12 GRADIENT DESCENT METHOD

The gradient-descent method is the backbone of learning algorithms. It is a powerful iterative procedure that allows us to approach minimizers of objective functions when closed-form expressions for these minimizers are not possible. Several variations will be described in this and the following chapters. We focus initially on objective functions that are *first-order differentiable*. In subsequent chapters we consider non-smooth functions that may have points of non-differentiability and introduce subgradient and proximal algorithms for their minimization. Although gradient-descent algorithms can be applied to both convex and non-convex functions, we will focus largely on convex objectives and examine their convergence properties. Later, in Chapter 24, we consider non-convex optimization problems.

12.1 EMPIRICAL AND STOCHASTIC RISKS

We consider an optimization problem of the following generic form:

$$w^\star \overset{\Delta}{=} \underset{w\in\mathbb{R}^M}{\text{argmin}}\;\; P(w) \tag{12.1}$$

where $P(w)$ refers to the objective function that we wish to minimize, $w \in \mathbb{R}^M$ is the independent variable, and $w^\star$ denotes a minimizing argument. In the context of learning algorithms, objective functions are called *risks* because they provide a measure of how much error or risk is incurred in using a solution w to make inference decisions.

12.1.1 Empirical Risks

The results in this chapter are applicable to convex risk functions, $P(w)$. In learning problems, $P(w)$ will generally be some function of N data points denoted by the notation $\{\gamma(m), h_m,\; m = 0, 1, \ldots, N-1\}$, where $\gamma \in \mathbb{R}$ is a scalar referred to as the *target* or *label* variable and $h \in \mathbb{R}^M$ is a vector referred to as the *feature* vector. In particular, $P(w)$ will often take the form of a *sample average* over this data, written as

$$P(w) = \frac{1}{N}\sum_{m=0}^{N-1} Q\Big(w;\gamma(m),h_m\Big), \quad \gamma(m)\in \mathrm{I\!R}, \quad h_m \in \mathrm{I\!R}^M \tag{12.2}$$

for some convex function $Q(w;\cdot,\cdot)$, referred to as the *loss*. The value $Q(w;\gamma(m),h_m)$ represents the loss at the $m-$th data pair $(\gamma(m),h_m)$. When $P(w)$ has the sample average form (12.2), we refer to it as an *empirical* risk; one that is defined directly from data measurements. In this way, problem (12.1) becomes an empirical risk minimization (ERM) problem of the form:

$$\boxed{w^\star \;\overset{\Delta}{=}\; \underset{w\in \mathrm{I\!R}^M}{\text{argmin}}\; \frac{1}{N}\sum_{m=0}^{N-1} Q\Big(w;\gamma(m),h_m\Big)} \quad \textbf{(empirical risk minimization)} \tag{12.3}$$

We will encounter several choices for the loss function in future chapters such as:

$$Q\Big(w;\gamma(m),h_m\Big) = \begin{cases} q(w) \;+\; (\gamma(m)-h_m^{\mathsf T}w)^2, & \text{(quadratic)}\\ q(w) \;+\; \ln\Big(1+e^{-\gamma(m)h_m^{\mathsf T}w}\Big), & \text{(logistic)}\\ q(w) \;+\; \max\Big\{0,-\gamma(m)h_m^{\mathsf T}w\Big\}, & \text{(Perceptron)}\\ q(w) \;+\; \max\Big\{0,1-\gamma(m)h_m^{\mathsf T}w\Big\}, & \text{(hinge)} \end{cases} \tag{12.4}$$

The (also convex) function $q(w)$ is called the *regularization* factor and it usually takes one of several forms, such as:

$$q(w) \;=\; \begin{cases} 0, & \text{(no regularization)}\\ \rho\|w\|^2, & (\ell_2-\text{regularization})\\ \alpha\|w\|_1, & (\ell_1-\text{regularization})\\ \alpha\|w\|_1+\rho\|w\|^2, & \text{(elastic-net regularization)} \end{cases} \tag{12.5}$$

where $\alpha > 0$ and $\rho > 0$. Other choices for $q(w)$ are possible. We explain in future Chapter 51 that the choice of $q(w)$ plays an important role in determining the form of the minimizer $w^\star$, such as forcing it to have a small norm or forcing it to be sparse and have many zero entries. Table 12.1 lists the empirical risk functions described so far.

Note that all loss functions in (12.4) depend on $\{h_m, w\}$ through the inner product $h_m^{\mathsf T}w$. Although unnecessary, this property will hold for most loss functions of interest in our treatment. It is customary to interpret this inner product as an estimate or prediction for $\gamma(m)$, written as

$$\widehat{\gamma}(m) = h_m^{\mathsf T}w, \qquad \textbf{(prediction)} \tag{12.6}$$

In this way, the loss functions in (12.4) can be interpreted as measuring the discrepancy between the labels $\{\gamma(m)\}$ and their predictions $\{\widehat{\gamma}(m)\}$. By seeking a minimizer $w^\star$ in (12.3), we are in effect seeking a model that "best" matches the $\{\widehat{\gamma}(m)\}$ to the $\{\gamma(m)\}$.

Table 12.1 Examples of empirical risks based on N data pairs $\{\gamma(m), h_m\}$, and where $q(w)$ denotes a convex regularization factor.

name	**empirical risk,** $P(w)$
least-squares	$q(w) + \frac{1}{N}\sum_{m=0}^{N-1}\left(\gamma(m) - h_m^\mathsf{T} w\right)^2$
logistic	$q(w) + \frac{1}{N}\sum_{m=0}^{N-1}\ln\left(1 + e^{-\gamma(m)h_m^\mathsf{T} w}\right)$
Perceptron	$q(w) + \frac{1}{N}\sum_{m=0}^{N-1}\max\left\{0,\ -\gamma(m)h_m^\mathsf{T} w\right\}$
hinge	$q(w) + \frac{1}{N}\sum_{m=0}^{N-1}\max\left\{0,\ 1 - \gamma(m)h_m^\mathsf{T} w\right\}$

12.1.2 Stochastic Risks

In many instances, the objective function $P(w)$ will not have an empirical form but will instead be *stochastic* in nature. In these cases, $P(w)$ will be defined as the expectation of the loss function:

$$P(w) = \mathbb{E}\, Q(w; \boldsymbol{\gamma}, \boldsymbol{h}) \tag{12.7}$$

Here, the expectation operator $\mathbb{E}$ is relative to the distribution of the data $\{\boldsymbol{\gamma}, \boldsymbol{h}\}$, now assumed to be randomly distributed according to some joint probability density function, $f_{\boldsymbol{\gamma},\boldsymbol{h}}(\gamma, h)$. In this way, problem (12.1) becomes one of the form:

$$\boxed{w^o \stackrel{\Delta}{=} \underset{w\in\mathbb{R}^M}{\text{argmin}}\ \mathbb{E}\, Q(w; \boldsymbol{\gamma}, \boldsymbol{h})} \qquad \textbf{(stochastic risk minimization)} \tag{12.8}$$

where we are denoting the minimizing argument by w^o. In analogy with Table 12.1, we list examples of stochastic risks in Table 12.2. Note that all loss functions in the table depend again on $\{\boldsymbol{h}, w\}$ through the inner product $\boldsymbol{h}^\mathsf{T} w$, which we also interpret as a prediction for $\boldsymbol{\gamma}$, written as $\widehat{\boldsymbol{\gamma}} = \boldsymbol{h}^\mathsf{T} w$. In this way, the loss functions in Table 12.2 measure the average discrepancy between the label $\boldsymbol{\gamma}$ and its prediction $\widehat{\boldsymbol{\gamma}}$ over the distribution of the data. By seeking the minimizer w^o in (12.8), we are in effect seeking a model w^o that "best" matches $\widehat{\boldsymbol{\gamma}}$ to $\boldsymbol{\gamma}$ in some average loss sense.

Remark 12.1. **(Notation for minimizers)** We will employ the following convention throughout our treatment to distinguish between the empirical and stochastic scenarios. We will denote the minimizer of an empirical risk by $w^\star$ and the minimizer of a stochastic risk by w^o:

$$w^\star : \text{ minimizers for empirical risks} \tag{12.9a}$$

$$w^o : \text{ minimizers for stochastic risks} \tag{12.9b}$$

In general though, when we are dealing with a generic optimization problem where $P(w)$ can refer to either an empirical or stochastic risk, we will denote the minimizing argument generically by $w^\star$, as was already done in (12.1).

■

Table 12.2 Examples of stochastic risks defined over the joint distribution of the data $\{\boldsymbol{\gamma}, \boldsymbol{h}\}$, and where $q(w)$ denotes the regularization factor.

name	**stochastic risk**, $P(w)$
mean-square-error	$q(w) + \mathbb{E}\,\left(\boldsymbol{\gamma} - \boldsymbol{h}^\mathsf{T} w\right)^2$
logistic	$q(w) + \mathbb{E}\,\ln\left(1 + e^{-\boldsymbol{\gamma}\boldsymbol{h}^\mathsf{T} w}\right)$
Perceptron	$q(w) + \mathbb{E}\,\max\left\{0,\, -\boldsymbol{\gamma}\boldsymbol{h}^\mathsf{T} w\right\}$
hinge	$q(w) + \mathbb{E}\,\max\left\{0,\, 1 - \boldsymbol{\gamma}\boldsymbol{h}^\mathsf{T} w\right\}$

One difficulty that arises in the minimization of stochastic risks of the form (12.8) is that the joint distribution of the data $\{\boldsymbol{\gamma}, \boldsymbol{h}\}$ is rarely known beforehand. This means that the expectation in (12.7) cannot be computed, which in turn means that the risk function $P(w)$ itself is not known! This situation is different from the empirical risk case (12.2) where $P(w)$ is defined in terms of N data pairs $\{\gamma(m), h_m\}$ and is therefore known. However, motivated by the ergodicity property (7.18), we can approximate the expectation in (12.7) and replace it by a sample average computed over a good number of data samples $\{\gamma(m), h_m\}$ arising from the unknown distribution. Using these samples, we can approximate the *stochastic* risk (12.7) by the *empirical* risk (12.2). For this reason, gradient-descent methods for minimizing empirical risks are equally applicable to the minimization of stochastic risks, as our presentation will reveal.

12.1.3 Generalization

The models $\{w^\star, w^o\}$ that result from minimizing empirical or stochastic risks will be used to perform inference on new feature vectors. If we denote a generic feature vector by h, then $w^\star$ can be used to predict its target or label by using $\widehat{\gamma} = h^\mathsf{T} w^\star$; likewise, for w^o. One important distinction arises in the performance of the two models $\{w^\star, w^o\}$:

(a) An empirical risk formulation of the form (12.3) determines the optimizer $w^\star$ that is implied by the *given* collection of N data points, $\{\gamma(m), h_m\}$. As such, the performance of the empirical model $w^\star$ in predicting future labels will be strongly dependent on how representative the original dataset $\{\gamma(m), h_m\}$

is of the space from which features and labels arise. This issue relates to the important question of "*generalization*," and will be discussed in greater detail in future Chapter 64. Intuitively, one model w_a is said to generalize better than another model w_b if w_a is able to perform more accurate predictions than w_b for new feature data. The concept of "generalization" is also referred to as *inductive inference* or *inductive reasoning* because it endows models with the ability to reason about new feature data based on experience learned from training data.

(b) In contrast, the stochastic risk formulation (12.8) seeks the optimizer w^o that is defined by the joint probability distribution of the data $\{\boldsymbol{\gamma}, \boldsymbol{h}\}$, and not by any finite collection of data points arising from this distribution. This is because the optimization criterion seeks to minimize the *average loss* over the joint pdf. The resulting model w^o is expected to perform better on average in predicting new labels. The challenge, however, as we are going to see, is that it is not possible to minimize stochastic risks directly because they require knowledge of the joint pdf of the data, and this information is rarely available. For this reason, solutions for the stochastic risk problem will often involve a step that reduces it to an empirical risk problem through an ergodic approximation, which is then minimized from a collection of data points. The bottom line is that, either way, whether we are dealing with empirical or stochastic risks, it is important to examine how well inference models generalize. We defer the technical details to Chapter 64.

12.2 CONDITIONS ON RISK FUNCTION

Three observations that are warranted at this stage:

(a) First, in many cases of interest in this and subsequent chapters, the risk $P(w)$ will have a *unique* global minimizer $w^\star$ since $P(w)$ will generally be strongly-convex. This is because the addition of regularization factors will often ensure strong convexity. We will examine this case in some detail. We will also comment on the case when $P(w)$ is only convex, as well as study nonconvex risks in future Chapter 24.

(b) Second, the development in this chapter is not limited to the risks and losses shown in the previous tables.

(c) Third, the risk function $P(w)$ need not be smooth (i.e., it need not be differentiable everywhere). For example, for the logistic risk in Table 12.1 we have

$$P(w) = q(w) \; + \; \frac{1}{N}\sum_{m=0}^{N-1} \ln\left(1 + e^{-\gamma(m)h_m^{\mathsf{T}} w}\right) \tag{12.10}$$

This function is differentiable for all w when $q(w) = \rho\|w\|^2$ but is not differentiable at $w = 0$ when $q(w) = \alpha\|w\|_1$ or $q(w) = \alpha\|w\|_1 + \rho\|w\|^2$. Likewise, for the hinge risk in Table 12.1 we have

$$P(w) = q(w) \; + \; \frac{1}{N}\sum_{m=0}^{N-1} \max\Big\{0,\, 1 - \gamma(m)h_m^{\mathsf{T}} w\Big\} \tag{12.11}$$

This function is not differentiable at all points w satisfying $1 = \gamma(m)h_m^{\mathsf{T}}w$. The function is also not differentiable at $w = 0$ when $q(w) = \alpha\|w\|_1$ or $q(w) = \alpha\|w\|_1 + \rho\|w\|^2$. Observe that non-differentiability can arise either from the regularization term or from the unregularized component of the risk. The recursive techniques for determining $w^\star$ will need to account for the possibility of points of non-differentiability. We focus in this chapter on the case in which $P(w)$ is first-order differentiable, and defer the case of non-smooth risks to future chapters.

Motivated by these considerations, we will consider in this chapter optimization problems of the form (12.1) where the risk function $P(w)$ satisfies two conditions:

(A1) (**Strong convexity**). $P(w)$ is $\nu-$strongly convex and first-order differentiable at all w so that, from definition (8.21),

$$P(w_2) \;\geq\; P(w_1) \;+\; \nabla_w P(w_1)(w_2 - w_1) \;+\; \frac{\nu}{2}\|w_2 - w_1\|^2 \tag{12.12a}$$

for every $w_1, w_2 \in \text{dom}(P)$ and some $\nu > 0$.

(A2) (**$\delta-$Lipschitz gradients**). The gradient vectors of $P(w)$ are $\delta-$Lipschitz:

$$\|\nabla_w P(w_2) - \nabla_w P(w_1)\| \;\leq\; \delta\,\|w_2 - w_1\| \tag{12.12b}$$

for any $w_1, w_2 \in \text{dom}(P)$, and where $\|\cdot\|$ denotes the Euclidean norm of its vector argument.

For reference, we know from the earlier results (8.29) and (10.20) derived for strongly-convex and $\delta-$Lipschitz functions that conditions **A1** and **A2** imply respectively:

$$(\textbf{A1}) \quad \Longrightarrow \quad \frac{\nu}{2}\|\widetilde{w}\|^2 \leq P(w) - P(w^\star) \leq \frac{1}{2\nu}\|\widetilde{w}\|^2 \tag{12.13a}$$

$$(\textbf{A2}) \quad \Longrightarrow \quad \frac{1}{2\delta}\|\widetilde{w}\|^2 \leq P(w) - P(w^\star) \leq \frac{\delta}{2}\|\widetilde{w}\|^2 \tag{12.13b}$$

where $\widetilde{w} = w^\star - w$. The upper bounds in both expressions indicate that whenever we bound $\|\widetilde{w}\|^2$ we will also be automatically bounding the excess risk, $P(w) - P(w^\star)$.

Example 12.1 (**Second-order differentiability**) Conditions (12.12a)–(12.12b) only require $P(w)$ to be first-order differentiable since the conditions are stated in terms of the gradient of the risk function. However, if $P(w)$ happens to be *second-order* differentiable over w, then we can combine both conditions into a single statement involving the Hessian matrix of $P(w)$. Recall from property (8.30) that strong-convexity is equivalent to $P(w)$ having a Hessian matrix that is uniformly bounded from *below* by ν, i.e.,

$$0 < \nu I_M \leq \nabla_w^2\, P(w), \quad \forall\, w \in \text{dom}(P) \tag{12.14a}$$

We also know from (10.32) that the $\delta-$Lipschitz condition (12.12b) is equivalent to the Hessian matrix being uniformly bounded from *above* by δ, i.e.,

$$\nabla_w^2\, P(w) \;\leq\; \delta I_M, \quad \forall\, w \in \text{dom}(P) \tag{12.14b}$$

Therefore, combining (12.14a) and (12.14b) we find that under second-order differentiability of $P(w)$, the two conditions (12.12a)–(12.12b) are equivalent to requiring the Hessian matrix of $P(w)$ to be uniformly bounded from below *and* from above as follows:

$$\boxed{0 < \nu I_M \leq \nabla_w^2\, P(w) \;\leq\; \delta I_M} \tag{12.15}$$

Clearly, condition (12.15) requires $\nu \leq \delta$. Several risks from Table 12.1 satisfy property (12.15). Here is one example — see also Prob. 12.2.

Example 12.2 **(Logistic empirical risk)** Consider the ℓ_2-regularized logistic risk from Table 12.1, namely,

$$P(w) = \rho\|w\|^2 \;+\; \frac{1}{N}\sum_{m=0}^{N-1} \ln\left(1 + e^{-\gamma(m)h_m^{\mathsf{T}} w}\right) \tag{12.16}$$

It can be verified that

$$\nabla_w^2\, P(w) = 2\rho I_M \;+\; \frac{1}{N}\sum_{m=0}^{N-1} h_m h_m^{\mathsf{T}}\, \underbrace{\frac{e^{-\gamma(m)h_m^{\mathsf{T}} w}}{\left(1 + e^{-\gamma(m)h_m^{\mathsf{T}} w}\right)^2}}_{\leq 1} \tag{12.17}$$

from which we conclude that

$$0 \;<\; \underbrace{2\rho}_{\triangleq\, \nu}\, I_M \;\leq\; \nabla_w^2\, P(w) \;\leq\; \underbrace{2\rho I_M + \lambda_{\max}\left(\frac{1}{N}\sum_{m=0}^{N-1} h_m h_m^{\mathsf{T}}\right) I_M}_{\triangleq\, \delta} \tag{12.18}$$

where the notation $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue of its symmetric matrix argument.

12.3 CONSTANT STEP-SIZES

We are now ready to motivate the gradient-descent method. We consider first the case in which a constant step-size is employed in the implementation of the algorithm. In a future section, we examine the case of iteration-dependent step-sizes.

12.3.1 Derivation of Algorithm

When $P(w)$ is first-order differentiable and strongly-convex, its unique global minimizer $w^\star$ satisfies:

$$\nabla_{w^\mathsf{T}} P(w)\Big|_{w=w^\star} = 0 \tag{12.19}$$

We are differentiating relative to w^T and not w in order to be consistent with our earlier convention from Chapter 2 that differentiation relative to a row vector results in a column vector. Equality (12.19) does not change if we scale the gradient vector by any positive scalar $\mu > 0$ and add and subtract $w^\star$, so that it also holds

$$w^\star \;=\; w^\star \;-\; \mu\,\nabla_{w^\mathsf{T}} P(w)\Big|_{w=w^\star} \tag{12.20}$$

This relation indicates that we can view the solution $w^\star$ as a *fixed point* for the mapping $f(w) : \mathbb{R}^M \rightarrow \mathbb{R}^M$ defined by

$$f(w) \;\stackrel{\Delta}{=}\; w - \mu \nabla_{w^\mathsf{T}} P(w) \tag{12.21}$$

The idea of the gradient-descent method is based on transforming the fixed-point equality (12.20) into a recursion, written as:

$$\boxed{w_n = w_{n-1} - \mu \nabla_{w^\mathsf{T}} P(w_{n-1}), \quad n \geq 0} \tag{12.22}$$

where the $w^\star$ on the left-hand side of (12.20) is replaced by w_n, while the $w^\star$ on the right-hand side of the same expression is replaced by w_{n-1}. The vectors $\{w_{n-1}, w_n\}$ represent two successive iterates that serve as estimates for $w^\star$. The scalar $\mu > 0$ is known as the *step-size* parameter and it is usually a small number. The gradient-descent algorithm is listed in (12.24). It is iterated over n until a maximum number of iterations is reached, or until the change in the weight iterate is small, or until the norm of the gradient vector is small:

$$n \leq n_{\max} \tag{12.23a}$$

$$\|w_n - w_{n-1}\|^2 \leq \epsilon, \quad \text{for some small } \epsilon \tag{12.23b}$$

$$\|\nabla_{w^\mathsf{T}} P(w_n)\| \;\leq \epsilon', \quad \text{for some small } \epsilon' \tag{12.23c}$$

Gradient-descent method for minimizing $P(w)$.

```
given gradient operator, $\nabla_{w^\mathsf{T}} P(w)$;
given a small step-size parameter $\mu > 0$;
start from an arbitrary initial condition, $w_{-1}$;          (12.24)
repeat until convergence over $n \geq 0$ :
    $w_n = w_{n-1} - \mu\,\nabla_{w^\mathsf{T}} P(w_{n-1})$
end
return $w^\star \leftarrow w_n$.
```

Recursion (12.24) starts from some initial condition, denoted by w_{-1} (usually the zero vector), and updates the iterate w_{n-1} along the negative direction of the gradient vector of $P(w)$ at w_{n-1}. The reason for the negative sign in front of μ in (12.22) is to ensure that the update to w_{n-1} is in the direction of the minimizer $w^\star$. This is because, by definition, the gradient vector of a function points in the direction towards which the function is increasing and, hence, the negative gradient points in the opposite direction. This is illustrated in Fig. 12.1. The panel on the left shows the mechanics of one update step, while the panel on the right shows the result of several successive steps.

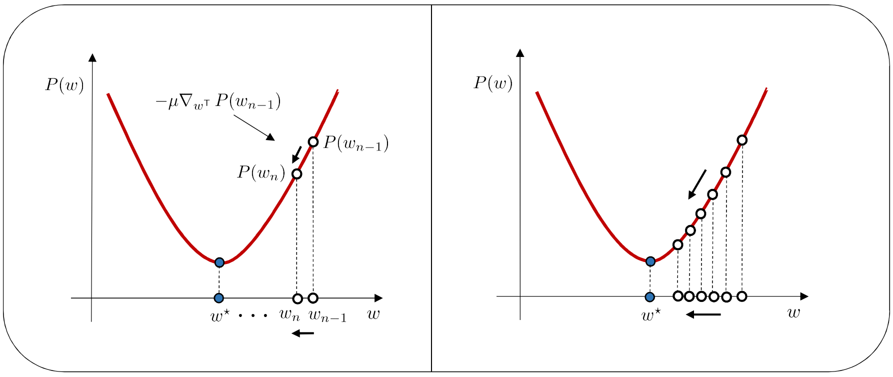
$P(w)$
$-\mu\nabla_{w^\mathsf{T}} P(w_{n-1})$
$P(w_{n-1})$
$P(w_n)$
$w^\star$ · · · w_n w_{n-1} w
$P(w)$
$w^\star$
w

Figure 12.1 The panel on the left shows the mechanics of one update step where w_{n-1} is updated in the direction of the minimizer $w^\star$. The panel on the right shows the result of several successive steps with the iterates approaching $w^\star$.

Example 12.3 **(Quadratic approximation)** There are many ways by which the gradient-descent method can be motivated. For example, we can motivate the same gradient-descent recursion (12.22) by minimizing a quadratic approximation to $P(w)$. Let w_{n-1} denote an estimate for $w^\star$ that is available at iteration $n-1$ and approximate the Hessian matrix by $\nabla_w^2 P(w_{n-1}) \approx \frac{1}{\mu} I_M$, for some $\mu > 0$. We consider a second-order expansion for $P(w)$ around w_{n-1} and pose the problem of updating w_{n-1} to w_n by solving:

$$w_n \;=\; \underset{w\in\mathbb{R}^M}{\text{argmin}} \left\{ P(w_{n-1}) + \nabla_w P(w_{n-1})(w - w_{n-1}) + \frac{1}{2\mu}\|w - w_{n-1}\|^2 \right\} \qquad (12.25)$$

Differentiating the right-hand side relative to w we find that the solution w_n is given by the relation:

$$w_n \;=\; w_{n-1} - \mu \nabla_{w^\mathsf{T}} P(w_{n-1}) \qquad (12.26)$$

Example 12.4 **(Batch gradient-descent)** If we apply the gradient-descent algorithm (12.22) to *empirical risk* functions of the form (12.2), then the gradient vector will have the form of a sample average expression:

$$\nabla_{w^{\mathsf{T}}} P(w) \;=\; \frac{1}{N}\sum_{m=0}^{N-1} \nabla_{w^{\mathsf{T}}} Q(w;\gamma(m),h_m) \tag{12.27}$$

In this case, we can be more explicit about the description of the gradient-descent method and write it in the form shown in (12.28). The reason for the designation "batch algorithm" is because each iteration of (12.28) employs the *entire* set of data, i.e., all N data pairs $\{\gamma(m),h_m\}$. Moreover, this dataset is used repeatedly until the algorithm approaches its limiting behavior. There are at least two disadvantages for these types of batch implementations:

(a) First, the entire set of N data pairs $\{\gamma(m),h_m\}$ needs to be available beforehand to be used at every iteration. For this reason, batch implementations cannot respond to streaming data, i.e., to data that arrive sequentially at every time instant.

(b) Second, the rightmost sum in (12.28) needs to be computed repeatedly at every iteration since its argument w_{n-1} is continuously changing. The computational cost can be prohibitive for large N.

Batch gradient-descent for minimizing empirical risks, $P(w)$.

```
given N data pairs $\{\gamma(m),h_m\}, m=0,1,\ldots,N-1$;
risk has empirical form $P(w)=\frac{1}{N}\sum_{m=0}^{N-1} Q(w;\gamma(m),h_m)$;
given gradient operator, $\nabla_{w^{\mathsf{T}}}Q(w;\gamma,h)$;
given a small step-size parameter $\mu>0$;                                  (12.28)
start from an arbitrary initial condition, $w_{-1}$;
repeat until convergence over $n \geq 0$ :
    $w_n = w_{n-1} - \mu\left(\frac{1}{N}\sum_{m=0}^{N-1}\nabla_{w^{\mathsf{T}}} Q(w_{n-1};\gamma(m),h_m)\right)$
end
return $w^\star \leftarrow w_n$.
```

We will explain in Chapter 16 how difficulties **(a)** and **(b)** can be addressed by resorting to *stochastic* gradient algorithms. In one implementation, the gradient sum in (12.27) is approximated by a *single* term, $\nabla_{w^{\mathsf{T}}} Q(w_{n-1},\boldsymbol{\gamma}(n),\boldsymbol{h}_n)$, where the pair $(\boldsymbol{\gamma}(n),\boldsymbol{h}_n)$ is selected at random from the dataset. In a second implementation, the gradient sum is approximated by a *mini-batch* where a small *subset* of the $N-$long data $\{\gamma(m),h_m\}$ is selected at random at every iteration and used in the update from w_{n-1} to w_n.

