_k$, where $\eta_k \in [0, 1]$.
5: **end for**

Step 4 is called the *convex averaging step*. Note that C being convex guarantees $\mathbf{x}_k \in C$ for all k values. To see why, we already know for the base case we initialize $\mathbf{x}_1 \in C$. Then, if we suppose $\mathbf{x}_{k_*} \in C$ we know $\mathbf{v}_{k_*} \in C$ by how we define $\mathbf{v}_{k_*}$ in algorithm 3. Since C is convex, we also know $\mathbf{x}_{k_*+1} = (1-\eta_{k_*})\mathbf{x}_{k_*} + \eta_{k_*}\mathbf{v}_{k_*} \in C$ since $\eta_{k_*} \in [0,1]$. By induction, we conclude $x_k \in C$, $\forall k$.

Geometric Illustration

Consider the probability simplex in $\mathbb{R}^2$ defined by $\Delta_2 = \{\mathbf{v} \in \mathbb{R}^2 : \mathbf{v}[1], \mathbf{v}[2] \geq 0, \mathbf{v}[1] + \mathbf{v}[2] \leq 1\}$. On the $\mathbb{R}^2$ plane, this looks like a triangle with vertices on $(0,0), (1,0), (0,1)$. Suppose $\nabla f(\mathbf{x}_k) = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \in \mathbb{R}^2$, then it can be verified that $\mathbf{v}_k = \arg\min_{\mathbf{v}\in C} \left\langle \mathbf{v}, \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $\forall k$. This makes sense intuitively if we interpret it in a game-theoretic context, where we suppose $\mathbf{v}[1]$ and $\mathbf{v}[2]$ represent how one allocates a total of "1" resources. If the person wants to minimize a certain linear objective function, they should put all their resources in the direction that decreases this objective function the most significantly. In this specific example, that would be the $\mathbf{v}[2]$ direction, since the objective function in this case would be $\left\langle \mathbf{v}, \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\rangle = \mathbf{v}[1] - \mathbf{v}[2]$. Hence, each step of the Frank-Wolfe method essentially converges to the $(0,1)$ vertex while also remaining in Δ_2. Meanwhile, if one were to implement the standard gradient descent algorithm on this problem, the point would keep moving in the $\nabla f = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ direction without bound.

Theorem 4 (Convergence of the Frank-Wolfe Method)**.** *Assume $f(\cdot)$ is a L-smooth convex function. Denote $D := \max_{\mathbf{x},\mathbf{y}\in C} \|\mathbf{x}-\mathbf{y}\|_2$ as the diameter of the set C. Let $\eta_k = \min\{1, \frac{2}{k}\} \in [0,1]$. Then, Frank-Wolfe has:*

$$f(\mathbf{x}_K) - f(\mathbf{x}_*) \leq \frac{2LD^2}{K}.$$

Proof. First, recall that by L-smoothness we have:

$$\begin{aligned} f(\mathbf{x}_{K+1}) &\leq f(\mathbf{x}_K) + \langle \nabla f(\mathbf{x}_K), \mathbf{x}_{K+1} - \mathbf{x}_K \rangle + \frac{L}{2}\|\mathbf{x}_{K+1} - \mathbf{x}_K\|^2 \\ &= f(\mathbf{x}_K) + \eta_K \langle \nabla f(\mathbf{x}_K), \mathbf{v}_K - \mathbf{x}_K \rangle + \frac{L\eta_K^2}{2}\|\mathbf{v}_K - \mathbf{x}_K\|^2 \qquad (15) \\ &\leq f(\mathbf{x}_K) + \eta_K \langle \nabla f(\mathbf{x}_K), \mathbf{v}_K - \mathbf{x}_K \rangle + \frac{L\eta_K^2}{2} D^2, \qquad (16) \end{aligned}$$

