# 2 Linear Regression

**Goal**

Understand linear regression for predicting a real response. Regularization and cross-validation.

**Alert 2.1: Convention**

Gray boxes are not required hence can be omitted for unenthusiastic readers.
This note is likely to be updated again soon.
Some notations and conventions in this note are different from those in the slides.

**Definition 2.2: Interpolation**

Given a sequence of pairs $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{R}^d \times \mathbb{R}^t : i = 1, \dots, n\}$, we want to find a function $f : \mathbb{R}^d \to \mathbb{R}^t$ so that for all $i$:

$$f(\mathbf{x}_i) \approx \mathbf{y}_i.$$

Most often, $t = 1$, i.e., each $\mathbf{y}_i$ is real-valued. However, we will indulge ourselves for treating any $t$ (since it brings very minimal complication).

The variable $t$ here stands for the number of responses (tasks), i.e., how many values we are interested in predicting (simultaneously).

**Theorem 2.3: Exact interpolation**

*For any finite number of pairs $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{R}^d \times \mathbb{R}^t : i = 1, \dots, n\}$ that satisfy $\mathbf{x}_i = \mathbf{x}_j \implies \mathbf{y}_i = \mathbf{y}_j$, there exist infinitely many functions $f : \mathbb{R}^d \to \mathbb{R}^t$ so that for all $i$:*

$$f(\mathbf{x}_i) = \mathbf{y}_i.$$

*Proof:* W.l.o.g. we may assume all $\mathbf{x}_i$'s are distinct. Lagrange polynomials give immediately such a claimed function. More generally, one may put a bump function within a small neighborhood of each $\mathbf{x}_i$ and then glue them together. In details, set $\mathcal{N}_i := \{\mathbf{z} : \|\mathbf{z} - \mathbf{x}_i\|_\infty < \delta\} \subseteq \mathbb{R}^d$. Clearly, $\mathbf{x}_i \in \mathcal{N}_i$ and for $\delta$ sufficiently small, $\mathcal{N}_i \cap \mathcal{N}_j = \emptyset$. Define

$$f_i(\mathbf{z}) = \begin{cases} \mathbf{y}_i e^{d/\delta^2} \prod_{j=1}^d \exp\left(-\frac{1}{\delta^2 - (z_j - x_{ji})^2}\right), & \text{if } \mathbf{z} \in \mathcal{N}_i \\ 0, & \text{otherwise} \end{cases}.$$

The function $f = \sum_i f_i$ again exactly interpolates our data $\mathcal{D}$. ■
The condition $\mathbf{x}_i = \mathbf{x}_j \implies \mathbf{y}_i = \mathbf{y}_j$ is clearly necessary, for otherwise there cannot be any function so that $\mathbf{y}_i = f(\mathbf{x}_i) = f(\mathbf{x}_j) = \mathbf{y}_j$ and $\mathbf{y}_i \neq \mathbf{y}_j$. Of course, when all $\mathbf{x}_i$'s are distinct, this condition is trivially satisfied.

**Exercise 2.4: From 1 to $\infty$**

Complete the proof of Theorem 2.3 with regard to the "infinite" part.

**Remark 2.5: Infinitely many choices...**

Theorem 2.3 has the following important implication: Given a finite training set $\mathcal{D}$, no matter how large its size might be, there exist infinitely many *smooth* (infinitely differentiable) functions $f$ that maps each $\mathbf{x}_i$ in $\mathcal{D}$ *exactly* to $\mathbf{y}_i$, i.e. they all achieve zero training "error." However, on a new test instance $\mathbf{x} \notin \mathcal{D}$, the predictions $\hat{\mathbf{y}} = f(\mathbf{x})$ of different choices of $f$ can be wildly different (in fact, we can make $f(\mathbf{x}) = \mathbf{y}$ for any $\mathbf{y} \in \mathbb{R}^t$).

Which function should we choose then?

**Definition 2.6: Least squares regression**

To resolve the difficulty in Remark 2.5, we need some statistical assumption on how our data is generated. In particular, we assume $(\mathsf{X}_i, \mathsf{Y}_i)$ are independently and identically distributed (i.i.d.) random samples from an unknown distribution $\mathbb{P}$. The test sample $(\mathsf{X}, \mathsf{Y})$ is also drawn independently and identically from the same distribution $\mathbb{P}$. We are interested in solving the least squares regression problem:

$$\min_{f:\mathbb{R}^d \to \mathbb{R}^t} \quad \mathbb{E}\|\mathsf{Y} - f(\mathsf{X})\|_2^2, \tag{2.1}$$

i.e., finding a function $f : \mathbb{R}^d \to \mathbb{R}^t$ so that $f(\mathsf{X})$ approximates $\mathsf{Y}$ well in expectation. (Strictly speaking we need the technical assumption that $f$ is measurable so that the above expectation is even defined.)