Example 12.5 (**Batch logistic regression**) Consider the ℓ_2-regularized logistic empirical risk from Table 12.1 along with its gradient vector:

$$P(w)=\rho\|w\|^2 \;+\; \frac{1}{N}\sum_{m=0}^{N-1}\ln\left(1+e^{-\gamma(m)h_m^{\mathsf{T}}w}\right) \tag{12.29a}$$

$$\nabla_{w^{\mathsf{T}}} P(w)=2\rho w \;-\; \frac{1}{N}\sum_{m=0}^{N-1}\frac{\gamma(m)h_m}{1+e^{\gamma(m)h_m^{\mathsf{T}}w}} \tag{12.29b}$$

The corresponding gradient-descent recursion (12.22) is given by:

$$w_n \;=\; (1-2\mu\rho)\,w_{n-1}+\mu\left(\frac{1}{N}\sum_{m=0}^{N-1}\frac{\gamma(m)h_m}{1+e^{\gamma(m)h_m^{\mathsf{T}}w_{n-1}}}\right),\quad n\geq 0 \tag{12.30}$$

The reason for the designation "logistic" is because the logistic loss in (12.29a) will arise when we study logistic regression problems in future Chapters 28 and 59. When ρ is zero, the above recursion simplifies to:

$$w_n \;=\; w_{n-1} + \mu\left(\frac{1}{N}\sum_{m=0}^{N-1}\frac{\gamma(m)h_m}{1+e^{\gamma(m)h_m^{\mathsf{T}}w_{n-1}}}\right), \quad n\geq 0 \tag{12.31}$$

Example 12.6 **(Mean-square-error stochastic risk)** Consider next an example involving a stochastic risk, say, one of the form:

$$P(w) = \rho\|w\|^2 \;+\; \mathbb{E}\,(\boldsymbol{\gamma} - \boldsymbol{h}^{\mathsf{T}}w)^2 \tag{12.32}$$

where $\boldsymbol{\gamma}\in\mathbb{R}$ and $\boldsymbol{h}\in\mathbb{R}^M$ are assumed to have zero means with second-order moments denoted by $\sigma_\gamma^2 = \mathbb{E}\,\boldsymbol{\gamma}^2$, $R_h = \mathbb{E}\,\boldsymbol{h}\boldsymbol{h}^{\mathsf{T}}$, and $r_{h\gamma} = \mathbb{E}\,\boldsymbol{h}\boldsymbol{\gamma}$. Then, it holds that

$$P(w) = \rho\|w\|^2 \;+\; \sigma_\gamma^2 - 2r_{h\gamma}^{\mathsf{T}}w + w^{\mathsf{T}}R_h w \tag{12.33a}$$

$$\nabla_{w^{\mathsf{T}}}P(w) = 2\rho w \;-2r_{h\gamma} + 2R_h w \tag{12.33b}$$

and the gradient-descent recursion (12.22) leads to

$$w_n \;=\; (1-2\mu\rho)\,w_{n-1} + 2\mu\,(r_{h\gamma} - R_h w_{n-1})\,, \quad n\geq 0 \tag{12.34}$$

Example 12.7 **(Batch least-squares)** Consider the ℓ_2-regularized least-squares empirical risk from Table 12.1 where

$$P(w) = \rho\|w\|^2 \;+\; \frac{1}{N}\sum_{m=0}^{N-1}(\gamma(m) - h_m^{\mathsf{T}}w)^2 \tag{12.35a}$$

$$\nabla_{w^{\mathsf{T}}}P(w) = 2\rho w \;-\; \frac{2}{N}\sum_{m=0}^{N-1}h_m(\gamma(m) - h_m^{\mathsf{T}}w) \tag{12.35b}$$

and the gradient-descent method reduces to

$$w_n \;=\; (1-2\mu\rho)\,w_{n-1} + 2\mu\left(\frac{1}{N}\sum_{m=0}^{N-1}h_m(\gamma(m) - h_m^{\mathsf{T}}w_{n-1})\right), \quad n\geq 0 \tag{12.36}$$

where $\mu > 0$ is a small step-size.

Example 12.8 **(Batch least-squares with offset)** In many inference problems, there will be a need to incorporate an offset parameter θ into the problem formulation. We illustrate this fact by considering a variation of the ℓ_2-regularized least-squares risk from the previous example:

$$(w^\star,\theta^\star) \;\stackrel{\Delta}{=}\; \underset{w\in\mathbb{R}^M,\theta\in\mathbb{R}}{\text{argmin}}\left\{\rho\|w\|^2 \;+\; \frac{1}{N}\sum_{m=0}^{N-1}\Big(\gamma(m) - h_m^{\mathsf{T}}w + \theta\Big)^2\right\} \tag{12.37}$$

where θ represents the scalar offset parameter. In this case, the prediction for the target γ corresponding to a feature h is computed by means of the affine relation $\widehat{\gamma} = h^{\mathsf{T}}w - \theta$. Observe that regularization is applied to w only and not to θ. We now have two parameters $\{w, \theta\}$ and, therefore,

$$P(w,\theta) = \rho\|w\|^2 + \frac{1}{N}\sum_{m=0}^{N-1}(\gamma(m) - h_m^{\mathsf{T}} w)^2 \tag{12.38a}$$

$$\nabla_{w^{\mathsf{T}}} P(w) = 2\rho w \; - \; \frac{2}{N}\sum_{m=0}^{N-1} h_m(\gamma(m) - h_m^{\mathsf{T}} w + \theta) \tag{12.38b}$$

$$\partial\, P(w,\theta)/\partial\theta = \frac{2}{N}\sum_{m=0}^{N-1}(\gamma(m) - h_m^{\mathsf{T}} w + \theta) \tag{12.38c}$$

The batch iteration (12.28) then becomes

$$\theta(n) = \theta(n-1) - 2\mu\left(\frac{1}{N}\sum_{m=0}^{N-1}\Big(\gamma(m) - h_m^{\mathsf{T}} w_{n-1} + \theta(n-1)\Big)\right) \tag{12.39a}$$

$$w_n = (1-2\mu\rho)\, w_{n-1} + 2\mu\left(\frac{1}{N}\sum_{m=0}^{N-1} h_m\Big(\gamma(m) - h_m^{\mathsf{T}} w_{n-1} + \theta(n-1)\Big)\right) \tag{12.39b}$$

We can combine the two recursions into a single relation by introducing the augmented variables of size $M+1$ each:

$$w' \;\triangleq\; \begin{bmatrix} -\theta \\ w \end{bmatrix}, \quad h' \;\triangleq\; \begin{bmatrix} 1 \\ h \end{bmatrix} \tag{12.40}$$

and writing

$$w'_n \;=\; \begin{bmatrix} 1 & \\ & (1-2\mu\rho)\, I_M \end{bmatrix} w'_{n-1} + 2\mu\left(\frac{1}{N}\sum_{m=0}^{N-1} h'_m\Big(\gamma(m) - (h'_m)^{\mathsf{T}} w'_{n-1}\Big)\right) \tag{12.41}$$

12.3.2 Convergence Analysis

The size of the step taken in (12.22) along the (negative) gradient direction is determined by μ. A small μ helps the iterates $\{w_n\}$ approach $w^\star$ in small steps, while a large μ can result in unstable behavior with the iterates bouncing back and forth around $w^\star$. Most convergence analysis specify bounds on how large μ can be to ensure the convergence of w_n to $w^\star$ as $n \to \infty$.

Theorem 12.1. (Convergence under constant step-sizes) *Consider the gradient-descent recursion (12.22) for minimizing a first-order differentiable risk function $P(w)$, where $P(w)$ is $\nu-$strongly-convex with $\delta-$Lipschitz gradients according to (12.12a)–(12.12b). Introduce the error vector $\widetilde{w}_n = w^\star - w_n$, which measures the difference between the $n-$th iterate and the global minimizer of $P(w)$. If the step-size μ satisfies (i.e., is small enough):*

$$0 < \mu < 2\nu/\delta^2 \tag{12.42}$$

then w_n and the excess risk converge exponentially fast in the following sense:

$$\|\widetilde{w}_n\|^2 \leq \lambda\, \|\widetilde{w}_{n-1}\|^2, \quad n \geq 0 \tag{12.43a}$$

$$P(w_n) - P(w^\star) \leq \frac{\delta}{2}\lambda^{n+1}\|\widetilde{w}_{-1}\|^2 \;=\; O(\lambda^n), \quad n \geq 0 \tag{12.43b}$$

where

$$\lambda \;\stackrel{\Delta}{=}\; 1 - 2\mu\nu + \mu^2\delta^2 \;\in [0,1) \tag{12.44}$$

Proof: We subtract $w^\star$ from both sides of (12.22) to get

$$\widetilde{w}_n \;=\; \widetilde{w}_{n-1} \;+\; \mu\,\nabla_{w^\mathsf{T}}\, P(w_{n-1}) \tag{12.45}$$

We compute the squared Euclidean norms (or energies) of both sides of the above equality and use the fact that $\nabla_{w^\mathsf{T}}\, P(w^\star) = 0$ to write

$$\begin{aligned}
&\|\widetilde{w}_n\|^2 \\
&= \|\widetilde{w}_{n-1}\|^2 + 2\mu\,(\nabla_{w^\mathsf{T}}\, P(w_{n-1}))^\mathsf{T}\,\widetilde{w}_{n-1} + \mu^2\,\|\nabla_{w^\mathsf{T}}\, P(w_{n-1})\|^2 \\
&= \|\widetilde{w}_{n-1}\|^2 + 2\mu\,(\nabla_{w^\mathsf{T}}\, P(w_{n-1}))^\mathsf{T}\,\widetilde{w}_{n-1} + \mu^2\,\|\nabla_{w^\mathsf{T}}\, P(w^\star) - \nabla_{w^\mathsf{T}}\, P(w_{n-1})\|^2 \\
&\overset{(12.12b)}{\leq} \;\|\widetilde{w}_{n-1}\|^2 + 2\mu\,(\nabla_{w^\mathsf{T}}\, P(w_{n-1}))^\mathsf{T}\,\widetilde{w}_{n-1} + \mu^2\delta^2\|\widetilde{w}_{n-1}\|^2
\end{aligned} \tag{12.46}$$

We appeal to the strong-convexity property (12.12a) and use $w_2 = w^\star$, $w_1 = w_{n-1}$ in step (a) below and $w_2 = w_{n-1}$, $w_1 = w^\star$ in step (b) to find that

$$\begin{aligned}
(\nabla_{w^\mathsf{T}}\, P(w_{n-1}))^\mathsf{T}\,\widetilde{w}_{n-1} &\overset{(a)}{\leq} P(w^\star) - P(w_{n-1}) - \frac{\nu}{2}\|\widetilde{w}_{n-1}\|^2 \\
&\overset{(b)}{\leq} -\frac{\nu}{2}\|\widetilde{w}_{n-1}\|^2 \;-\; \frac{\nu}{2}\|\widetilde{w}_{n-1}\|^2 \\
&= -\nu\|\widetilde{w}_{n-1}\|^2
\end{aligned} \tag{12.47}$$

Substituting into (12.46) gives

$$\|\widetilde{w}_n\|^2 \;\leq\; (1 - 2\mu\nu + \mu^2\delta^2)\,\|\widetilde{w}_{n-1}\|^2 \tag{12.48}$$

which coincides with (12.43a)–(12.44). Iterating we find that

$$\|\widetilde{w}_n\|^2 \;\leq\; \lambda^{n+1}\,\|\widetilde{w}_{-1}\|^2 \tag{12.49}$$

which highlights the exponential convergence of $\|\widetilde{w}_n\|^2$ to zero. We next verify that condition (12.42) ensures $0 \leq \lambda < 1$ using the same argument from Fig. 11.5. We plot the coefficient $\lambda(\mu)$ as a function of μ in Fig. 12.2. The minimum value of $\lambda(\mu)$ occurs at location $\mu = \nu/\delta^2$ and is equal to $1 - \nu^2/\delta^2$. This value is nonnegative since $0 < \nu \leq \delta$. It is clear from the figure that $0 \leq \lambda < 1$ for $\mu \in (0, \frac{2\nu}{\delta^2})$.

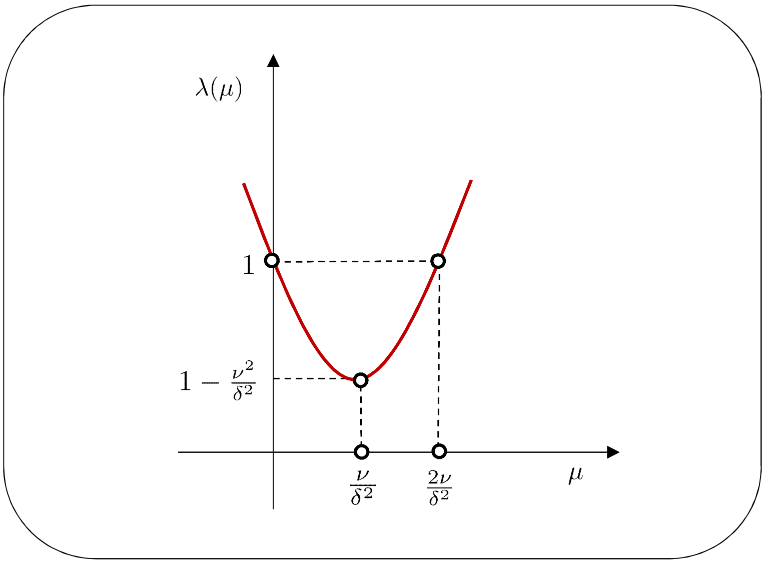

Figure 12.2 Plot of the function $\lambda(\mu) = 1 - 2\nu\mu + \mu^2\delta^2$ given by (12.44). It shows that the function $\lambda(\mu)$ assumes values below one in the range $0 < \mu < 2\nu/\delta^2$.

To establish (12.43b), we first note that $P(w_n) \geq P(w^\star)$ since $w^\star$ is the minimizer of $P(w)$. Using the upper bound (12.13b) we have

$$0 \leq P(w_n) - P(w^\star) \leq \frac{\delta}{2}\|\widetilde{w}_n\|^2 \overset{(12.48)}{\leq} \frac{\delta}{2}\lambda^{n+1}\|\widetilde{w}_{-1}\|^2 \qquad (12.50)$$

■

Remark 12.2 (**Exponential or linear convergence).** Recursions evolving according to a dynamics of the form (12.43a), such as $a(n) \leq \lambda\, a(n-1)$ for some $\lambda \in [0, 1)$, are said to converge *exponentially fast* since, by iterating, we get $a(n) \leq \lambda^{n+1} a(-1)$. This expression shows that $a(n)$ decays to zero exponentially at the rate λ^n. This mode of convergence is also referred to as *linear* convergence because, when plotted on a semi-log scale, the curve $\ln a(n) \times n$ will be linear in n with slope $\ln\lambda$, namely,

$$\ln a(n) \leq (n+1)\ln\lambda + \text{cte} \qquad (12.51)$$

■

Remark 12.3 (**Big-O, little-o, and big-Θ notation).** The statement (12.43b) uses the big-O notation. In other locations, we will employ the little-o notation. We therefore compare their meanings in this remark. We already explained in the earlier Remark 11.3 that the big-O notation is used to compare the asymptotic growth rate of two sequences. Thus, writing $a_n = O(b_n)$, with a big O for a sequence b_n with positive entries, means that there exists some constant $c > 0$ and index n_o such that $|a_n| \leq cb_n$ for all $n > n_o$. This also means that the decay rate of a_n is at least as fast or faster than b_n. For example, writing $a_n = O(1/n)$ means that the samples of the sequence a_n decay asymptotically at a rate that is comparable to or faster than $1/n$. Sometimes, one may use the big-Θ notation, which is more specific than the big-O notation in that it bounds the sequence $|a(n)|$ both from above and from below. Thus, writing $a_n = \Theta(b_n)$ now means that there exist two constants $c_1 > 0$ and $c_2 > 0$, and an index n_o, such that $c_1 b_n \leq |a_n| \leq c_2 b_n$ for all $n > n_o$. This means that the decay rate of the sequence a_n

is comparable to the decay rate of b_n. For instance, writing $a_n = \Theta(1/n)$ would now mean that the samples of the sequence a_n decay asymptotically at the rate $1/n$.

On the other hand, the little-o notation, $a_n = o(b_n)$, means that, asymptotically, the sequence a_n decays faster than the sequence b_n so that it should hold $|a_n|/b_n \to 0$ as $n \to \infty$. In this case, the notation $a_n = o(1/n)$ implies that the samples of a_n decay at a faster rate than $1/n$. Table 12.3 summarizes these definitions.

Table 12.3 Interpretation of the big-O, little-o, and big-Θ notation.

notation	interpretation
$a_n = O(b_n)$	$\lvert a_n\rvert \le cb_n,\ n > n_o$
$a_n = \Theta(b_n)$	$c_1 b_n \le \lvert a_n\rvert \le c_2 b_n,\ n > n_o$
$a_n = o(b_n)$	$\lvert a_n\rvert / b_n \to 0$ as $n \to \infty$

■

Example 12.9 **(A more relaxed bound on μ)** The result of Theorem 12.1 establishes the exponential convergence of the squared weight-error, $\|\widetilde{w}_n\|^2$, and the excess risk, $P(w_n) - P(w^\star)$, towards zero for sufficiently small step-sizes, μ. In most instances, these results are sufficient since our objective is often to verify whether the iterative algorithms approach their desired limits. This conclusion is established in Theorem 12.1 under the bound $\mu < 2\nu/\delta^2$. We can relax the result and show that convergence will continue to occur for $\mu < 2/\delta$. We do by exploiting a certain *co-coercivity* property that is satisfied by convex functions with $\delta-$Lipschitz gradients. Specifically, we know from the result of Prob. 10.4 that:

$$\Big(\nabla_{w^\mathsf{T}} P(w_2) - \nabla_{w^\mathsf{T}} P(w_1)\Big)^\mathsf{T} (w_2 - w_1) \ge \frac{1}{\delta}\,\|\nabla_w P(w_2) - \nabla_w P(w_1)\|^2 \tag{12.52}$$

We use this inequality in (12.46) as follows:

$$\begin{aligned}
&\|\widetilde{w}_n\|^2\\
&= \|\widetilde{w}_{n-1}\|^2 - 2\mu\Big(\nabla_{w^\mathsf{T}} P(w^\star) - \nabla_{w^\mathsf{T}} P(w_{n-1})\Big)^\mathsf{T}\widetilde{w}_{n-1} + \mu^2\,\|\nabla_{w^\mathsf{T}} P(w_{n-1})\|^2\\
&\overset{(12.52)}{\le} \|\widetilde{w}_{n-1}\|^2 - 2\mu\Big(\nabla_{w^\mathsf{T}} P(w^\star) - \nabla_{w^\mathsf{T}} P(w_{n-1})\Big)^\mathsf{T}\widetilde{w}_{n-1} +\\
&\qquad\quad + \mu^2\delta\Big(\nabla_{w^\mathsf{T}} P(w^\star) - \nabla_{w^\mathsf{T}} P(w_{n-1})\Big)^\mathsf{T}\widetilde{w}_{n-1}\\
&= \|\widetilde{w}_{n-1}\|^2 - (2\mu - \mu^2\delta)\Big(\nabla_{w^\mathsf{T}} P(w^\star) - \nabla_{w^\mathsf{T}} P(w_{n-1})\Big)^\mathsf{T}\widetilde{w}_{n-1}\\
&= \|\widetilde{w}_{n-1}\|^2 + (2\mu - \mu^2\delta)(\nabla_{w^\mathsf{T}} P(w_{n-1}))^\mathsf{T}\widetilde{w}_{n-1}\\
&\overset{(12.47)}{\le} \|\widetilde{w}_{n-1}\|^2 - (2\mu - \mu^2\delta)\nu\|\widetilde{w}_{n-1}\|^2\\
&= \underbrace{(1 - 2\mu\nu + \mu^2\nu\delta)}_{\triangleq\,\lambda'}\|\widetilde{w}_{n-1}\|^2
\end{aligned} \tag{12.53}$$

This result is consistent with (12.48) since $\lambda' \le \lambda$ in view of $\nu \le \delta$. Working with λ', we obtain the bound $0 < \mu < 2/\delta$ for stability with convergence occurring at $O((\lambda')^n)$.

Example 12.10 **(Convergence analysis based on excess-risk)** The convergence analysis used to establish Theorem 12.1 was based on examining the evolution of the squared error, $\|\widetilde{w}_n\|^2$, and from there we were able to conclude how the excess risk term evolves with time. We will adopt this approach uniformly throughout our presentation. However, we remark here that we can arrive at similar conclusions by working directly with the risk function. To do so, we exploit two properties of the risk function: its strong convexity and the fact that it has Lipschitz gradients. These properties were shown before to induce certain bounds on the risk.

For instance, using the $\nu-$strong convexity of $P(w)$, we use property (8.29) to deduce that

$$P(w^\star) \geq P(w_{n-1}) - \frac{1}{2\nu}\|\nabla_w P(w_{n-1})\|^2 \tag{12.54}$$

On the other hand, from the $\delta-$Lipschitz property on the gradients of $P(w)$, we use result (10.13) to write

$$\begin{aligned} P(w_n) \quad &\leq \quad P(w_{n-1}) + (\nabla_w P(w_{n-1}))\,(w_n - w_{n-1}) + \frac{\delta}{2}\|w_n - w_{n-1}\|^2 \\ &\overset{(12.22)}{=} P(w_{n-1}) - \mu\Big(1 - \frac{\mu\delta}{2}\Big)\|\nabla_w P(w_{n-1})\|^2 \end{aligned} \tag{12.55}$$

Subtracting $P(w^\star)$ from both sides of this inequality and using (12.54) we obtain

$$P(w_n) - P(w^\star) \leq \underbrace{(1 - 2\mu\nu + \mu^2\nu\delta)}_{\lambda'}\Big(P(w_{n-1}) - P(w^\star)\Big) \tag{12.56}$$

This result is consistent with (12.53) and convergence again occurs for $0 < \mu < 2/\delta$ at the rate $O((\lambda')^n)$.

Regret analysis

The statement of Theorem 12.1 examines the convergence behavior of the squared weight error, $\|\widetilde{w}_n\|^2$, and the risk value $P(w_n)$. Another common performance measure for learning algorithms is the *average regret*. It is defined over a window of N iterations and computes the deviation of the accumulated risk relative to the minimal risk:

$$\boxed{\mathcal{R}(N) \;\triangleq\; \frac{1}{N}\sum_{n=0}^{N-1} P(w_{n-1}) - P(w^\star)} \qquad \textbf{(average regret)} \tag{12.57}$$

The sum involves all risk values over the first N iterations. Using (12.43b) we find that the regret decays at the rate of $1/N$ since

$$\begin{aligned} \mathcal{R}(N) &\leq \frac{1}{N}\frac{\delta\|\widetilde{w}_{-1}\|^2}{2}\sum_{n=0}^{N-1}\lambda^{n+1} \\ &= \frac{1}{N}\frac{\delta\|\widetilde{w}_{-1}\|^2}{2}\frac{(1-\lambda^N)\lambda}{1-\lambda} \\ &= O(1/N) \end{aligned} \tag{12.58}$$

This calculation shows that we can transform bounds on the excess risk $P(w_n) - P(w^\star)$ into bounds on the average regret. For this reason, we will continue to derive excess risk bounds throughout our analysis of learning algorithms, with the understanding that they can be easily transformed into regret bounds.

Remark 12.4. (Regret analysis and convexity) There is another useful bound for the average regret for convex risk functions. Using property (8.4) we have

$$P(w_{n-1}) - P(w^\star) \leq -(\nabla_{w^\mathsf{T}} P(w_{n-1}))^\mathsf{T} \widetilde{w}_{n-1} \tag{12.59}$$

so that we can bound (12.57) by

$$\mathcal{R}(N) \leq -\frac{1}{N} \sum_{n=0}^{N-1} (\nabla_{w^\mathsf{T}} P(w_{n-1}))^\mathsf{T} \widetilde{w}_{n-1}, \quad (\textbf{for convex risks}) \tag{12.60}$$

For this reason, it is also customary to study $\mathcal{R}(N)$ by bounding the inner product $(\nabla_{w^\mathsf{T}} P(w_{n-1}))^\mathsf{T} \widetilde{w}_{n-1}$ and its cumulative sum. Examples to this effect will be encountered later in Sec. 16.5 and Appendix 17.A in the context of stochastic optimization algorithms.

■

Example 12.11 **(Dependence of convergence on problem dimension)** Result (12.58) may suggest at first sight that the regret bound is not dependent on the parameter dimension, M. However, the bound is scaled by the Lipschitz constant δ and this constant is implicitly dependent on M. This is because the value of δ depends on the norm used in (12.12b). For most of our treatment, we will be working with the Euclidean norm, but there are important cases where the gradient Lipschitz property will hold for other norms. For example, assume for the sake of argument that the gradients of the risk function $P(w)$ happen to be $\delta-$Lipschitz relative to some other norm, such as the $\ell_\infty-$norm. In this case, expression (12.12b) will be replaced by

$$\|\nabla_w P(w_2) - \nabla_w P(w_1)\|_\infty \;\leq\; \delta \, \|w_2 - w_1\|_\infty \tag{12.61}$$

Using the norm inequalities:

$$\|x\|_2 \leq \sqrt{M}\, \|x\|_\infty, \quad \|x\|_\infty \leq \|x\|_2 \tag{12.62}$$

for any $x \in \mathbb{R}^M$, relation (12.61) can be transformed into an inequality involving the ℓ_2-norm, as in (12.12b):

$$\|\nabla_w P(w_2) - \nabla_w P(w_1)\| \;\leq\; \sqrt{M}\delta \, \|w_2 - w_1\| \tag{12.63}$$

with a new δ value that is scaled by $\sqrt{M}$. If we were to write the regret and performance bounds derived so far in the chapter using this new δ, then the results will be scaled by $\sqrt{M}$ and become dependent on the problem dimension. This fact is problematic for large dimensional inference problems. Later, in Sec. 15.3, we will motivate the *mirror-descent* algorithm, which addresses this problem for a class of constrained optimization problems and leads to performance bounds that are independent of M.

Convexity versus strong-convexity

The statement of Theorem 12.1 assumes *strongly-convex* risk functions $P(w)$ with $\delta-$Lipschitz gradients satisfying (12.12a)–(12.12b). The theorem establishes in (12.43a) the exponential convergence of $\|\widetilde{w}_n\|^2$ to zero at the rate λ^n. It also establishes in (12.43b) that $P(w_n)$ converges exponentially at the same rate to $P(w^\star)$. We will express these conclusions by adopting the following notation:

$$\begin{cases} \|\widetilde{w}_n\|^2 & \leq & O(\lambda^n), \\ P(w_n) - P(w^\star) & \leq & O(\lambda^n), \end{cases} \quad (\textbf{for strongly-convex } P(w)) \tag{12.64a}$$

This means that $O(\ln(1/\epsilon))$ iterations are needed for the risk value $P(w_n)$ to get $\epsilon-$close to $P(w^\star)$. We will be dealing largely with strongly-convex risks $P(w)$, especially since regularization will ensure strong convexity in many cases of interest. Nevertheless, when $P(w)$ happens to be only convex (but not necessarily strongly-convex) then, following an argument similar to the derivation of (11.71), we can establish that convergence in this case will be *sublinear* (rather than linear). Specifically, it will hold for $\mu < 1/\delta$ that the successive risk values approach the minimum value at the slower rate of $1/n$:

$$P(w_n) - P(w^\star) \;\leq\; O(1/n), \qquad (\textbf{for convex } P(w)) \tag{12.64b}$$

This result is established in Prob. 12.13. In this case, $O(1/\epsilon)$ iterations will be needed for the risk value $P(w_n)$ to get $\epsilon-$close to $P(w^\star)$.

12.4 ITERATION-DEPENDENT STEP-SIZES

Although recursion (12.22) employs a *constant* step-size μ, one can also consider iteration-dependent step-sizes, denoted by $\mu(n)$, and write:

$$\boxed{w_n = w_{n-1} - \mu(n)\nabla_{w^\mathsf{T}} P(w_{n-1}), \quad n \geq 0} \tag{12.65}$$

The ability to vary the step-size with n provides an opportunity to control the size of the gradient step, for example, by using larger steps during the initial stages of learning and smaller steps later. There are several ways by which the step-size sequence $\mu(n)$ can be selected.

