
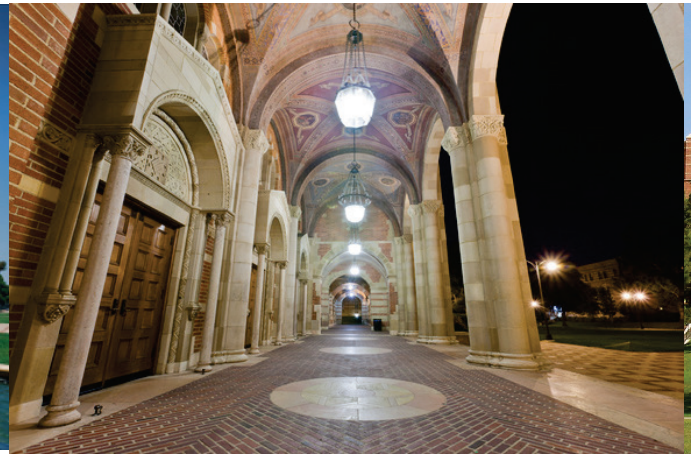


EE210A: Adaptation and Learning

Professor Ali H. Sayed

LECTURE #08

STEEPEST DESCENT METHOD

Sections in order: 8.1-8.3 and 9.1-9.8

MOTIVATION

Now there are situations where a designer may be interested in other performance criteria, other than the mean-square error criterion. Several examples to this effect are provided in the problems at the end of this part (e.g., Probs. III.12–III.18).

In most of these cases it is generally not possible to describe the optimal solution $\hat{x}$ in *closed-form* in terms of the moments of the underlying variables, and it often becomes necessary to approximate the optimal solution iteratively.

The iterative procedure would start from an initial guess for the solution and then improve upon it from one iteration to another. The purpose of this chapter is to describe one class of iterative schemes known as steepest-descent methods, which is at the core of most adaptive filtering techniques.

MOTIVATION

The steepest-descent methods will be initially motivated by showing how they apply to the *already-studied* case of linear least-mean-squares estimation. By focusing on a situation that is familiar to the reader, and one for which the optimal solution is already known, we will be able to highlight some of the abilities (and deficiencies) of iterative schemes. In particular, we will be able to show, even for the linear estimation problem, that steepest-descent methods are of independent value in their own right.

For instance, they will help us avoid the need to invert R_y in order to determine K_o in the solution of the normal equations $K_o R_y = R_{xy}$. Such matrix inversions are challenging from a complexity point of view (requiring of the order of N^3 computations for an $N \times N$ matrix R_y); they are also challenging for ill-conditioned matrices R_y, namely, for matrices that are close to singular and that have a large ratio of largest to smallest eigenvalues. Once the main idea of steepest-descent has been examined in the context of linear estimation, we shall then show how to extend the technique to other estimation problems, with more involved performance criteria.

MOTIVATION

Steepest-descent methods are not studied in this chapter only because they provide a mechanism for solving more involved estimation problems. In addition to this useful objective, these methods are also important because they will serve as the launching pad for the development of adaptive filters in Chapters 10–14. It is because of this latter objective that, from now on and until the end of this textbook, we shall adopt a notation that is more specific, and also more suited, to the study of adaptive filters.

NOTATION

Notation. In Parts I (*Optimal Estimation*) and II (*Linear Estimation*) of this book, we adopted the $\{x, y\}$ notation, as is common in estimation theory, for the variable to be estimated and for the observation vector. The variables $\{x, y\}$ were general and they could refer to scalars or vectors. The results of the earlier chapters are of broad interest and they are not exclusive to the study of adaptive filters. However, from now on, we shall develop the theory of adaptive filters in greater detail. In this context, we will be mostly interested in the case in which x is a *scalar* and y is a *row* vector. Moreover, the $\{x, y\}$ variables will have specific meanings attached to them. For instance, x will denote the so-called "desired signal" and we shall replace it by the letter d, which will be a scalar. The observation vector y, on the other hand, will be a row vector and it will be denoted by u. In this way, we are now interested in estimating d from u. Some motivation for our choice of the *row* vector notation for u appears in the Notation section in the opening pages of this book.

$\diamond$

LINEAR ESTIMATION REVIEW

8.1 LINEAR ESTIMATION PROBLEM

So let $\boldsymbol{d}$ be a zero-mean *scalar-valued* random variable with variance σ_d^2,

$$\boxed{\mathsf{E}\,\boldsymbol{d} = 0, \qquad \sigma_d^2 = \mathsf{E}\,|\boldsymbol{d}|^2}$$

and let $\boldsymbol{u}$ be a $1 \times M$ zero-mean random row vector with a positive-definite covariance matrix denoted by R_u

$$\boxed{R_u \triangleq \mathsf{E}\,\boldsymbol{u}^*\boldsymbol{u}} \qquad \text{(a square matrix)}$$

The variables $\{\boldsymbol{d}, \boldsymbol{u}\}$ are allowed to be complex-valued for generality, which, as we saw in several examples in Chapters 1–6, is usually a necessity in digital communications applications.