In reality, we do not know the distribution $\mathbb{P}$ of $(\mathsf{X}, \mathsf{Y})$ hence will not be able to compute the expectation, let alone minimizing it. Instead, we use the training set $\mathcal{D} = \{(\mathsf{X}_i, \mathsf{Y}_i) : i = 1, \dots, n\}$ to *approximate* the expectation:

$$\min_{f:\mathbb{R}^d \to \mathbb{R}^t} \quad \hat{\mathbb{E}}\|\mathsf{Y} - f(\mathsf{X})\|_2^2 := \frac{1}{n}\sum_{i=1}^n \|\mathsf{Y}_i - f(\mathsf{X}_i)\|_2^2. \tag{2.2}$$

By law of large numbers, for any *fixed* function $f$, we indeed have

$$\frac{1}{n}\sum_{i=1}^n \|\mathsf{Y}_i - f(\mathsf{X}_i)\|_2^2 \xrightarrow{\text{as } n\to\infty} \mathbb{E}\|\mathsf{Y} - f(\mathsf{X})\|_2^2.$$

**Remark 2.7: Properties of (conditional) expectation**

- The expectation $\mathbb{E}[\mathsf{Y}]$ intuitively is the (elementwise) average value of the random variable $\mathsf{Y}$.
- The conditional expectation $\mathbb{E}[\mathsf{Y}|\mathsf{X}]$ is an (equivalent class of) real-valued function(s) of $\mathsf{X}$.
- $\mathbb{E}[f(\mathsf{X})|\mathsf{X}] = f(\mathsf{X})$ (almost everywhere). Intuitively, conditioned on $\mathsf{X}$, $f(\mathsf{X})$ is not random hence the (conditional) expectation is vacuous.
- Law of total expectation: $\mathbb{E}[\mathsf{Y}] = \mathbb{E}[\mathbb{E}(\mathsf{Y}|\mathsf{X})]$ for any random variable $\mathsf{Y}$ and $\mathsf{X}$. Intuitively, the left hand side is the average value of $\mathsf{Y}$, while the right hand side is the average of averages: $\mathbb{E}(\mathsf{Y}|\mathsf{X})$ is the average of $\mathsf{Y}$ for a given realization of $\mathsf{X}$. For instance, take $\mathsf{Y}$ to be the height of a human being and $\mathsf{X}$ to be gender. Then, the average height of a human (left hand side) is equal to the average of the average heights of man and woman (right hand side).

**Definition 2.8: Regression function**

For a moment let us assume the distribution $\mathbb{P}$ is known to us, so we can at least in theory solve the least squares regression problem (2.1). It turns out there exists an optimal solution whose closed-form expression can be derived as follows:

$$\begin{aligned}
\mathbb{E}\|\mathsf{Y}-f(\mathsf{X})\|_2^2 &= \mathbb{E}\|\mathsf{Y}-\mathbb{E}(\mathsf{Y}|\mathsf{X})+\mathbb{E}(\mathsf{Y}|\mathsf{X})-f(\mathsf{X})\|_2^2 \\
&= \mathbb{E}\|\mathsf{Y}-\mathbb{E}(\mathsf{Y}|\mathsf{X})\|_2^2+\mathbb{E}\|\mathbb{E}(\mathsf{Y}|\mathsf{X})-f(\mathsf{X})\|_2^2+2\mathbb{E}[\langle \mathsf{Y}-\mathbb{E}(\mathsf{Y}|\mathsf{X}),\mathbb{E}(\mathsf{Y}|\mathsf{X})-f(\mathsf{X})\rangle] \\
&= \mathbb{E}\|\mathsf{Y}-\mathbb{E}(\mathsf{Y}|\mathsf{X})\|_2^2+\mathbb{E}\|\mathbb{E}(\mathsf{Y}|\mathsf{X})-f(\mathsf{X})\|_2^2+2\mathbb{E}\Big[\mathbb{E}[\langle \mathsf{Y}-\mathbb{E}(\mathsf{Y}|\mathsf{X}),\mathbb{E}(\mathsf{Y}|\mathsf{X})-f(\mathsf{X})\rangle\,|\mathsf{X}]\Big] \\
&= \mathbb{E}\|\mathsf{Y}-\mathbb{E}(\mathsf{Y}|\mathsf{X})\|_2^2+\mathbb{E}\|\mathbb{E}(\mathsf{Y}|\mathsf{X})-f(\mathsf{X})\|_2^2+2\mathbb{E}\Big[\langle \mathbb{E}[\mathsf{Y}-\mathbb{E}(\mathsf{Y}|\mathsf{X})|\mathsf{X}],\mathbb{E}(\mathsf{Y}|\mathsf{X})-f(\mathsf{X})\rangle\Big] \\
&= \mathbb{E}\|\mathsf{Y}-\mathbb{E}(\mathsf{Y}|\mathsf{X})\|_2^2+\mathbb{E}\|\mathbb{E}(\mathsf{Y}|\mathsf{X})-f(\mathsf{X})\|_2^2,
\end{aligned}$$

whence the regression function

$$m(\mathsf{X}) := \mathbb{E}(\mathsf{Y}|\mathsf{X}) \tag{2.3}$$

eliminates the second nonnegative term while the first nonnegative term is not affected by any $f$ at all.