12.4.1 Vanishing Step-Sizes

The convergence analysis in the next Theorem 12.2 assumes step-sizes that satisfy either one of the following two conditions:

$$(\textbf{condition I}) \qquad \sum_{n=0}^{\infty} \mu^2(n) < \infty \quad \text{and} \quad \sum_{n=0}^{\infty} \mu(n) = \infty \tag{12.66a}$$

$$(\textbf{condition II}) \qquad \lim_{n\to\infty} \mu(n) = 0 \quad \text{and} \quad \sum_{n=0}^{\infty} \mu(n) = \infty \tag{12.66b}$$

Clearly, any sequence that satisfies (12.66a) also satisfies (12.66b). In either case, the step-size sequence vanishes asymptotically but the rate of decay of $\mu(n)$ to zero should not be too fast (so that the sequence is not absolutely summable). For example, step-size sequences of the form:

$$\mu(n) = \frac{\tau}{(n+1)^c}, \quad \text{for any } \tau > 0 \text{ and } \tfrac{1}{2} < c \leq 1 \tag{12.67}$$

satisfy (12.66b). The choice $c = 1$ is common. There are other choices for $\mu(n)$, besides sequences that satisfy (12.66a) or (12.66b), that can ensure convergence of the gradient-descent method. We will illustrate this fact further ahead when we examine *backtracking* in Sec. 12.4.2. There, we will introduce another sufficient requirement on $\mu(n)$ to guarantee convergence known as the *Armijo condition*. Other examples of convergent gradient-descent methods with iteration-dependent step-sizes include the alternating projection algorithm from Sec. 12.6 and Kaczmarz's method from Prob. 12.34. For now, we continue with the popular conditions (12.66a)– (12.66b). The following result shows that the convergence rate is not exponential any longer and is slower than under constant step-sizes.

THEOREM 12.2. (Convergence under vanishing step-sizes) *Consider the gradient-descent recursion (12.65) for minimizing a first-order differentiable risk function $P(w)$, where $P(w)$ is $\nu-$strongly-convex with $\delta-$Lipschitz gradients according to (12.12a)–(12.12b). If the step-size sequence $\mu(n)$ satisfies either (12.66a) or (12.66b), then w_n converges to the global minimizer, $w^\star$. In particular, when the step-size sequence is chosen as $\mu(n) = \tau/(n+1)$, the convergence rate is on the order of*

$$\|\widetilde{w}_n\|^2 \leq O(1/n^{2\nu\tau}) \tag{12.68a}$$

$$P(w_n) - P(w^\star) \leq O(1/n^{2\nu\tau}) \tag{12.68b}$$

for large enough n.

Proof: The argument that led to (12.48) will similarly lead to

$$\|\widetilde{w}_n\|^2 \leq \lambda(n)\,\|\widetilde{w}_{n-1}\|^2 \tag{12.69}$$

where now $\lambda(n) = 1 - 2\nu\mu(n) + \delta^2\mu^2(n)$. We split $2\nu\mu(n)$ into the sum of two factors and write

$$\lambda(n) = 1 - \nu\mu(n) - \nu\mu(n) + \delta^2\mu^2(n) \tag{12.70}$$

Now, since $\mu(n) \to 0$ under (12.66a) or (12.66b), we conclude that for large enough $n > n_o$, the value of $\mu^2(n)$ will be smaller than $\mu(n)$. Therefore, a large enough time index, n_o, exists such that the following two conditions are satisfied:

$$\nu\mu(n) \geq \delta^2\mu^2(n), \qquad 0 < 1 - \nu\mu(n) \leq 1, \qquad n > n_o \tag{12.71}$$

It follows that

$$\lambda(n) \;\leq\; 1 - \nu\mu(n), \qquad n > n_o \tag{12.72}$$

and, hence,

$$\|\widetilde{w}_n\|^2 \;\leq\; (1 - \nu\mu(n))\,\|\widetilde{w}_{n-1}\|^2, \qquad n > n_o \tag{12.73}$$

Iterating over n we can write (assuming a finite n_o exists for which $\|\widetilde{w}_{n_o}\| \neq 0$, otherwise the algorithm would have converged):

$$\lim_{n\to\infty} \left(\frac{\|\widetilde{w}_n\|^2}{\|\widetilde{w}_{n_o}\|^2} \right) \leq \prod_{n=n_o+1}^{\infty} (1 - \nu\mu(n)) \tag{12.74}$$

or, equivalently,

$$\lim_{n\to\infty} \ln \left(\frac{\|\widetilde{w}_n\|^2}{\|\widetilde{w}_{n_o}\|^2} \right) \leq \sum_{n=n_o+1}^{\infty} \ln (1 - \nu\mu(n)) \tag{12.75}$$

Now, using the following property for the natural logarithm function:

$$\ln(1-y) \;\leq\; -y, \quad \text{for all } 0 \leq y < 1 \tag{12.76}$$

and letting $y = \nu\mu(n)$, we have that

$$\ln(1 - \nu\mu(n)) \;\leq\; -\nu\mu(n), \quad n > n_o \tag{12.77}$$

so that

$$\sum_{n=n_o+1}^{\infty} \ln(1-\nu\mu(n)) \;\leq\; -\sum_{n=n_o+1}^{\infty} \nu\mu(n) \;=\; -\nu \left(\sum_{n=n_o+1}^{\infty} \mu(n) \right) \;=\; -\infty \tag{12.78}$$

since the step-size series is assumed to be divergent under (12.66a) or (12.66b) . We conclude that

$$\lim_{n\to\infty} \ln \left(\frac{\|\widetilde{w}_n\|^2}{\|\widetilde{w}_{n_o}\|^2} \right) = -\infty \tag{12.79}$$

so that $\widetilde{w}_n \to 0$ as $n \to \infty$.

We next examine the rate at which this convergence occurs for step-size sequences of the form $\mu(n) = \tau/(n+1)$. Note first that these sequences satisfy the following two conditions

$$\sum_{n=0}^{\infty} \mu(n) = \infty, \qquad \sum_{n=0}^{\infty} \mu^2(n) \;=\; \tau^2 \left(\sum_{n=1}^{\infty} \frac{1}{n^2} \right) \;=\; \beta\tau^2 < \infty \tag{12.80}$$

for $\beta = \pi^2/6$. Again, since $\mu(n) \to 0$ and $\mu^2(n)$ decays faster than $\mu(n)$, we know that for some large enough $n > n_1$, it will hold that

$$2\nu\mu(n) \geq \delta^2\mu^2(n) \tag{12.81}$$

and, hence,

$$0 < \lambda(n) \leq 1, \quad n > n_1 \tag{12.82}$$

We can now repeat the same steps up to (12.79) using $y = 2\nu\mu(n) - \delta^2\mu^2(n)$ to conclude that

$$\begin{aligned}
\ln\left(\frac{\|\widetilde{w}_n\|^2}{\|\widetilde{w}_{n_1}\|^2}\right) &\leq \sum_{m=n_1+1}^{n} \ln\left(1-2\nu\mu(m)+\delta^2\mu^2(m)\right) \\
&\leq -\sum_{m=n_1+1}^{n} \left(2\nu\mu(m)-\delta^2\mu^2(m)\right) \\
&= -2\nu\left(\sum_{m=n_1+1}^{n}\mu(m)\right)+\delta^2\left(\sum_{m=n_1+1}^{n}\mu^2(m)\right) \\
&\leq -2\nu\left(\sum_{m=n_1+1}^{n}\mu(m)\right)+\beta\tau^2\delta^2 \\
&= -2\nu\tau\left(\sum_{m=n_1+2}^{n+1}\frac{1}{m}\right)+\beta\tau^2\delta^2 \\
&\overset{(a)}{\leq} -2\nu\tau\left(\int_{n_1+2}^{n+2}\frac{1}{x}dx\right)+\beta\tau^2\delta^2 \\
&= 2\nu\tau\ln\left(\frac{n_1+2}{n+2}\right)+\beta\tau^2\delta^2 \\
&= \ln\left(\frac{n_1+2}{n+2}\right)^{2\nu\tau}+\beta\tau^2\delta^2
\end{aligned} \tag{12.83}$$

where in step (a) we used the following integral bound, which reflects the fact that the area under the curve $f(x)=1/x$ over the interval $x\in[n_1+2,n+2]$ is upper bounded by the sum of the areas of the rectangles shown in Figure 12.3:

$$\int_{n_1+2}^{n+2}\frac{1}{x}dx \;\leq\; \sum_{m=n_1+2}^{n+1}\frac{1}{m} \tag{12.84}$$

We conclude from (12.83) that

$$\begin{aligned}
\|\widetilde{w}_n\|^2 &\leq \left\{e^{\left(\ln\left(\frac{n_1+2}{n+2}\right)^{2\nu\tau}+\beta\tau^2\delta^2\right)}\right\}\|\widetilde{w}_{n_1}\|^2, \quad i>i_1 \\
&= e^{\beta\tau^2\delta^2}\,\|\widetilde{w}_{n_1}\|^2\left(\frac{n_1+2}{n+2}\right)^{2\nu\tau} \\
&= O(1/n^{2\nu\tau})
\end{aligned} \tag{12.85}$$

as claimed. Result (12.68b) follows by noting from (12.50) that

$$0\leq P(w_n)-P(w^\star)\leq\frac{\delta}{2}\|\widetilde{w}_n\|^2 \tag{12.86}$$

■

Convexity versus strong-convexity

The statement of Theorem 12.2 assumes *strongly-convex* risks $P(w)$ with $\delta-$Lipschitz gradients. In Prob. 12.15 we relax these conditions and limit $P(w)$ to being convex (as opposed to strongly-convex) and Lipschitz as opposed to gradient-Lipschitz, i.e.,

$$\|P(w_1)-P(w_2)\|\leq\delta\|w_1-w_2\|, \quad \forall\, w_1,w_2\in\text{dom}(P) \tag{12.87}$$

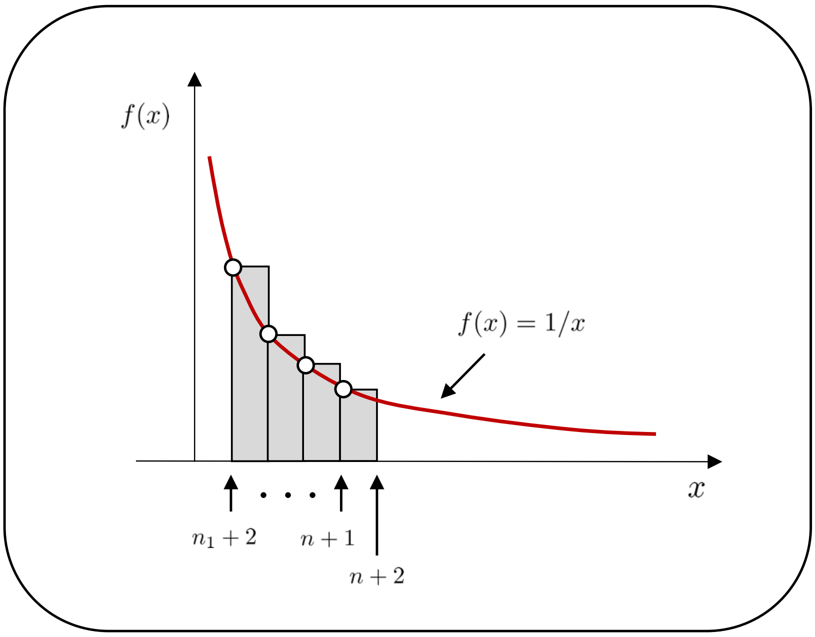

Figure 12.3 The area under the curve $f(x) = 1/x$ over the interval $x \in [n_1 + 2, n + 2]$ is upper bounded by the sum of the areas of the rectangles shown in the figure.

We know from property (10.41) that the condition of a Lipschitz function translates into bounded gradient vectors, so that we are in effect requiring $\|\nabla_w P(w)\| \leq \delta$. Assume we run the gradient-descent recursion (12.65) for N iterations using a decaying step-size sequence of the form $\mu(n) = c/\sqrt{n+1}$ for some positive constant c, and let w^{best} denote the iterate that results in the smallest risk value, namely,

$$w^{\text{best}} \stackrel{\Delta}{=} \underset{0 \leq n \leq N-1}{\text{argmin}} \; P(w_n) \tag{12.88}$$

Then, we show in Prob. 12.15 that

$$P(w^{\text{best}}) - P(w^\star) = O\Big(\ln(N)/N\Big) \tag{12.89}$$

Example 12.12 **(Steepest-descent algorithm)** We motivate another choice for the step-size sequence $\mu(n)$ by seeking the steepest-descent direction along which the update of w_{n-1} should be performed. We use the same reasoning from Example 6.13, which dealt with the line search method.

Starting from an iterate w_{n-1}, our objective is to determine a small adjustment to it, say, $w_n = w_{n-1} + \delta w$, by solving

$$\delta w^o = \underset{\delta w \in \mathbb{R}^M}{\text{argmin}} \Big\{ P(w_{n-1} + \delta w) \Big\}, \quad \text{subject to } \frac{1}{2}\|\delta w\|^2 \leq \epsilon \tag{12.90}$$

We introduce a Lagrange multiplier $\lambda \geq 0$ and consider the unconstrained formulation

$$\delta w^o = \operatorname*{argmin}_{\delta w \in \mathbb{R}^M} \left\{ P(w_{n-1} + \delta w) \; + \; \lambda\Big(\frac{1}{2}\|\delta w\|^2 - \epsilon\Big) \right\} \tag{12.91}$$

To solve the problem, we introduce the *first-order* Taylor series expansion:

$$P(w_n) \approx P(w_{n-1}) + \nabla_w P(w_{n-1})\delta w \tag{12.92}$$

so that the cost appearing in (12.91) is approximated by

$$\text{cost} \approx P(w_{n-1}) + \nabla_w P(w_{n-1})\delta w + \lambda\Big(\frac{1}{2}\|\delta w\|^2 - \epsilon\Big) \tag{12.93}$$

To minimize the right-hand side over δw, and to find λ, we repeat the argument from Example 6.13 to arrive at the same conclusion:

$$w_n = w_{n-1} - \underbrace{\frac{\sqrt{2\epsilon}}{\|\nabla_{w^{\mathsf{T}}} P(w_{n-1})\|}}_{\triangleq\, \mu(n)} \nabla_{w^{\mathsf{T}}} P(w_{n-1}) \tag{12.94}$$

The term multiplying $\nabla_{w^{\mathsf{T}}} P(w_{n-1})$ plays the role of an iteration-dependent step-size. In this case, the step-size is chosen to result in the "largest" descent possible per iteration.

Example 12.13 **(Comparing constant and vanishing step-sizes)** We return to the logistic algorithm (12.31) and simulate its performance under both constant and vanishing step-sizes. Figure 12.4 plots a learning curve for the algorithm using parameters

$$\rho = 2, M = 10, N = 200, \quad \mu = 0.001 \tag{12.95}$$

For this simulation, the data $\{\gamma(m), h_m\}$ are generated randomly as follows. First, a random parameter model $w^a \in \mathbb{R}^{10}$ is selected, and a random collection of feature vectors $\{h_m\}$ are generated, say, with zero-mean and unit-variance Gaussian entries. Then, for each h_m, the label $\gamma(m)$ is set to either $+1$ or -1 according to the following construction:

$$\gamma(m) = +1 \;\text{ if }\; \left(\frac{1}{1 + e^{-h_m^{\mathsf{T}} w^a}}\right) \geq 0.5; \;\text{ otherwise } \gamma(m) = -1 \tag{12.96}$$

We will explain in future expression (59.5a) that construction (12.96) amounts to generating data $\{\gamma(m), h_m\}$ that satisfy a logistic probability model. The gradient-descent recursion (12.30) is run for 2000 iterations on the data $\{\gamma(m), h_m\}$. The resulting weight iterate, denoted by $w^\star$, is shown in the bottom plot of the figure and the value of the risk function at this weight iterate is found to be

$$P(w^\star) \approx 0.6732 \tag{12.97}$$

The two plots in the top row display the learning curve $P(w_n)$ relative to the minimum value $P(w^\star)$, both in linear scale (on the left) and in normalized logarithmic scale on the right (according to construction (11.65)). The plot on the right in the top row reveals the linear convergence of $P(w_n)$ towards $P(w^\star)$ under constant step-sizes, as anticipated by result (12.43b).

Figure 12.5 repeats the simulation using the same logistic data $\{\gamma(m), h_m\}$ albeit with a decaying step-size sequence of the form:

$$\mu(n) = \tau/(n+1), \quad \tau = 0.1 \tag{12.98}$$

The gradient-descent recursion (12.65) is now repeated for 4000 iterations with μ replaced by $\mu(n)$, and the resulting learning curve is compared against the curve generated under the constant step-size regime from the previous simulation. The plot on the left

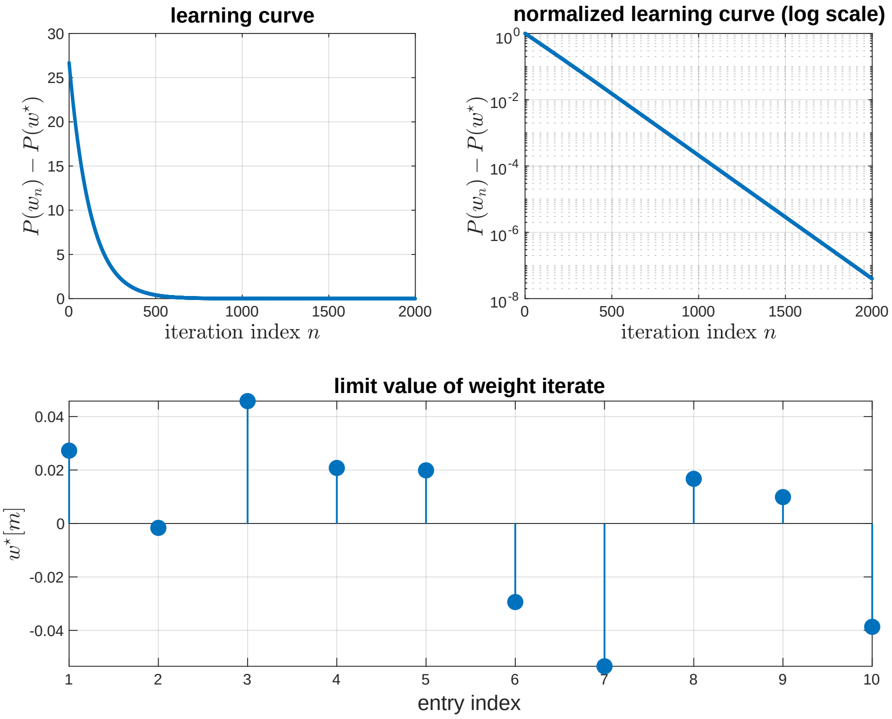

Figure 12.4 (*Top*) Learning curves $P(w_n)$ relative to the minimum risk value $P(w^\star)$ in linear scale (on the left) and in normalized logarithmic scale (on the right). This latter plot confirms the linear convergence of the risk value towards $P(w^\star)$. (*Bottom*) Limiting value of the weight iterate w_n, which tends to the minimizer $w^\star$ according to result (12.43a).

shows the learning curves in normalized logarithmic scale; it is clear that the convergence rate under decaying step-sizes is much slower (it starts converging faster but ultimately becomes slower). The plot on the right illustrates this effect; it shows the limiting value $w^\star$ that was determined under constant step-size learning in Fig. 12.4 after 2000 iterations along with the weight iterate that is obtained under the decaying step-size after 4000 iterations. It is clear that convergence has not been attained yet in the latter case, and many more iterations would be needed; this is because $\mu(n)$ becomes vanishingly small as n increases.

12.4.2 Backtracking Line Search

There are other methods to select the step-size sequence $\mu(n)$, besides (12.66a) or (12.66b). One method is the *backtracking line search* technique. Recall that the intent is to move from w_{n-1} to w_n in a manner that reduces the risk function at $w_{n-1} + \delta w$; it is also desirable to take "larger" steps when possible. Motivated by these considerations, at every iteration n, the backtracking method runs a

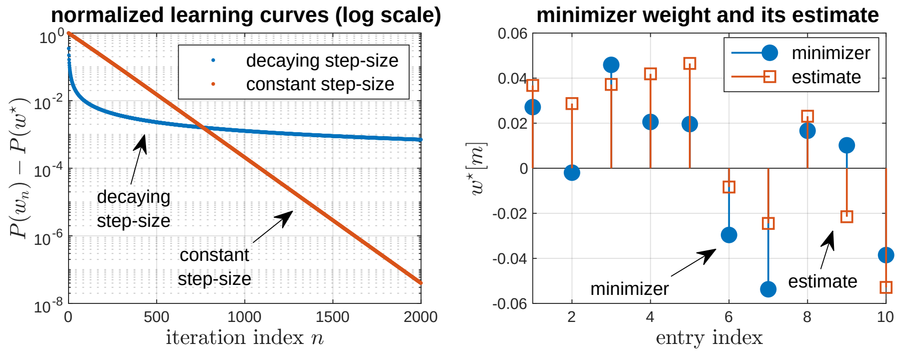

Figure 12.5 (*Left*) Learning curves $P(w_n)$ relative to the minimum risk value $P(w^\star)$ in normalized logarithmic scale for both cases of constant and decaying step-sizes. (*Right*) After 4000 iterations, the weight iterate w_n in the decaying step-size implementation has not converged yet.

separate search to select $\mu(n)$ for that iteration. It starts from some large initial value and repeatedly shrinks it until a convenient value is found. The procedure is motivated as follows.

Starting from the iterate w_{n-1}, we introduce a first-order Taylor series approximation for the risk function around w_{n-1}:

$$P(w) \approx P(w_{n-1}) + \nabla_w P(w_{n-1})(w - w_{n-1}) \tag{12.99}$$

If we take a gradient-descent step from w_{n-1} to w_n with some generic step-size value μ, i.e.,

$$w_n = w_{n-1} - \mu \nabla_{w^\mathsf{T}} P(w_{n-1}) \tag{12.100}$$

then, substituting into (12.99), the new risk value is approximately

$$P(w_n) \approx P(w_{n-1}) - \mu \left\|\nabla_w P(w_{n-1})\right\|^2 \tag{12.101}$$

The backtracking line search method selects μ to ensure that the decrease in the risk value is at least a fraction of the amount suggested above, say,

$$P\Big(w_{n-1} - \mu\,\nabla_{w^\mathsf{T}}\, P(w_{n-1})\Big) \;\leq\; P(w_{n-1}) - \alpha\mu\|\nabla_w\, P(w_{n-1})\|^2 \tag{12.102}$$

for some $0 < \alpha \leq \frac{1}{2}$. This is achieved as shown in listing (12.103). Typical values for the parameters are $\beta = 0.2$, $\alpha = 0.01$, and $\mu_0 = 1$.

Gradient-descent with backtracking line search for minimizing $P(w)$.

```
given gradient operator, $\nabla_{w^\mathsf{T}} P(w)$;
given $0 < \beta < 1$, $0 < \alpha < 1/2$, $\mu_0 > 0$;
start from an arbitrary initial condition, $w_{-1}$;
repeat until convergence over $n \geq 0$ :
    $j = 0$;
    while $P\Big(w_{n-1} - \mu_j \nabla_{w^\mathsf{T}} P(w_{n-1})\Big) > P(w_{n-1}) - \alpha\mu_j \|\nabla_w P(w_{n-1})\|^2$
        shrink step-size to $\mu_{j+1} = \beta\mu_j$;
        $j \leftarrow j + 1$;
    end
    set $\mu(n) = \mu_j$;
    $w_n = w_{n-1} - \mu(n)\nabla_{w^\mathsf{T}}\, P(w_{n-1})$;
end
return $w^\star \leftarrow w_n$.
```

(12.103)

Once $\mu(n)$ is selected at step n (say, $\mu(n) = \mu$ for some value μ), the risk function will satisfy

$$P(w_n) \leq P(w_{n-1}) - \alpha\mu\|\nabla_w P(w_{n-1})\|^2 \tag{12.104}$$

This result is known as the *Armijo condition*, which is usually stated in the following more abstract form. Consider an update step δw and introduce the function:

$$\phi(\mu) \overset{\Delta}{=} P(w + \mu\,\delta w) \tag{12.105}$$

where μ is some step-size parameter that we wish to determine in order to update w to $w + \mu\,\delta w$. The Armijo condition chooses μ to satisfy:

$$\boxed{\begin{array}{l}(\textbf{Armijo condition})\\ \phi(\mu) \leq \phi(0) + \alpha\mu\,\phi'(0), \quad \text{for some } 0 < \alpha < 1/2\end{array}} \tag{12.106}$$

where $\phi'(\mu)$ denotes the derivative of $\phi(\mu)$ relative to μ. It is easy to verify that this condition reduces to (12.104) for the case of the gradient-descent algorithm where $\delta w = -\nabla_{w^\mathsf{T}} P(w)$. The step-sizes $\mu(n)$ that result from the backtracking procedure (12.103) satisfy the above Armijo condition at every step, n. We show next that the Armijo condition is sufficient to ensure convergence of the gradient-descent algorithm.

Theorem 12.3. (Convergence under Armijo condition) *Consider the gradient-descent recursion (12.65) for minimizing a first-order differentiable risk function $P(w)$, where $P(w)$ is $\nu-$strongly-convex with $\delta-$Lipschitz gradients according to (12.12a)–(12.12b). If the step-size sequence $\mu(n)$ is chosen to satisfy the Armijo condition (12.104) at every step, then the excess risk converges exponentially fast, namely,*

$$P(w_n) - P(w^\star) \leq O(\lambda^n), \quad (\textit{for strongly-convex } P(w)) \tag{12.107}$$

for some $\lambda \in [0,1)$. If $P(w)$ is only convex, then

$$P(w_n) - P(w^\star) \leq O(1/n), \quad (\textit{for convex } P(w)) \tag{12.108}$$

Proof: First, we call upon property (10.13) for the $\delta-$Lipschitz gradient of $P(w)$, which allows us to write (using $z \leftarrow w_n$, $z_1 \leftarrow w_{n-1}$, and $z - z_1 \leftarrow -\mu_j \nabla_{w^\mathsf{T}} P(w_{n-1})$):

$$P\Big(w_{n-1} - \mu_j \nabla_{w^\mathsf{T}} P(w_{n-1})\Big) \leq P(w_{n-1}) - \mu_j\Big(1 - \frac{\delta\mu_j}{2}\Big)\|\nabla_w\, P(w_{n-1}\|^2 \tag{12.109}$$

According to the backtracking construction (12.103), the search for the step-size parameter will stop when

$$P\Big(w_{n-1} - \mu_j \nabla_{w^\mathsf{T}} P(w_{n-1})\Big) \;\leq P(w_{n-1}) - \alpha\mu_j\|\nabla_w P(w_{n-1})\|^2 \tag{12.110}$$

Combining with (12.109), we find that for the search to stop it is sufficient to require

$$\mu_j\Big(1 - \frac{\delta\mu_j}{2}\Big) > \alpha\mu_j \tag{12.111}$$

Since, by choice, $\alpha < 1/2$, the search is guaranteed to stop when

$$\mu_j\Big(1 - \frac{\delta\mu_j}{2}\Big) > \frac{1}{2}\mu_j \iff \mu_j < 1/\delta \tag{12.112}$$

This argument shows that the exit condition for the backtracking construction will be satisfied whenever $\mu_j < 1/\delta$. Using this condition in (12.109), and noting that the argument of $P(\cdot)$ on the left-hand side becomes w_n at the exit point, we find that at that point:

$$P(w_n) \leq P(w_{n-1}) - \frac{\mu_j}{2}\|\nabla_w\, P(w_{n-1}\|^2 \tag{12.113}$$

On the other hand, using the $\nu-$strong convexity of $P(w)$, we apply the upper bound from property (8.29) to deduce that:

$$P(w^\star) \geq P(w_{n-1}) - \frac{1}{2\nu}\|\nabla_w\, P(w_{n-1})\|^2 \tag{12.114}$$

Subtracting $P(w^\star)$ from both sides of inequality (12.113) and using (12.114) we obtain

$$P(w_n) - P(w^\star) \leq (1 - \mu_j\nu)\Big(P(w_{n-1}) - P(w^\star)\Big) \tag{12.115}$$

Now recall that we launch the backtracking search from the initial condition $\mu_0 = 1$. Two scenarios are possible: either $1/\delta > 1$ or $1/\delta \leq 1$. In the first case, the backtracking search will stop right away at $\mu_j = \mu_0 = 1$ since the condition $\mu_j < 1/\delta$ will be met. In the second case, the step-size will be scaled down repeatedly by β until the first time it goes below $1/\delta$, at which point the search stops. In this case, the final μ_j will satisfy $\mu_j \geq \beta/\delta$. Therefore, it holds that

$$\mu_j \geq \min\{1, \beta/\delta\} \tag{12.116}$$

Substituting into (12.115) we obtain

$$P(w_n) - P(w^\star) \leq \underbrace{\Big(1 - \min\{\nu, \nu\beta/\delta\}\Big)}_{\triangleq\, \lambda}\Big(P(w_{n-1}) - P(w^\star)\Big) \tag{12.117}$$

from which we deduce exponential convergence of $P(w_n)$ to $P(w^\star)$. For convex risk functions $P(w)$, we can establish a conclusion similar to (12.64b) by following an argument similar to the derivation of (11.71). This result is established in Prob. 12.16.

