1 Outline

In this lecture, we cover

- gradient descent for strongly convex functions,
- regularization for linear regression,
- lasso: least absolute shrinkage and selection operator.

2 Gradient Descent for Strongly Convex Functions

We say that a function is strongly convex in the ℓ_2-norm if there exists some $\alpha > 0$ such that

$$f(x) - \frac{\alpha}{2}\|x\|_2^2$$

is convex. More precisely, we say that f is α-strongly convex in the norm $\|\cdot\|_2$. If f is α-strongly convex, then we have

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\alpha}{2}\|y - x\|_2^2.$$

This inequality implies that a strongly convex function is lower bounded by a quadratic function as in the following figure. That means that, when a point is far from an optimal solution, the gradient

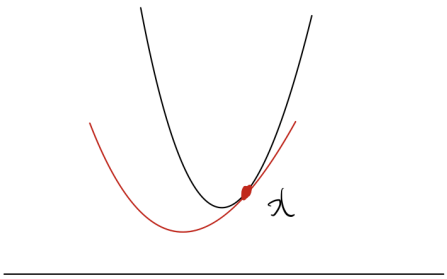

Figure 5.1: Quadratic lower bound on a strongly convex function

at this point has to be large. Hence, when applying gradient descent or the subgradient method, this leads to a faster convergence.

Theorem 5.1. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be L-Lipschitz continuous and α-strongly convex in the ℓ_2-norm, and let $\{x_t : t = 1, \ldots, T\}$ be the sequence of iterates generated by gradient descent with step size*

$$\eta_t = \frac{2}{\alpha(t+1)}$$

for each t. Then

$$f\left(\sum_{t=1}^{T} \frac{2t}{T(T+1)} x_t\right) - f(x^*) \leq \frac{2L^2}{\alpha(T+1)}$$

where x^* is an optimal solution to $\min_{x \in \mathbb{R}^d} f(x)$.

If function f is β-smooth and α-strongly convex in the ℓ_2-norm, then it follows that

$$\frac{\alpha}{2}\|y-x\|_2^2 \le (f(y)-f(x)) - \nabla f(x)^\top (y-x) \le \frac{\beta}{2}\|y-x\|_2^2.$$

Here, we call $\kappa = \beta/\alpha$ the **condition number** of f. In fact, when f is both smooth and strongly convex, it leads to a drastic improvement in the convergence rate. The convergence rate depends on the condition number κ.

Theorem 5.2. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be β-smooth and α-strongly convex in the ℓ_2-norm, and let $\{x_t : t = 1, \dots, T+1\}$ be the sequence of iterates generated by gradient descent with step size*

$$\eta_t = \frac{2}{\alpha+\beta}$$

for each t. Then

$$f(x_{T+1}) - f(x^*) \le \frac{\beta}{2}\exp\left(-\frac{4T}{\kappa+1}\right)\|x_1 - x^*\|_2^2$$

where x^* is an optimal solution to $\min_{x\in\mathbb{R}^d} f(x)$.

Note that $\exp(-4/(\kappa+1)) < 1$, and therefore, the convergence rate is $O(c^T)$ where $c = \exp(-4/(\kappa+1)) < 1$. Hence, we achieve a linear rate of convergence, and after $T = O(\log(1/\epsilon))$ iterations, we have

$$f(x_{T+1}) - f(x^*) \le \epsilon.$$

3 Linear Regression Revisited

In this section, we consider linear regression again in the context of smoothness and strong convexity. We assume that the relationship between the vector $x \in \mathbb{R}^d$ of features and the response variable $y \in \mathbb{R}$ is modeled using a linear equation given by

$$y = \theta_{\text{true}}^\top x + \epsilon$$

where:

- $\theta_{\text{true}} \in \mathbb{R}^d$ is the coefficient vector,
- $\epsilon \in \mathbb{R}$ is the noise term representing unexplained variation.