With hindsight, it is not surprising the regression function $m$ is an optimal solution for the least squares regression problem: it basically says given $(\mathsf{X},\mathsf{Y})$, we set $m(\mathsf{X})=\mathsf{Y}$ if there is a unique value of $\mathsf{Y}$ asscoiated with $\mathsf{X}$ (which of course is optimal), while if there are multiple values of $\mathsf{Y}$ associated to the given $\mathsf{X}$, then we simply average them.

The constant term $\mathbb{E}\|\mathsf{Y}-\mathbb{E}(\mathsf{Y}|\mathsf{X})\|_2^2$ describes the difficulty of our regression problem: no function $f$ can reduce it.

**Definition 2.9: Overfitting, underfitting, and regularization**

All regression methods are in one way or another trying to approximate the regression function (2.3). In reality, we only have access to an i.i.d. training set, but if we solve (2.2) naively we will run into again the difficulty in Remark 2.5: we achieve very small (or even zero) error on the training set but the performance on a test sample can be very bad. This phenomenon is known as overfitting, i.e. we are taking our training set "too seriously." Remember in machine learning we are not interested in doing well on the training set at all (even though training data is all we got, oh life!); training set is used only as a means to get good performance on (future) test set.

Overfitting arises here because we are considering all functions $f:\mathbb{R}^d\to\mathbb{R}^t$, which is a *huge* class, while we only have limited (finite) training examples. In other words, we do not have enough training data to support our ambition. To address overfitting, we will restrict ourselves to a subclass $\mathcal{F}_n$ of functions $\mathbb{R}^d\to\mathbb{R}^t$ and solve:

$$\min_{f\in\mathcal{F}_n}\ \frac{1}{n}\sum_{i=1}^n\|\mathsf{Y}_i-f(\mathsf{X}_i)\|^2. \tag{2.4}$$

In other words, we regularize our choice of candidate functions $f$ to avoid fitting too well on the training set. Typically, $\mathcal{F}_n$ grows as $n$ increases, i.e. with more training data we could allow us to consider more candidate functions. On the other hand, if there are too few candidate functions in $\mathcal{F}_n$ (for instance when $\mathcal{F}_n$ consists of all constant functions), then we may not be able to do well even on the training set. This phenomemon is known as underfitting. Generally speaking, doing too well or too badly on the training set are both indications of poor design (either more data needs to be collected or a larger/smaller function class needs to be considered).

**Remark 2.10: Approximation vs. estimation and bias vs. variance**

It is possible that the regression function $m$ (see (2.3)), the object we are trying to find, is not in the chosen function class $\mathcal{F}_n$ at all. This results in the so-called approximation error (which solely depends on the function class $\mathcal{F}_n$ but not training data), or bias in statistical terms. Clearly, the "larger" $\mathcal{F}_n$ is, the smaller the approximation error. On the other hand, finding the "optimal" $f \in \mathcal{F}_n$ based on (2.4) is more challenging for a larger $\mathcal{F}_n$ (when $n$ is fixed). This is called estimation error, e.g. the error due to using a finite training set (random sampling). Typically, the "larger" $\mathcal{F}_n$ is, the larger the estimation error is. Often, with a "larger" $\mathcal{F}_n$, the performance on the test set has more variation (had we repeated with different training data) .

Much of the work on regression is about how to balance between the approximation error and estimation error, a.k.a. the bias and variance trade-off.

**Definition 2.11: Linear least squares regression**

The simplest choice for the function class $\mathcal{F}_n \equiv \mathcal{F}$ is perhaps the class of linear/affine functions (recall Definition 1.13). Adopting this choice leads to the linear least squares regression problem:

$$\min_{W \in \mathbb{R}^{t\times d}, \mathbf{b}\in\mathbb{R}^t} \quad \frac{1}{n}\sum_{i=1}^{n} \|\mathsf{Y}_i - W\mathsf{X}_i - \mathbf{b}\|_2^2, \tag{2.5}$$

where recall that $\mathsf{X}_i \in \mathbb{R}^d$ and $\mathsf{Y}_i \in \mathbb{R}^t$.

**Exercise 2.12: Linear functions may not exactly interpolate**

Show that Theorem 2.3 fails if we are only allowed to use linear/affine functions.