■

12.5 COORDINATE-DESCENT METHOD

The gradient-descent algorithm described in the earlier sections minimizes the risk function $P(w)$ over the *entire* vector $w \in \mathbb{R}^M$. One alternative technique is the *coordinate-descent* approach, which optimizes $P(w)$ over a *single* entry of w at a time while keeping all other entries fixed. The individual entries of w are called *coordinates* and, hence, the designation "coordinate-descent."

12.5.1 Derivation of Algorithm

In its traditional form, the coordinate-descent technique writes $P(w)$ as an explicit function of the individual entries of $w = \text{col}\{w_m\}$ for $m = 1, 2, \ldots, M$:

$$P(w) = P(w_1, \ldots, w_m, \ldots, w_M) \tag{12.118}$$

and minimizes it over each argument separately. Listing (12.119) describes the algorithm when the minimization can be carried out in closed-form. The weight iterate at iteration $n-1$ is denoted by w_{n-1} and its coordinates by $\{w_{n-1,m}\}$. At every iteration n, the algorithm cycles through the coordinates and updates each $w_{n-1,m}$ to $w_{n,m}$ by minimizing $P(w)$ over w_m while keeping all other coordinates of indexes $m' \neq m$ fixed at their most *recent* values. Observe in particular that once the first coordinate w_1 is updated to $w_{n,1}$ in the first step, this new value is used as argument in the second step that updates the second coordinate to $w_{n,2}$. The process continues in this manner by using the updated coordinates from the previous steps as arguments in subsequent steps.

Traditional coordinate-descent for minimizing $P(w)$.

```
let $w = \text{col}\{w_m\},\ m = 1, 2, \ldots, M$;
start from an arbitrary initial condition $w_{-1} = \text{col}\{w_{m,-1}\}$.
repeat until convergence over $n \geq 0$:
    $w_{n-1} = \text{col}\{w_{n-1,m}\}$ is available at start of iteration;
    for each coordinate $m = 1, 2, \ldots, M$ compute:
        $w_{n,1} = \underset{w_1 \in \mathbb{R}}{\text{argmin}}\ P(\,\underline{w_1}, w_{n-1,2}, w_{n-1,3}, \ldots, w_{n-1,M})$
        $w_{n,2} = \underset{w_2 \in \mathbb{R}}{\text{argmin}}\ P(w_{n,1}, \underline{w_2}, w_{n-1,3}, w_{n-1,4}, \ldots, w_{n-1,M})$        (12.119)
        $w_{n,3} = \underset{w_3 \in \mathbb{R}}{\text{argmin}}\ P(w_{n,1}, w_{n,2}, \underline{w_3}, w_{n-1,4}, \ldots, w_{n-1,M})$
                •
                •
        $w_{n,M} = \underset{w_M \in \mathbb{R}}{\text{argmin}}\ P(w_{n,1}, w_{n,2}, \ldots, w_{n,M-1}, \underline{w_M}\,)$
    end
end
$w_n = \text{col}\{w_{n,m}\}_{m=1}^{M}$
return $w^\star \leftarrow w_n$.
```

The coordinate-descent procedure can be motivated as follows. Consider a convex risk $P(w)$ and let $w^\star$ denote a global minimizer so that $\nabla_w P(w^\star) = 0$. This also means that

$$\left.\frac{\partial P(w)}{\partial w_m}\right|_{w_m = w_m^\star} = 0 \tag{12.120}$$

so that $P(w)$ is minimized over each coordinate. Specifically, for any step λ and basis vector $e_m \in \mathbb{R}^M$, it will hold that:

$$P(w^\star + \lambda e_m) \geq P(w^\star), \quad m = 1, 2, \ldots, M \tag{12.121}$$

which justifies searching for $w^\star$ by optimizing separately over the coordinates of w. Unfortunately, this property is lost when $P(w)$ is not differentiable — see Prob. 12.27. This fact highlights one of the weaknesses of the coordinate-descent method. We will revisit this issue in the next chapter and explain for what type of non-differentiable risks the coordinate-descent construction will continue to work.

Example 12.14 **(Coordinate-descent for ℓ_2-regularized least-squares)** Consider the regularized least-squares problem:

$$w^\star \;\stackrel{\Delta}{=}\; \underset{w\in\mathbb{R}^M}{\text{argmin}} \left\{\rho\|w\|^2 \;+\; \frac{1}{N}\sum_{\ell=0}^{N-1}\Big(\gamma(\ell)-h_\ell^{\mathsf T}w\Big)^2\right\} \tag{12.122}$$

We are using the subscript ℓ to index the data points $\{\gamma(\ell), h_\ell\}$ to avoid confusion with the subscript m used to index the individual coordinates of w. We denote the individual entries of h_ℓ by $\text{col}\{h_{\ell,m}\}$ for $m = 1, 2, \ldots, M$. We also use the notation w_{-m} and $h_{\ell,-m}$ to refer to the vectors w and h_ℓ with their $m-$th entries excluded. Then, as a function of w_m, the risk can be written in the form:

$$\begin{aligned} P(w) &= \rho\|w\|^2 \;+\; \frac{1}{N}\sum_{\ell=0}^{N-1}\Big(\gamma(\ell)-h_{\ell,-m}^{\mathsf T}w_{-m}-h_{\ell,m}w_m\Big)^2 \\ &\stackrel{(a)}{=} \rho\, w_m^2 \;+\; \underbrace{\left(\frac{1}{N}\sum_{\ell=0}^{N-1}h_{\ell,m}^2\right)}_{\stackrel{\Delta}{=}\, a_m} w_m^2 \;- \\ &\qquad\qquad \underbrace{\frac{2}{N}\left(\sum_{\ell=0}^{N-1}h_{\ell,m}\Big(\gamma(\ell)-h_{\ell,-m}^{\mathsf T}w_{-m}\Big)\right)}_{\stackrel{\Delta}{=}\, 2c_m} w_m + \text{cte} \\ &\stackrel{(b)}{=} (\rho+a_m)w_m^2 - 2c_m w_m + \text{cte} \end{aligned} \tag{12.123}$$

where terms independent of w_m are collected into the constant factor in step (a). In step (b), we introduced the scalars $a_m \geq 0$ and c_m for compactness of notation. Minimizing $P(w)$ over w_m we get

$$\widehat{w}_m = c_m/(\rho + a_m) \tag{12.124}$$

and arrive at listing (12.125).

Traditional coordinate-descent algorithm for solving (12.122).

```
given $N$ data points $\{\gamma(\ell), h_\ell\}$, $\ell = 0, 1, \ldots, N-1$;
start from an arbitrary initial condition $w_{-1} = 0$.
repeat until convergence over $n \geq 0$:
    iterate is $w_{n-1} = \text{col}\{w_{n-1,m}\}_{m=1}^M$
    repeat for each coordinate $m = 1, 2, \ldots, M$:
        $a_m = \frac{1}{N}\sum_{\ell=0}^{N-1} h_{\ell,m}^2$
        $c_m = \frac{1}{N}\sum_{\ell=0}^{N-1} h_{\ell,m}\big(\gamma(\ell) - h_{\ell,-m}^{\mathsf T} w_{n-1,-m}\big)$
        $w_{n,m} = c_m/(\rho + a_m)$
        $w_{n-1,m} \leftarrow w_{n,m}$,   (use updated coordinate in next step)
    end
end
$w_n = \text{col}\{w_{n,m}\}_{m=1}^M$
return $w^\star \leftarrow w_n$.
```

(12.125)

Observe that the expressions for a_m and c_m depend on all data points. Moreover, at each iteration n, all coordinates of w_n are updated. We discuss next some simplifications.

12.5.2 Randomized Implementation

In practice, the minimization of $P(w)$ over the coordinates $\{w_m\}$ is often difficult to solve in closed-form. In these cases, it is customary to replace the minimization in (12.119) by a gradient-descent step of the form:

$$w_{n,m} = w_{n-1,m} - \mu \left.\frac{\partial P(w)}{\partial w_m}\right|_{w=w_{n-1}}, \quad m = 1, 2, \ldots, M \tag{12.126}$$

While implementation (12.119) cycles through *all* coordinates of w at each iteration n, there are popular variants that limit the update to a single coordinate per iteration. The coordinate may be selected in different ways, for example, uniformly at random or as the coordinate corresponding to the maximal absolute gradient value. Description (12.127) is the randomized version of coordinate-descent.

Randomized coordinate-descent for minimizing $P(w)$.

```
let $w = \text{col}\{w_m\},\ m = 1, 2, \ldots, M$;
start from an arbitrary initial condition $w_{-1}$.
repeat until convergence over $n \geq 0$:
    $w_{n-1} = \text{col}\{w_{n-1,m}\}$ is available at start of iteration
    select an index $m^o$ at random within $1 \leq m \leq M$
    update $w_{n,m^o} = w_{n-1,m^o} - \mu \left.\frac{\partial P(w)}{\partial w_{m^o}}\right|_{w=w_{n-1}}$
    keep $w_{n,m} = w_{n-1,m}$, for all $m \neq m^o$
end
$w_n = \text{col}\{w_{n,m}\}_{m=1}^{M}$
return $w^\star \leftarrow w_n$.
```
(12.127)

This implementation can be viewed as a variation of gradient descent. At every iteration n, we select some basis vector e_{m^o} at random with probability $1/M$ and use it to construct the scaling diagonal matrix $D_n = \text{diag}\{e_{m^o}\}$, with a single unit entry on its diagonal at the m^o-th location. Note that D_n is a random matrix, and therefore we will write $\boldsymbol{D}_n$ using the boldface notation to highlight this fact. The selection of $\boldsymbol{D}_n$ at iteration n is performed independently of any other variables in the optimization problem. Then, the update generated by the algorithm can be written in vector form as follows:

$$\boldsymbol{w}_n = \boldsymbol{w}_{n-1} - \mu \boldsymbol{D}_n \nabla_{w^{\mathsf{T}}} P(\boldsymbol{w}_{n-1}) \tag{12.128}$$

where the variables $\{\boldsymbol{w}_n, \boldsymbol{w}_{n-1}\}$ are also *random* in view of the randomness in $\boldsymbol{D}_n$. In particular, observe that on average:

$$\mathbb{E}\,\boldsymbol{D}_n = \frac{1}{M}\sum_{m=1}^{M} \text{diag}\{e_m\} \;=\; \frac{1}{M} I_M \;=\; \mathbb{E}\,\boldsymbol{D}_n^2 \tag{12.129}$$

The following statement establishes the convergence of the randomized algorithm. In contrast to the earlier arguments in this chapter on the convergence of the gradient-descent implementation, we now need to take the randomness of the weight iterates into account.

Theorem 12.4. (Convergence of randomized coordinate-descent) *Consider the randomized coordinate-descent algorithm (12.127) for minimizing a first-order differentiable risk function $P(w)$, where $P(w)$ is $\nu-$strongly-convex with $\delta-$Lipschitz gradients according to (12.12a)–(12.12b). Introduce the error vector $\widetilde{\boldsymbol{w}}_n = w^\star - \boldsymbol{w}_n$, which measures the difference between the $n-$th iterate and the global minimizer of $P(w)$. If the step-size μ satisfies (i.e., is small enough):*

$$0 < \mu < 2\nu/\delta^2 \tag{12.130}$$

then the mean-square-error, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_n\|^2$, and the average excess risk converge exponentially fast in the sense that

$$\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_n\|^2 \;\le\; \lambda\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{n-1}\|^2, \;\; n \ge 0 \tag{12.131a}$$

$$\mathbb{E}\,P(\boldsymbol{w}_n) - P(w^\star) \;\le\; \frac{\delta}{2}\lambda^{n+1}\|\widetilde{w}_{-1}\|^2 \;=\; O(\lambda^n) \tag{12.131b}$$

where

$$\lambda \;=\; 1 - \frac{2\mu\nu}{M} + \frac{\mu^2\delta^2}{M} \;\in [0,1) \tag{12.132}$$

Proof: We subtract $w^\star$ from both sides of (12.128) to get

$$\widetilde{\boldsymbol{w}}_n \;=\; \widetilde{\boldsymbol{w}}_{n-1} \;+\; \mu\,\boldsymbol{D}_n\,\nabla_{w^{\mathsf{T}}}\,P(\boldsymbol{w}_{n-1}) \tag{12.133}$$

We compute the squared Euclidean norms (or energies) of both sides and use the fact that $\nabla_{w^{\mathsf{T}}}\,P(w^\star) = 0$ to write

$$\|\widetilde{\boldsymbol{w}}_n\|^2 = \|\widetilde{\boldsymbol{w}}_{n-1}\|^2 + 2\mu\left(\nabla_{w^{\mathsf{T}}}\,P(\boldsymbol{w}_{n-1})\right)^{\mathsf{T}} \boldsymbol{D}_n\widetilde{\boldsymbol{w}}_{n-1} + \mu^2\,\|\nabla_{w^{\mathsf{T}}}\,P(\boldsymbol{w}_{n-1})\|^2_{\boldsymbol{D}_n^2} \tag{12.134}$$

where the notation $\|x\|^2_A$ stands for $x^{\mathsf{T}}Ax$. Conditioning on $\widetilde{\boldsymbol{w}}_{n-1}$ and taking expectations of both sides gives

$$\begin{aligned}
&\mathbb{E}\left(\|\widetilde{\boldsymbol{w}}_n\|^2 \,|\, \widetilde{\boldsymbol{w}}_{n-1}\right) \\
&\leq \|\widetilde{w}_{n-1}\|^2 + 2\mu\left(\nabla_{w^{\mathsf{T}}} P(w_{n-1})\right)^{\mathsf{T}} (\mathbb{E}\,\boldsymbol{D}_n)\widetilde{w}_{n-1} + \mu^2 \left\|\nabla_{w^{\mathsf{T}}} P(w_{n-1})\right\|^2_{\mathbb{E}\,\boldsymbol{D}_n^2} \\
&\overset{(12.129)}{=} \|\widetilde{w}_{n-1}\|^2 + \frac{2\mu}{M}\left(\nabla_{w^{\mathsf{T}}} P(w_{n-1})\right)^{\mathsf{T}} \widetilde{w}_{n-1} + \frac{\mu^2}{M}\left\|\nabla_{w^{\mathsf{T}}} P(w_{n-1})\right\|^2 \\
&\overset{(12.47)}{\leq} \|\widetilde{w}_{n-1}\|^2 - \frac{2\mu\nu}{M}\|\widetilde{w}_{n-1}\|^2 + \frac{\mu^2\delta^2}{M}\|\widetilde{w}_{n-1}\|^2 \qquad (12.135)
\end{aligned}$$

Taking expectations again to eliminate the conditioning over $\widetilde{\boldsymbol{w}}_{n-1}$ we arrive at

$$\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_n\|^2 \;\leq\; \left(1 - \frac{2\mu\nu}{M} + \frac{\mu^2\delta^2}{M}\right)\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{n-1}\|^2 \qquad (12.136)$$

Comparing with (12.48) we find that the recursion is in terms of the mean-square error. The structure of the above recursion is similar to (12.43a) and we arrive at the conclusions stated in the theorem.

■

We conclude fthat the weight iterate $\boldsymbol{w}_n$ converges in the mean-square-error sense to $w^\star$. Referring to the earlier diagram from Fig. 3.11 on the convergence of random sequences, we conclude that this fact implies that $\boldsymbol{w}_n$ converges to $w^\star$ in probability. Moreover, comparing expression (12.132) for λ with (12.44) in the gradient-descent case we find that

$$\lambda \approx 1 - 2\mu\nu, \qquad \text{(for gradient-descent)} \qquad (12.137a)$$

$$\lambda \approx 1 - \frac{2\mu\nu}{M}, \qquad \text{(for randomized coordinate-descent)} \qquad (12.137b)$$

which suggests that randomized coordinate descent converges at a slower pace. This is understandable because only one entry of the weight iterate is updated at every iteration. However, if we compare the performance of both algorithms by considering one iteration for gradient descent against M iterations for randomized coordinate descent, then the decay of the squared weight-error vectors will occur at comparable rates.

Remark 12.5. (**Bound on step-size)** If we follow the argument from Example 12.10 based on the risk function, we can relax the upper bound on μ in (12.130) to $\mu < 2/\delta$ and replace λ by

$$\lambda \;=\; 1 - \frac{2\mu\nu}{M} + \frac{\mu^2\nu\delta}{M} \;\in [0,1) \qquad (12.138)$$

This fact is exploited in the proof of convergence for the Gauss-Southwell variant in the next section.

■

Example 12.15 (**Randomized coordinate-descent for regularized least-squares**) Consider the ℓ_2-regularized least-squares problem (12.122). In this case, we can determine the optimal coordinate $w^o_{n,m}$ in closed-form at every iteration n. Using

$$\partial P(w)/\partial w_m = 2(\rho + a_m)w_m - 2c_m, \quad m = 1, 2, \ldots, M \qquad (12.139)$$

we find that the corresponding randomized coordinate-descent implementation is given by listing (12.140).

Randomized coordinate-descent for solving (12.122).

```
given $N$ data points $\{\gamma(\ell), h_\ell\},\ \ell = 0, 1, \ldots, N-1$;
start from an arbitrary initial condition $w_{-1} = 0$.
repeat until convergence over $n \geq 0$:
    iterate is $w_{n-1} = \text{col}\{w_{n-1,m}\}_{m=1}^{M}$
    select an index $m^o$ at random within $1 \leq m \leq M$
    $a_{m^o} = \frac{1}{N}\sum_{\ell=0}^{N-1} h_{\ell,m^o}^2$
    $c_{m^o} = \frac{1}{N}\sum_{\ell=0}^{N-1} h_{\ell,m^o}\Big(\gamma(\ell) - h_{\ell,-m^o}^{\mathsf{T}} w_{n-1,-m^o}\Big)$
    update $w_{n,m^o} = c_{m^o}/(\rho + a_{m^o})$
    keep $w_{n,m} = w_{n-1,m}$, for all $m \neq m^o$
end
$w_n = \text{col}\{w_{n,m}\}_{m=1}^{M}$
return $w^\star \leftarrow w_n$.
```
(12.140)

We can describe the algorithm in vector form by introducing the vector and matrix quantities:

$$\gamma_N \triangleq \begin{bmatrix} \gamma(0) \\ \gamma(1) \\ \vdots \\ \gamma(N-1) \end{bmatrix}, \quad H_N = \begin{bmatrix} h_0^{\mathsf{T}} \\ h_1^{\mathsf{T}} \\ \vdots \\ h_{N-1}^{\mathsf{T}} \end{bmatrix} \tag{12.141}$$

where γ_N is $N \times 1$ and H_N is $N \times M$. We let x_{m^o} denote the column of index m^o in H_N and write $H_{N,-m^o}$ to refer to the data matrix H_N with its m^o−th column excluded. That is, $H_{N,-m^o}$ has dimensions $N \times (M-1)$. Then, it can be verified that — see Prob. 12.30:

$$w_{n,m^o} = \frac{1}{\rho + \frac{1}{N}\|x_{m^o}\|^2} \times \frac{1}{N} x_{m^o}^{\mathsf{T}}\Big(\gamma_N - H_{N,-m^o}\, w_{n-1,-m^o}\Big) \tag{12.142}$$

We illustrate the operation of the algorithm by generating a random model $w^o \in {\rm I\!R}^{10}$ with $M = 10$, and a collection of $N = 200$ random feature vectors $\{h_n\}$. The entries of w^o are selected randomly from a Gaussian distribution with mean zero and unit variance; likewise for the entries of the feature vectors. We also generate noisy target signals:

$$\gamma(n) = h_n^{\mathsf{T}} w^o + v(n) \tag{12.143}$$

where $v(n)$ are realizations of zero-mean Gaussian noise with variance $\sigma_v^2 = 0.01$. We set the step-size parameter to $\mu = 0.01$ and the regularization parameter to $\rho = 2/N$. If we differentiate the risk function (12.122) relative to w, it is straightforward to determine that the minimizer is given by:

$$w^\star = (\rho N I_M + H_N^{\mathsf{T}} H_N)^{-1} H_N^{\mathsf{T}} \gamma_N \tag{12.144}$$

Substituting $w^\star$ into the risk function we find the minimal risk value, $P(w^\star) = 0.0638$. The learning curves in Fig. 12.6 are plotted relative to this value; the curves in the right plot in the first row are normalized by the maximum value of $P(w_n) - P(w^\star)$ so that they start from the value one. The learning curves for the coordinate descent implementation are downsampled by a factor $M = 10$ since, on average, it takes 10 iterations for all entries of the weight vector to be updated (whereas, under the gradient-descent implementation, all entries are updated at every single iteration). The downsampling allows for a fair comparison of the convergence rates of the two methods. It is observed from the results in the figure that both methods are able to estimate $w^\star$ and that their learning curves practically coincide with each other.

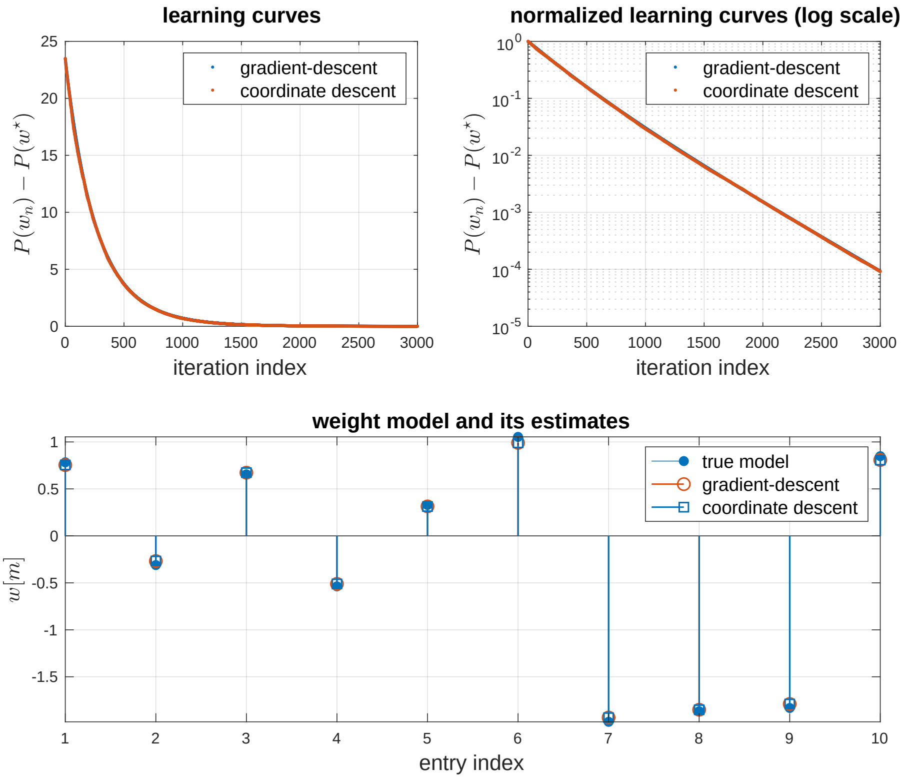

Figure 12.6 (*Top*) Learning curves $P(w_n)$ relative to the minimum risk value $P(w^\star)$ in regular and normalized logarithmic scales for gradient-descent and randomized coordinate-descent; the learning curve for the latter is downsampled and plotted every $M = 10$ iterations. (*Bottom*) The original and estimated parameter models.

12.5.3 Gauss-Southwell Implementation

We consider next a coordinate-descent implementation where at every iteration n, the coordinate corresponding to the maximal absolute gradient is updated. This variant is known as the Gauss-Southwell (GS) rule.

Gauss-Southwell coordinate-descent for minimizing $P(w)$.

```
let $w = \mathrm{col}\{w_m\},\ m = 1, 2, \ldots, M$;
start from an arbitrary initial condition $w_{-1}$.
repeat until convergence over $n \geq 0$:
    $w_{n-1} = \mathrm{col}\{w_{n-1,m}\}$ is available at start of iteration
    select $m^o = \underset{1 \leq m \leq M}{\mathrm{argmax}} \left| \frac{\partial P(w)}{\partial w_m} \right|$ evaluated at $w = w_{n-1}$        (12.145)
    update $w_{n,m^o} = w_{n-1,m^o} - \mu \left. \frac{\partial P(w)}{\partial w_{m^o}} \right|_{w = w_{n-1}}$
    keep $w_{n,m} = w_{n-1,m}$, for all $m \neq m^o$
end
$w_n = \mathrm{col}\{w_{n,m}\}_{m=1}^{M}$
return $w^\star \leftarrow w_n$.
```

Motivation

We motivate implementation (12.145) by explaining that the update direction corresponds to a steepest-descent choice. We follow an argument similar to Example 12.12 except that the bound is now imposed on the ℓ_1-norm of the perturbation rather than on its Euclidean norm.

Specifically, starting from an iterate w_{n-1}, we seek to determine a small adjustment to it, say,

$$w_n = w_{n-1} + \delta w \tag{12.146}$$

by solving

$$\delta w^o = \underset{\delta w \in \mathbb{R}^M}{\mathrm{argmin}} \Big\{ P(w_{n-1} + \delta w) \Big\}, \quad \text{subject to } \|\delta w\|_1 \leq \mu' \tag{12.147}$$

for some $\mu' > 0$. We introduce the *first-order* Taylor series expansion:

$$P(w_n) \approx P(w_{n-1}) + \nabla_w P(w_{n-1}) \delta w \tag{12.148}$$

and approximate the problem by solving

$$y^o = \underset{y \in \mathbb{R}^M}{\mathrm{argmin}} \Big\{ \mu' \, \nabla_w P(w_{n-1}) y \Big\}, \quad \text{subject to } \|y\|_1 \leq 1 \tag{12.149}$$

where we introduced the change of variables $y = \delta w / \mu'$. We recall from the result of part (c) in Prob. 1.26 that for any vectors $\{x, y\}$ of matching dimensions, the ℓ_1 and ℓ_∞ norms satisfy:

$$\|x\|_\infty = \sup_{\|y\|_1 \leq 1} \Big\{ x^\mathsf{T} y \Big\} \tag{12.150}$$

Problem (12.149) has the same form (if we negate the argument and replace max by min):

$$y^o = \underset{\|y\|_1 \leq 1}{\text{argmax}} \; \Big\{ -\mu' \, \nabla_w P(w_{n-1}) y \Big\} \tag{12.151}$$

It follows that the optimal value of (12.151) is equal to $\mu' \, \|\nabla_w P(w_{n-1})\|_\infty$, which is the maximum absolute entry of the gradient vector scaled by μ'. Let m^o denote the index of this maximum absolute entry. Then, the maximal value of (12.151) is attained if we select

$$y^o = -\, e_{m^o} \, \text{sign}\left(\frac{\partial P(w_{n-1})}{\partial w_{m^o}} \right) \tag{12.152}$$

Taking a step along this direction leads to the update

$$w_{n,m^o} = w_{n-1,m^o} - \mu' \, \text{sign}\left(\frac{\partial P(w_{n-1})}{\partial w_{m^o}} \right) \tag{12.153}$$

which updates w_{n-1} along the descent direction determined by the maximal absolute entry of the gradient vector; in a manner “similar” to (12.145).