LINEAR ESTIMATION REVIEW

The $M \times 1$ cross-covariance vector of $\{\boldsymbol{d}, \boldsymbol{u}\}$ is denoted by

$$\boxed{R_{du} \triangleq \mathsf{E}\, \boldsymbol{d}\boldsymbol{u}^*} \qquad \text{(a column vector)}$$

We then consider the problem of estimating $\boldsymbol{d}$ from $\boldsymbol{u}$ in the linear least-mean-squares sense as follows:

$$\boxed{\min_{w} \quad \mathsf{E}\, |\boldsymbol{d} - \boldsymbol{u}w|^2} \tag{8.1}$$

where w is $M \times 1$ and is known as the weight vector.

Remark 8.1 (Row vector notation) Observe that since we choose $\boldsymbol{u}$ to be a row vector and the unknown w to be a column vector, the inner-product between $\boldsymbol{u}$ and w is simply written as $\boldsymbol{u}w$ with no transposition or conjugation symbols needed.

> We adopt this convention throughout our treatment of adaptive filters in this and subsequent chapters.

All vectors, from this chapter onwards, will be column vectors with the notable exception of $\boldsymbol{u}$, which will be a row vector.

SOLUTION

Using the Orthogonality Principle

Alternatively, we can solve (8.1) more directly by invoking the orthogonality principle of linear least-mean-squares estimation. Specifically, from Thm. 4.1, we know that the optimal weight vector w^o should lead to an error variable, $\boldsymbol{d} - \boldsymbol{u}w^o$, that is orthogonal to the observation vector $\boldsymbol{u}$, i.e., it must hold that $\boldsymbol{d} - \boldsymbol{u}w^o \perp \boldsymbol{u}$ or, equivalently,

$$\mathsf{E}\,\boldsymbol{u}^*(\boldsymbol{d} - \boldsymbol{u}w^o) \;=\; 0 \tag{8.6}$$

which means that w^o should satisfy the normal equations $R_{du} - R_u w^o = 0$, and we are back to (8.4). Likewise, the resulting m.m.s.e. can be obtained from the orthogonality condition as follows:

$$\begin{aligned}
\text{m.m.s.e.} &= \mathsf{E}\,|\boldsymbol{d} - \boldsymbol{u}w^o|^2 \\
&= \mathsf{E}\,(\boldsymbol{d} - \boldsymbol{u}w^o)(\boldsymbol{d} - \boldsymbol{u}w^o)^* \\
&= \mathsf{E}\,(\boldsymbol{d} - \boldsymbol{u}w^o)\boldsymbol{d}^* \qquad \text{(because of (8.6))} \\
&= \sigma_d^2 - R_{ud}R_u^{-1}R_{du}
\end{aligned}$$

SOLUTION

Theorem 8.1 (Optimal linear estimator) All random variables are zero-mean. Consider a scalar variable d and a row vector $\boldsymbol{u}$ with $R_u = \mathsf{E}\,\boldsymbol{u}^*\boldsymbol{u} > 0$. The linear least-mean-squares estimator of d given $\boldsymbol{u}$ is $\hat{d} = \boldsymbol{u}w^o$ where

$$w^o = R_u^{-1} R_{du}$$

The resulting minimum mean-square error is m.m.s.e. $= \sigma_d^2 - R_{ud} R_u^{-1} R_{du}$.

STEEPESE-DESCENT

8.2 STEEPEST-DESCENT METHOD

$$J(w) \triangleq \mathsf{E}\,|\boldsymbol{d}-\boldsymbol{u}w|^2 = \mathsf{E}\,(\boldsymbol{d}-\boldsymbol{u}w)(\boldsymbol{d}-\boldsymbol{u}w)^* \tag{8.7}$$

$$\boxed{J(w) = \sigma_d^2 - R_{du}^* w - w^* R_{du} + w^* R_u w} \tag{8.8}$$

We already know that $J(w)$ has a unique global minimum at $w^o = R_u^{-1} R_{du}$ with minimum value given by (8.11). Figure 4.1 shows a typical plot of $J(w)$ for the case in which w is two-dimensional and real-valued.