Note that the equation has no bias term for simplicity. As before, we infer the true coefficient vector θ_{true} using the method of least squares, which minimizes the average of squared differences between the observed and predicted values of y. Namely, given a set of n data $(x_1, y_1), \dots, (x_n, y_n)$, we solve

$$\min_\theta \quad \frac{1}{n}\sum_{i=1}^n \left(y_i - \theta^\top x_i\right)^2 \quad = \quad \min_\theta \quad \frac{1}{n}\|Y - X\theta\|_2^2. \tag{5.1}$$

Here, Y denotes the vector whose components are $y_1, \dots, y_n$, and X denotes the matrix whose rows are $x_1^\top, \dots, x_n^\top$. Note that

$$f(\theta) := \frac{1}{n}\|Y - X\theta\|_2^2 = \frac{1}{n}\theta^\top X^\top X\theta - \frac{2}{n}Y^\top X\theta + \frac{1}{n}Y^\top Y.$$

Since $X^\top X$ is positive semidefinite, it follows that the MSE loss $f(\theta)$ is convex. Moreover, $f(\theta)$ is α-strongly convex and β-smooth in the ℓ_2-norm with

$$\alpha = \frac{1}{n}\lambda_{\min}(X^\top X) \quad \text{and} \quad \beta = \frac{1}{n}\lambda_{\max}(X^\top X)$$

where $\lambda_{\min}(X^\top X)$ and $\lambda_{\max}(X^\top X)$ denote the minimum and maximum eigenvalues of $X^\top X$. As long as X is a nonzero matrix, we have $\lambda_{\max}(X^\top X) > 0$. However, we can have $\lambda_{\min}(X^\top X) = 0$ when the rank of $X^\top X$ is lower than the number of features d.

Data-Rich Regime Recall that n is the number of data and d is the number of features. When $n \geq d$, then it is possible that X is of full column rank, in which case $X^\top X$ is invertible. If $X^\top X$ is invertible, it is positive definite, and therefore, we have $\alpha = \lambda_{\min}(X^\top X)/n > 0$. In this case, the MSE loss $f(\theta)$ is indeed strongly convex. Another Remark is that if $X^\top X$ is invertible, then

$$\theta_{\text{opt}}^{\text{rich}} := \text{argmin}_\theta \frac{1}{n}\|Y - X\theta\|_2^2 = (X^\top X)^{-1}X^\top y$$

because

$$\nabla f(\theta) = \frac{2}{n}X^\top(X\theta - y).$$

Data-Poor Regime When $n < d$, then the rank of X is less than d, which means that $X^\top X$ is not of full rank and thus $X^\top X$ is not invertible. In this case, we have $\alpha = \lambda_{\min}(X^\top X)/n = 0$, and therefore, the MSE loss $f(\theta)$ is not strongly convex. When $X^\top X$ is not invertible, we have

$$\theta_{\text{opt}}^{\text{poor}} := \text{argmin}_\theta \frac{1}{n}\|Y - X\theta\|_2^2 = (X^\top X)^\dagger X^\top y$$

where $(X^\top X)^\dagger$ denotes the Moore-Penrose pseudo-inverse of $X^\top X$.

3.1 Gradient Descent for Minimizing the MSE Loss

We generated a random instance with 75 feature variables and 100 data samples. To consider a data-poor regime, we randomly selected 30 samples from the data set. Recall that θ_{true} denotes the true coefficient vector in the linear model $y = \theta_{\text{true}}^\top x + \epsilon$.

The following figures map loss convergence patterns under the data-rich and data-poor regimes. The figures show that gradient descent quickly minimizes the MSE loss under both regimes.

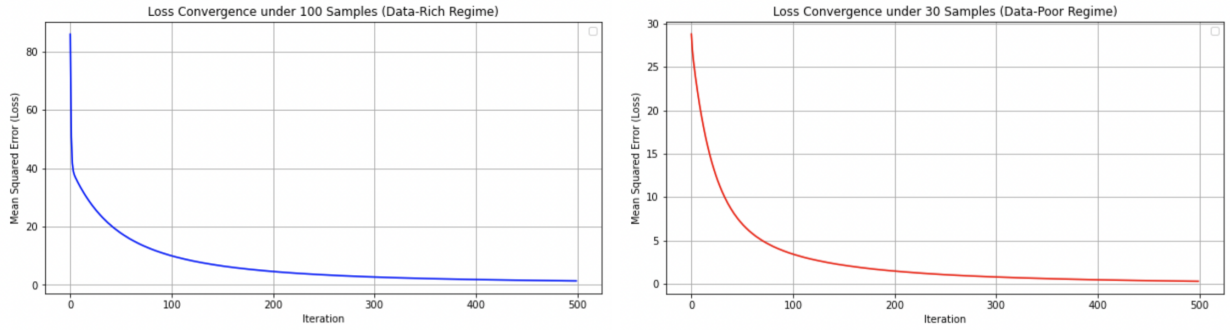

Figure 5.2: Loss convergence patterns under the data-rich regime (Left) and the data-poor regime (Right)