Convergence

The Gauss-Southwell implementation (12.145) can again be viewed as a variation of gradient-descent. At every iteration n, we construct the diagonal matrix $D_n = \text{diag}\{e_{m^o}\}$, where m^o is the index of the entry in the gradient vector at w_{n-1} with the largest absolute value. The matrix D_n is not random anymore, as was the case in the randomized coordinate-descent implementation. Instead, its value depends on w_{n-1} and the update can be written in vector form as follows:

$$w_n = w_{n-1} + \mu D_n \nabla_{w^\mathsf{T}} P(w_{n-1}) \tag{12.154}$$

In the analysis for the randomized algorithm, we were able to remove the effect of the matrix $\boldsymbol{D}_n$ through expectation. This is not possible here because D_n is now deterministic and dependent on w_{n-1}. Nevertheless, a similar convergence analysis is applicable with one minor adjustment. We continue to assume that the risk function $P(w)$ is $\nu-$strongly convex as in (12.12a), but require $P(w)$ to have $\delta-$Lipschitz gradients relative to each coordinate, namely,

(1) (**Strong convexity**). $P(w)$ is $\nu-$strongly convex and first-order differentiable:

$$P(w_2) \;\geq\; P(w_1) \;+\; \nabla_{w^\mathsf{T}} P(w_1)(w_2 - w_1) \;+\; \frac{\nu}{2}\|w_2 - w_1\|^2 \tag{12.155a}$$

for every $w_1, w_2 \in \text{dom}(P)$ and some $\nu > 0$.

(2) (**$\boldsymbol{\delta}-$Lipschitz gradients relative to each coordinate**). The gradient vectors of $P(w)$ are $\delta-$Lipschitz relative to each coordinate, meaning that:

$$\left| \frac{\partial}{\partial w_m} P(w + \alpha e_m) - \frac{\partial}{\partial w_m} P(w) \right| \leq \delta |\alpha| \tag{12.155b}$$

for any $w \in \text{dom}(P)$, $\alpha \in \mathbb{R}$, and where e_m denotes the $m-$th basis vector in $\mathbb{R}^M$.

Theorem 12.5. (Convergence of Gauss-Southwell coordinate-descent) *Consider the Gauss-Southwell coordinate-descent algorithm (12.145) for minimizing a first-order differentiable risk function $P(w)$, where $P(w)$ is $\nu-$strongly-convex with $\delta-$Lipschitz gradients relative to each coordinate according to (12.155a)–(12.155b). Introduce the error vector $\widetilde{w}_n = w^\star - w_n$, which measures the difference between the $n-$th iterate and the global minimizer of $P(w)$. If the step-size μ satisfies (i.e., is small enough):*

$$0 < \mu < 2/\delta \tag{12.156}$$

then the risk value converges exponentially fast as follows:

$$P(w_n) - P(w^\star) \;\le\; \lambda\Big(P(w_{n-1}) - P(w^\star)\Big) \tag{12.157}$$

where

$$\lambda \;=\; 1 - \frac{2\mu\nu}{M} + \frac{\mu^2\nu\delta}{M} \;\in [0,1) \tag{12.158}$$

Proof: We follow an argument similar to Example 12.10 based on the risk function. In view of the $\nu-$strong convexity of $P(w)$, we first use property (8.29) to deduce that

$$P(w^\star) \ge P(w_{n-1}) - \frac{1}{2\nu}\|\nabla_w\, P(w_{n-1})\|^2 \tag{12.159}$$

Next, using the coordinate-wide $\delta-$Lipschitz property (12.155b) and the result of Prob. 10.1 we write

$$\begin{aligned}
P(w_n) \quad &\le \quad P(w_{n-1}) + \frac{\partial P(w_{n-1})}{\partial w_{m^o}}(w_{n,m^o} - w_{n-1,m^o}) + \frac{\delta}{2}(w_{n,m^o} - w_{n-1,m^o})^2 \\
&\overset{(12.127)}{=} P(w_{n-1}) - \mu\left(\frac{\partial P(w_{n-1})}{\partial w_{m^o}}\right)^2 + \frac{\mu^2\delta}{2}\left(\frac{\partial P(w_{n-1})}{\partial w_{m^o}}\right)^2 \\
&= \quad P(w_{n-1}) - \mu\Big(1 - \frac{\mu\delta}{2}\Big)\left(\frac{\partial P(w_{n-1})}{\partial w_{m^o}}\right)^2
\end{aligned} \tag{12.160}$$

Now note the bound

$$\left(\frac{\partial P(w_{n-1})}{\partial w_{m^o}}\right)^2 \overset{(a)}{=} \|\nabla_w\, P(w_{n-1})\|_\infty^2 \overset{(b)}{\ge} \frac{1}{M}\|\nabla_w\, P(w_{n-1})\|_2^2 \tag{12.161}$$

where step (a) is by construction and step (b) is the property of norms $\|x\|_2 \le \sqrt{M}\|x\|_\infty$ for any $M-$dimensional vector x. Subtracting $P(w^\star)$ from both sides of (12.160) and using (12.159) we obtain after grouping terms:

$$P(w_n) - P(w^\star) \le \Big(1 - \frac{2\mu\nu}{M} + \frac{\mu^2\nu\delta}{M}\Big)\Big(P(w_{n-1}) - P(w^\star)\Big) \tag{12.162}$$

■

12.6 ALTERNATING PROJECTION ALGORITHM[1]

We end this chapter with one application of the gradient-descent methodology to the derivation of a popular *alternating projection* algorithm. The method can be used to check whether two convex sets have a nontrivial intersection and to retrieve points from that intersection. It can also be used to verify whether a convex optimization problem with constraints has feasible solutions.

Consider two closed convex sets $\mathcal{C}_1$ and $\mathcal{C}_2$ in $\mathbb{R}^M$ and assume we are interested in determining a point $w^\star$ in their intersection, $\mathcal{C}_1 \cap \mathcal{C}_2$. Let w denote some arbitrary point. The distance from w to any of the sets is denoted by $\text{dist}(w, \mathcal{C})$ and defined as the smallest Euclidean distance to the elements in $\mathcal{C}$:

$$\text{dist}(w, \mathcal{C}) \;\stackrel{\Delta}{=}\; \min_{c \in \mathcal{C}} \|c - w\|_2 \tag{12.163}$$

To determine $w^\star$, we formulate the optimization problem — see Prob. 12.22:

$$\begin{aligned} P(w) &\stackrel{\Delta}{=} \max\Big\{\text{dist}(w, \mathcal{C}_1),\ \text{dist}(w, \mathcal{C}_2)\Big\} && \text{(12.164a)}\\ w^\star &= \underset{w \in \mathbb{R}^M}{\text{argmin}}\ P(w) && \text{(12.164b)} \end{aligned}$$

For every w, the cost function $P(w)$ measures its distance to the set that is furthest away from it. The minimization seeks a point w with the smallest maximal distance. This formulation is motivated by the result of Prob. 9.13, which showed that provided the sets intersect:

$$P(w^\star) = 0 \iff w^\star \in \mathcal{C}_1 \cap \mathcal{C}_2 \tag{12.165}$$

That is, we will succeed in finding a point in the intersection if, and only if, the minimal cost value turns out to be zero.

Now, for any w outside the intersection set, it will generally be further away from one of the sets. Let ℓ be the index of this set so that $P(w) = \text{dist}(w, \mathcal{C}_\ell)$. Let $\mathcal{P}_{C_\ell}(w)$ denote the projection of w onto $\mathcal{C}_\ell$. We can now evaluate the gradient of $P(w)$ by using the result of Prob. 8.32, which shows that the gradient of $\text{dist}(w, \mathcal{C}_\ell)$ relative to w is given by

$$\nabla_{w^{\mathsf{T}}} \text{dist}(w, \mathcal{C}_\ell) \;=\; \frac{w - \mathcal{P}_{C_\ell}(w)}{\|w - \mathcal{P}_{C_\ell}(w)\|_2} \tag{12.166}$$

We can therefore use this expression to write the following gradient-descent re-

[1] This section can be skipped on a first reading.

cursion to minimize $P(w)$ with an iteration-dependent step-size:

```
for each iteration $n$ do:
    $w_{n-1}$ is the iterate at step $n-1$;
    let $\ell$ denote the index of the convex set furthest from it;
    let $\mathcal{P}_{C_\ell}(w_{n-1})$ denote the projection of $w_{n-1}$ onto this set;
    set the step-size to $\mu(n) = \|w_{n-1} - \mathcal{P}_{C_\ell}(w_{n-1})\|$  (= distance to $\mathcal{C}_\ell$);
    update $w_n = w_{n-1} - \mu(n)\,\nabla_{w^\mathsf{T}}\,\text{dist}(w_{n-1}, \mathcal{C}_\ell)$;
end
```

(12.167)

We can simplify the last step as follows:

$$\begin{aligned} w_n &= w_{n-1} - \mu(n)\,\nabla_{w^\mathsf{T}}\,\text{dist}(w_{n-1}, \mathcal{C}_\ell) \\ &= w_{n-1} \;-\; \|w_{n-1} - \mathcal{P}_{C_\ell}(w_{n-1})\| \times \frac{w_{n-1} - \mathcal{P}_{C_\ell}(w_{n-1})}{\|w_{n-1} - \mathcal{P}_{C_\ell}(w_{n-1})\|} \\ &= w_{n-1} - \Big(w_{n-1} - \mathcal{P}_{C_\ell}(w_{n-1})\Big) \end{aligned} \tag{12.168}$$

That is, the gradient-descent step reduces to the following projection step:

$$\boxed{w_n = \mathcal{P}_{C_\ell}(w_{n-1})} \tag{12.169}$$

Thus, for example, if $\mathcal{C}_1$ happens to be the convex set that is furthest from w_{n-1}, then w_n will be its projection onto $\mathcal{C}_1$. For the next iteration, $\mathcal{C}_2$ will be the set that is furthest away from w_n and we will project onto $\mathcal{C}_2$. In this way, we arrive at a procedure that involves projecting onto the two convex sets alternately until $w^\star$ is attained. This construction is illustrated in Fig. 12.7.

We can describe the alternating projection procedure as generating two sequences of vectors, $\{a_n, b_n\}$, one in $\mathcal{C}_1$ and the other in $\mathcal{C}_2$. Assume we start from an initial condition $a_{-1} \in \mathcal{C}_1$; if the initial condition is outside $\mathcal{C}_1$, we can always project it onto $\mathcal{C}_1$ first and take that projection as the initial condition. Then, we can alternate as shown in listing (12.171) for $n \geq 0$ or, equivalently,

$$a_n = \mathcal{P}_{\mathcal{C}_1}\Big(\mathcal{P}_{C_2}(a_{n-1})\Big), \quad n \geq 0 \tag{12.170}$$

In this way, the sequence of vectors $\{a_n,\ n \geq -1\}$ will belong to $\mathcal{C}_1$ while the sequence of vectors $\{b_n,\ n \geq 0\}$ will belong to $\mathcal{C}_2$.

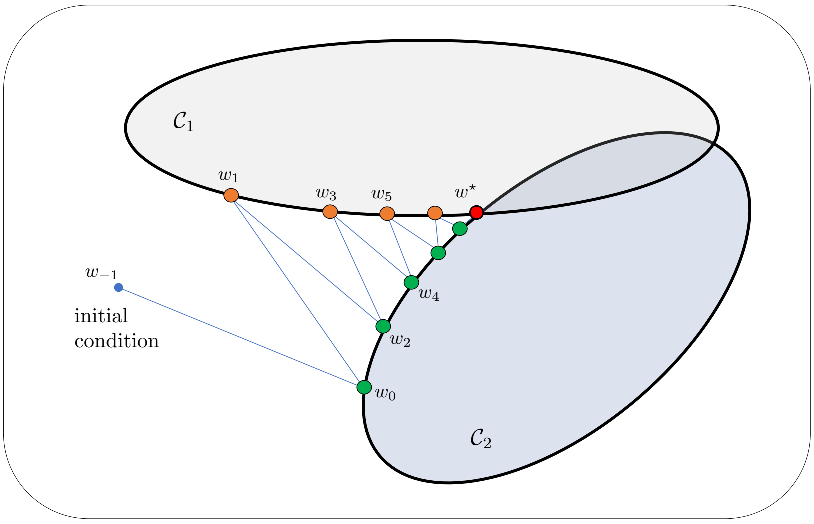

Figure 12.7 Illustration of the alternating projection procedure over two convex sets. Starting from an initial condition, the algorithm successively alternates the projections on the sets.

Alternating projection algorithm.

```
given two closed convex sets $\mathcal{C}_1$ and $\mathcal{C}_2$;
given projection operators onto $\mathcal{C}_1$ and $\mathcal{C}_2$;
start from arbitrary $a_{-1} \in \mathcal{C}_1$;
objective:
   if $\mathcal{C}_1 \cap \mathcal{C}_2 \neq \emptyset$: find a point $w^\star$ in the intersection;
   else find points $\{a^\star \in \mathcal{C}_1, b^\star \in \mathcal{C}_2\}$ closest to each other;      (12.171)
repeat until convergence over $n \geq 0$:
   $b_n = \mathcal{P}_{C_2}(a_{n-1})$
   $a_n = \mathcal{P}_{C_1}(b_n)$
end
if $\|a_n - b_n\|$ small, return $w^\star \leftarrow a_n$;
   else return $a^\star \leftarrow a_n$, $b^\star \leftarrow b_n$.
end
```

We examine next the convergence of the algorithm; its behavior will depend on whether the sets $\mathcal{C}_1$ and $\mathcal{C}_2$ have a nontrivial intersection:

(a) Assume first that $\mathcal{C}_1 \cap \mathcal{C}_2 \neq \emptyset$. Then, we will verify that the sequences a_n

and b_n will converge to the *same* limit point $w^\star \in \mathcal{C}_1 \cap \mathcal{C}_2$. That is, both sequences will converge to a point in the intersection set.

(b) Assume next that $\mathcal{C}_1 \cap \mathcal{C}_2 = \emptyset$ so that the sets $\mathcal{C}_1$ and $\mathcal{C}_2$ do not intersect. The algorithm converges now in a different manner. To describe the behavior, we define the distance between the two convex sets as the smallest distance attainable between any points in the sets, namely,

$$\text{dist}(\mathcal{C}_1, \mathcal{C}_2) \triangleq \min_{x \in \mathcal{C}_1, y \in \mathcal{C}_2} \|x - y\| \tag{12.172}$$

Then, we show below that the sequences a_n and b_n will converge to limit points $a^\star \in \mathcal{C}_1$ and $b^\star \in \mathcal{C}_2$, respectively, such that the distance between them attains the distance between the sets:

$$\|a^\star - b^\star\| = \text{dist}(\mathcal{C}_1, \mathcal{C}_2) \tag{12.173}$$

This fact is illustrated schematically in Fig. 12.8.

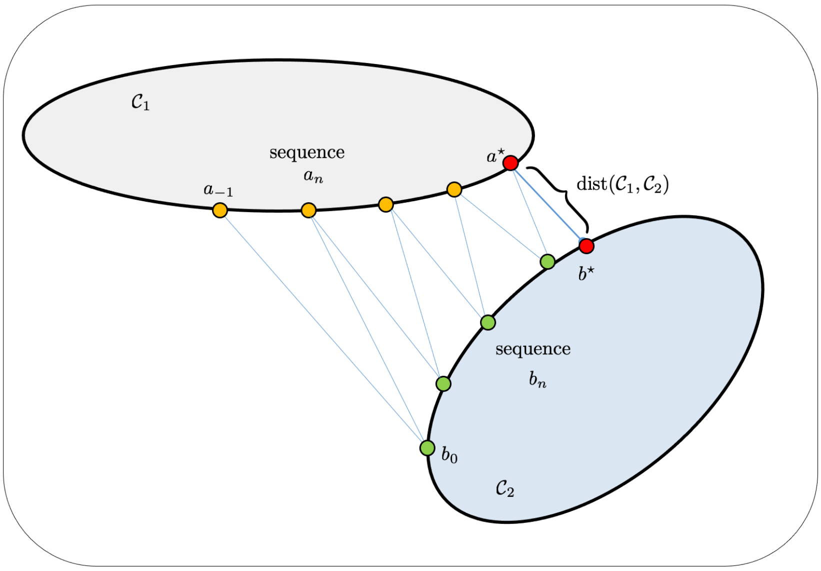

Figure 12.8 The sequences $\{a_n, b_n\}$ generated by the alternating projection algorithm converge to limit points $\{a^\star, b^\star\}$ that are closest to each other from both convex sets.

Theorem 12.6. (Convergence of alternating projection) *Consider the alternating projection algorithm (12.171) and two closed convex sets $\mathcal{C}_1$ and $\mathcal{C}_2$:*

(a) When $\mathcal{C}_1 \cap \mathcal{C}_2 \neq \emptyset$, the sequences $\{a_n, b_n\}$ converge to the same limit point $w^\star \in \mathcal{C}_1 \cap \mathcal{C}_2$.

(b) When $\mathcal{C}_1 \cap \mathcal{C}_2 = \emptyset$, the sequences $\{a_n, b_n\}$ converge to limit points $\{a^\star \in \mathcal{C}_1,\ b^\star \in \mathcal{C}_2\}$ that are closest to each other.

Proof: Assume initially that the sets intersect and let $w^\star$ denote some arbitrary point in their intersection. Then, obviously, projecting $w^\star$ onto either set results in $w^\star$ again:

$$w^\star = \mathcal{P}_{\mathcal{C}_1}(w^\star), \quad w^\star = \mathcal{P}_{\mathcal{C}_2}(w^\star) \tag{12.174}$$

We now call upon the non-expansive property (9.70) of projection operators, namely, the fact that, for any convex set $\mathcal{C}$:

$$\|\mathcal{P}_{\mathcal{C}}(x) - \mathcal{P}_{\mathcal{C}}(y)\| \le \|x-y\|, \quad \forall\, x, y \in \mathcal{C} \tag{12.175}$$

Applying this property to the sequences $\{a_n, b_n\}$ generated by (12.171) we get (using $x = a_{n-1}$ in the first line and $x = b_n$ in the second line while $y = w^\star$ in both lines):

$$\|\mathcal{P}_{\mathcal{C}_2}(a_{n-1}) - w^\star\| \;=\; \|b_n - w^\star\| \le \|a_{n-1} - w^\star\| \tag{12.176a}$$

$$\|\mathcal{P}_{\mathcal{C}_1}(b_n) - w^\star\| = \|a_n - w^\star\| \le \|b_n - w^\star\| \tag{12.176b}$$

It follows that a_n is closer to $w^\star$ than b_n, and b_n is closer to $w^\star$ than a_{n-1}. More importantly, by combining both inequalities, we observe that the sequence of squared distances $\|a_n - w^\star\|^2$ is decreasing and bounded from below by zero since it satisfies

$$0 \le \|a_n - w^\star\|^2 \le \|a_{n-1} - w^\star\|^2 \;\le\; \ldots \;\le\; \|a_{-1} - w^\star\|^2 \tag{12.177}$$

We conclude that the sequence of projections $a_n \in \mathcal{C}_1$ converges to some limit point denoted by $a^\star$. Since $\mathcal{C}_1$ is closed by assumption, this point belongs to $\mathcal{C}_1$, i.e., $a^\star \in \mathcal{C}_1$. A similar argument shows that

$$0 \le \|b_n - w^\star\|^2 < \|b_{n-1} - w^\star\|^2 \;<\; \ldots \;<\; \|b_0 - w^\star\| < \|a_{-1} - w^\star\|^2 \tag{12.178}$$

so that the sequence of projections $b_n \in \mathcal{C}_2$ converges to some limit point denoted by $b^\star \in \mathcal{C}_2$. The limit points satisfy

$$a^\star = \mathcal{P}_{\mathcal{C}_1}(b^\star), \quad b^\star = \mathcal{P}_{\mathcal{C}_2}(a^\star) \tag{12.179}$$

We next apply the inner-product property (9.66) for projections, namely, the fact that

$$(x - \mathcal{P}_{\mathcal{C}}(x))^{\mathsf{T}}(c - \mathcal{P}_{\mathcal{C}}(x)) \le 0, \quad \forall\, c \in \mathcal{C} \tag{12.180}$$

Therefore, the limit points $\{a^\star, b^\star\}$ satisfy

$$(b^\star - a^\star)^{\mathsf{T}}(c_1 - a^\star) \le 0, \quad \forall\, c_1 \in \mathcal{C}_1 \tag{12.181a}$$

$$(a^\star - b^\star)^{\mathsf{T}}(c_2 - b^\star) \le 0, \quad \forall\, c_2 \in \mathcal{C}_2 \tag{12.181b}$$

Adding gives

$$\|b^\star - a^\star\|^2 \le (b^\star - a^\star)^{\mathsf{T}}(c_2 - c_1) \overset{(a)}{\le} \|b^\star - a^\star\|\,\|c_2 - c_1\| \tag{12.182}$$

where step (a) is by Cauchy-Schwarz. It follows that

$$\|b^\star - a^\star\| \le \|c_2 - c_1\|, \quad \forall\, c_1 \in \mathcal{C}_1,\ c_2 \in \mathcal{C}_2 \tag{12.183}$$

When the sets $\mathcal{C}_1$ and $\mathcal{C}_2$ have a nontrivial intersection, the right-hand side can be made equal to zero by selecting $c_1 = c_2 = w^\star$, from which we conclude that $b^\star = a^\star$. But since $b^\star \in \mathcal{C}_2$ and $a^\star \in \mathcal{C}_1$ by construction, the limit point satisfying $b^\star = a^\star$ must belong to the intersection of both sets. On the other hand, when the intersection is an empty set, we conclude that $\|b^\star - a^\star\|$ attains the smallest distance between any two points in $\mathcal{C}_1$ and $\mathcal{C}_2$.

■

The alternating projection method suffers from one inconvenience when the intersection set $\mathcal{C}_1 \cap \mathcal{C}_2$ has more than one point. Starting from an initial vector

a_{-1}, the method generates two sequences $\{a_n \in \mathcal{C}_1\}$ and $\{b_n \in \mathcal{C}_2\}$ that are only guaranteed to converge to some *arbitrary* point in the intersection. We describe a modification in the comments at the end of the chapter, known as *Dykstra method*, which allows the algorithm to converge to the point $w^\star$ in the intersection that is closest to the initial condition a_{-1} — see listing (12.208).

12.7 COMMENTARIES AND DISCUSSION

Method of gradient descent. In Sec. 12.3 we motivated the gradient-descent recursion (12.22) for minimizing differentiable convex functions. The method is credited to the French mathematician **Augustine Cauchy (1789–1857)**, who proposed it as an iterative procedure for locating the roots of a function. Consider a function $P(x, y, z)$ of three scalar parameters (x, y, z), and assume that $P(x, y, z)$ has a unique root at some location $(x^\star, y^\star, z^\star)$ where $P(x^\star, y^\star, z^\star) = 0$. The objective is to identify this location. Cauchy (1847) worked with nonnegative and continuous functions $P(x, y, z)$ so that finding their roots corresponds to finding minimizers for the function. He argued that starting from some initial guess for $(x^\star, y^\star, z^\star)$, one can repeatedly move to new locations (x, y, z) where the values of the function continue to decrease — a description of Cauchy's argument is given by Lemaréchal (2012). Cauchy identified the direction of the update in terms of the negative gradient of the function. If we let (P_x, P_y, P_z) denote the partial derivatives of $P(x, y, z)$ relative to its individual arguments, then Cauchy proposed the following recursive form:

$$x(n) = x(n-1) - \mu\, P_x\Big(x(n-1), y(n-1), z(n-1)\Big) \tag{12.184a}$$

$$y(n) = y(n-1) - \mu\, P_y\Big(x(n-1), y(n-1), z(n-1)\Big) \tag{12.184b}$$

$$z(n) = z(n-1) - \mu\, P_z\Big(x(n-1), y(n-1), z(n-1)\Big) \tag{12.184c}$$

where μ is a small positive parameter. If we introduce the vector notation $w = \text{col}\{x, y, z\}$, this construction can be rewritten as

$$w_n = w_{n-1} - \mu\, \nabla_{w^{\mathsf T}} P(w_{n-1}) \tag{12.185}$$

which is the gradient-descent step we considered in (12.22). Cauchy did not analyze the convergence of his procedure. Convergence studies appeared later, e.g., in works by Curry (1944) and Goldstein (1962). For further discussion on gradient and *steepest-descent* methods, the reader may refer to Polyak (1987), Fletcher (1987), Nash and Sofer (1996), Luenberger and Ye (2008), Bertsekas (1995), Bertsekas and Tsitsiklis (1997), and Sayed (2014a). The convergence analysis given in the text for gradient-descent algorithms for both constant and decaying step-sizes follows the presentations by Polyak (1987) and Sayed (2014a). The argument in Sec. 12.4.2 for the convergence of the backtracking method is based on the analysis in Boyd and Vandenberghe (2004); see also Curry (1944), Wolfe (1969,1971), Goldstein (1966), and Kelley (1996). The Armijo condition (12.106) is from Armijo (1966). Some further applications of gradient-descent to batch algorithms appear in Bottou (1998), Bottou and LeCun (2004), Le Roux, Schmidt, and Bach (2012), and Cevher, Becker, and Schmidt (2014).

Momentum acceleration methods. In Probs. 12.9 and 12.11 we describe two popular methods to accelerate the convergence of gradient-descent methods by incorporating *momentum* terms into the update recursion. One method is known as the *heavy-ball* implementation or *Polyak momentum acceleration* and is due to Polyak (1964,1987) and Polyak and Juditsky (1992). This method modifies the gradient-descent recursion

(12.185) by adding a driving term that is proportional to the difference of the last two iterates, namely,

$$w_n = w_{n-1} - \mu \nabla_{w^\mathsf{T}} P(w_{n-1}) \; + \; \beta(w_{n-1} - w_{n-2}), \quad n \geq 0 \tag{12.186}$$

The scalar $0 \leq \beta < 1$ is called the *momentum parameter*. It is shown in Prob. 12.10 that recursion (12.186) can be described in the equivalent form:

(**Polyak momentum acceleration**)

$$\left\{ \begin{array}{l} b_n = \nabla_{w^\mathsf{T}} P(w_{n-1}) \\ \bar{b}_n = \beta \bar{b}_{n-1} + b_n, \quad \bar{b}_{-1} = 0 \\ w_n = w_{n-1} - \mu \bar{b}_n \end{array} \right. \tag{12.187}$$

which helps clarify the role of the momentum term. In this description, the gradient vector is denoted by b_n. It is seen that b_n is smoothed over time into $\bar{b}_n$, and the smoothed direction $\bar{b}_n$ is used to update the weight iterate from w_{n-1} to w_n. By doing so, momentum helps reinforce search directions with more pronounced progress towards the location of the sought-after minimizer.