where (15) and (16) follow from the fact that the update is $\mathbf{x}_{K+1}-\mathbf{x}_K = \eta_K(\mathbf{v}_K-\mathbf{x}_K)$ and $D \geq \max_{\mathbf{x},\mathbf{v}\in C}\|\mathbf{x}-\mathbf{v}\|^2$. Then pick $\mathbf{x}_* \in \arg\min_{\mathbf{x}\in C} f(\mathbf{x})$. By recalling $\mathbf{v}_k = \arg\min_{\mathbf{v}\in C}\langle \mathbf{v}, \nabla f(\mathbf{x}_k)\rangle$, we know

$$\langle \nabla f(\mathbf{x}_K), \mathbf{v}_K\rangle \leq \langle \nabla f(\mathbf{x}_K), \mathbf{z}\rangle \quad \forall \mathbf{z} \in C.$$

By setting $\mathbf{z} = \mathbf{x}_*$, this is implies

$$\langle \nabla f(\mathbf{x}_K), \mathbf{v}_K - \mathbf{x}_K\rangle \leq \langle \nabla f(\mathbf{x}_K), \mathbf{x}_* - \mathbf{x}_K\rangle. \tag{17}$$

Furthermore, by the convexity of f we know

$$f(\mathbf{x}_*) \geq f(\mathbf{x}_K) + \langle \nabla f(\mathbf{x}_K), \mathbf{x}_* - \mathbf{x}_K\rangle. \tag{18}$$

We can use (18) in (17), then plug this estimate into (16) and rearrange to obtain

$$f(\mathbf{x}_{K+1}) - f(\mathbf{x}_*) \leq (1-\eta_K)(f(\mathbf{x}_K) - f(\mathbf{x}_*)) + \frac{LD^2\eta_K^2}{2}. \tag{19}$$

Before we proceed, we state the following lemma which can be proven via induction:

Lemma 2. *Let $\{\delta_k\}$ be a sequence that satisfies the recurrence*

$$\delta_{k+1} \leq \delta_k(1-\eta_k) + \eta_k^2 c_0.$$

Then taking $\eta = \min\{1, \frac{2}{k}\}$, we get

$$\delta_k \leq \frac{4c_0}{k}.$$

For the proof of this lemma, see Lemma 7.2 in Chapter 7 of [Hazan (2016)].
Then, by setting $\delta_K = f(\mathbf{x}_K) - f(\mathbf{x}_*)$ and $c_0 = \frac{LD^2}{2}$, we can apply Lemma 2 to (19) and obtain

$$f(\mathbf{x}_K) - f(\mathbf{x}_*) \leq \frac{2LD^2}{K},$$

which concludes the proof. □

3.1 Application of Frank-Wolfe: Matrix Completion

First, we introduce the nuclear norm of a matrix that is useful to explain the matrix completion example of the Frank-Wolfe method.

Nuclear Norm: The nuclear norm of a matrix $A \in \mathbb{R}^{m\times n}$ denoted as $\|A\|_\sigma$ is defined as the sum of all singular values of the matrix, i.e.

$$\|A\|_\sigma = \sum_{i=1}^{l} \sigma_i(A),$$

where $l = \min(m, n)$. By the singular value decomposition, if $A = U\Sigma V^T$, then

$$\Sigma = \begin{bmatrix} \sigma_1(A) & & \\ & \sigma_2(A) & \\ & & \ddots \end{bmatrix}.$$

Matrix completion

The matrix completion problem is illustrated through a realistic example. Let's imagine a scenario with a fixed number of people and different fruits. Each person has a different rating or preference for a fruit. Figure 2 shows a matrix that represents the preference of 5 people for 7 different fruits. Let M denote the matrix in Figure 2. Imagine that some entries of the preference matrix M are collected as shown in black boxes in Figure 2. Let's denote the partially collected or given matrix as $P_O(M)$. The preference of i-th person for j-th fruit $P_O(M)_{i,j}$ is given as

$$P_O(M)_{i,j} = \begin{cases} M_{i,j} & \text{if } (i,j) \text{ is observed} \\ 0 & \text{otherwise.} \end{cases}$$

Mathematically, we are given $P_O(M)$. The matrix completion problem is to complete unknown entries of $P_O(M)$. The matrix completion problem is formulated as

$$\min_{X \in R^{m\times n}: \|X\|_\sigma \le r} f(X), \quad \text{where } f(X) := \frac{1}{2}\|X - P_O(M)\|_2^2. \tag{20}$$

The constrained optimization problem is to solve a linear equation over the set of observed entries with the aim of keeping the nuclear norm of the completed matrix X less than r. This constraint makes sure that X does not overfit the observed values. The matrix completion problem is to find the minimizer of Euclidian distance from $P_O(M)$ with the nuclear norm less than r.