STEEPEST-DESCENT

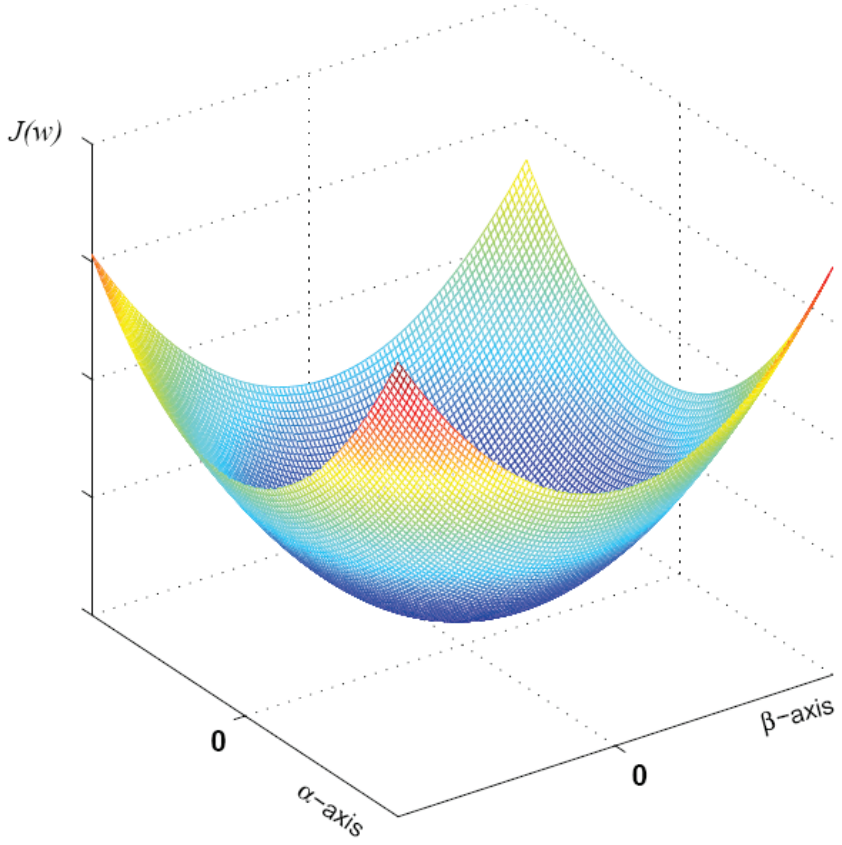

FIGURE 8.1 A typical plot of the quadratic cost function $J(w)$ when w is two-dimensional and real-valued, say $w = \text{col}\{\alpha, \beta\}$.

SEARCH DIRECTION

Now given $J(w)$, and without assuming any prior knowledge about the location of its minimizing argument w^o, we wish to devise a procedure that starts from an initial guess for w^o and then improves upon it in a recursive manner until ultimately converging to w^o. The procedure that we seek is one of the form

$$(\text{new guess}) \;=\; (\text{old guess}) \;+\; (\text{a correction term})$$

or, more explicitly,

$$\boxed{w_i = w_{i-1} + \mu\, p, \quad i \geq 0} \tag{8.12}$$

where we are writing w_{i-1} to denote a guess for w^o at iteration $(i-1)$, and w_i to denote the updated guess at iteration i. The vector p is an *update direction* vector that we should choose adequately, along with the positive scalar μ, in order to guarantee convergence of w_i to w^o. The scalar μ is called the *step-size* parameter since it affects how small or how large the correction term is. In (8.12), and in all future developments in this book, it is assumed that the index i runs from 0 onwards, so that the initial condition is specified at $i = -1$. Usually, but not always, the initial condition w_{-1} is taken to be zero.

SEARCH DIRECTION

The criterion for selecting μ and p is to enforce, if possible, the condition $J(w_i) < J(w_{i-1})$. In this way, the value of the cost function at the successive iterations will be monotonically decreasing. To show how this condition can be enforced, we start by relating $J(w_i)$ to $J(w_{i-1})$. Evaluating $J(w)$ at $w_i = w_{i-1} + \mu p$ and expanding we get

$$\begin{aligned} J(w_i) &= \sigma_d^2 - R_{du}^*(w_{i-1} + \mu p) - (w_{i-1} + \mu p)^* R_{du} + (w_{i-1} + \mu p)^* R_u (w_{i-1} + \mu p) \\ &= J(w_{i-1}) + \mu(w_{i-1}^* R_u - R_{du}^*)p + \mu p^*(R_u w_{i-1} - R_{du}) + \mu^2 p^* R_u p \end{aligned} \quad (8.13)$$

We can rewrite this equality more compactly by observing from expression (8.8) that the gradient vector of $J(w)$ with respect to w is equal to

$$\nabla_w J(w) = w^* R_u - R_{du}^* \quad (8.14)$$

This means that the term $w_{i-1}^* R_u - R_{du}^*$ that appears in (8.13) is simply the value of the gradient vector at $w = w_{i-1}$, i.e.,

$$w_{i-1}^* R_u - R_{du}^* = \nabla_w J(w_{i-1})$$

SEARCH DIRECTION

Similarly, the matrix R_u appearing in $\mu^2 p^* R_u p$ is equal to the Hessian matrix of $J(w)$, i.e.,

$$\mu^2 p^* R_u p = \mu^2 p^* \left[\nabla_w^2 J(w_{i-1})\right] p$$

We can then rewrite (8.13) as

$$\boxed{J(w_i) \;=\; J(w_{i-1}) \;+\; 2\mu\, \mathsf{Re}\left[\nabla_w J(w_{i-1}) p\right] \;+\; \mu^2 p^* R_u\, p} \qquad (8.15)$$

in terms of the real part of the inner product $\nabla_w J(w_{i-1})p$.