A second momentum method is known as *Nesterov momentum acceleration* and is due to Nesterov (1983,2004,2005). It modifies the gradient-descent recursion in the following manner:

$$w_n = w_{n-1} - \mu \nabla_{w^\mathsf{T}} P\Big(w_{n-1} + \beta(w_{n-1} - w_{n-2})\Big) \; + \; \beta(w_{n-1} - w_{n-2}), \quad n \geq 0 \tag{12.188}$$

Compared with Polyak momentum (12.186), we find that the main difference is that the gradient vector is evaluated at the intermediate iterate $w_{n-1} + \beta(w_{n-1} - w_{n-2})$. It is shown in Prob. 12.12 that recursion (12.188) can be described in the equivalent form:

(**Nesterov momentum acceleration**)

$$\left\{ \begin{array}{l} w'_{n-1} = w_{n-1} - \mu\beta \bar{b}_{n-1} \\ b'_n = \nabla_{w^\mathsf{T}} P(w'_{n-1}) \\ \bar{b}_n = \beta \bar{b}_{n-1} + b'_n, \quad \bar{b}_{-1} = 0 \\ w_n = w_{n-1} - \mu \bar{b}_n \end{array} \right. \tag{12.189}$$

That is, we first adjust w_{n-1} to the intermediate value w'_{n-1} and denote the gradient at this location by b'_n. We smooth this gradient over time and use the smoothed direction $\bar{b}_n$ to update the weight iterate.

When the risk function $P(w)$ is ν-strongly convex and has δ-Lipschitz gradients, both momentum methods succeed in accelerating the gradient descent method to attain a faster exponential convergence rate, and this rate has been proven to be optimal for problems with smooth $P(w)$ and cannot be attained by the standard gradient descent method — see Polyak (1987) and Nesterov (2004). Specifically, it is shown in these references that the convergence of the squared error $\|\widetilde{w}_n\|^2$ to zero occurs for these acceleration methods at the rate — see Prob. 12.9:

$$\|\widetilde{w}_n\|^2 \leq \left(\frac{\sqrt{\delta} - \sqrt{\nu}}{\sqrt{\delta} + \sqrt{\nu}} \right)^2 \|\widetilde{w}_{n-1}\|^2 \tag{12.190}$$

In contrast, in Theorem 2.1.15 of Nesterov (2005) and Theorem 4 in Section 1.4 of Polyak (1987), the fastest rate for the gradient-descent method is shown to be — see Prob. 12.5:

$$\|\widetilde{w}_n\|^2 \leq \left(\frac{\delta - \nu}{\delta + \nu} \right)^2 \|\widetilde{w}_{n-1}\|^2 \tag{12.191}$$

It can be verified that

$$\frac{\sqrt{\delta}-\sqrt{\nu}}{\sqrt{\delta}+\sqrt{\nu}} < \frac{\delta-\nu}{\delta+\nu} \tag{12.192}$$

for $\nu < \delta$. This inequality confirms that the momentum algorithms can achieve faster rates in minimizing strongly-convex risks $P(w)$ with Lipschitz gradients and that this faster rate cannot be attained by standard gradient descent.

Newton and quasi-Newton methods. In Sec. 12.3 we motivated the gradient-descent recursion for minimizing a first-order differentiable convex risk function. This technique is a *first-order method* since it relies solely on gradient calculations. There are other iterative techniques that can be used for the same purpose. If we examine the derivation that led to the gradient-descent recursion (12.22) starting from (12.20), we observe that the search direction may be scaled by any positive-definite matrix, say, by A^{-1}, so that the corresponding iteration (12.22) would become

$$w_n = w_{n-1} - \mu\, A^{-1}\, \nabla_{w^{\mathsf{T}}}\, P(w_{n-1}), \;\; n \geq 0 \tag{12.193}$$

The step-size parameter μ can be incorporated into A if desired. This construction is referred to as the *quasi-Newton method*. If A is diagonal with individual positive entries, $A = \text{diag}\{a(1), a(2), \ldots, a(M)\}$, one for each entry of w, then we end up with a gradient-descent implementation where a separate step-size is used for each individual entry of the weight iterate. The value of A can also vary with n. A second popular procedure is Newton method (also called Newton-Raphson) described below, which is a *second-order method* since it uses the Hessian matrix of the risk function. A third procedure is the *natural gradient method* encountered in (6.131), and which replaces the Hessian matrix by the Fisher information matrix. We elaborate here on Newton method.

The Newton-Raphson method is named after the English mathematician and physicist **Isaac Newton (1643–1727)**, whose contribution appeared in Wallis (1685), and also after Newton's contemporary Raphson (1697). Both Newton and Raphson were interested in finding roots of polynomials. According to the account by Kollerstrom (1992), Newton's original method was not iterative, while Raphson's method was not expressed in differential form. It was the British mathematician **Thomas Simpson (1710-1761)** who introduced the current form of the method in Simpson (1740) — see the account by Christensen (1996). Thus, consider a $\nu-$strongly convex second-order differentiable risk function, $P(w)$. Let δw denote a small perturbation vector. We approximate the value $P(w + \delta w)$ in terms of a second-order Taylor series expansion of $P(w)$ around w as follows:

$$P(w+\delta w) \approx P(w) + \Big(\nabla_{w^{\mathsf{T}}} P(w)\Big)^{\mathsf{T}} \delta w + \frac{1}{2}\,(\delta w)^{\mathsf{T}}\, \nabla_w^2 P(w) \delta w \tag{12.194}$$

We select δw to minimize the difference $P(w + \delta w) - P(w)$. We differentiate the expression on the right-hand side with respect to δw and set the result to zero leading to:

$$\nabla_{w^{\mathsf{T}}} P(w) + \nabla_w^2 P(w) \delta w = 0 \tag{12.195}$$

When $P(w)$ is $\nu-$strongly convex, its Hessian matrix satisfies $\nabla_w^2 P(w) \geq \nu I$ and is therefore positive-definite and invertible. It follows that the optimal perturbation is given by

$$\delta w^o = -\left(\nabla_w^2 P(w)\right)^{-1} \nabla_{w^{\mathsf{T}}}\, P(w) \tag{12.196}$$

Starting from the iterate w_{n-1}, Newton method updates it to $w_n = w_{n-1} + \delta w^o$ leading to listing (12.197). Often, a small correction ϵI_M is added to the Hessian matrix to avoid degenerate situations and ensure the inverse operation is valid.

Newton method for minimizing a risk function $P(w)$.

```
given gradient operator, $\nabla_{w^{\mathsf{T}}} P(w)$;
given Hessian operator, $\nabla_w^2 P(w)$;
given small $\epsilon > 0$;
start from arbitrary $w_{-1}$.                                  (12.197)
repeat until convergence over $n \geq 0$:
    $A_{n-1} = \epsilon I_M + \nabla_w^2 P(w_{n-1})$
    $w_n = w_{n-1} - A_{n-1}^{-1} \nabla_{w^{\mathsf{T}}} P(w_{n-1})$
end
return $w^\star \leftarrow w_n$.
```

We can verify that δw^o plays the role of a "descent" direction by noting that — see also Prob. 12.7:

$$\begin{aligned}
&P(w+\delta w^o) - P(w)\\
&= \left(\nabla_{w^{\mathsf{T}}} P(w)\right)^{\mathsf{T}} \delta w^o + \frac{1}{2}\left(\delta w\right)^{\mathsf{T}} \nabla_w^2 P(w) \delta w^o\\
&= -\left(\nabla_{w^{\mathsf{T}}} P(w)\right)^{\mathsf{T}} \left(\nabla_w^2 P(w)\right)^{-1} \nabla_w P(w) +\\
&\qquad \frac{1}{2}\left(\nabla_{w^{\mathsf{T}}} P(w)\right)^{\mathsf{T}} \left(\nabla_w^2 P(w)\right)^{-1} \nabla_w^2 P(w) \left(\nabla_w^2 P(w)\right)^{-1} \nabla_{w^{\mathsf{T}}} P(w)\\
&= -\left(\nabla_{w^{\mathsf{T}}} P(w)\right)^{\mathsf{T}} \left(\nabla_w^2 P(w)\right)^{-1} \nabla_w P(w) +\\
&\qquad \frac{1}{2}\left(\nabla_{w^{\mathsf{T}}} P(w)\right)^{\mathsf{T}} \left(\nabla_w^2 P(w)\right)^{-1} \nabla_{w^{\mathsf{T}}} P(w)\\
&= -\frac{1}{2}\left(\nabla_{w^{\mathsf{T}}} P(w)\right)^{\mathsf{T}} \left(\nabla_w^2 P(w)\right)^{-1} \nabla_w P(w)\\
&< 0
\end{aligned} \tag{12.198}$$

BFGS method. Newton method (12.197) selects the update term δw^o in terms of the inverse of the Hessian matrix of the risk function, namely,

$$\delta w^o = -\left(\nabla_w^2 P(w_{n-1})\right)^{-1} \nabla_{w^{\mathsf{T}}} P(w_{n-1}) \tag{12.199}$$

The BFGS method, named after the initials of its independent developers Broyden (1970), Fletcher (1970), Goldfarb (1970), and Shanno (1970), is a quasi-Newton method that approximates the Hessian matrix. It employs instead

$$\delta w^o = -B_{n-1}^{-1} \nabla_{w^{\mathsf{T}}} P(w_{n-1}) \tag{12.200}$$

for some positive-definite matrix B_{n-1} updated recursively:

$$B_n = B_{n-1} + \alpha a_n a_n^{\mathsf{T}} + \beta b_n b_n^{\mathsf{T}} \tag{12.201}$$

through the addition of two rank-one matrices to B_{n-1}. The scalars $\{\alpha, \beta\}$ are chosen to enforce a certain constraint on the successive matrices $\{B_n\}$ as follows. Introduce first the vectors:

$$a_n \overset{\Delta}{=} \nabla_{w^{\mathsf{T}}} P(w_n) - \nabla_{w^{\mathsf{T}}} P(w_{n-1}) \tag{12.202a}$$

$$z_n \overset{\Delta}{=} w_n - w_{n-1} \tag{12.202b}$$

$$b_n \overset{\Delta}{=} B_{n-1} z_n \tag{12.202c}$$

Note that a_n is the difference between two successive gradient vectors, while z_n is the difference between two successive iterates. The vector b_n is the result of transforming

z_n by B_{n-1}. The scalars $\{\alpha, \beta\}$ are selected to ensure the following constraint (also called the *secant* equation):

$$B_n(w_n - w_{n-1}) = \nabla_{w^\mathsf{T}} P(w_n) - \nabla_{w^\mathsf{T}} P(w_{n-1}) \tag{12.203}$$

That is, $B_n z_n = a_n$. This condition is reminiscent of the classical *secant method* for finding the roots of a function $f(x)$ over $x \in \mathbb{R}$, which is described by the recursion

$$x_{n+1} = x_n - f(x_n) \times \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \tag{12.204}$$

The BFGS method seeks a root for the equation $\nabla_{w^\mathsf{T}} P(w) = 0$. It does so by extending construction (12.204) to the vector case through the imposition of (12.203). It is easily verified, by multiplying the update relation for B_n from the left by z_n, that (12.203) is satisfied for the choices:

$$\alpha = \frac{1}{z_n^\mathsf{T} a_n}, \quad \beta = -\frac{1}{z_n^\mathsf{T} B_{n-1} z_n} \tag{12.205}$$

We arrive at listing (12.206) where the initial matrix can be selected as $B_{-1} = I_M$. Problem 12.35 derives an expression that update B_{n-1}^{-1} to B_n^{-1} directly and shows that the successive matrices B_n are positive-definite. For more details, see Fletcher (1987), Kelley (1996), and Nocedal and Wright (2006).

BFGS method for minimizing a risk function $P(w)$.

```
given gradient operator, $\nabla_{w^\mathsf{T}} P(w)$;
given small step-size, $\mu > 0$;
start from arbitrary $w_{-1}$ and $B_{-1} > 0$.
repeat until convergence over $n \geq 0$:
    $w_n = w_{n-1} - \mu B_{n-1}^{-1} \nabla_{w^\mathsf{T}} P(w_{n-1})$
    $z_n = w_n - w_{n-1}$
    $a_n = \nabla_{w^\mathsf{T}} P(w_n) - \nabla_{w^\mathsf{T}} P(w_{n-1})$
    $b_n = B_{n-1} z_n$
    $B_n = B_{n-1} + \frac{1}{z_n^\mathsf{T} a_n} a_n a_n^\mathsf{T} - \frac{1}{z_n^\mathsf{T} B_{n-1} z_n} b_n b_n^\mathsf{T}$
end
return $w^\star \leftarrow w_n$.
```

(12.206)

Method of coordinate-descent. The coordinate-descent method is a simple and effective technique for the solution of optimization problems; it relies on a sequence of coordinate-wise steps to reduce an $M-$dimensional problem to M one-dimensional problems. In each step, the optimization is carried out over a single entry (or coordinate) of the parameter vector w, while holding all other entries fixed at their current estimated values. Ideally, when the optimization can be carried out in closed form, the coordinate-descent construction minimizes $P(w)$ over its individual coordinates in sequence before repeating the iteration, as was shown in listing (12.119). In practice though, the optimization step is generally difficult to compute analytically and it is replaced by a gradient-descent step. The classical implementation of coordinate-descent cycles through all coordinates, while more popular variants update one coordinate per iteration. This coordinate is selected either uniformly at random, which is one of the most popular variants, or as the coordinate that corresponds to the maximal absolute gradient value. For more details, the reader may refer to Nesterov (2012), which studies the case of smooth functions under both convexity and strong convexity conditions. The work by Richtárik and Takác (2011) simplifies the results and considers the case of smooth plus separable risks. There are many variants of coordinate-descent implementations, including the important case where blocks of coordinates (rather than a single coordinate) are updated at the same time — see, e.g., the convergence analysis in Tseng (2001) where the future separable form (14.123) for non-smooth risks is studied in some detail. In the block case, the parameter w is divided into sub-blocks, say, $w = \text{blkcol}\{w_1, w_2, \ldots, w_B\}$ where each w_b is now a sub-vector with multiple entries in it. Then, the same constructions we described in the body of the chapter will apply working with block sub-vectors rather than single coordinates as shown, for example, in listing (12.207).

Block coordinate-descent for minimizing a risk function $P(w)$.

```
let $w = \text{blkcol}\{w_1, w_2, \ldots, w_B\}$;
start from an arbitrary initial condition $w_{-1}$.
repeat until convergence over $n \geq 0$:
    $w_{n-1}$ with $B$ blocks $\text{blkcol}\{w_{n-1,b}\}$ is available at start of iteration;
    for each block $b = 1, 2, \ldots, B$ compute:
        $w_{n,b} = \underset{w_b}{\text{argmin}}\; P\Big(w_{n-1,1}, \ldots, \boxed{w_b}, \ldots, w_{n-1,B}\Big)$
    end
end
return $w^\star \leftarrow w_n$.
```
(12.207)

The idea of estimating one component at a time while fixing all other components at their current values appears already in the classical Gauss-Seidel approach to solving linear systems of equations — see, e.g., Golub and Van Loan (1996). The Gauss-Seidel approach is accredited to the German mathematicians **Carl Friedrich Gauss (1777-1855)** and Seidel (1874). Gauss described the method 50 years prior to Seidel in a correspondence from 1823 — see the collection of works by Gauss (1903). In more recent times, some of the earliest references on the use of coordinate-descent in optimization include Hildreth (1957), where block coordinate descent was first introduced, in addition to Warga (1963) and Ortega and Rheinboldt (1970). The Gauss-Southwell (GS) rule that amounts to selecting the gradient component with the largest absolute value in (12.145) is also due to Gauss (1903) and Southwell (1940). Some analysis on comparing the convergence rates of the GS and randomized rules appear in Nutini *et al.* (2015), where it is argued that the GS rule leads to improved convergence — see Prob. 12.19; this argument is consistent with the steepest-descent derivation of the GS rule in Sec. 12.5.3 and is related to the derivation used in the proof of Theorem 12.5. More recent applications of coordinate-descent in computer tomography, machine learning, statistics, and multi-agent optimization appear, for example, in Luo and Tseng (1992a), Sauer and Bouman (1993), Fu (1998), Daubechies, Defrise, and De Mol (2004), Friedman *et al.* (2007), Wu and Lange (2008), Chang, Hsieh, and Lin (2008), Tseng and Yun (2009, 2010), Beck and Tetruashvili (2013), Lange, Chi, and Zhou (2014), Wright (2015), Shi *et al.* (2017), Wang *et al.* (2018), Fercoq and Bianchi (2019), and the many references therein.

Alternating projection method. We described in Sec. 12.6 the *alternating projection algorithm* (12.171) for finding points in the intersection of two closed convex sets, $\mathcal{C}_1$ and $\mathcal{C}_2$; the technique is also known as the *successive projection method.* It has found applications in a wide range of fields including statistics, optimization, medical imaging, machine learning, and finance. The algorithm was originally developed by the Hungarian-American mathematician **John von Neumann (1903–1957)** in 1933 in unpublished lecture notes, which appeared later in press in the works by von Neumann (1949,1950) on operator theory. von Neumann's work focused on the intersection of two affine subspaces in Hilbert space (such as hyperplanes), and it was subsequently extended by Halperin (1962) to the intersection of multiple affine subspaces and by Bregman (1965) to the intersection of multiple closed convex sets. For the benefit of the reader, a set $\mathcal{S} \subset \mathbb{R}^M$ is an affine subspace if every element $s \in \mathcal{S}$ can be written as $s = p + v$, for some fixed p and where $v \in \mathcal{V}$ denotes an arbitrary element from a vector space. For example, the set $\mathcal{S} = \{x \,|\, a^{\mathsf{T}} x = b\}$ is an affine subspace. If we let $\widehat{x}$ denote any solution to $a^{\mathsf{T}} x = b$, then any element $s \in \mathcal{S}$ can be written as $s = \widehat{x} + v$ for any $v \in \mathcal{N}(a)$.

The proof of Theorem 12.6 assumes initially that the convex sets have a nontrivial intersection. When the sets do not intersect, the argument reveals through expression (12.183), as was discovered by Cheney and Goldstein (1959), that the alternating projection algorithm converges to points that are closest to each other from both sets. This fact was illustrated schematically in Fig. 12.8 and it has many useful applications. In particular, the result shows that the alternating projection method can be used to check whether a collection of convex sets intersect or not (such as checking whether the constraints in a convex optimization problem of the form (9.1) admit feasible solutions). It can also be used to determine the minimum distance between two convex sets. Moreover,we can use any hyperplane that is orthogonal to the segment connecting $a^\star$ and $b^\star$ to separate the two convex sets from each other: one set will be on one side of the hyperplane while the other set will be on the other side. This observation is useful for classification problems, as we will explain in later chapters when we discuss linearly separable datasets. The presentation in the chapter benefited from useful overviews on the alternating projection method, including proofs for its convergence properties, given in the works by Cheney and Goldstein (1959), Bregman (1967), Gubin, Polyak, and Raik (1967), Combettes (1993), Bauschke and Borwein (1996), Escalante and Raydan (2011), and Dattoro (2016). Although we have focused on finding points in the intersection of *two* convex sets, the algorithm can be applied to a larger number of sets by projecting sequentially onto the sets, one at a time, with minimal adjustments to the arguments, leading to what is known as the *cyclic projection algorithm* — see Prob. 12.34.

The alternating projection method suffers from one inconvenience when the intersection set $\mathcal{C}_1 \cap \mathcal{C}_2$ has more than one point. Starting from an initial vector a_{-1}, the method generates two sequences $\{a_n \in \mathcal{C}_1\}$ and $\{b_n \in \mathcal{C}_2\}$ that are only guaranteed to converge to some *arbitrary* point in the intersection. An elegant variation is Dykstra algorithm listed in (12.208) and which was developed by Dysktra (1983); see also Boyle and Dykstra (1986). The same method was rediscovered by Han (1988).

Dykstra alternating projection algorithm.

```
given two closed convex sets $\mathcal{C}_1$ and $\mathcal{C}_2$;
given projection operators onto $\mathcal{C}_1$ and $\mathcal{C}_2$;
start from $a_{-1} \in \mathcal{C}_1$, and set $c_{-1} = d_0 = 0$;
objective:
   if $\mathcal{C}_1 \cap \mathcal{C}_2 \neq \emptyset$: find projection $w^\star$ of $a_{-1}$ onto the intersection;
   else find points $\{a^\star \in \mathcal{C}_1, b^\star \in \mathcal{C}_2\}$ closest to each other;
repeat until convergence over $n \geq 0$:                                   (12.208)
   $b_n = \mathcal{P}_{C_2}(a_{n-1} + c_{n-1})$
   $c_n = a_{n-1} + c_{n-1} - b_n$,        (residual)
   $a_{n+1} = \mathcal{P}_{C_1}(b_n + d_n)$
   $d_{n+1} = b_n + d_n - a_{n+1}$,        (residual)
end
if $\|a_n - b_n\|$ small, return $w^\star \leftarrow a_n$;
   else return $a^\star \leftarrow a_n$, $b^\star \leftarrow b_n$.
end
```

The Dykstra method ensures convergence to the point $w^\star$ in the intersection that is closest to a_{-1}, namely,

$$\|a_{-1} - w^\star\| \leq \|a_{-1} - w\|, \quad \text{for any } w \in \mathcal{C}_1 \cap \mathcal{C}_2 \tag{12.209}$$

That is, the method ends up determining the projection of a_{-1} onto $\mathcal{C}_1 \cap \mathcal{C}_2$. Compared with the classical formulation (12.171), the Dykstra method (12.208) introduces two auxiliary vectors $\{c_n, d_n\}$ and performs the calculations shown in the table repeatedly starting from the same initial vector $a_{-1} \in \mathcal{C}_1$ and using $c_{-1} = d_0 = 0$.

Using the traditional alternating projection recursions, and the Dykstra variation, we are able to solve the following two types of problems:

$$(\textbf{feasibility problem}) : \text{finding } w^\star \in \mathcal{C}_1 \cap \mathcal{C}_2 \tag{12.210}$$

$$(\textbf{projection problem}) : \text{finding } \mathcal{P}_{C_1 \cap C_2}(x), \text{ for a given } x. \tag{12.211}$$

Both problems are solved by working independently with the projection operators $\mathcal{P}_{\mathcal{C}_1}(x)$ and $\mathcal{P}_{\mathcal{C}_2}(x)$, and by applying them alternately starting from x.

Zeroth-order optimization. The gradient-descent algorithm described in the body of the chapter is an example of a first-order optimization method, which requires the availability of the gradient information, $\nabla_w P(w)$, in order to perform the update (12.22). In some situations of interest, it may not be possible to evaluate the gradient function either because the risk $P(w)$ may not have a closed analytical form or is unknown to the designer altogether — see, e.g., Brent (2002) and Conn, Scheinberg, and Vicente (2009) for examples. The latter situation arises, for example, in adversarial learning scenarios, studied in future Chapter 71, where the designer wishes to misguide the operation of a learning algorithm and is only able to perform function evaluations of the risk function, $P(w)$, at different data samples. We provide a brief review of zeroth-order optimization, also known as *derivative-free optimization* in Appendix 12.A. This technique enables the designer to approximate the gradient vector by relying solely on function evaluations. One of the earliest references on the use of finite difference approximations for gradient evaluations is Kiefer and Wolfowitz (1952). We explain in the appendix how gradient vectors can be approximated by means of two function evaluations in a manner that satisfies the useful unbiasedness property (12.218). The proof given for this latter property follows Nesterov and Spokoiny (2017) and Flaxman, Kalai, and McMahan (2005). There have been several works in more recent years on the performance of optimization algorithms based on such constructions. They are slower to converge than traditional gradient descent and have been shown to require at least M times more iterations to converge. The slowdown in performance is due to the error variance in estimating the true gradient vector. Results along these lines can be found in Wibisono *et al.* (2012), Nesterov and Spokoiny (2017), and Liu *et al.* (2018). Overviews of gradient-free optimization appear in Rios and Sahinidis (2013), Duchi *et al.* (2015), Larson, Menickelly, and Wild (2019), and Liu *et al.* (2020).

PROBLEMS

12.1 Establish that for $\nu-$strongly convex risk functions $P(w)$ with $\delta-$Lipschitz gradients it holds that

$$\frac{2}{\delta}\left(P(w) - P(w^\star)\right) \;\leq\; \|\widetilde{w}\|^2 \leq \frac{2}{\nu}\left(P(w) - P(w^\star)\right)$$

12.2 Show that the ℓ_2-regularized least-squares risk listed in Table 12.1 satisfies condition (12.15). Determine the values for $\{\nu, \delta\}$.

12.3 Assume all we know is that $P(w)$ is twice-differentiable over w and satisfies condition (12.15). Establish the validity of Theorem 12.1.

12.4 Assume that $P(w)$ is twice-differentiable over w and satisfies (12.15).

(a) Use the mean-value relation (10.10) to show that the error vector satisfies a recursion of the form $\widetilde{w}_n = (I_M - \mu H_{n-1})\widetilde{w}_{n-1}$ where

$$H_{n-1} \stackrel{\Delta}{=} \int_0^1 \nabla_w^2 \, P(w^\star - t\widetilde{w}_{n-1})dt$$

(b) Show that the conclusions of Theorem 12.1 continue to hold over the wider step-size interval $\mu < 2/\delta$, which is independent of ν.

12.5 Problems 12.5–12.9 are motivated by results from Polyak (1987, Chs. 1, 3). Consider the same setting of Prob. 12.4. Show that convergence of $\|\widetilde{w}_n\|^2$ to zero also occurs at an exponential rate that is given by $\lambda_2 = \max\{(1-\mu\delta)^2, (1-\mu\nu)^2\}$. Conclude that the convergence rate is fastest when the step-size is chosen as $\mu^o = 2/(\nu+\delta)$ for which $\lambda_2^o = (\delta-\nu)^2/(\delta+\nu)^2$. Roughly, how many iterations are needed for the squared error, $\|\widetilde{w}_n\|^2$, to fall below a small threshold value, ϵ?

12.6 Let $P(w)$ be a real-valued first-order differentiable risk function whose gradient vector satisfies the $\delta-$Lipschitz condition (12.12b). The risk $P(w)$ is *not* assumed convex. Instead, we assume that it is lower-bounded, namely, $P(w) \geq L$ for all w and for some finite value L. Consider the gradient-descent algorithm (12.22). Show that if the step-size μ satisfies $\mu < 2/\delta$, then the sequence of iterates $\{w_n\}$ satisfies the following two properties:

(a) $P(w_n) \leq P(w_{n-1})$.

(b) $\lim_{n\to\infty} \nabla_w P(w_n) = 0$.

12.7 Let $P(w)$ denote a real-valued $\nu-$strongly convex and twice-differentiable cost function with $w \in \mathbb{R}^M$. Assume the Hessian matrix of $P(w)$ is $\delta-$Lipschitz continuous, i.e., $\left\|\nabla_w^2 P(w_2) - \nabla_w^2 P(w_1)\right\| \leq \delta\|w_2 - w_1\|$. The global minimizer of $P(w)$ is sought by means of Newton method:

$$w_n = w_{n-1} - \left(\nabla_w^2 \, P(w_{n-1})\right)^{-1} \nabla_{w^\mathsf{T}} \, P(w_{n-1}), \quad n \geq 0$$

which employs the inverse of the Hessian matrix. The initial condition is denoted by w_{-1}. Let $\lambda \stackrel{\Delta}{=} \left(\delta/2\nu^2\right)^2 \|\nabla_w P(w_{-1})\|^2$, and assume $\lambda < 1$. Show that $\|\widetilde{w}_n\|^2$ converges to zero at the rate

$$\|\widetilde{w}_n\|^2 \leq \left(\frac{2\nu^2}{\delta}\right) \lambda^{2^n}$$

Conclude that the convergence rate is now dependent on the quality of the initial condition.