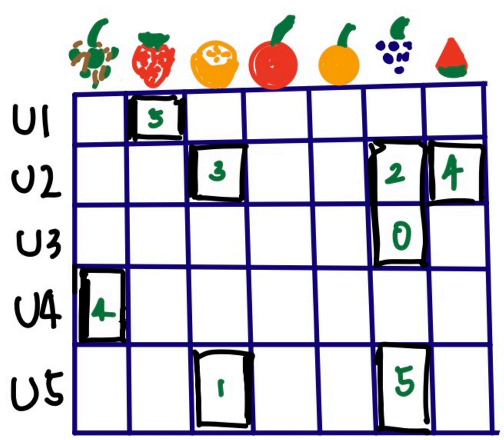


Figure 2: Fruit preference matrix of 5 users for 7 different fruits.

The update of Frank-Wolfe

Taking the gradient of the objective function $f(X)$ in (20) yields to

$$\nabla f(X) = X - P_O(M) \in \mathbb{R}^{m\times n}.$$

Then, the linear optimization step becomes

$$\mathbf{v}_k = \underset{\|\mathbf{v}\|_\sigma \le r}{\text{argmin}}\, \langle \nabla f(X_k), \mathbf{v}\rangle. \tag{21}$$

Let's denote $-\nabla f(X) = U\Sigma W^\top$ the singular value decomposition, where $U \in \mathbb{R}^{m\times l}$, $\Sigma \in \mathbb{R}^{l\times l}$, and $W \in \mathbb{R}^{n\times l}$ and $l = \min(m,n)$. The solution to (21) becomes

$$\mathbf{v}_k = r\mathbf{u}_1\mathbf{w}_1^\top, \tag{22}$$

where $\mathbf{u}_1 \in \mathbb{R}^m$ and $\mathbf{w}_1 \in \mathbb{R}^n$ is the top left and right singular vector. The complexity to calculate $\mathbf{v}_k$ is in the order of $\tilde{\mathcal{O}}(m \times n)$ since only the top left, right singular vectors and the top singular value are calculated.

We introduce the definition of a nuclear-norm ball expression to sketch out the steps to provide reasoning in the result (22). A nuclear-norm ball is defined as

$$\{Y \in \mathbb{R}^{m\times n} : \|Y\|_\sigma \le r\} = r \cdot \mathbf{conv}\{\mathbf{u}\mathbf{w}^\top : \mathbf{u} \in \mathbb{R}^m, \mathbf{w} \in \mathbb{R}^n, \|\mathbf{u}\|_2 = \|\mathbf{w}\|_2 = 1\}. \tag{23}$$

The linear oracle outputs

$$\begin{aligned}
\underset{V\in\mathbb{R}^{m\times n}:\|V\|_\sigma \le r}{\arg\min} \langle V, Y\rangle &= r \cdot \underset{\mathbf{u}\in\mathbb{R}^m,\mathbf{w}\in\mathbb{R}^n:\|\mathbf{u}\|_2=\|\mathbf{w}\|_2=1}{\arg\max} \langle \mathbf{u}\mathbf{w}^\top, -Y\rangle \\
&= r \cdot \underset{\mathbf{u}\in\mathbb{R}^m,\mathbf{w}\in\mathbb{R}^n:\|\mathbf{u}\|_2=\|\mathbf{w}\|_2=1}{\arg\max} \text{tr}\left(\left(\mathbf{u}\mathbf{w}^\top\right)^\top (-Y)\right) \\
&= r \cdot \underset{\mathbf{u}\in\mathbb{R}^m,\mathbf{w}\in\mathbb{R}^n:\|\mathbf{u}\|_2=\|\mathbf{w}\|_2=1}{\arg\max} \mathbf{u}^\top(-Y)\mathbf{w} \\
&= r \cdot \mathbf{u}_1\mathbf{w}_1^\top.
\end{aligned}$$