Now the last term on the right-hand side of (8.13) is positive for all nonzero p since $R_u > 0$. Therefore, a *necessary* condition for

$$J(w_i) \;<\; J(w_{i-1}) \qquad (8.16)$$

is to require the update direction p to satisfy

$$\boxed{\mathsf{Re}\left[\nabla_w J(w_{i-1}) p\right] \;<\; 0} \qquad (8.17)$$

SEARCH DIRECTION

$$\boxed{\text{Re}\,[\nabla_w J(w_{i-1})p] \;<\; 0} \tag{8.17}$$

This condition guarantees that the second term on the right-hand side of (8.15) is strictly negative. The selection of a vector p according to (8.17) will depend on whether $\nabla_w J(w_{i-1})$ is zero or not. If the gradient vector is zero, then $R_u w_{i-1} = R_{du}$, and thus w_{i-1} already coincides with the desired solution w^o. In this situation, recursion (8.12) would have attained w^o and p should be selected as $p = 0$.

When, on the other hand, the gradient vector at w_{i-1} is nonzero, there are many choices of vectors p that satisfy (8.17). For example, any p of the form

$$\boxed{p \;=\; -B\,[\nabla_w J(w_{i-1})]^*} \tag{8.18}$$

for any Hermitian positive-definite matrix B will do (this choice will also give $p = 0$ when $\nabla_w J(w_{i-1}) = 0$). To see this, note that for any such p, the inner product in (8.17) is real-valued and evaluates to

$$\nabla_w\, J(w_{i-1})p \;=\; -\,[\nabla_w\, J(w_{i-1})]\,B\,[\nabla_w\, J(w_{i-1})]^*$$

which is negative in view of the positive-definiteness of B.

SEARCH DIRECTION

which is negative in view of the positive-definiteness of B. The special choice $B = \mathrm{I}$ is very common and it corresponds to the update direction

$$\boxed{p = -[\nabla_w J(w_{i-1})]^* = R_{du} - R_u w_{i-1}} \tag{8.19}$$

This choice for p reduces (8.12) to the recursion

$$\boxed{w_i = w_{i-1} + \mu[R_{du} - R_u w_{i-1}], \quad i \geq 0, \quad w_{-1} = \text{initial guess}} \tag{8.20}$$

The update direction (8.19) has a useful and intuitive interpretation. Recall that the gradient vector at any point of a cost function points toward the direction in which the function is increasing. Now (8.19) is such that, at each iteration, it chooses the update direction p to point in the *opposite* direction of the (conjugate) gradient vector. For this reason, we refer to (8.20) as a steepest-descent method; the successive weight vectors $\{w_i\}$ are obtained by descending along a path of decreasing cost values. The choice of the step-size μ is crucial and, if not chosen with care, it can destroy this desirable behavior. Choosing p according to (8.19) is only a necessary condition for (8.16) to hold; it is not sufficient as we still need to choose μ properly, as we proceed to explain.

CONVERGENCE CONDITION

Introduce the weight-error vector

$$\boxed{\tilde{w}_i \stackrel{\Delta}{=} w^o - w_i}$$

It measures the difference between the weight estimate at time i and the optimal weight vector, w^o, which we are attempting to reach.

Subtracting both sides of the steepest-descent recursion (8.20) from w^o we obtain

$$\tilde{w}_i = \tilde{w}_{i-1} - \mu[R_{du} - R_u w_{i-1}]$$

with initial weight-error vector $\tilde{w}_{-1} = w^o - w_{-1}$. Using the fact that w^o satisfies the normal equations $R_u w^o = R_{du}$, we replace R_{du} in the above recursion by $R_u w^o$ and arrive at the weight-error recursion:

$$\boxed{\tilde{w}_i = [\mathrm{I} - \mu R_u]\tilde{w}_{i-1}\ , \quad i \geq 0, \qquad \tilde{w}_{-1} = \text{ initial condition}} \tag{8.21}$$

CONVERGENCE CONDITION

This is a homogeneous difference equation with coefficient matrix $(\mathrm{I} - \mu R_u)$. Therefore, a necessary and sufficient condition for the error vector $\tilde{w}_i$ to tend to zero, regardless of the initial condition $\tilde{w}_{-1}$, is to require that all of the eigenvalues of the matrix $(\mathrm{I} - \mu R_u)$ be strictly less than one in magnitude. That is, $(\mathrm{I} - \mu R_u)$ must be a *stable* matrix. This conclusion is a special case of a general result. For any homogeneous recursion of the form $y_i = Ay_{i-1}$, it is well-known that the successive vectors y_i will tend to zero regardless of the initial condition y_{-1} if, and only if, all eigenvalues of A are strictly inside the unit disc. The argument that we give below establishes the result for the special case $A = \mathrm{I} - \mu R_u$. For generic matrices A, the proof is left as an exercise to the reader; see Prob. III.23.