**Definition 2.13: Matrix norms**

For a matrix $W \in \mathbb{R}^{t\times d}$, we define its Frobenius norm

$$\|W\|_{\mathsf{F}} = \sqrt{\sum_{ij} w_{ij}^2},$$

which is essentially the matrix analogue of the vector Euclidean norm $\|\mathbf{w}\|_2$.

Another widely used matrix norm is the spectral norm

$$\|W\|_{\text{sp}} = \max_{\|\mathbf{x}\|_2=1} \|W\mathbf{x}\|_2,$$

which coincides with the largest singular value of $W$.

It is known that

- $\|W\|_{\text{sp}} \le \|W\|_{\mathsf{F}} \le \sqrt{\text{rank}(W)}\|W\|_{\text{sp}}$,
- $\|W\mathbf{x}\|_2 \le \|W\|_{\text{sp}}\|\mathbf{x}\|_2$.

**Remark 2.14: Padding again**

We apply the same padding trick as in Remark 1.17. Define $\mathbf{x}_i = \binom{\mathsf{X}_i}{1}$ and $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{p\times n}$, $Y = [\mathsf{Y}_1, \dots, \mathsf{Y}_n] \in \mathbb{R}^{t\times n}$, and $\mathbf{W} = [W, \mathbf{b}] \in \mathbb{R}^{t\times p}$, where of course $p = d+1$. We can then rewrite the linear

least squares problem in the following (equivalent but prettier) matrix format:

$$\min_{\mathbf{W}\in\mathbb{R}^{t\times p}} \quad \tfrac{1}{n}\|Y-\mathbf{W}\mathbf{X}\|_{\mathsf{F}}^2. \tag{2.6}$$

**Remark 2.15: Pseudo-inverse**

Recall that the Moore-Penrose pseudo-inverse $A^\dagger$ of any matrix $A\in\mathbb{R}^{p\times n}$ is the unique matrix $G\in\mathbb{R}^{n\times p}$ so that

$$AGA=A,\quad GAG=G,\quad (AG)^\top=AG,\quad (GA)^\top=GA.$$

In particular, if $A=USV^\top$ is the thin SVD (singular value decomposition) of $A$, then $A^\dagger=VS^{-1}U^\top$. If $A$ is in fact invertible, then $A^\dagger=A^{-1}$.

We can use pseudo-inverse to solve the linear least squares problem (2.6). Indeed, it is known that $W^\star:=A^\dagger CB^\dagger$ is a solution for the minimization problem:

$$\min_W \quad \|AWB-C\|_{\mathsf{F}}^2.$$

In fact, $W^\star$ is the unique solution that enjoys the minimum Frobenius norm. See this lecture note for proofs and more related results.

Equipped with the above result, we know that $\mathbf{W}^\star=Y\mathbf{X}^\dagger$ is a closed-form solution for the linear least squares problem (2.6).

**Definition 2.16: Gradient and Hessian (for the brave hearts)**

Recall that the gradient of a smooth function $f:\mathbb{R}^d\to\mathbb{R}^t$ at $\mathbf{w}$ is defined as the linear mapping $f'(\mathbf{w}):\mathbb{R}^d\to\mathbb{R}^t$ so that:

$$\lim_{\mathbf{0}\neq\Delta\mathbf{w}\to\mathbf{0}}\frac{\|f(\mathbf{w}+\Delta\mathbf{w})-f(\mathbf{w})-[f'(\mathbf{w})](\Delta\mathbf{w})\|}{\|\Delta\mathbf{w}\|}=0, \tag{2.7}$$

where say the norm $\|\cdot\|$ is the Euclidean norm. Or equivalently in big-o notation:

$$\|f(\mathbf{w}+\Delta\mathbf{w})-f(\mathbf{w})-[f'(\mathbf{w})](\Delta\mathbf{w})\|=o(\|\Delta\mathbf{w}\|).$$

As $\mathbf{w}$ varies, we may think of the gradient as the (nonlinear) mapping:

$$f':\mathbb{R}^d\to\mathcal{L}(\mathbb{R}^d,\mathbb{R}^t),\quad \mathbf{w}\mapsto\{f'(\mathbf{w}):\mathbb{R}^d\to\mathbb{R}^t\}$$

where $\mathcal{L}(\mathbb{R}^d,\mathbb{R}^t)$ denotes the class of linear mappings from $\mathbb{R}^d$ to $\mathbb{R}^t$, or equivalently the class of matrices $\mathbb{R}^{t\times d}$.

We can iterate the above definition. In particular, replacing $f$ with $f'$ we define the Hessian $f'':\mathbb{R}^d\to\mathcal{L}(\mathbb{R}^d,\mathcal{L}(\mathbb{R}^d,\mathbb{R}^t))\simeq\mathcal{B}(\mathbb{R}^d\times\mathbb{R}^d,\mathbb{R}^t)$ of $f$ as the derivative of the derivative $f'$, where $\mathcal{B}(\mathbb{R}^d\times\mathbb{R}^d,\mathbb{R}^t)$ denotes the class of bilinear mappings from $\mathbb{R}^d\times\mathbb{R}^d$ to $\mathbb{R}^t$.