Let us verify whether gradient descent returns solutions that converge to the optimal model minimizing the MSE loss. Recall that $\theta_{\text{opt}}^{\text{rich}} = (X^\top X)^{-1} X^\top y$ is the model minimizing the MSE loss under the data-rich regime while $\theta_{\text{opt}}^{\text{poor}} = (X^\top X)^\dagger X^\top y$ is the model minimizing the MSE loss under the data-poor regime. Figure 5.3 reports the distances between models θ generated by gradient descent and the optimal model under each regime. Here, the purple line shows the squared norm of

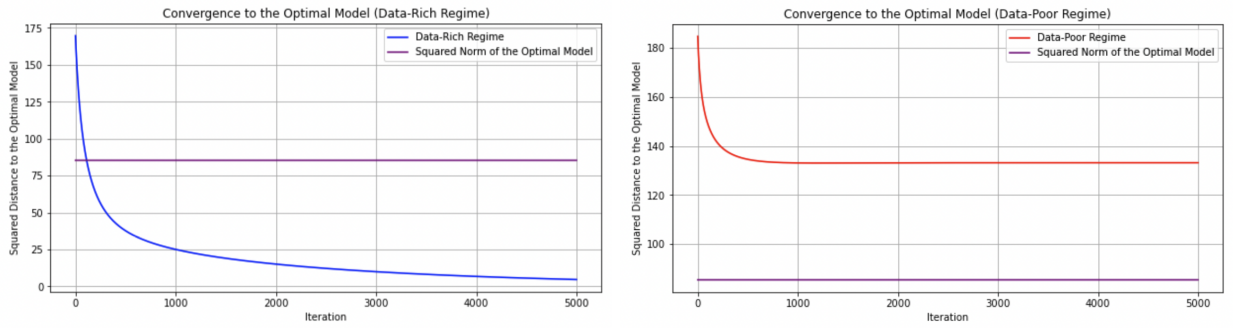

Figure 5.3: Convergence to the optimal model under the data-rich regime (Left) and the data-poor regime (Right)

$\theta_{\text{opt}}^{\text{rich}}$ and that of $\theta_{\text{opt}}^{\text{poor}}$, given by $\|\theta_{\text{opt}}^{\text{rich}}\|_2^2$ and $\|\theta_{\text{opt}}^{\text{poor}}\|_2^2$, respectively. The red line depicts $\|\theta - \theta_{\text{opt}}^{\text{poor}}\|_2^2$ under the data-poor regime, while the blue one shows $\|\theta - \theta_{\text{opt}}^{\text{rich}}\|_2^2$ under the data-rich regime. Figure 5.3 shows that the solution deduced by gradient descent under the data-rich regime indeed seems to converge to the optimal vector minimizing the MSE loss, but that under the data-poor regime does not. This is because the MSE loss is no longer strongly convex under the data-poor regime.

In Figure 5.4, we report the distances between each model θ generated by gradient descent and the true coefficient vector θ_{true}. Here, the green line shows the squared norm of θ_{true}, given by $\|\theta_{\text{true}}\|_2^2$.

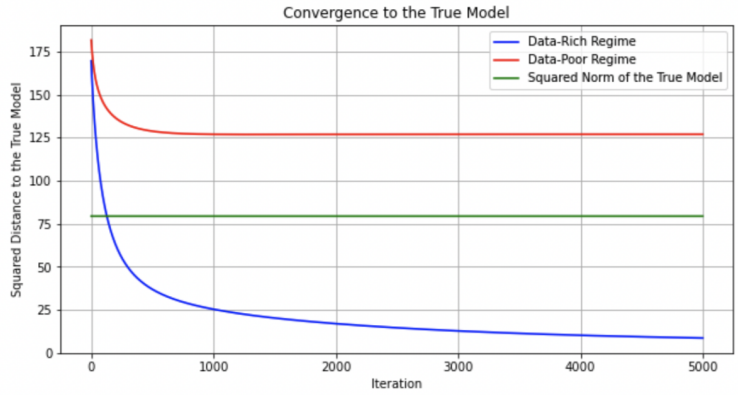

Figure 5.4: Convergence to the true model

The red line depicts $\|\theta - \theta_{\text{true}}\|_2^2$ under the data-poor regime, while the blue one shows $\|\theta - \theta_{\text{true}}\|_2^2$ under the data-rich regime. Figure 5.4 shows that the solution deduced by gradient descent under the data-rich regime indeed seems to converge to the actual true coefficient vector, but that under the data-poor regime does not.