The update of PGD

Let's denote $(\mathbf{z})_+ \stackrel{\Delta}{=} \max\{0, \mathbf{z}\}$ and the singular-value decomposition of $Y = \sum_{i=1}^{\min(m,n)} \sigma_i\, \mathbf{u}_i\mathbf{w}_i \in \mathbb{R}^{m\times n}$. Then, the projection of Y onto a nuclear norm-ball with r is defined as

$$\text{Proj}_{\|\cdot\|_\sigma \le r}[Y] = \sum_{i=1}^{\min(m,n)} (\sigma_i - \lambda)_+ \mathbf{u}_i\mathbf{w}_i,$$

where λ is the solution to $\sum_{i=1}^{\min(m,n)}(\sigma_i - \lambda)_+ = r$. Since all the singular values $\min(m,n)$ are calculated, the complexity of the projection step in the PGD is in the order of $\tilde{\mathcal{O}}(m \times n \times \min(m, n))$.

Remark: The complexity of each update in the Frank-Wolfe is $\tilde{\mathcal{O}}(m \times n)$ which is much less than the complexity of each update in the PGD, $\tilde{\mathcal{O}}(m \times n \times \min(m, n))$

Comparison to the projection on a l_1 norm ball
Example: (without closed-form solution)
Let $C := \{\mathbf{x} \in \mathbb{R}^d : ||\mathbf{x}||_1 \leq 1\}$.
Denote $(\mathbf{z})_+ \triangleq \max\{0, \mathbf{z}\}$.
Then, we have for the *Characterization of* $\text{Proj}_C(\mathbf{y})$ *when* $\mathbf{y} \notin C$

$$\hat{\mathbf{x}}[i] = \text{sign}(\mathbf{y}[i]) \, (|\mathbf{y}[i]| - \lambda)_+,$$

where λ is the solution to $\sum_{i=1}^d (|\mathbf{y}[i]| - \lambda)_+ = 1$.

(Frank-Wolfe) Faster rate than $O(1/K)$ when $f(\cdot)$ is smooth and strongly convex?

- Negative example [Lan (2014)]:

  If C is a probability simplex, i.e., $C := \{x \in \mathbb{R}^d : \sum_{i=1}^d x[i] = 1, x[i] \geq 0\}$.

  $$K = \Omega\left(\max\left(\frac{L}{\epsilon}, \frac{d}{2}\right)\right).$$

- Positive example [Wang (2023)]:

  When C is a μ-strongly convex **set** w.r.t. a norm $\|\cdot\|$, i.e., $x, z \in C$ implies that a ball centered at $\alpha x + (1-\alpha)z$ with a radius in $\alpha(1-\alpha)\frac{\mu}{2}\|x - z\|^2$ is in C, where $\alpha \in [0, 1]$.

  Example: l_p norm with $p \in (1, 2]$.

Bibliographic notes

For more examples and discussions, see [Combettes (2021)] and Chapter 7 of [Hazan (2016)].

References

[Wang (2023)] Jun-Kun Wang, Jacob Abernethy, Kfir Y Levy. No-regret dynamics in the Fenchel game: A unified framework for algorithmic convex optimization. Mathematical Programming, 2023

[Hazan (2016)] Elad Hazan. Introduction to Online Convex Optimization. 2016.

[Combettes (2021)] Cyrille W. Combettes, Sebastian Pokutta. Complexity of Linear Minimization and Projection on Some Sets. 2021.

[Lan (2014)] Guanghui Lan. The Complexity of Large-scale Convex Programming under a Linear Optimization Oracle. 2014.