**Definition 2.17: Gradient and Hessian through partial derivatives**

Let $f:\mathbb{R}^d\to\mathbb{R}$ be a real-valued smooth function. We can define its gradient through partial derivatives:

$$[\nabla f(\mathbf{w})]_j=\frac{\partial f}{\partial w_j}(\mathbf{w}).$$

Note that the gradient $\nabla f(\mathbf{w})\in\mathbb{R}^d$ has the same size as the input $\mathbf{w}$.

Similarly, we can define the Hessian through partial derivatives:

$$[\nabla^2 f(\mathbf{w})]_{ij} = \frac{\partial^2 f}{\partial w_i \partial w_j}(\mathbf{w}) = \frac{\partial^2 f}{\partial w_j \partial w_i}(\mathbf{w}) = [\nabla^2 f(\mathbf{w})]_{ji},$$

where the second equality holds as long as $f$ is twice-differentiable. Note that the Hessian is a symmetric matrix $\nabla^2 f(\mathbf{w}) \in \mathbb{R}^{d\times d}$ with the same number of rows/columns as the size of the input $\mathbf{w}$.

**Remark 2.18: Matrix input**

If $f : \mathbb{R}^{t\times p} \to \mathbb{R}$ takes a matrix as input, then its gradient is computed similarly as in Definition 2.17:

$$[\nabla f(\mathbf{W})]_{ij} = \frac{\partial f}{\partial w_{ij}}(\mathbf{W}).$$

Again, the gradient $\nabla f(\mathbf{W}) \in \mathbb{R}^{t\times p}$ has the same size as the input $\mathbf{W}$.

In principle, computing the Hessian is completely similar, although we will need 4 indices and the notation can get quite messy quickly. Fortunately, we will rarely find ourselves facing this challenge.

**Example 2.19: Quadratic function**

Consider the following quadratic function

$$f : \mathbb{R}^d \to \mathbb{R}, \quad \mathbf{w} \mapsto \mathbf{w}^\top Q\mathbf{w} + \mathbf{p}^\top \mathbf{w} + \alpha, \tag{2.8}$$

where $Q \in \mathbb{R}^{d\times d}$, $\mathbf{p} \in \mathbb{R}^d$, and $\alpha \in \mathbb{R}$. We can write explicitly:

$$f(\mathbf{w}) = \alpha + \sum_{k=1}^d w_k \left[p_k + \sum_{l=1}^d q_{kl} w_l\right],$$

whence follows

$$\begin{aligned}
[\nabla f(\mathbf{w})]_j = \frac{\partial f}{\partial w_j}(\mathbf{w}) &= \sum_{k=1}^d \left[\frac{\partial w_k}{\partial w_j}\left(p_k + \sum_{l=1}^d q_{kl} w_l\right) + w_k \frac{\partial\left(p_k + \sum_{l=1}^d q_{kl} w_l\right)}{\partial w_j}\right] \\
&= \left(p_j + \sum_{l=1}^d q_{jl} w_l\right) + \sum_{k=1}^d w_k q_{kj} \\
&= p_j + [(Q + Q^\top)\mathbf{w}]_j.
\end{aligned}$$

That is, collectively

$$\nabla f(\mathbf{w}) = \mathbf{p} + (Q + Q^\top)\mathbf{w}.$$

Similarly,

$$[\nabla^2 f(\mathbf{w})]_{ij} = \frac{\partial^2 f}{\partial w_i \partial w_j}(\mathbf{x}) = \frac{\partial\left(p_j + \sum_{l=1}^d q_{jl} w_l + \sum_{k=1}^d w_k q_{kj}\right)}{\partial w_i} = q_{ji} + q_{ij}.$$

That is, collectively

$$\nabla^2 f(\mathbf{w}) \equiv Q + Q^\top.$$

The formula further simplifies if we assume $Q$ is symmetric, i.e. $Q = Q^\top$ (which is usually the case).

As demonstrated above, using partial derivatives to derive the gradient and Hessian is straightforward but tedious. Fortunately, we need only do this once: derive and memorize a few, and then resort to the chain rule.