One way to establish that $(\mathrm{I} - \mu R_u)$ must be a stable matrix is the following. Since R_u is a positive-definite Hermitian matrix, its eigen-decomposition has the form (cf. App. B.1):

$$\boxed{R_u = U\Lambda U^*} \tag{8.22}$$

where Λ is diagonal with positive entries, $\Lambda = \mathrm{diag}\{\lambda_k\}$, and U is unitary, i.e., it satisfies $UU^* = U^*U = \mathrm{I}$. The columns of U, say $\{q_k\}$, are the orthonormal eigenvectors of R_u, namely, each q_k satisfies

$$R_u q_k = \lambda_k q_k, \qquad \|q_k\|^2 = 1$$

CONVERGENCE CONDITION

Now define the transformed weight-error vector

$$\boxed{x_i \triangleq U^* \tilde{w}_i} \tag{8.23}$$

Since U is unitary and, hence, invertible, x_i and $\tilde{w}_i$ determine each other uniquely. The vectors $\{x_i, \tilde{w}_i\}$ also have equal Euclidean norms since

$$\|x_i\|^2 = x_i^* x_i = \tilde{w}_i^* \underbrace{UU^*}_{\mathrm{I}} \tilde{w}_i = \tilde{w}_i^* \tilde{w}_i = \|\tilde{w}_i\|^2$$

Therefore, if x_i tends to zero then $\tilde{w}_i$ tends to zero and vice-versa. This means that we can instead seek a condition on μ to force x_i to tend to zero. It is more convenient to work with x_i because it satisfies a difference equation similar to (8.21), albeit one with a *diagonal* coefficient matrix. To see this, we multiply (8.21) by U^* from the left, and replace R_u by $U\Lambda U^*$ and I by UU^*, to get

CONVERGENCE CONDITION

$$\boxed{x_i = [\mathrm{I} - \mu\Lambda]x_{i-1}, \quad x_{-1} = U^* \tilde{w}_{-1} = \text{initial condition}} \tag{8.24}$$

The coefficient matrix for this difference equation is now diagonal and equal to $(\mathrm{I}-\mu\Lambda)$. It follows that the evolution of the individual entries of x_i are decoupled. Specifically, if we denote these individual entries by $x_i = \mathrm{col}\{x_1(i), x_2(i), \ldots, x_M(i)\}$, then (8.24) shows that the k-th entry of x_i satisfies

$$x_k(i) = (1 - \mu\lambda_k)x_k(i-1)$$

Iterating this recursion from time -1 up to time i gives

$$\boxed{x_k(i) = (1 - \mu\lambda_k)^{i+1}\, x_k(-1), \quad i \geq 0} \tag{8.25}$$

where $x_k(-1)$ denotes the k-th entry of the initial condition x_{-1}.

CONVERGENCE CONDITION

We refer to the coefficient $(1-\mu\lambda_k)$ as the *mode* associated with $x_k(i)$. Now in order for $x_k(i)$ to tend to zero regardless of $x_k(-1)$, the mode $(1-\mu\lambda_k)$ must have less than unit magnitude. This condition is both necessary and sufficient. Therefore, in order for all the entries of the transformed vector x_i to tend to zero, the step-size μ must satisfy

$$|1-\mu\lambda_k| < 1, \qquad \text{for all } k = 1, 2, \ldots, M \tag{8.26}$$

The modes $\{1-\mu\lambda_k\}$ are the eigenvalues of the coefficient matrix $(\mathrm{I}-\mu R_u)$ in (8.21), and we have therefore established our initial claim that all eigenvalues of this matrix must be less than one in magnitude in order for $\tilde{w}_i$ to converge to zero. The condition (8.26) is of course equivalent to choosing μ such that

$$\boxed{0 < \mu < 2/\lambda_{\max}}$$

where $\lambda_{\max}$ denotes the largest eigenvalue of R_u.

STEEPEST-DESCENT

Theorem 8.2 (Steepest-descent algorithm) Consider a zero-mean random variable $\boldsymbol{d}$ with variance σ_d^2 and a zero-mean random row vector $\boldsymbol{u}$ with $R_u = \mathsf{E}\,\boldsymbol{u}^*\boldsymbol{u} > 0$. Let $\lambda_{\max}$ denote the largest eigenvalue of R_u. The solution w^o of the linear least-mean-squares estimation problem

$$\min_{w} \; \mathsf{E}\,|\boldsymbol{d} - \boldsymbol{u}w|^2$$

can be obtained recursively as follows. Start with any initial guess w_{-1}, choose any step-size μ that satisfies $0 < \mu < 2/\lambda_{\max}$, and iterate for $i \geq 0$:

$$w_i = w_{i-1} + \mu[R_{du} - R_u w_{i-1}]$$

Then $w_i \to w^o$ as $i \to \infty$.

8.3 MORE GENERAL COST FUNCTIONS

With the above statement, we have achieved our original goal of deriving an iterative procedure for solving the least-mean-squares estimation problem

$$\min_{w} \; \mathsf{E}\,|\boldsymbol{d} - \boldsymbol{u}w|^2 \tag{8.27}$$

The ideas developed for this case can be applied to more general optimization problems, say

$$\min_{w} J(w)$$

with cost functions $J(w)$ that are *not* necessarily quadratic in w (see, e.g., Probs. III.15–III.18). The update recursion in these cases would continue to be of the form

$$\boxed{w_i = w_{i-1} - \mu\,[\nabla_w J(w_{i-1})]^*} \tag{8.28}$$

in terms of the gradient vector of $J(\cdot)$, and using sufficiently small step-sizes.