12.8 Consider the gradient-descent recursion (12.65) where the step-size sequence is selected as

$$\mu(n) = \frac{\tau}{(n+1)^q}, \qquad 1/2 < q \leq 1, \quad \tau > 0$$

(a) Verify that the step-size sequence satisfies conditions (12.66b).

(b) Follow the proof of Theorem 12.2 to determine the rate of convergence of $\|\widetilde{w}_n\|^2$ to zero.

(c) For a fixed τ, which value of q in the range $1/2 < q \leq 1$ results in the fastest convergence rate?

12.9 Let $P(w)$ be a real-valued risk function, assumed $\nu-$strongly convex, first-order differentiable at all $w \in \text{dom}(P)$, and with $\delta-$Lipschitz gradients as in (12.12a)–(12.12b). Consider a *heavy-ball* implementation of gradient-descent algorithm, which is a form of *momentum acceleration*, also known as Polyak momentum method, and given by:

$$w_n = w_{n-1} - \mu\nabla_{w^\mathsf{T}} \, P(w_{n-1}) \; + \; \beta(w_{n-1} - w_{n-2}), \quad n \geq 0$$

where the past iterate w_{n-2} is also used in the update equation, and $0 \leq \beta < 1$ is called the momentum parameter. Assume the initial conditions w_{-1} and w_{-2} lie sufficiently close to $w^\star$, i.e., $\|\widetilde{w}_{-1}\|^2 < \epsilon$ and $\|\widetilde{w}_{-2}\|^2 < \epsilon$ for some small enough ϵ.

(a) Show that if $0 < \mu < 2(1+\beta)/\delta$, then $\|\widetilde{w}_n\|^2$ converges to zero at the exponential rate $O(\lambda_3^n)$. Identify λ_3 and show that optimal values for $\{\mu, \beta, \lambda_3\}$ are

$$\mu^o = \frac{4}{\left(\sqrt{\delta}+\sqrt{\nu}\right)^2}, \qquad \beta^o = \left(\frac{\sqrt{\delta}-\sqrt{\nu}}{\sqrt{\delta}+\sqrt{\nu}}\right)^2, \qquad \lambda_3^o = \beta^o$$

(b) Let $\kappa = \delta/\nu$. Large values for κ indicate ill-conditioned Hessian matrices, $\nabla_w^2 P(w)$, since their spectra will lie over wider intervals. Let λ_2^o denote the optimal rate of convergence when $\beta = 0$. We already know from Prob. 12.5 that $\lambda_2^o = (\delta - \nu)^2/(\delta+\nu)^2$. Argue that for large κ:

$$\lambda_2^o \approx 1 - 2/\kappa, \qquad \lambda_3^o \approx 1 - 2/\sqrt{\kappa}$$

Compare the number of iterations that are needed for $\|\widetilde{w}_n\|^2$ to fall below a threshold ϵ for both cases of $\beta = 0$ and $\beta = \beta^o$.

12.10 Show that the momentum method that updates w_{n-1} to w_n in Prob. 12.9 can be described in the equivalent form:

$$\begin{aligned} b_n &\stackrel{\Delta}{=} \nabla_{w^\mathsf{T}} P(w_{n-1}) \\ \bar{b}_n &= \beta \bar{b}_{n-1} + b_n, \quad \bar{b}_{-1} = 0 \\ w_n &= w_{n-1} - \mu \bar{b}_n \end{aligned}$$

Remark. We will encounter this construction later in Sec. 17.5 when we study adaptive gradient methods with momentum acceleration.

12.11 Consider the same setting of Prob. 12.9. A second momentum implementation is Nesterov momentum method, which is given by the following recursion:

$$w_n = w_{n-1} - \mu \nabla_{w^\mathsf{T}} P\Big(w_{n-1} + \beta(w_{n-1} - w_{n-2})\Big) \; + \; \beta(w_{n-1} - w_{n-2}), \quad n \geq 0$$

Compared with Polyak momentum from Prob. 12.10 we find that the main difference is that the gradient vector is evaluated at the intermediate iterate $w_{n-1}+\beta(w_{n-1}-w_{n-2})$. Study the convergence properties of Nesterov method in a manner similar to Prob. 12.9.

Remark. For more details on this implementation and its convergence properties, see Nesterov (1983,2004) and Yu, Jin, and Yang (2019).

12.12 Show that the Nesterov momentum method that updates w_{n-1} to w_n in Prob. 12.11 can be described in the equivalent form:

$$\begin{aligned} w'_{n-1} &\stackrel{\Delta}{=} w_{n-1} - \mu\beta \bar{b}_{n-1} \\ b'_n &\stackrel{\Delta}{=} \nabla_{w^\mathsf{T}} P(w'_{n-1}) \\ \bar{b}_n &= \beta \bar{b}_{n-1} + b'_n, \quad \bar{b}_{-1} = 0 \\ w_n &= w_{n-1} - \mu \bar{b}_n \end{aligned}$$

Remark. We will encounter this construction later in Sec. 17.5 when we study adaptive gradient methods with momentum acceleration.

12.13 Refer to the gradient-descent recursion (12.22) and assume that $P(w)$ is only convex (but not necessarily strongly-convex) with a $\delta-$Lipschitz gradient satisfying (12.12b). Let $\mu < 1/\delta$.

(a) Use property (11.120) for convex functions with $\delta-$Lipschitz gradients to argue that

$$P(w_n) \leq P(w_{n-1}) - \frac{\mu}{2}\|\nabla_w P(w_{n-1})\|^2$$

Conclude that $P(w_n)$ is non-increasing.

(b) Use part (a) to show that, for any $z \in \mathbb{R}^M$, it holds

$$P(w_n) \;\leq P(z) + \Big(\nabla_{w^\mathsf{T}} P(w_{n-1})\Big)(w_{n-1} - z) - \frac{\mu}{2}\|\nabla_w P(w_{n-1})\|^2$$

(c) Show that $P(w_n) - P(w^\star) \leq \frac{1}{2\mu}(\|\widetilde{w}_{n-1}\|^2 - \|\widetilde{w}_n\|^2)$.

(d) Conclude that $P(w_n) - P(w^\star) \leq \frac{1}{2\mu n}\|\widetilde{w}_0\|^2$ so that (12.64b) holds.

12.14 Refer to the gradient-descent recursion (12.22) and assume that $P(w)$ is only convex and $\delta-$Lipschitz, namely,

$$\|P(w_1) - P(w_2)\| \leq \delta\|w_1 - w_2\|, \quad \forall w_1, w_2 \in \text{dom}(P)$$

Observe that we are now assuming that $P(w)$ itself is Lipschitz rather than its gradient. We know from property (10.41) that the condition of a Lipschitz function translates into bounded gradient vectors, so that $\|\nabla_w P(w)\| \leq \delta$. Let $w^\star$ be a minimizer for $P(w)$ and assume $\|\widetilde{w}_{-1}\| \leq W$, where w_{-1} is the initial condition for the gradient-descent recursion.

(a) Use the convexity of $P(w)$, the gradient-descent recursion, and the bounded gradients, to verify that

$$P(w_{n-1}) - P(w^\star) \leq \frac{\mu\delta^2}{2} + \frac{1}{2\mu}\|\widetilde{w}_{n-1}\|^2 - \frac{1}{2\mu}\|\widetilde{w}_n\|^2$$

(b) Sum over the first N iterations to verify that

$$\frac{1}{N}\sum_{n=0}^{N-1} P(w_n) - P(w^\star) \leq \frac{\mu\delta^2}{2} + \frac{W^2}{2N\mu}$$

(c) Show that the upper bound is minimized for $\mu^o = \frac{W}{\delta\sqrt{N}}$ and and apply Jensen inequality to $P(w)$ to conclude that

$$P\Big(\frac{1}{N}\sum_{n=0}^{N-1} w_n\Big) - P(w^\star) \leq W\delta/\sqrt{N}$$

In other words, the risk value evaluated at the average iterate approaches the minimal value $P(w^\star)$ at the rate $O(1/\sqrt{N})$.

(d) Let w^{best} denote the iterate value that results in the smallest risk from among all iterates, namely, $w^{\text{best}} = \underset{0\leq n\leq N-1}{\text{argmin}}\; P(w_n)$. Conclude further that

$$P(w^{\text{best}}) - P(w^\star) \leq W\delta/\sqrt{N}$$

12.15 Refer to the gradient-descent algorithm (12.65) with decaying step-sizes. Repeat the argument from parts (a) and (b) of Prob. 12.14 to establish that

$$\sum_{n=0}^{N-1} \mu(n)\Big(P(w_n) - P(w^\star)\Big) \leq \frac{\delta}{2}\sum_{n=0}^{N-1}\mu^2(n) + \frac{W^2}{2}$$

Select $\mu(n) = \frac{c}{\sqrt{n+1}}$ for some constant c and show that

$$P(w^{\text{best}}) - P(w^\star) = O\Big(\ln(N)/\sqrt{N}\Big)$$

12.16 Refer to the backtracking line search method (12.103) and assume now that $P(w)$ is only convex. Repeat the argument of Prob. 12.13 to establish that

$$P(w_n) - P(w^\star) \leq \frac{1}{2\mu_{\text{bt}}\, n}\|\widetilde{w}_0\|^2$$

where $\mu_{\text{bt}} = \min\{1, \beta/\delta\}$.

12.17 Show that step-size sequences of the form (12.67) satisfy condition (12.66b).

12.18 Consider the gradient-descent recursion (12.65) with the step-size sequence selected as $\mu(n) = \tau/(n+1)^q$ where $\frac{1}{2} < q \leq 1$ and $\tau > 0$.

(a) Verify that the step-size sequence satisfies condition (12.66b).
(b) Follow the proof of Theorem 12.2 to determine the rate of convergence of $\|\widetilde{w}_n\|^2$ to zero.
(c) For a fixed τ, which value of q results in the fastest rate of convergence?

12.19 Refer to the statement of Theorem 12.5 for the Gauss-Southwell coordinate descent algorithm. The value of $\lambda \in [0,1)$ determines the convergence rate of the algorithm. We can establish a tighter result with a smaller $\lambda' \in [0,1)$, which would imply a faster rate than the one suggested by the theorem as follows. We know from the result of Prob. 8.60 that $P(w)$ is also ν_1-strongly convex relative to the infinity norm, i.e., expression (12.155a) implies

$$P(w_2) \;\geq\; P(w_1) \;+\; \Big(\nabla_{w^{\mathsf{T}}}\, P(w_1)\Big)^{\mathsf{T}}(w_2 - w_1) \;+\; \frac{\nu_1}{2}\|w_2 - w_1\|_\infty^2$$

where ν_1 satisfies $\frac{\nu}{M} \leq \nu_1 \leq \nu$. Repeat the argument in the proof of Theorem 12.5 to show that the algorithm continues to be stable for $\mu < 2/\delta$ with the excess risk now evolving according to

$$P(w_n) - P(w^\star) \;\leq\; \lambda'\Big(P(w_n) - P(w^\star)\Big)$$

where $\lambda' \;\overset{\Delta}{=}\; 1-2\mu\nu_1+\mu^2\nu_1\delta$. Verify that $\lambda' < \lambda$ when $\nu_1 > \nu/M$, and conclude that this result suggests a faster rate of convergence for the Gauss-Southwell coordinate descent recursion than the randomized coordinate-descent recursion. *Remark.* The reader may see Nutini *et al.* (2015) for a related discussion.

12.20 Consider a risk function $P(w)$ and let w_{n-1} denote an estimate for the minimizer of $P(w)$ at iteration $n-1$. Assume we are able to construct a function $G(w, w_{n-1})$ that satisfies the two conditions:

$$G(w, w_{n-1}) = P(w), \quad G(w, w_{n-1}) \geq P(w_{n-1}),\ \forall w$$

We say that $G(w, w_{n-1})$ "majorizes" $P(w)$ by bounding it from above. Note that the definition of $G(w)$ depends on w_{n-1}. Now set w_n to the minimizer of $G(w, w_{n-1})$, i.e.,

$$w_n = \underset{w\in\mathbb{R}^M}{\text{argmin}}\; G(w, w_{n-1})$$

Show that $P(w_n) \leq P(w_{n-1})$. That is, the updates constructed in this manner lead to non-increasing risks. This method of design is referred to as *majorization-minimization*; one example is encountered in future Example 58.2 in the context of the multiplicative update algorithm for nonnegative matrix factorization.

12.21 Consider the ℓ_2-regularized mean-square-error risk:

$$w^o = \underset{w\in\mathbb{R}^M}{\text{argmin}}\; \Big(\rho\|w\|^2 \;+\; \mathbb{E}\,(\boldsymbol{\gamma} - \boldsymbol{h}^{\mathsf{T}} w)^2\Big)$$

(a) Denote the risk in the above optimization problem by $P(w)$. Verify that $P(w)$ is quadratic in w and given by $P(w) = \sigma_\gamma^2 - 2r_{\gamma h}^{\mathsf{T}} w + w^{\mathsf{T}}(\rho I + R_h)w$.
(b) Show that $P(w)$ is $\nu-$strongly-convex and find a value for ν.
(c) Show that $P(w)$ has $\delta-$Lipschitz gradients and find a value for δ.
(d) Show that $w^o = (\rho I + R_h)^{-1} r_{h\gamma}$.
(e) In this case we know a closed-form expression for the optimal solution w^o. Still, let us determine its value iteratively. Show that the gradient-descent recursion (12.22) applied to the above $P(w)$ leads to

$$w_n = (1 - 2\rho\mu)w_{n-1} + 2\mu(r_{h\gamma} - R_h w_{n-1}), \;\; n \geq 0$$

(f) Let $\widetilde{w}_n = w^o - w_n$. Verify that $\widetilde{w}_n = ((1 - 2\rho\mu)I_M - 2\mu R_h)\widetilde{w}_{n-1}$.
(g) Show that $\widetilde{w}_n$ converges to zero for step-sizes μ satisfying $\mu < 2/(\rho+\lambda_{\max})$, where $\lambda_{\max}$ denotes the maximum eigenvalue of R_h (which is also equal to its spectral radius).

12.22 Is the objective function $P(w)$ defined by (12.164a) convex in w?

12.23 Refer to Newton recursion (12.197) for the minimization of $P(w)$ with $\epsilon = 0$. Introduce the change of variables $w = Az$ and write down Newton method for minimizing the function $Q(z) = P(Az)$ over z, namely,

$$z_n = z_{n-1} - \left(\nabla_w^2 Q(z_{n-1})\right)^{-1} \nabla_{z^\mathsf{T}} Q(z_{n-1})$$

Multiply by A from the left and show that the result reduces to Newton's original recursion over w_n for minimizing $P(w)$. Conclude that Newton method is invariant to affine scaling.

12.24 Consider the optimization problem (9.42) with linear equality constraints and the corresponding saddle-point formulation (9.44). Apply gradient-descent on w and gradient-ascent on the dual variable β by using

$$\begin{aligned} w_n &= w_{n-1} - \mu_w \nabla_{w^\mathsf{T}} P(w_{n-1}) - \mu_w B^\mathsf{T} \beta_{n-1},\ n \geq 0 \\ \beta_n &= \beta_{n-1} + \mu_\beta (Bw_n - c) \end{aligned}$$

where $\mu_w > 0$ and $\mu_\beta > 0$ are step-size parameters and w_{-1} and β_{-1} are arbitrary initial conditions satisfying $\beta_{-1} = Bw_{-1}$ or $\beta_{-1} = 0$. Note that the updated iterate w_n is used in the recursion for β_n instead of w_{n-1}; we say that we are implementing the recursions in an *incremental* form. Assume $P(w)$ is $\nu-$strongly convex and has $\delta-$Lipschitz gradients, i.e., for any w_1, w_2:

$$\|\nabla_{w^\mathsf{T}} P(w_1) - \nabla_{w^\mathsf{T}} P(w_2)\| \leq \delta \|w_1 - w_2\|$$

Consider the unique saddle-point $(w^\star, \beta_d^\star)$ defined by Lemma 9.2 and introduce the error quantities $\widetilde{w}_n = w^\star - w_n$ and $\widetilde{\beta}_n = \beta_d^\star - \beta_n$. Show that, for $\mu_w < 1/\delta$ and $\mu_\beta < \nu/\sigma_{\max}^2(B)$, the above recursions converge linearly to $(w^\star, \beta_d^\star)$, namely,

$$\|\widetilde{w}_n\|_{c_w}^2 + \|\widetilde{\beta}_n\|_{c_\beta}^2 \ \leq\ \rho \left(\|\widetilde{w}_{n-1}\|_{c_w}^2 + \|\widetilde{\beta}_{n-1}\|_{c_\beta}^2 \right)$$

where $c_\beta = \mu_w/\mu_\beta > 0$, $c_w = 1 - \mu_w \mu_\beta \sigma_{\max}^2(B) > 0$, and

$$\rho \ \stackrel{\Delta}{=}\ \max\Big\{ 1 - \mu_w \nu (1 - \mu_w \delta),\ 1 - \mu_w \mu_\beta \sigma_{\min}^2(B) \Big\} < 1$$

where $\sigma_{\min}(B)$ is the smallest nonzero singular value of B. *Remark*. For more details, the reader may refer to Alghunaim and Sayed (2020) and the discussion therein.

12.25 Consider the same setting of Prob. 12.24 except that the saddle-point $(w^\star, \beta_d^\star)$ is now sought by means of the following recursions

$$\begin{aligned} w_n &= w_{n-1} - \mu_w \nabla_{w^\mathsf{T}} P(w_{n-1}) - \mu_w B^\mathsf{T} \beta_{n-1},\ n \geq 0 \\ \beta_n &= \beta_{n-1} + \mu_\beta (Bw_{n-1} - c) \end{aligned}$$

The main difference is that w_{n-1} is used in the recursion for β_n instead of w_n. The above recursions, with w_{n-1} instead of w_n, are due to Arrow and Hurwicz (1956) and are known as the *Arrow-Hurwicz algorithm*. We will encounter an instance of it in future Example 44.10 when studying Markov decision processes. We will also comment on the history of the algorithm at the concluding remarks of that future chapter.

(a) Introduce the modified cost function $P_a(w) = P(w) - \frac{\mu_\beta}{2}\|Bw - c\|^2$. Write down the incremental recursions of Prob. 12.24 that would correspond to the problem of minimizing $P_a(w)$ subject to $Bw = c$. Verify that these recursions can be transformed into the Arrow-Hurwicz algorithm for $P(w)$ given above.

(b) Use the result of Prob. 12.24 to conclude that the Arrow-Hurwicz recursions also converge linearly to the unique saddle-point $(w^\star, \beta_d^\star)$ for $\mu_w < 1/(\delta + \mu_\beta \sigma_{\max}^2(B))$ and $\mu_\beta < \nu/2\sigma_{\max}^2(B)$.

12.26 Consider a $\nu-$strongly convex risk function $P(w)$ and apply the following gradient-descent algorithm for its minimization:

$$\boldsymbol{w}_n = \boldsymbol{w}_{n-1} - \mu \boldsymbol{D}_n \nabla_{w^{\mathsf{T}}} P(\boldsymbol{w}_{n-1})$$

where $\boldsymbol{D}_n$ is a diagonal matrix; each of its diagonal entries is either one with probability p or zero with probability $1-p$. Repeat an analysis similar to the proof of Theorem 12.4 to determine conditions for the convergence of this scheme.

12.27 Consider a non-smooth convex risk function $P(w) : \mathbb{R}^M \rightarrow \mathbb{R}$ for which a point $w^\star$ is found that satisfies (12.121). Does it follow that $w^\star$ is a global minimizer for $P(w)$? Consider the risk $P(w) = \|w\|^2 + |w_1 - w_2|$ and the point $(w_1, w_2) = (0.5, 0.5)$.

12.28 The following example is from Powell (1973). Consider a risk function over $\mathbb{R}^3$ of the form

$$P(w) = -w_1 w_2 - w_2 w_3 - w_1 w_3 + \sum_{k=1}^{3} \Big(|w_k| - 1 \Big)_+^2$$

where the notation $(x)_+ = x$ if $x \geq 0$ and zero otherwise.

(a) Is $P(w)$ convex over w?

(b) Verify that $P(w)$ has two minimizers at locations $\text{col}\{1,1,1\}$ and $\text{col}\{-1,-1,-1\}$.

(c) Choose the initial condition w_{-1} close to other vertices of the unit cube, other than the minimizers. Write down the corresponding coordinate-descent algorithm (12.119). Does it converge?

(d) Write down the corresponding gradient-descent algorithm. Does it converge?

12.29 Consider the optimization problem

$$W^\star = \underset{W}{\text{argmin}} \; \|X - WZ\|_{\text{F}}^2$$

where X is $M \times N$, W is $M \times K$, and Z is $K \times N$.

(a) Verify that the cost function can be rewritten as

$$P(W) = \text{Tr}\Big\{ (X - WZ)^{\mathsf{T}} (X - WZ) \Big\}$$

(b) Let $B = XZ^{\mathsf{T}}$ and $A = ZZ^{\mathsf{T}}$. Let also W_m denote the estimate for W at iteration m. Follow a coordinate-descent argument similar to the one leading to (12.125) to estimate the individual columns of W and show that the resulting algorithm involves an update of the form:

$$W_m = W_{m-1} + (B - W_{m-1}A)\text{diag}(A^{-1})$$

12.30 Refer to expression (12.142) and assume $\rho = 0$. Show that it can be rewritten in the equivalent form:

$$w_{n,m^o} \;=\; w_{n-1,m^o} + \frac{1}{\|x_{m^o}\|^2} x_{m^o}^{\mathsf{T}} \Big(\gamma_N - H_N w_{n-1} \Big)$$

12.31 The alternating projection algorithm (12.171) can be written in the equivalent form $a_n = \mathcal{P}_{\mathcal{C}_1}(\mathcal{P}_{\mathcal{C}_2}(a_{n-1}))$. Show that the cascade projection operator $\mathcal{P}(x) = \mathcal{P}_{\mathcal{C}_1}(\mathcal{P}_{\mathcal{C}_2}(x))$ is non-expansive, i.e.,

$$\|\mathcal{P}(x) - \mathcal{P}(y)\| \;\leq\; \|x - y\|, \quad \forall x, y \in \mathcal{C}_1$$

12.32 The method of *averaged projections* is an alternative to the alternating projection algorithm (12.171). It starts from an initial vector w_{-1} and updates it recursively as follows by averaging its projections on the two convex sets:

$$w_n = \frac{1}{2}\Big(\mathcal{P}_{\mathcal{C}_1}(w_{n-1}) \;+\; \mathcal{P}_{\mathcal{C}_2}(w_{n-1}) \Big), \;\; n \geq 0$$

Assume the intersection $\mathcal{C}_1 \cap \mathcal{C}_2$ is non-empty. Establish convergence of the sequence w_n to some point $w^\star$ in this intersection.

12.33 The Dykstra method (12.208) is a variation of the alternating projection algorithm. Show that starting from an initial vector a_{-1}, it converges to the unique point $w^\star \in \mathcal{C}_1 \cap \mathcal{C}_2$ that is closest to a_{-1} from within the intersection set. *Remark.* The reader may refer to Dykstra (1983), Boyle and Dykstra (1986), Combettes and Pesquet (2011), and Dattoro (2016) for a related discussion.

12.34 We wish to determine a solution to the linear system of equations $Hw = d$, where $H \in \mathbb{R}^{N\times M}$. This objective can be recast as the problem of determining the intersection of a collection of affine subspaces. We denote the rows of H by $\{h_k^\mathsf{T}\}$ and the entries of d by $\{\theta_k\}$ for $k = 1, 2, \ldots, N$, and consider the hyperplanes $\mathcal{H}_k = \{z \,|\, h_k^\mathsf{T} z - \theta_k = 0\}$. Starting from an initial condition w_{-1}, assume we apply a cyclic projection algorithm and project successively onto the $\mathcal{H}_k$ and keep repeating the procedure. Use the result of Prob. 9.3 for projections onto hyperplanes to verify that this construction leads to the following so-called Kaczmarz algorithm:

$$\left\{\begin{array}{l} \text{current projection is } w_{n-1} \\ \text{current selected row of } H \text{ is } h_k \text{ and selected entry of } d \text{ is } \theta_k \\ \text{update } w_n = w_{n-1} - \dfrac{(h_k^\mathsf{T} w_{n-1} - \theta_k)}{\|w_{n-1}\|^2} w_{n-1} \end{array}\right.$$

Remark. These recursions correspond to the classical method of Kaczmarz (1937) for the solution of linear systems of equations. We will encounter a randomized version later in Prob. 16.7 and apply it to the solution of least-squares problems.

12.35 Refer to the BFGS algorithm (12.206). Verify that $z_n^\mathsf{T} a_n > 0$ for strictly convex risks $P(w)$. Use the matrix inversion lemma to show that

$$B_n^{-1} = \left(I_M - \frac{1}{z_n^\mathsf{T} a_n} z_n a_n^\mathsf{T}\right) B_{n-1}^{-1} \left(I_M - \frac{1}{z_n^\mathsf{T} a_n} z_n a_n^\mathsf{T}\right)^\mathsf{T} + \frac{1}{z_n^\mathsf{T} a_n} z_n z_n^\mathsf{T}$$

Conclude that $B_n > 0$ when $P(w)$ is strictly convex and $z_n \neq 0$.

12.36 Continuing with the BFGS algorithm (12.206), introduce the second-order approximation for the risk function around w_n:

$$\widehat{P}(w) = P(w_n) + \nabla_w P(w_n)(w - w_n) + \frac{1}{2}(w - w_n)^\mathsf{T} B_n (w - w_n)$$

Use the secant condition to show that the gradient vectors of $P(w)$ and its approximation coincide at the locations w_n and w_{n-1}, i.e.,

$$\nabla_w \widehat{P}(w_n) = \nabla_w P(w_n), \quad \nabla_w \widehat{P}(w_{n-1}) = \nabla_w P(w_{n-1})$$

12.37 The material in Probs. 12.37–12.39 is motivated by results from Nesterov and Spokoiny (2017). Refer to the smoothed function (12.216) relative to the Gaussian distribution $f_{\boldsymbol{x}}(x) = \mathcal{N}_{\boldsymbol{x}}(0, I_M)$. Verify that:

(a) $P_\alpha(w) \geq P(w)$ for any $\alpha > 0$.

(b) $P_\alpha(w)$ is convex when $P(w)$ is convex.

(c) If $P(w)$ is $\delta-$Lipschitz then $P_\alpha(w)$ is $\delta_\alpha-$Lipschitz with $\delta_\alpha \leq \delta$.

(d) If $P(w)$ has $\delta-$Lipschitz gradients then $P_\alpha(w)$ has $\delta_\alpha-$Lipschitz gradients with $\delta_\alpha \leq \delta$.

12.38 Refer to the smoothed function (12.216) relative to the Gaussian distribution $f_{\boldsymbol{x}}(x) = \mathcal{N}_{\boldsymbol{x}}(0, I_M)$. Assume $P(w)$ is $\delta-$Lipschitz, i.e., $|P(w_1) - P(w_2)| \leq \delta\|w_1 - w_2\|$ for all $w_1, w_2 \in \text{dom}(P)$. Show that $P_\alpha(w)$ is first-order differentiable and its gradient vector is $\delta_\alpha-$Lipschitz, i.e.,

$$\|\nabla_w P_\alpha(w_1) - \nabla_w P_\alpha(w_2)\| \leq \delta_\alpha \|w_1 - w_2\|$$

with $\delta_\alpha = \delta\sqrt{M}/\alpha$.