**Example 2.20: Quadratic function (for the brave hearts)**

Another way to derive the gradient and Hessian is to guess and then verify the definition (2.7). Using the quadratic function (2.8) again as an example. We need to verify:

$$\begin{aligned} o(\|\Delta\mathbf{w}\|) &= \|f(\mathbf{w}+\Delta\mathbf{w}) - f(\mathbf{w}) - [\nabla f(\mathbf{w})](\Delta\mathbf{w})\| \\ &= \|(\mathbf{w}+\Delta\mathbf{w})^\top Q(\mathbf{w}+\Delta\mathbf{w}) + \mathbf{p}^\top(\mathbf{w}+\Delta\mathbf{w}) - \mathbf{w}^\top Q\mathbf{w} - \mathbf{p}^\top\mathbf{w} - [\nabla f(\mathbf{w})](\Delta\mathbf{w})\| \\ &= \|\mathbf{w}^\top Q\Delta\mathbf{w} + (\Delta\mathbf{w})^\top Q\mathbf{w} + \mathbf{p}^\top\Delta\mathbf{w} - [\nabla f(\mathbf{w})](\Delta\mathbf{w}) + (\Delta\mathbf{w})^\top Q(\Delta\mathbf{w})\|, \end{aligned}$$

whence we guess

$$\nabla f(\mathbf{w}) = \mathbf{p} + (Q + Q^\top)\mathbf{w}$$

so that we can cancel out the first four terms (that are all linear in $\Delta\mathbf{w}$). It is then easy to verify that the remaining term indeed satisfies

$$\|(\Delta\mathbf{w})^\top Q(\Delta\mathbf{w})\| = o(\|\Delta\mathbf{w}\|).$$

Similarly, we can guess and verify the Hessian:

$$\begin{aligned} o(\|\Delta\mathbf{w}\|) &= \|\nabla f(\mathbf{w}+\Delta\mathbf{w}) - \nabla f(\mathbf{w}) - [\nabla^2 f(\mathbf{w})](\Delta\mathbf{w})\| \\ &= \|(Q+Q^\top)\Delta\mathbf{w} - [\nabla^2 f(\mathbf{w})](\Delta\mathbf{w})\|, \end{aligned}$$

from which it is clear that $\nabla^2 f(\mathbf{w}) = Q + Q^\top$ would do.

**Theorem 2.21: Fermat's necessary condition for extrema (recalled)**

*A necessary condition for* $\mathbf{w}$ *to be a local minimizer of a differentiable function* $f : \mathbb{R}^d \to \mathbb{R}$ *is*

$$\nabla f(\mathbf{w}) = \mathbf{0}.$$

*(Such points are called stationary, a.k.a. critical.) For convex* $f$ *the necessary condition is also sufficient.*

*Proof:* See the lecture note. ■

Take $f(w) = w^3$ and $w = 0$ we see that this necessary condition is not sufficient for nonconvex functions. For local maximizers, we simply negate the function and apply the theorem to $-f$ instead.

**Definition 2.22: Normal equation**

We may now solve the linear least squares problem (2.6). Simply compute its gradient (by following Example 2.19) and set it to $\mathbf{0}$, as suggested by Theorem 2.21. Upon simplifying, we obtain the so-called normal equation (for the unknown variable $\mathbf{W} \in \mathbb{R}^{t\times p}$):

$$\mathbf{W}\mathbf{X}\mathbf{X}^\top = Y\mathbf{X}^\top, \text{ or after transposing } \mathbf{X}\mathbf{X}^\top\mathbf{W}^\top = \mathbf{X}Y^\top. \tag{2.9}$$

This is a system of linear equations, which we can solve using standard numerical linear algebra toolboxes (Cholesky decomposition in this case). The time complexity is $O(p^3 + p^2 n + pnr)$. For large $n$ or $p$, we may use iterative algorithms (e.g. conjugate gradient) to directly but approximately solve (2.6).

**Exercise 2.23: Linear least squares is linear**

Based on the original linear least squares formula (2.6), or the normal equation (2.9) (or the more direct solution $Y\mathbf{X}^\dagger$ using pseudo-inverse), prove the following equivariance property of linear least squares:

- If we apply a nonsingular transformation $T \in \mathbb{R}^{p\times p}$ to $\mathbf{X}$ (or $X$), what would happen to the linear least squares solution $\mathbf{W}$?
- If we apply a nonsingular transformation $T \in \mathbb{R}^{t\times t}$ to $Y$, what would happen to the linear least squares solution $\mathbf{W}$?

**Definition 2.24: Prediction**

Once we have the linear least squares solution $\hat{\mathbf{W}} = (\hat{W}, \hat{\mathbf{b}})$, we perform prediction on a (future) test sample $\mathsf{X}$ naturally by:

$$\hat{\mathsf{Y}} := \hat{W}\mathsf{X} + \hat{\mathbf{b}}.$$

We measure the "goodness" of our prediction $\hat{\mathsf{Y}}$ by:

$$\|\hat{\mathsf{Y}} - \mathsf{Y}\|_2^2,$$

which is usually averaged over a test set.