TRANSIENT BEHAVIOR

In order to gain further insight into the workings of steepest-descent methods, we shall continue to examine recursion (8.20), namely,

$$\boxed{w_i = w_{i-1} + \mu[R_{du} - R_u w_{i-1}], \quad i \geq 0, \quad w_{-1} = \text{initial guess}} \tag{9.1}$$

which pertains to the quadratic cost function (8.8). In particular, we shall now study more closely the manner by which the weight-error vector $\tilde{w}_i$ of (8.21) tends to zero. We repeat the weight-error vector recursion here for ease of reference,

$$\boxed{\tilde{w}_i = [\mathrm{I} - \mu R_u]\tilde{w}_{i-1}\,, \quad i \geq 0, \quad \tilde{w}_{-1} = \text{ initial condition}} \tag{9.2}$$

along with its transformed version (8.24):

$$\boxed{x_i = [\mathrm{I} - \mu\Lambda]x_{i-1}, \quad x_{-1} = U^* \tilde{w}_{-1} = \text{initial condition}} \tag{9.3}$$

9.1 MODES OF CONVERGENCE

To begin with, it is clear from (9.3) that the form of the exponential decay of the $k-$th entry of x_i, namely, $x_k(i)$, to zero depends on the value of the mode $1-\mu\lambda_k$. For instance, the sign of $1-\mu\lambda_k$ determines whether the convergence of $x_k(i)$ to zero occurs with or without *oscillation*. When $0 \leq 1-\mu\lambda_k < 1$ the decay of $x_k(i)$ to zero is monotonic. On the other hand, when $-1 < 1-\mu\lambda_k < 0$ the decay of $x_k(i)$ to zero is oscillatory.

Example 9.1 (Exponential decay)

Consider a two-dimensional data vector $\boldsymbol{u}$, i.e., $M = 2$ and R_u is 2×2. Assume the eigenvalues of R_u are $\lambda_{\min} = 1$ and $\lambda_{\max} = 4$. Then μ must satisfy $\mu < 2/\lambda_{\max} = 1/2$ for convergence of the steepest-descent method (9.1) to be guaranteed. If we choose $\mu = 2/5$, then the resulting modes $\{1-\mu\lambda_k\}$ will be $1-\mu\lambda_{\max} = -3/5 < 0$ and $1-\mu\lambda_{\min} = 3/5 > 0$. In this case, both entries of the transformed vector x_i will tend to zero; however, one entry will converge monotonically (the one associated with $\lambda_{\min}$) while the other entry will converge in an oscillatory manner (the one associated with $\lambda_{\max}$). This situation is illustrated in Fig. 9.1.

EXPONENTIAL DECAY

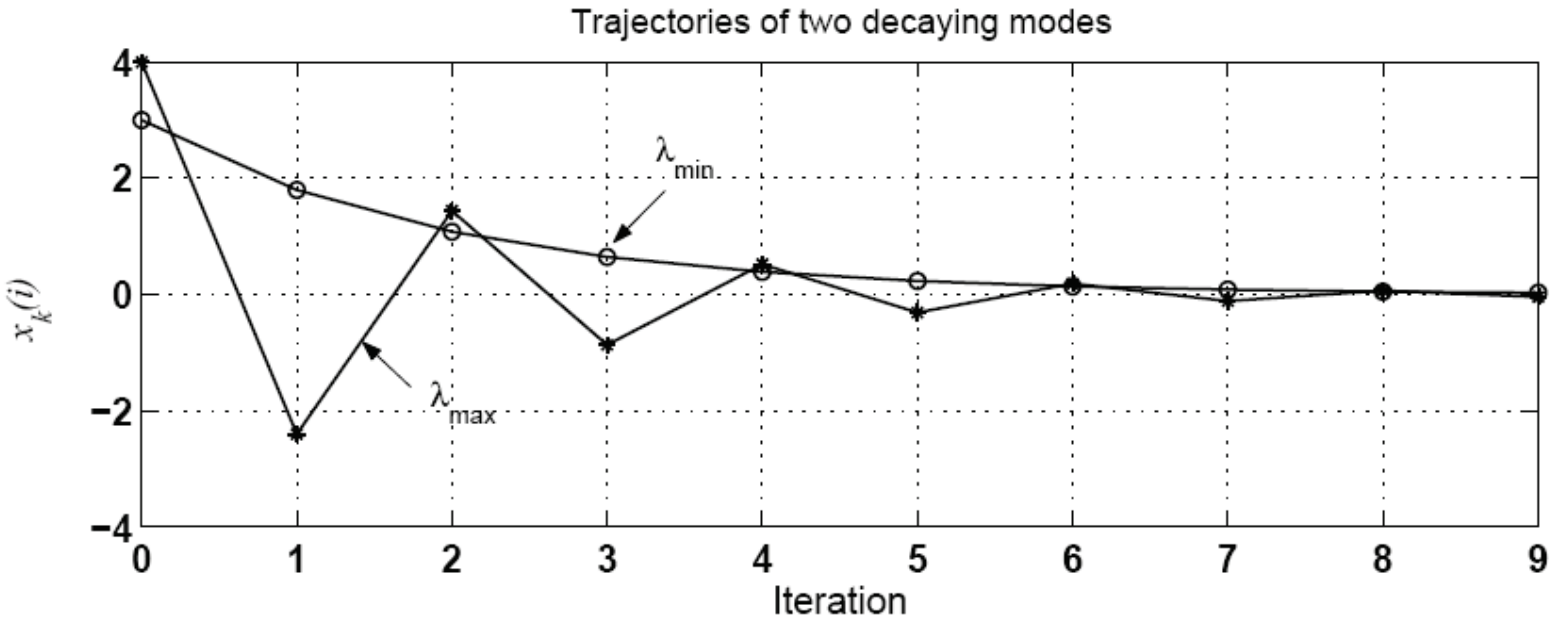

FIGURE 9.1 Two exponentially decaying modes from Ex. 9.1.

