1 Outline

In this lecture, we study

- Frank-Wolfe algorithm,
- Applications in LASSO and matrix completion,
- Introduction to online convex optimization.

2 Projection-free method

We saw that projected gradient descent minimizes a smooth function with a convergence rate of $O(1/T)$. There are some issues.

1. The projection step onto the feasible set C can be expensive.
2. We have used the ℓ_2 norm to define smoothness.

Each projection step essentially amounts to solving an optimization problem, which can be difficult depending on the structure of C. Even for the case when C is a polyhedron, the projection onto C can an expensive procedure. The second point is that in our analysis of gradient descent for smooth functions, there are parts that do need smoothness with respect to the ℓ_2 norm. It is often the case that smoothness in the ℓ_2 norm is implied by smoothness in another norm, e.g., the ℓ_1 norm.

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq \sqrt{d}\|\nabla f(x) - \nabla f(y)\|_\infty \leq \sqrt{d}\beta\|x - y\|_1 \leq d\beta\|x - y\|_2.$$

The implication of this inequality is the following. Even if a function is smooth in the ℓ_1 norm with a tiny smoothness parameter β, the smoothness parameter with respect to the ℓ_2 norm can blow up by a factor of dimension d, in which case we lose the desired dimension-free property.

2.1 Constrained optimization formulation of LASSO

We studied the folllowing formulation of LASSO given n data points n data points $(x_1, y_1), \ldots, (x_n, y_n)$.

$$\min_{\beta} \quad \frac{1}{n}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

where

- $y = (y_1, \ldots, y_n)^\top$ and the rows of X are $x_1^\top, \ldots, x_n^\top$,
- the quadratic term

  $$\frac{1}{n}\sum_{i=1}^{n}(y_i - \beta^\top x_i)^2 \quad = \quad \frac{1}{n}\|y - X\beta\|_2^2$$

  is the mean squared error of regressor β.

Recall that the formulation is the *Lagrangian form* of LASSO. The constrained form is given by

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n}\|y - X\beta\|_2^2 \\ \text{subject to} \quad & \|\beta\|_1 \leq t \end{aligned}$$

where t is a parameter determining the degree of regularization. Hence, the constrained formulation is a constrained convex optimization problem

$$\min_{C} \quad \frac{1}{n}\|y - X\beta\|_2^2$$

where

$$C = \left\{ \beta \in \mathbb{R}^d : \ \|\beta\|_1 \leq t \right\}.$$

Here, C is a polytope. We may run projected gradient descent to this problem, but it requires projection onto polytope C which amounts to solving a quadratic program. On the other hand, we know that the problem of optimizing a linear function over polytope C is a linear program, for which we have fast solution methods.

2.2 Conditional gradient method: Frank-Wolfe algorithm

Motivated by the issues of projected gradient descent, we consider 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. A pseudo-code of the method is given as follows.

Algorithm 1 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. This is in contrast to projected gradient descent which has a quadratic objective for each projection step. For this reason, the conditional gradient method is called "projection-free".

Another difference compared to projected gradient descent is that the direction we take for an update can be different from $-\nabla f$. We provide Figure 13.1 for a pictorial description of the update rule. 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 x_{t+1}.

Definition 13.1. We say that a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is β-*smooth* with respect to a norm $\|\cdot\|$ for some $\beta > 0$ if

$$\|\nabla f(x) - \nabla f(y)\|_* \leq \beta \|x - y\|$$

holds for any $x, y \in \mathbb{R}^d$ where $\|\cdot\|_*$ denotes the dual norm of $\|\cdot\|$.

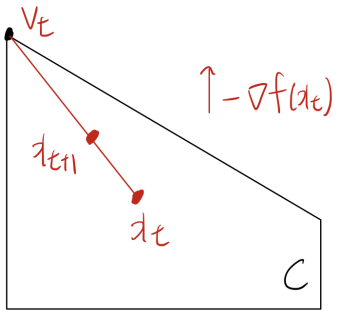

Figure 13.1: Illustration of an update from conditional gradient descent

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 13.2. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex function that is β-smooth with respect to 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*

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

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. □

2.3 Low-rank matrix completion

Let $A \in \mathbb{R}^{n\times p}$ be a partially observable matrix, of which the missing entries are filled with 0. We assume that even the non-zero entries of A are some noisy observations of the true values.

Such a matrix A arises in movie rating systems, in which case the rows of A correspond to the users and the columns correspond to the list of movies. Hence, n is the number of users and p is the number of movies. Here, one reasonable assumption is that the movie ratings of users depend on a small set of features and criteria. One way to model this is to impose that the true rating matrix A^* satisfies

$$A \cong A^* = UV^\top$$

where U is an $n \times k$ matrix and V is a $p \times k$ matrix with $k < n, p$. By $A^* = UV^\top$, the true rating matrix A^* has rank at most k. Then, to infer the true matrix A^*, we may attempt to solve the following problem.

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|A - X\|_F^2 \\ \text{subject to} \quad & \operatorname{rank}(X) \leq k \end{aligned}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, which essentially extends the ℓ_2 norm over vectors to matrices. Here, the low-rank constraint $\operatorname{rank}(X) \leq k$ is non-convex, so we may consider the following problem instead.