3.2 ℓ_2-Regularized Least Squares

We discussed that the MSE loss under the data-poor regime is not strongly convex. In practice, it is often desirable to add an ℓ_2-regularization term, which makes the resulting loss function strongly convex. To be more precise, we consider

$$\min_{\theta} \quad \frac{1}{n}\|Y - X\theta\|_2^2 + \lambda\|\theta\|_2^2 \tag{5.2}$$

for some positive λ. Note that the regularized loss is α-strongly convex and β-smooth in the ℓ_2-norm with

$$\alpha = \frac{1}{n}\lambda_{\min}(X^\top X) + \lambda \quad \text{and} \quad \beta = \frac{1}{n}\lambda_{\max}(X^\top X) + \lambda.$$

Hence, as long as $\lambda > 0$, the regularized loss is strongly convex. As $X^\top X + \alpha I$ is positive definite, the model minimizing the regularized loss is given by

$$\theta_{\text{opt}} := \text{argmin}_\theta \frac{1}{n}\|Y - X\theta\|_2^2 + \lambda\|\theta\|_2^2 = (X^\top X + \lambda I)^{-1}X^\top y.$$

In Figure 5.5, we report the distances between each model θ generated based on the regularized loss and the true coefficient vector θ_{true}. Here, the green line shows the squared norm of θ_{true}, given

Figure 5.5: Convergence to the true model under regularization

by $\|\theta_{\text{true}}\|_2^2$. The orange line depicts $\|\theta - \theta_{\text{true}}\|_2^2$ for the regularized loss, while the red one shows $\|\theta - \theta_{\text{true}}\|_2^2$ for the original MSE loss. We see that

Let us also check convergence to the optimal model minimizing the regularized loss. Recall that $\theta_{\text{opt}} = (X^\top X + \lambda I)^{-1}X^\top y$ is the model minimizing the regularized loss. Here, the purple line shows the squared norm of θ_{opt} given by $\|\theta_{\text{opt}}\|_2^2$. The orange line depicts $\|\theta - \theta_{\text{opt}}\|_2^2$ for the regularized loss, while the red one shows $\|\theta - \theta_{\text{opt}}\|_2^2$ for the original MSE loss.

4 LASSO: Least Absolute Shrinkage and Selection Operator

Recall the formulation of LASSO, given by

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

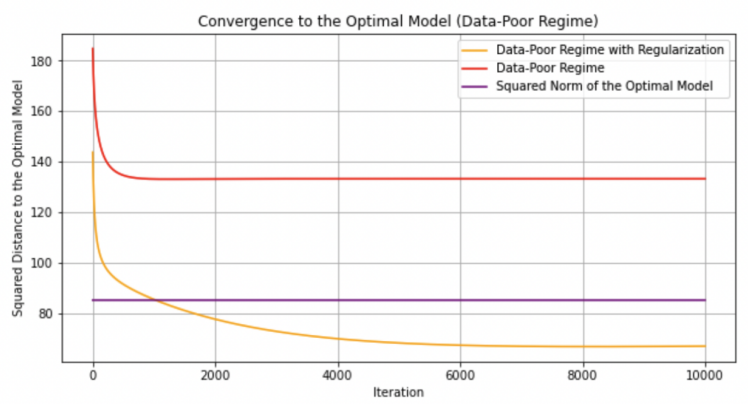

Figure 5.6: Convergence to the optimal model under regularization

Here, the objective function is non-differentiable because of the ℓ_1-regularization term $\lambda\|\theta\|_1$, and therefore, it is non-smooth. On the other hand, the objective is convex, and we have a characterization of the subdifferential of $\|\theta\|_1$, so we can simply apply the subgradient method. To bound the additive error by ϵ, the subgradient method requires $O(1/\epsilon^2)$ iterations.