with the largest magnitude determines the entry of x_i that decays to zero at the slowest rate. The above example shows that the fastest and slowest rates of convergence are *not* necessarily the ones that are associated with the largest and smallest eigenvalues of R_u, respectively. For the numerical values used in the example, both $\lambda_{\min}$ and $\lambda_{\max}$ lead to modes $\{1 - \mu\lambda_k\}$ with identical magnitudes (equal to $3/5$). Consider the following alternative example.

FASTEST RATE OF DECAY

Example 9.2 (Fastest rate of decay)

Assume again that $M = 2$ and that $\lambda_{\min} = 1$ and $\lambda_{\max} = 3$. Then μ must satisfy $\mu < 2/\lambda_{\max} = 2/3$. Choose $\mu = 7/12$. Then $1 - \mu\lambda_{\max} = -9/12 < 0$ and $1 - \mu\lambda_{\min} = 5/12 > 0$. This shows that the entry of x_i that is associated with $\lambda_{\min}$ (rather than $\lambda_{\max}$) will decay at the fastest rate.

$\diamond$

9.3 WEIGHT-ERROR VECTOR CONVERGENCE

Let us now examine the convergence behavior of the weight-error vector. Since, as indicated by (8.23), $\tilde{w}_i = Ux_i$, it follows that $\tilde{w}_i$ is a linear combination of the columns of U, and the coefficients of this linear combination are the entries of x_i. Using (8.25) we then get

$$\tilde{w}_i = \sum_{k=1}^{M} q_k \, x_k(i) = \sum_{k=1}^{M} (1-\mu\lambda_k)^{i+1} \, q_k \, x_k(-1) \tag{9.6}$$

This expression shows that the convergence of $\tilde{w}_i$ to zero is also determined by the slowest converging mode among the $\{1-\mu\lambda_k\}$; once the faster modes have died out relative to the slowest mode, it is the slowest mode that ultimately determines the convergence rate of $\tilde{w}_i$ to zero. Assume that this slowest mode of convergence corresponds to an eigenvalue λ_{k_o}. Then (9.6) shows that in the limit, as $i \to \infty$, $\tilde{w}_i$ tends to zero along the direction of the associated eigenvector, q_{k_o}:

$$\tilde{w}_i \longrightarrow (1-\mu\lambda_{k_o})^{i+1} x_{k_o}(-1) \cdot q_{k_o} \quad \text{as} \quad i \longrightarrow \infty$$

9.4 TIME CONSTANTS

It is customary to describe the rate of convergence of a steepest-descent algorithm in terms of its time constants, which are defined as follows.

Recall that for an exponential function $f(t) = e^{-t/\tau}$, the time constant is τ and it corresponds to the time required for the value of the function to decay by a factor of e since

$$f(t+\tau) = e^{-(t+\tau)/\tau} = f(t)/e$$

Now, for an exponential discrete-time sequence of the form (cf. (8.25)):[3]

$$|x_k(i)|^2 = (1-\mu\lambda_k)^2\,|x_k(i-1)|^2, \quad i \geq 0$$

the value of $|x_k(i)|^2$ decays by $(1-\mu\lambda_k)^2$ at each iteration. Let T denote the time interval between one iteration and another, and let us fit a decaying exponential function through the points of the sequence $\{|x_k(i)|^2\}$.

TIME CONSTANTS

. Denote the function by $f(t) = e^{-t/\tau_k}$, with a time constant τ_k to be determined. Then we must have

$$\begin{aligned} f(t)|_{t=(i-1)T} &= |x_k(i-1)|^2 = e^{-(i-1)T/\tau_k} \\ f(t)|_{t=iT} &= (1-\mu\lambda_k)^2 |x_k(i-1)|^2 = e^{-iT/\tau_k} \end{aligned}$$

Dividing one expression by the other leads to $e^{-T/\tau_k} = (1-\mu\lambda_k)^2$ or, equivalently,

$$\tau_k \triangleq \frac{-T}{2\ln|1-\mu\lambda_k|} \qquad \text{(measured in units of time)}$$

This value measures the time that is needed for the value of $|x_k(i)|^2$ to decay by a factor of e, which corresponds to a decrease of the order of $10\log e \approx 4.4$ dB. It is common to normalize the value of τ_k to be independent of T. Thus let $\bar{\tau}_k = \tau_k/T$. Then

$$\boxed{\bar{\tau}_k \triangleq \frac{-1}{2\ln|1-\mu\lambda_k|}} \qquad \text{(measured in iterations)} \qquad (9.7)$$