**Alert 2.25: Calibration**

Note that we used the squared loss $(y, \hat{y}) \mapsto (y - \hat{y})^2$ in training linear least squares regression, see (2.5). Thus, naturally, when evaluating the linear least squares solution $\hat{\mathbf{W}}$ on a test set, we should use the same squared loss. If we use a different loss, such as the absolute error $|\hat{y} - y|$, then our training procedure may be suboptimal. Be consistent in terms of the training objective and the test measure!

Nevertheless, sometimes it might be necessary to use one loss $\ell_1(\hat{y}, y)$ for training and another one $\ell_2(\hat{y}, y)$ for testing. For instance, $\ell_1$ may be easier to handle computationally. The theory of calibration studies when a minimizer under the loss $\ell_1$ remains optimal under a different loss $\ell_2$. See the (heavy) paper of Steinwart (2007) and some particular refinement in Long and Servedio (2013) for more details.

Steinwart, I. (2007). "How to compare different loss functions and their risks". *Constructive Approximation*, vol. 26, no. 2, pp. 225–287.

Long, P. M. and R. A. Servedio (2013). "Consistency versus Realizable $H$-Consistency for Multiclass Classification". In: *Proceedings of the 31st International Conference on Machine Learning (ICML)*.

**Definition 2.26: Ridge regression with Tikhonov regularization (Tikhonov63; Hoerl and Kennard 1970)**

The class of linear functions may still be too large, leading linear least squares to overfit or instable. We can then put some extra restriction, such as the Tikhonov regularization in ridge regression:

$$\min_{\mathbf{W}\in\mathbb{R}^{t\times p}} \quad \tfrac{1}{n}\|Y - \mathbf{W}\mathbf{X}\|_{\mathsf{F}}^2 + \lambda\|\mathbf{W}\|_{\mathsf{F}}^2, \tag{2.10}$$

where $\lambda \geq 0$ is the regularization constant (hyperparameter) that balances the two terms.

To understand ridge regression, consider

- when $\lambda$ is small, thus we are neglecting the second regularization term, and the solution resembles that of the ordinary linear least squares solution;
- when $\lambda$ is large, thus we are neglecting the first data term, and the solution degenerates to $\mathbf{0}$.

In the literature, the following variant that chooses *not* to regularize the bias term $\mathbf{b}$ is also commonly used:

$$\min_{\mathbf{W}=[W,\mathbf{b}]} \quad \tfrac{1}{n}\|Y - \mathbf{W}\mathbf{X}\|_{\mathsf{F}}^2 + \lambda\|W\|_{\mathsf{F}}^2. \tag{2.11}$$

Hoerl, A. E. and R. W. Kennard (1970). "Ridge regression: Biased estimation for nonorthogonal problems". *Technometrics*, vol. 12, no. 1, pp. 55–67.

**Exercise 2.27: Solution for ridge regression**

Prove that the unique solution for (2.10) is given by the following normal equation:

$$(\mathbf{X}\mathbf{X}^\top + n\lambda I)\mathbf{W}^\top = \mathbf{X}Y^\top,$$

where $I$ is the identity matrix of appropriate size.

- Follow the derivation in Definition 2.22 by computing the gradient and setting it to $\mathbf{0}$.
- Use the trick you learned in elementary school, completing the square, to reduce ridge regression to the ordinary linear least squares regression.
- Data augmentation: ridge regression is equivalent to the ordinary linear least squares regression (2.6) with

$$\mathbf{X} \leftarrow [\mathbf{X}, \sqrt{n\lambda}I_{p\times p}], \quad Y \leftarrow [Y, \mathbf{0}_{t\times p}].$$

How about the solution for the variant (2.11)?

**Remark 2.28: Equivalence between regularization and constraint**

The regularized problem

$$\min_{\mathbf{W}} \quad \ell(\mathbf{W}) + \lambda \cdot r(\mathbf{W}) \tag{2.12}$$

is "equivalent" to the following constrained problem:

$$\min_{r(\mathbf{W})\le\gamma} \quad \ell(\mathbf{W}) \tag{2.13}$$

in the sense that

- for any $\lambda \ge 0$ and any solution $\mathbf{W}$ of (2.12) there exists some $\gamma$ so that $\mathbf{W}$ remains to be a solution for (2.13);
- under mild conditions, for any $\gamma$ there exists some $\lambda \ge 0$ so that any solution $\mathbf{W}$ of (2.13) remains to be a solution for (2.12).

The regularized problem (2.12) is computationally more appealing as it is *unconstrained* while the constrained problem (2.13) is more intuitive and easier to analyze theoretically. Using this equivalence we see that ridge regression essentially performs linear least squares regression over the restricted class of linear functions whose weights are no larger than some constant $\gamma$.

**Exercise 2.29: Is ridge regression linear?**

Redo Exercise 2.23 with ridge regression.