If you take a closer look at the objective, it consists of two part. One part is smooth, and the other part is something whose subdifferential is well understood. Can we use this structure to obtain a better algorithm? The main subject of this section is developing an algorithm that converges to an ϵ-approximate solution after $O(1/\epsilon)$ iterations.

4.1 Projection and Proximal Operator

We studied the projected gradient descent method, where at each step, we take a projection to the constraint set. When the constraint set is given by $\mathcal{X}$, the projection operator is given by

$$\text{Proj}_{\mathcal{X}}(x) = \underset{u\in\mathcal{X}}{\text{argmin}} \frac{1}{2}\|u-x\|_2^2 = \underset{u\in\mathbb{R}^d}{\text{argmin}} \left\{ I_{\mathcal{X}}(u) + \frac{1}{2}\|u-x\|_2^2 \right\}$$

where $I_{\mathcal{X}}(u)$ is the indicator function of $\mathcal{X}$. This definition is proper as there is a unique minimizer for the optimization problem. Hence, the projection operator is defined by the indicator function and the proximity term $(1/2)\|u-x\|_2^2$. The proximal operator is a generalization of the projection operator replacing the indicator function by other general functions.

The proximal operator with respect to a convex function h is defined as follows.

$$\text{Prox}_h(x) = \underset{u\in\mathbb{R}^d}{\text{argmin}} \left\{ h(u) + \frac{1}{2}\|u-x\|_2^2 \right\}.$$

Again the definition is proper because the objective of the optimization problem is strongly convex. Hence, for any $\eta > 0$,

$$\text{Prox}_{\eta h}(x) = \underset{u\in\mathbb{R}^d}{\text{argmin}} \left\{ h(u) + \frac{1}{2\eta}\|u-x\|_2^2 \right\}.$$

As projected gradient descent proceeds with the update rule

$$x_{t+1} = \text{Proj}_{\mathcal{X}}\left\{x_t - \eta\nabla f(x_t)\right\},$$

we can defined the proximal gradient method with the update rule

$$x_{t+1} = \text{Prox}_{\eta h}(x_t - \eta \nabla f(x_t)).$$

In particular, when we take the indicator function I_C for h, the proximal gradient method reduces to the projeced gradient descent method.

Lemma 5.3. *$u = prox_{\eta h}(x)$ if and only if $x - u \in \eta \partial h(u)$.*

Proof. Note that $u = \text{prox}_{\eta h}(x)$ means that u minimizes $h(u) + (1/2\eta)\|u - x\|_2^2$. By the optimality condition, it is equivalent to $0 \in \partial h(u) + \{(1/\eta)(u - x)\}$, and this is equivalent to $x - u \in \eta \partial h(u)$. □

4.2 Shrinkage Operator

Consider $h(x) = \|x\|_1$. Then

$$\text{prox}_{\eta h}(x) = \underset{u \in \mathbb{R}^d}{\text{argmin}} \left\{ \|u\|_1 + \frac{1}{2\eta}\|u - x\|_2^2 \right\}.$$

Let $u = \text{prox}_{\eta h}(x)$. Then, by Lemma 5.3,

$$x - u \in \eta \partial \|u\|_1.$$

Recall that $g \in \partial \|u\|_1$ if and only if

$$g_i = \begin{cases} \text{sign}(u_i), & \text{if } u_i \neq 0, \\ \text{a value in } [-1, 1], & \text{if } u_i = 0. \end{cases}$$

Based on this, we can argue that $x - u \in \eta \partial \|u\|_1$ if and only if

$$u_i = \begin{cases} x_i - \eta, & \text{if } x_i \geq \eta, \\ 0, & \text{if } -\eta \leq x_i \leq \eta. \\ x_i + \eta, & \text{if } x_i \leq -\eta. \end{cases}$$

Moreover, $x - u \in \eta \partial \|u\|_1$ if and only if

$$u_i = \max\{0, |x_i| - \eta\} \cdot \text{sign}(x_i).$$

For example,

$$\text{prox}_h((3, 1, -2)^\top) = (2, 0, -1)^\top.$$

Note that when $h = \|x\|_1$, the corresponding proximal operator "shrinks" the vector. For this reason, the operator is called the self-thresholding operator or the shrinkage operator.