12.39 Refer to the smoothed function (12.216) relative to the Gaussian distribution $f_{\boldsymbol{x}}(x) = \mathcal{N}_{\boldsymbol{x}}(0, I_M)$. Assume $P(w)$ has $\delta-$Lipschitz gradients, i.e.,

$$\|\nabla_w P(w_1) - \nabla_w P(w_2)\| \leq \delta \|w_1 - w_2\|$$

for all $w_1, w_2 \in \text{dom}(P)$. Establish the following inequalities:
(a) $|P_\alpha(w) - P(w)| \leq \alpha^2 M\delta/2$.
(b) $\|\nabla_w P_\alpha(w) - \nabla_w P(w)\| \leq \alpha\delta(M+3)^{3/2}/2$.
(c) $\|\nabla_w P(w)\|^2 \leq 2\|\nabla_w P_\alpha(w)\|^2 + \alpha^2\delta^2(M+6)^3/2$.
Show further that the second-order moment of the gradient approximation (12.214) satisfies

$$\mathbb{E}_{\boldsymbol{u}}\|\widehat{\nabla_w P}(w)\|^2 \leq 2(M+4)\|\nabla_w P(w)\|^2 + \alpha^2\delta^2(M+6)^3/2$$

Conclude that the error variance in estimating the gradient vector is bounded by the sum of two components: one varies with α^2 and decays with α while the other is independent of α but depends on the problem dimension M.
12.40 Verify equality (12.226), which relates the integral over a ball to the integral over its spherical surface.
12.41 Consider the following one-point estimate for the gradient vector in place of (12.214):

$$\widehat{\nabla_{w^\mathsf{T}} P}(w) = \frac{\beta}{\alpha} P(w + \alpha\, u)u$$

Verify that the unbiasedness property (12.218) continues to hold.

12.A ZEROTH-ORDER OPTIMIZATION

The gradient-descent algorithm described in the body of the chapter is an example of a first-order optimization method, which requires the availability of the gradient information, $\nabla_w P(w)$, in order to perform the update:

$$w_n = w_{n-1} - \mu\nabla_{w^\mathsf{T}} P(w_{n-1}), \quad n \geq 0 \tag{12.212}$$

where we are assuming a constant step-size for illustration purposes. In some situations of interest, it may not be possible to evaluate the gradient function either because the risk $P(w)$ may not have a closed analytical form or is unknown to the designer altogether. This latter situation will arise, for example, in adversarial learning scenarios, studied in future Chapter 71, where the designer wishes to misguide the operation of a learning algorithm but is only able to perform function evaluations, $P(w)$. Zeroth-order optimization is a technique that enables the designer to approximate the gradient vector by relying on function evaluations. We provide a brief overview of the methodology in this appendix.

Two-point gradient estimate

One way to approximate $\nabla_{w^\mathsf{T}} P(w)$ is to construct a *two-point estimate* as follows. Let $u \in \mathbb{R}^M$ denote a realization for a random vector that is selected according to some predefined distribution. Two choices are common:

$$\boldsymbol{u} \sim \mathcal{N}_{\boldsymbol{u}}(0, I_M), \quad (\textbf{Gaussian distribution}) \tag{12.213a}$$

$$\boldsymbol{u} \sim \mathcal{U}(\mathbb{S}), \quad (\textbf{uniform distribution on the unit sphere, } \mathbb{S}) \tag{12.213b}$$

where $\mathbb{S}$ denotes the unit sphere in $\mathbb{R}^M$ of radius one and centered at the origin. In both cases, the variable $\boldsymbol{u}$ has zero mean. Let $\alpha > 0$ denote a small parameter known as the *smoothing factor*. Then, we construct

$$\boxed{\widehat{\nabla_{w^\mathsf{T}} P}(w) = \frac{\beta}{\alpha}\Big(P(w+\alpha\, u) - P(w)\Big)u} \tag{12.214}$$

where the value of the scalar β depends on which mechanism is used to generate the directional vector u:

$$\beta = \begin{cases} 1, & \text{if } \boldsymbol{u} \sim \mathcal{N}_{\boldsymbol{u}}(0, I_M) \\ M, & \text{if } \boldsymbol{u} \sim \mathcal{U}(\mathbb{S}) \end{cases} \tag{12.215}$$

Note that expression (12.214) requires two function evaluations to approximate the gradient vector.

Smoothed risk function

Construction (12.214) has one useful property. We introduce the following smoothed version of $P(w)$, which is dependent on α and where the integration is over the domain of the variable x:

$$P_\alpha(w) \;\stackrel{\Delta}{=}\; \int_{x\in\mathcal{X}} P(w+\alpha x) f_{\boldsymbol{x}}(x)dx \;=\; \mathbb{E}_{\boldsymbol{x}}\Big\{P(w+\alpha\boldsymbol{x})\Big\} \tag{12.216}$$

The distribution of $\boldsymbol{x} \in \mathbb{R}^M$ depends on the mechanism used to generate $\boldsymbol{u}$, namely,

$$f_{\boldsymbol{x}}(x) \;\stackrel{\Delta}{=}\; \begin{cases} \mathcal{N}_{\boldsymbol{x}}(0, I_M), & (\textbf{Gaussian distribution}) \\ \mathcal{U}(\mathbb{B}), & (\textbf{unit ball}) \end{cases} \tag{12.217}$$

where the symbol $\mathbb{B}$ denotes the unit ball in $\mathbb{R}^M$ of radius one and centered at the origin (all vectors $x \in \mathbb{B}$ will have $\|x\| \le 1$). The sphere $\mathbb{S}$ is the surface of this ball, and the ball is the interior of the sphere. The smoothed function $P_\alpha(w)$ is differentiable (even when $P(w)$ is not) with its gradient vector satisfying:

$$\boxed{\nabla_{w^\mathsf{T}} P_\alpha(w) \;=\; \mathbb{E}_{\boldsymbol{u}}\Big\{\widehat{\nabla_{w^\mathsf{T}} P}(w)\Big\}} \tag{12.218}$$

where the expectation is over the distribution used to generate $\boldsymbol{u}$. This result shows that construction (12.214) provides an unbiased estimate for the gradient of $P_\alpha(w)$. Additional properties for the smoothed function are listed in Prob. 12.37.

Proof of (12.218): Consider first the Gaussian case. Then, it holds that

$$\begin{aligned} P_\alpha(w) &\overset{(12.216)}{=} \int_{x\in\mathcal{X}} P(w+\alpha x)\frac{1}{\sqrt{(2\pi)^M}}\exp\Big\{-\frac{1}{2}\|x\|^2\Big\}dx \\ &= \frac{1}{\alpha}\frac{1}{\sqrt{(2\pi)^M}}\int_{y\in\mathcal{Y}} P(y)\exp\Big\{-\frac{1}{2\alpha^2}\|y-w\|^2\Big\}dy, \quad \text{using } y = w+\alpha x \end{aligned} \tag{12.219}$$

Differentiating relative to w gives

$$\begin{aligned}
\nabla_{w^{\mathsf T}} P_\alpha(w) &= \frac{1}{\alpha}\frac{1}{\sqrt{(2\pi)^M}}\int_{y\in\mathcal{Y}} P(y)\exp\Big\{-\frac{1}{2\alpha^2}\|y-w\|^2\Big\}\times\frac{1}{\alpha^2}(y-w)dy \\
&= \frac{1}{\sqrt{(2\pi)^M}}\int_{x\in\mathcal{X}} P(w+\alpha x)\exp\Big\{-\frac{1}{2}\|x\|^2\Big\}\times\frac{1}{\alpha^2}(\alpha x)dx \\
&= \frac{1}{\alpha}\int_{x\in\mathcal{X}} P(w+\alpha x)x\,\mathcal{N}_{\boldsymbol{x}}(0,I_M)dx \\
&= \frac{1}{\alpha}\mathbb{E}_{\boldsymbol{x}}\Big\{P(w+\alpha\boldsymbol{x})\boldsymbol{x}\Big\} \\
&= \frac{1}{\alpha}\mathbb{E}_{\boldsymbol{x}}\Big\{P(w+\alpha\boldsymbol{x})\boldsymbol{x}\Big\} - \underbrace{\frac{1}{\alpha}\mathbb{E}_{\boldsymbol{x}}\Big\{P(w)\boldsymbol{x}\Big\}}_{=0} \\
&= \mathbb{E}_{\boldsymbol{x}}\left\{\frac{1}{\alpha}\Big(P(w+\alpha\boldsymbol{x})-P(w)\Big)\boldsymbol{x}\right\} \\
&\overset{(12.214)}{=} \mathbb{E}_{\boldsymbol{u}}\Big\{\widehat{\nabla_{w^{\mathsf T}} P}(w)\Big\}
\end{aligned} \tag{12.220}$$

as claimed, where the last equality is because $\boldsymbol{u}$ and $\boldsymbol{x}$ have the same Gaussian distribution.

Consider next the case in which $\boldsymbol{u}$ is selected uniformly from the unit sphere, $\mathbb{S}$. Then, in this case,

$$\begin{aligned}
P_\alpha(w) &\overset{(12.216)}{=} \mathbb{E}_{\boldsymbol{x}\in\mathbb{B}}\Big\{P(w+\alpha\boldsymbol{x})\Big\} \\
&= \mathbb{E}_{\boldsymbol{y}\in\alpha\mathbb{B}}\Big\{P(w+\boldsymbol{y})\Big\},\quad\text{using change of variables } \boldsymbol{y}=\alpha\boldsymbol{x} \\
&= \frac{1}{\text{vol}(\alpha\mathbb{B})}\int_{y\in\alpha\mathbb{B}} P(w+y)dy
\end{aligned} \tag{12.221}$$

where vol($\alpha\mathbb{B}$) denotes the volume of the ball of radius α; we recall that the volume of a ball in $\mathbb{R}^M$ centered at the origin with radius α is given by

$$\text{vol}(\alpha\mathbb{B}) = \frac{\pi^{M/2}\,\alpha^M}{\Gamma(\frac{M}{2}+1)} \tag{12.222}$$

in terms of the Gamma function. Moreover, it also holds that

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{u}\in\mathbb{S}}\Big\{P(w+\alpha\boldsymbol{u})\boldsymbol{u}\Big\} &= \mathbb{E}_{\boldsymbol{z}\in\alpha\mathbb{S}}\Big\{P(w+\boldsymbol{z})\frac{\boldsymbol{z}}{\alpha}\Big\},\quad\text{using change of variables } \boldsymbol{z}=\alpha\boldsymbol{u} \\
&= \mathbb{E}_{\boldsymbol{z}\in\alpha\mathbb{S}}\Big\{P(w+\boldsymbol{z})\frac{\boldsymbol{z}}{\|\boldsymbol{z}\|}\Big\},\quad\text{since } \|\boldsymbol{z}\|=\alpha \\
&= \frac{1}{\text{surf}(\alpha\mathbb{S})}\int_{z\in\alpha\mathbb{S}} P(w+z)\frac{z}{\|z\|}dz
\end{aligned} \tag{12.223}$$

where surf($\alpha\mathbb{S}$) denotes the surface area of the ball of radius α; we recall that the surface area of a ball in $\mathbb{R}^M$ centered at the origin with radius α is given by

$$\text{surf}(\alpha\mathbb{S}) = \frac{2\pi^{M/2}\,\alpha^{M-1}}{\Gamma(\frac{M}{2})} \tag{12.224}$$

Using the property $\Gamma(z+1)=z\Gamma(z)$ for Gamma functions, we conclude that

$$\frac{\text{surf}(\alpha\mathbb{S})}{\text{vol}(\alpha\mathbb{B})} = \frac{M}{\alpha} \tag{12.225}$$

From the divergence theorem in calculus, which allows us to relate integration over a volume to the integral over its surface, it can be verified that — see Prob. 12.40:

$$\nabla_{w^{\mathsf{T}}}\left\{\int_{y\in\alpha\mathbb{B}} P(w+y)dy\right\} = \int_{z\in\alpha\mathbb{S}} P(w+z)\frac{z}{\|z\|}dz \tag{12.226}$$

Collecting terms we conclude that (12.218) holds.

■

An alternative two-point estimate for the gradient in place of (12.214) is the symmetric version

$$\widehat{\nabla_{w^{\mathsf{T}}} P}(w) = \frac{\beta}{2\alpha}\Big(P(w+\alpha\, u) - P(w-\alpha u)\Big)u \tag{12.227}$$

where the arguments of the risk function are $w \pm \alpha u$ and the factor in the denominator is 2α. In Prob. 12.41 we consider another example.

Zeroth-order algorithm

We can now list a zeroth-order algorithm for minimizing a risk function $P(w) : \mathbb{R}^M \rightarrow \mathbb{R}$. The original gradient-descent recursion (12.24) is replaced by (12.228). In the listing, we denote the distribution from which the directional vectors $\boldsymbol{u}$ are sampled by $f_{\boldsymbol{u}}(u)$; it can refer either to the Gaussian distribution $\mathcal{N}_{\boldsymbol{u}}(0, I_M)$ or the uniform distribution $\mathcal{U}(\mathbb{S})$, as described by (12.213a)–(12.213b).

Zeroth-order gradient-based method for minimizing $P(w)$.

```
given a small step-size parameter $\mu > 0$;
given a small smoothing factor $\alpha > 0$;
select the sampling distribution $f_{\boldsymbol{u}}(u)$ and set $\beta \in \{1, M\}$;
start from an arbitrary initial condition, $w_{-1}$.
repeat until sufficient convergence over $n \geq 0$ :
    sample $u_n \sim f_{\boldsymbol{u}}(u)$
    $\widehat{\nabla_{w^{\mathsf{T}}} P}(w_{n-1}) = \frac{\beta}{\alpha}\Big(P(w_{n-1} + \alpha\, u_n) - P(w_{n-1})\Big)u_n$
    $w_n = w_{n-1} - \mu\, \widehat{\nabla_{w^{\mathsf{T}}} P}(w_{n-1})$
end
return $w^\star \leftarrow w_n$.
```

(12.228)

REFERENCES

Alghunaim, S. A. and A. H. Sayed (2020), "Linear convergence of primal-dual gradient methods and their performance in distributed optimization," *Automatica*, vol. 117, pp. 109003.

Armijo, L. (1966), "Minimization of functions having Lipschitz continuous first partial derivatives," *Pacific J. Math.*, vol. 16, no. 1, pp. 1–3.

Arrow, K. J. and L. Hurwicz (1956), "Reduction of constrained maxima to saddle-point problems," *Proc. 3rd Berkeley Symp. on Math. Statist. and Prob.*, pp. 1–20.

Bauschke, H. H. and Borwein, J. M. (1996), "On projection algorithms for solving convex feasibility problems," *SIAM Review*, vol. 38, no. 3, pp. 367–426.

Beck, A. and L. Tetruashvili (2013), "On the convergence of block coordinate descent type methods," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2037–2060.

Bertsekas, D. P. (1995), *Nonlinear Programming*, Athena Scientific, MA.

Bertsekas, D. P. and J. N. Tsitsiklis (1997), *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, Singapore.

Bottou, L. (1998), "Online algorithms and stochastic approximations," in *Online Learning and Neural Networks*, D. Saad, *Ed.*, Cambridge University Press.

Bottou, L. and Y. LeCun (2004), "Large scale online learning," *Proc. Advances in Neural Information Processing Systems* (NIPS), vol. 16, pp. 217–224, MIT Press, Cambridge, MA.

Boyd, S. and L. Vandenberghe (2004), *Convex Optimization*, Cambridge University Press.

Boyle, J. P. and R. L. Dykstra (1986), "A method for finding projections onto the intersection of convex sets in Hilbert spaces," *Lecture Notes in Statistics*, vol. 37, pp. 28–47.

Bregman, L. M. (1965), "The method of successive projection for finding a common point of convex sets," *Soviet Mathematics*, vol. 6, pp. 688–692.

Bregman, L. M. (1967), "The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 3, pp. 200–217.

Brent, R. (2002), *Algorithms for Minimization without Derivatives*, Prentice Hall, NJ.

Broyden, C. G. (1970), "The convergence of a class of double-rank minimization algorithms," *J. Institute of Mathematics and Its Applications*, vol. 6, pp. 76–90.

Cauchy, A.-L. (1847), "Methode générale pour la résolution des systems déquations simultanes," *Comptes Rendus Ilebdomadaires des Séances de lÁcademic des Sciences*, vol. 25, pp. 536–538.

Cevher, V., S. Becker, and M. Schmidt (2014), "Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics," *IEEE Signal Processing Magazine,* vol. 31, no. 5, pp. 32–43.

Chang, K. W., C. J. Hsieh, and C. J. Lin (2008), "Coordinate descent method for large-scale L2-loss linear SVM," *J. Machine Learning Research*, vol. 9, pp. 1369–1398.

Cheney, W. and A. Goldstein (1959), "Proximity maps for convex sets," *Proc. AMS*, vol. 10, pp. 448–450.

Christensen, C. (1996), "Newton's method for resolving affected equations," vol. 27, no. 5, pp. 330–340.

Combettes, P. L. (1993), "The foundations of set theoretic estimation," *Proc. IEEE*, vol. 81, no. 2, pp. 182–208.

Combettes, P. L. and J.-C. Pesquet (2011), "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. H. Bauschke *et al.*, *Eds.*, pp. 185-212. Springer, NY.

Conn, A. R., K. Scheinberg, and L. N. Vicente (2009), *Introduction to Derivative-Free Optimization*, SIAM, PA.

Curry, H. B. (1944), "The method of steepest descent for nonlinear minimization problems," *The Quarterly Journal of Mechanics and Applied Mathematics*, vol. 2, pp. 258–261.

Dattoro, J. (2016), *Convex Optimization and Euclidean Distance Geometry*, Meboo Publishing, USA. Available online.

Daubechies, I., M. Defrise, and C. De Mol (2004), "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. LVII, pp. 1413–1457.

Duchi, J. C., M. I. Jordan, M. J. Wainwright, and A. Wibisono (2015), "Optimal rates for zero-order convex optimization: The power of two function evaluations," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2788–2806.

Dykstra, R. L. (1983), "An algorithm for restricted least squares regression," *J. American Statistical Association*, vol. 78, no. 384, pp. 837–842.

Escalante, R. and M. Raydan (2011), *Alternating Projection Methods*, SIAM, PA.

Fercoq O. and P. Bianchi (2019), "A coordinate-descent primal-dual algorithm with large step size and possibly nonseparable functions," *SIAM Journal on Optimization*, vol. 29, no. 1, pp. 100–134.

Flaxman, A. D., A. T. Kalai, and H. B. McMahan (2005), "Online convex optimization in the bandit setting: Gradient descent without a gradient," *Proc. Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 385–394, Vancouver, BC.

Fletcher, R. (1970), "A new approach to variable metric algorithms," *Computer Journal*, vol. 13, no. 3, pp. 317–322.

Fletcher, R. (1987), *Practical Methods of Optimization*, 2nd edition, Wiley, NY.

Friedman, J. H., T. Hastie, H. Höfling, and R. Tibshirani (2007), "Pathwise coordinate optimization," *The Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332.

Fu, W. J. (1998), "Penalized regressions: The bridge versus the Lasso," *J. Computational and Graphical Statistics*, vol. 7, no. 3, pp. 397–416.

Gauss, C. F. (1903), *Carl Friedrich Gauss Werke,* Akademie der Wissenschaften, Gottingen.

Goldfarb, D. (1970), "A family of variable metric updates derived by variational means," *Mathematics of Computation*, vol. 24, no. 109, pp. 23–26.

Goldstein, A. A. (1962), "Cauchy's method of minimization," *Numer. Math.*, vol. 4, no. 2, pp. 146–150.

Goldstein, A. A. (1966), "Minimizing functionals on normed-linear spaces," *SIAM J. Control*, vol. 4, pp. 91–89.

Golub, G. H. and C. F. Van Loan (1996), *Matrix Computations*, 3rd edition, The John Hopkins University Press, MD.

Gubin, L. G., B. T. Polyak, and E. V. Raik (1967), "The method of projections for finding the common point of convex sets," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 6, pp. 1–24.

Halperin, I. (1962), "The product of projection operators," *Acta. Sci. Math.*, vol. 23, pp. 96–99.

Han, S.-P. (1988), "A successive projection method," *Math. Programming*, vol. 40, pp. 1–14.

Hildreth, C. (1957), "A quadratic programming procedure," *Naval Research Logistics Quarterly*, vol. 4, pp. 79–85. See also erratum in same volume on page 361.

Kaczmarz, S. (1937), "Angenäherte Auflösung von Systemen linearer Gleichungen," *Bull. Int. Acad. Polon. Sci. Lett.* A, pp. 335–357.

Kelley, C. T. (1996), *Iterative Methods for Optimization*, SIAM, PA.

Kiefer, J. and J. Wolfowitz (1952), "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 462–466, 1952.

Kollerstrom, N. (1992), "Thomas Simpson and Newton's method of approximation: An enduring myth," *The British Journal for the History of Science*, vol. 25, no. 3, pp. 347–354.

Lange, K., E. C. Chi, and H. Zhou (2014), "A brief survey of modern optimization for statisticians," *International Statistical Review*, vol. 82, no. 1, pp. 46–70.

Larson, J., M. Menickelly, and S. M. Wild (2019), "Derivative-free optimization methods," *Acta Numerica*, vol. 28, pp. 287–404.

Liu, S., P.-Y. Chen, B. Kailkhura, G. Zhang, A. Hero, and P. Varshney (2020), "A primer on zeroth-order optimization in signal processing and machine learning," *IEEE Signal Processing Magazine*, vol 37, issue 5, pp. 43–54.

Liu, S., B. Kailkhura, P.-Y. Chen, P. Ting, S. Chang, and L. Amini (2018), "Zeroth-order stochastic variance reduction for nonconvex optimization,"*Proc. Advances Neural Information Processing Systems* (NIPS), pp. 3727–3737.

Lemaréchal, C. (2012), "Cauchy and the gradient method," *Documenta Mathematica*, Extra Volume ISMP, pp. 251–254.

Le Roux, N., M. Schmidt, and F. Bach (2012), "A stochastic gradient method with an exponential convergence rate for finite training sets," *Proc. Advances Neural Information Processing Systems* (NIPS), pp. 2672–2680, Lake Tahoe, 2012.

Luenberger, D. G. and Y. Ye (2008), *Linear and Nonlinear Programming*, Springer, NY.

Luo, Z. Q. and P. Tseng (1992a), “On the convergence of the coordinate descent method for convex differentiable minimization,” *J. Optim. Theory Appl.*, vol. 72, pp. 7–35.

Nash, S. G. and A. Sofer (1996), *Linear and Nonlinear Programming*, McGraw-Hill, NY.

Nesterov, Y. (1983), “A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$,” *Doklady AN USSR*, vol. 269, pp. 543–547.

Nesterov, Y. (2004), *Introductory Lectures on Convex Optimization*, Springer, NY.

Nesterov, Y. (2005), “Smooth minimization of non-smooth functions,” *Math. Programming*, vol. 103, no. 1, pp. 127–152.

Nesterov, Y. (2012), “Efficiency of coordinate descent methods on huge-scale optimization problems,” *SIAM J. Optim.*, vol. 22, no. 2, pp. 341–362.

Nesterov, Y. and V. Spokoiny (2017), “Random gradient-free minimization of convex functions,” *Found. Comput. Math.*, vol. 17, no. 2, pp. 527–566.

Nocedal, J. and S. J. Wright (2006), *Numerical Optimization*, Springer, NY.

Nutini, J., M. Schmidt, I. Laradji, M. Friedlander, and H. Koepke (2015), “Coordinate descent converges faster with the Gauss-Southwell rule than random selection,” *Proc. International Conference on Machine Learning* (ICML), pp. 1632–1641, Lille, France.

Ortega, J. M. and W. Rheinboldt (1970), *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press.

Polyak, B. T. (1964), “Some methods of speeding up the convergence of iteration methods,” *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17.

Polyak, B. T. (1987), *Introduction to Optimization*, Optimization Software, NY.

Polyak, B. T. and A. Juditsky (1992), “Acceleration of stochastic approximation by averaging,” *SIAM J. Control and Optim.*, vol. 30, no. 4, pp. 838–855.

Powell, M. J. D. (1973), “On search directions for minimization algorithms,” *Math. Program.*, vol. 4, pp. 193–201.

Raphson, J. (1697), *Analysis aequationum universalis seu ad aequationes algebraicas resolvendas methodus generalis, and expedita, ex nova infinitarum serierum methodo, deducta ac demonstrata,* publisher Th. Braddyll.

Richtárik, P. and M. Takác (2011), “Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function,” *Mathematical Programming*, Series A, 144, no. 1–2, pp. 1–38.

Rios, L. M. and N. V. Sahinidis (2013), “Derivative-free optimization: A review of algorithms and comparison of software implementations,” *J. Global Optimization*, vol. 56, no. 3, pp. 1247–1293.

Sauer, K. and C. Bouman (1993), “A local update strategy for iterative reconstruction from projections,” *IEEE Trans. Signal Processing*, vol. 41, no. 2, pp. 534–548.

Sayed, A. H. (2014a), *Adaptation, Learning, and Optimization over Networks,* Foundations and Trends in Machine Learning, NOW Publishers, vol. 7, no. 4–5, pp. 311–801.

Seidel, L. (1874), Abh. Bayer. Akad. Wiss. Math.-Naturwiss. Kl., vol. 11, no. 3, pp. 81–108.

Shanno, D. F. (1970), “Conditioning of quasi-Newton methods for function minimization,” *Mathematics of Computation*, vol. 24, no. 111, pp. 647–656.

Shi, H.-J. M., S. Tu, Y. Xu, and W. Yin (2017), “A primer on coordinate descent algorithms,” *available online at arXiv:1610.00040.*

Simpson, T. (1740), *Essays on Several Curious and Useful Subjects in Speculative and Mix'd Mathematicks*, London.

Southwell, R. V. (1940), *Relaxation Methods in Engineering Science – A Treatise on Approximate Computation*, Oxford University Press.

Tseng, P. (2001), “Convergence of a block coordinate descent method for nondifferentiable minimization,” *J. Optimization Theory and Applications*, vol. 109, pp. 475–494.

Tseng, P. and S. Yun (2009), “A coordinate gradient descent method for nonsmooth separable minimization,” *Math. Program.*, vol. 117, pp. 387–423.

Tseng, P. and S. Yun (2010), “A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training," *Comput. Optim. Appl.*, vol. 47, pp. 179–206.

von Neumann, J. (1949), "On rings of operators. Reduction theory," *Annals of Mathematics*, pp. 401–485.

von Neumann, J. (1950), *Functional Operators II: The Geometry of Orthogonal Spaces*, vol. 22, *Annal. Math. Studies.* Reprinted from lecture notes first distributed in 1933.

Wallis, J. (1685), *A Treatise of Algebra, both Historical and Practical. Shewing the Original, Progress, and Advancement thereof, from time to time, and by what Steps it hath attained to the Heighth at which it now is*, printed by John Playford, London.

Wang, C., Y. Zhang, B. Ying, and A. H. Sayed (2018), "Coordinate-descent diffusion learning by networked agents," *IEEE Trans. Signal Process.*, vol. 66, no. 2, pp. 352–367.

Warga, J. (1963), "Minimizing certain convex functions," *SIAM Journal on Applied Mathematics,* vol. 11, pp. 588–593.

Wibisono, A., M. J Wainwright, M. I. Jordan, and J. C. Duchi (2012), "Finite sample convergence rates of zero-order stochastic optimization methods," *Proc. Advances Neural Information Processing Systems* (NIPS), pp. 1439–1447.

Wolfe, P. (1969), "Convergence conditions for ascent methods," *SIAM Review*, vol. 11, no. 2, pp. 226–235.

Wolfe, P. (1971), "Convergence conditions for ascent methods II: Some corrections,"*SIAM Review* vol. 13, pp. 185–188.

Wright, S. J. (2015), "Coordinates descent algorithms," *Math. Program.*, vol. 151, pp. 3–34.

Wu, T. T. and K. Lange (2008), "Coordinate descent algorithms for LASSO penalized regression," *The Annals of Applied Statistics*, vol. 2, no. 1, pp. 224–244.

Yu, H., R. Jin, and S. Yang (2019), "On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization," *Proceedings of Machine Learning Research* (PMLR), vol. 97, pp. 7184–7193.