**Remark 2.30: Choosing the regularization constant**

The regularization constant $\lambda \geq 0$ balances the data term and the regularization term in ridge regression (2.10). A typical way to select $\lambda$ is through a validation set $\mathcal{V}$ that is different from the training set $\mathcal{D}$ and the test set $\mathcal{T}$. For each $\lambda$ in a candidate set $\Lambda \subseteq \mathbb{R}_+$ (chosen by experience or trial and error), we train our algorithm on $\mathcal{D}$ and get $\hat{\mathbf{W}}_\lambda$, evaluate the performance of $\hat{\mathbf{W}}_\lambda$ on $\mathcal{V}$ (according to some metric $\mathsf{perf}$), choose the best $\hat{\mathbf{W}}_\lambda$ (w.r.t. $\mathsf{perf}$) and finally apply it on the test set $\mathcal{T}$. Importantly,

- One should never "see" the test set during training and parameter tuning; otherwise we are cheating and bound to overfit.
- The validation set $\mathcal{V}$ should be different from both the training set and the test set.
- We can only use the validation set once in parameter tuning. Burn After Reading! (However, see the recent work of Dwork et al. (2017) and the references therein for data re-usage.)
- As mentioned in Alert 2.25, the performance measure $\mathsf{perf}$ should ideally align with the training objective.

Dwork, C., V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth (2017). "Guilt-free Data Reuse". *Communications of the ACM*, vol. 60, no. 4, pp. 86–93.

**Algorithm 2.31: Cross-validation**

When data is limited, we do not have the luxury to afford a separate validation set, in which case we may use the cross-validation procedure for hyperparameter selection. Essentially we split the training set and hold a (small) part out as validation set. However, to avoid random fluctuations (especially when the validation set is small), we repeat the split $k$ times and average the results.

**Algorithm:** Cross-Validation

**Input:** candidate regularization constants $\Lambda \subseteq \mathbb{R}$, performance metric $\mathsf{perf}$, $k \in \mathbb{N}$, training data $\mathcal{D}$
**Output:** best regularization constant $\lambda^\star$

randomly partition $\mathcal{D}$ into similarly sized $k$ subsets $\mathcal{D}_1, \dots, \mathcal{D}_k$ `// e.g.` $|\mathcal{D}_l| = \lfloor |\mathcal{D}|/k \rfloor, \forall l < k$
**for** $\lambda \in \Lambda$ **do**
  $p(\lambda) \leftarrow 0$
  **for** $l = 1, \dots, k$ **do**
    train on $\mathcal{D}_{\neg l} := \cup_{j \neq l} \mathcal{D}_j = \mathcal{D} \setminus \mathcal{D}_l$ and get $\hat{\mathbf{W}}_{\lambda, \neg l}$
    $p(\lambda) \leftarrow p(\lambda) + \mathsf{perf}(\hat{\mathbf{W}}_{\lambda, \neg l}, \mathcal{D}_l)$ `// evaluate` $\hat{\mathbf{W}}_{\lambda,\neg l}$ `on the holdout set` $\mathcal{D}_l$
$\lambda^\star \leftarrow \operatorname{argmin}_{\lambda \in \Lambda} p(\lambda)$ `// assuming the smaller perf the better`

With the "optimal" $\lambda^\star$ at hand, we re-train on the entire training set $\mathcal{D}$ to get $\hat{\mathbf{W}}_{\lambda^\star}$ and then evaluate it on the test set using the same performance measure $\mathsf{perf}(\hat{\mathbf{W}}_{\lambda^\star}, \mathcal{T})$.

In the extreme case where $k = |\mathcal{D}|$, each time we train on the entire training set except one data instance. This is called leave-one out cross-validation. Typically we set $k = 10$ or $k = 5$. Note that the larger $k$ is, the more expensive (computationally) cross-validation is.

As an example, for ridge regression, we can set the performance measure as:

$$\mathsf{perf}(\mathbf{W}, \mathcal{D}) := \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \|\mathsf{Y}_i - W\mathsf{X}_i - \mathbf{b}\|_2^2 = \frac{1}{|\mathcal{D}|} \|Y - \mathbf{W}\mathbf{X}\|_{\mathsf{F}}^2,$$

where recall that

$$\mathbf{X} = \begin{bmatrix} \mathsf{X}_1 & \cdots & \mathsf{X}_{|\mathcal{D}|} \\ 1 & \cdots & 1 \end{bmatrix}, \quad Y = [\mathsf{Y}_1, \ldots, \mathsf{Y}_{|\mathcal{D}|}], \quad \text{and } \mathbf{W} = [W, \mathbf{b}].$$

**Exercise 2.32: Can we use training objective as performance measure?**

Explain if we can use the regularized training objective $\frac{1}{|\mathcal{D}|}\|Y - \mathbf{W}\mathbf{X}\|_{\mathsf{F}}^2 + \lambda\|\mathbf{W}\|_{\mathsf{F}}^2$ as our performance measure $\mathsf{perf}$?