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|A - X\|_F^2 \\ \text{subject to} \quad & \|X\|_{\text{nuc}} \leq k \end{aligned} \tag{13.1}$$

where $\|\cdot\|_{\text{nuc}}$ denotes the nuclear norm. Here, (13.1) is a constrained convex optimization problem where the constraint set is given by

$$C = \left\{X \in \mathbb{R}^{n\times p} : \ \|X\|_{\text{nuc}} \leq k\right\}.$$

To solve (13.1), the first approach is to apply projected gradient descent over the constraint set C. It is known that projection onto the constraint set C amounts to computing the singular value decomposition of A [DSSSC08].

The second approach is to use the Frank-Wolfe algorithm. GIven a current iterate matrix X_t, the Frank-Wolfe update step considers

$$V_t \in \operatorname{argmin}\left\{\operatorname{Tr}((A - X_t)^\top V) : \ \|V\|_{\text{nuc}} \leq k\right\}.$$

The associated minimization problem is equivalent to computing the top left and right singular vectors of $X_t - A$, for which we may apply the classical power method [JS10]. Here, the power method does not compute the full singular value decomposition, so it runs faster than the projection operation.

3 Introduction to online convex optimization

We have so far discussed formulations and algorithms for convex optimization. In this section, we consider a different yet closely related setting. Online convex optimization (OCO) is an online learning problem, that is to make a squence of predictions based on the history of past decisions and their results. The framework of OCO is closely related to game theory, statistical learning theory, and stochastic modelling as well as convex optimization. The contents of this section are based on the text of Hazan [Haz16].

Let us provide some applications of the OCO framework.

- (Online spam filtering) We receive emails repeatedly, for each of which we apply an existing spam-filtering system. A spam-filtering system has a list of words and expressions, based on which, it can predict whether an email is spam or valid. When an email that is classified as valid by the existing filter tunrs out to be spam, we have to update the filter so that we can filter similar spam emails later.
- (Online advertisement selection) A web browser selects a collection of online advertisements for its ad slots. The web browser posts a catalog of online ads and observes their popularity from users by the click-through rates. Later, the browser can change its ad selection based on its prediction about the user demands.

3.1 Online binary classification

Let us consider a mathematical model to establish an email spam filtering system. Recall that we used the support vector machine (SVM) for binary classification. Just to remind you what it was, we find a pair of a coefficient vector w and a right-hand side value b to use the hyperplane $w^\top x = b$ to classify data points. Given a feature vector x, we assign it label $\mathrm{sign}(w^\top x - b)$ where $\mathrm{sign}(c)$ has value 1 if $c \geq 0$ and value -1 if $c < 0$. When a training set of multiple data is available, we can find such a classifier (w, b) by solving a convex optimization problem whose objective is to minimize the hinge loss.

However, in some scenarios, data points dynamically arrive so that we gradually accumulate the data. In such cases, we may adjust our model over time, and the learning process continues. To be more specific, let us consider the online binary classification problem described as follows. An email is represented by its feature vector $x \in \mathbb{R}^d$ and label $y \in \{-1, 1\}$. The feature vector can encode words and expressions written in it, while the label indicates whether the email is spam or not. Let's say that $y = 1$ indicates spam and $y = -1$ indicates valid. For each time slot t, we repeat the following procedure.

- The spam filtering system prepares a classifier (w_t, b_t) based on the past emails represented by $(x_1, y_1), \ldots, (x_{t-1}, y_{t-1}) \in \mathbb{R}^d \times \{-1, 1\}$.
- New email with feature vector x_t arrives.
- The spam filter predicts that its label is $\mathrm{sign}(w_t^\top x_t - b)$, while the true label of the email is y_t.
- The spam filter incurs a loss of $\max\{0, 1 - y_t(w_t^\top x_t - b)\}$.

After T emails, the cumulative loss is given by

$$\sum_{t=1}^{T} \max\{0, 1 - y_t(w_t^\top x_t - b)\}.$$

Compared to a best classifier, we incur

$$\sum_{t=1}^{T} \max\{0, 1 - y_t(w_t^\top x_t - b)\} - \min_{(w,b)\in\mathbb{R}^d\times\mathbb{R}} \sum_{t=1}^{T} \max\{0, 1 - y_t(w^\top x_t - b)\}$$

more loss. Denoting the loss function at each time t as

$$f_t(w, b) = \max\{0, 1 - y_t(w^\top x_t - b)\},$$

the excess cumulative loss is rewritten as

$$\sum_{t=1}^{T} f_t(w_t, b_t) - \min_{(w,b)\in\mathbb{R}^d\times\mathbb{R}} \sum_{t=1}^{T} f_t(w, b).$$

Therefore, the online binary classification problem is an instance of online convex optimization where the best fixed decision corresponds to the best spam classifier.

3.2 Adversarial multi-armed bandits

Suppose that we have d slot machines (or bandits). Then, at each t, the player chooses which slot machine to play. Here, let $i_t \in \{1, \dots, d\}$ be the machine that the player choses at time t. Then the reward of playing machine $i \in \{1, \dots, d\}$ at time t is given by $r_{i,t}$, which is revealed only after a play. Then we may compare the player's cumulative reward against the total reward of the best slot machine as follows.

$$\max_{i\in\{1,\dots,d\}} \sum_{t=1}^{T} r_{t,i} - \sum_{t=1}^{T} r_{t,i_t}.$$

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. 2.3

[FW56] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956. 2.2

[Haz16] Elad Hazan. Introduction to online convex optimization. *Found. Trends Optim.*, 2(3–4):157–325, aug 2016. 3

[JS10] Martin Jaggi and Marek Sulovský. A simple algorithm for nuclear norm regularized problems. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 471–478, Madison, WI, USA, 2010. Omnipress. 2.3