TIME CONSTANTS

$$\bar{\tau}_k \triangleq \frac{-1}{2\ln|1-\mu\lambda_k|} \quad \text{(measured in iterations)} \tag{9.7}$$

This normalized value measures the approximate number of *iterations* that is needed for the value of $|x_k(i)|^2$ to decay by approximately 4.4 dB. For sufficiently small step-sizes (say, for $\mu\lambda_k \ll 1$), we have $\ln|1-\mu\lambda_k| \approx -\mu\lambda_k$ and we can approximate the expression for $\bar{\tau}_k$ by

$$\bar{\tau}_k \approx \frac{1}{2\mu\lambda_k} \quad \text{(iterations)}$$

Usually, the largest $\{\bar{\tau}_k, k = 1, 2, \ldots, M\}$ is taken as indicative of the time constant of the steepest-descent method.

9.5 LEARNING CURVE

Besides modes and time constants, it is also customary to characterize the convergence performance of a steepest-descent method in terms of its learning curve. Recall that our original problem is to determine the vector w that minimizes $J(w) = \mathsf{E}\,|\boldsymbol{d} - \boldsymbol{u}w|^2$. The steepest-descent recursion (9.1) provides successive iterates w_i with cost values $J(w_i) = \mathsf{E}\,|\boldsymbol{d} - \boldsymbol{u}w_i|^2$. Since, by choosing the step-size μ such that $\mu < 2/\lambda_{\max}$, we are guaranteed a sequence $\{w_i\}$ that converges to the optimal solution w^o, the same condition on μ also guarantees that the successive values $J(w_i)$ will converge to the minimum value of $J(w)$, namely (cf. (8.11)):

$$J(w_i) \longrightarrow J_{\min} = \sigma_d^2 - R_{ud}R_u^{-1}R_{du} \quad \text{as} \quad i \longrightarrow \infty$$

LEARNING CURVE

It turns out, as we now verify, that the decay of $J(w_i)$ to $J_{\min}$ is always monotonic. To see this, we recall from (8.10) that

$$J(w) \;=\; J_{\min} \;+\; (w - w^o)^* R_u (w - w^o) \tag{9.8}$$

i.e.,

$$\boxed{J(w_i) = J_{\min} \;+\; \tilde{w}_i^* R_u \tilde{w}_i} \tag{9.9}$$

The term $\tilde{w}_i^* R_u \tilde{w}_i$ represents the *excess mean-square error* at iteration i and it will be denoted by

$$\boxed{\xi(w_i) \;\stackrel{\Delta}{=}\; J(w_i) - J_{\min} \;=\; \tilde{w}_i^* R_u \tilde{w}_i} \tag{9.10}$$

It measures how far the cost at iteration i is from the minimum cost, $J_{\min}$.

LEARNING CURVE

If we replace $\tilde{w}_i$ by Ux_i, and use the eigen-decomposition (8.22), we obtain

$$J(w_i) \;=\; J_{\min} + \sum_{k=1}^{M} \lambda_k |x_k(i)|^2 \;=\; J_{\min} \;+\; \sum_{k=1}^{M} \lambda_k (1-\mu\lambda_k)^{2(i+1)} \; |x_k(-1)|^2$$

which confirms, under the requirement $0 < \mu < 2/\lambda_{\max}$, that $J(w_i) \to J_{\min}$ as $i \to \infty$, irrespective of the initial weight-error vector $\tilde{w}_{-1}$. Moreover, the convergence is both exponential *and* monotonic; it is monotonic since, for any k, the coefficient $\lambda_k(1-\mu\lambda_k)^2$ is positive.

LEARNING CURVE

The evolution of $J(w_i)$ as a function of i provides useful information about the learning behavior of a steepest-descent algorithm. For future reference, we shall adopt the following definition.

Definition 9.1 (Learning curve) The learning curve of a steepest-descent method associated with a cost function $J(w)$ is denoted by $J(i)$ and defined as $J(i) = J(w_{i-1})$ for $i \geq 0$. In particular, for the quadratic cost function $J(w)$ in (8.7), we obtain that its learning curve is given by

$$J(i) = \mathsf{E}\,|e(i)|^2 \quad \text{where} \quad e(i) = d - uw_{i-1}$$

is the so-called *a priori* output estimation error. In this case, the learning curve is also called the *mean-square-error* (MSE) curve.

LEARNING CURVE

Observe that the initial value of $J(i)$ is $J(0) = J(w_{-1})$. In general, the value of the learning curve at an iteration i is a measure of the cost that would result if we freeze the weight estimate at the value obtained at the prior iteration. Correspondingly, in the mean-square-error case, the learning curve is defined in terms of the variance of the *a priori* error $e(i)$ (which uses w_{i-1} and not w_i). Figure 9.4 shows a typical learning curve for the steepest-descent algorithm (9.1) with $M = 3$, $\lambda_{\min} = 0.3$, $\lambda_{\max} = 1$, and $\mu = 1.5385$. The modes $\{1 - \mu\lambda_k\}$ for this simulation are at $\{0.5385, 0.0769, -0.5385\}$.

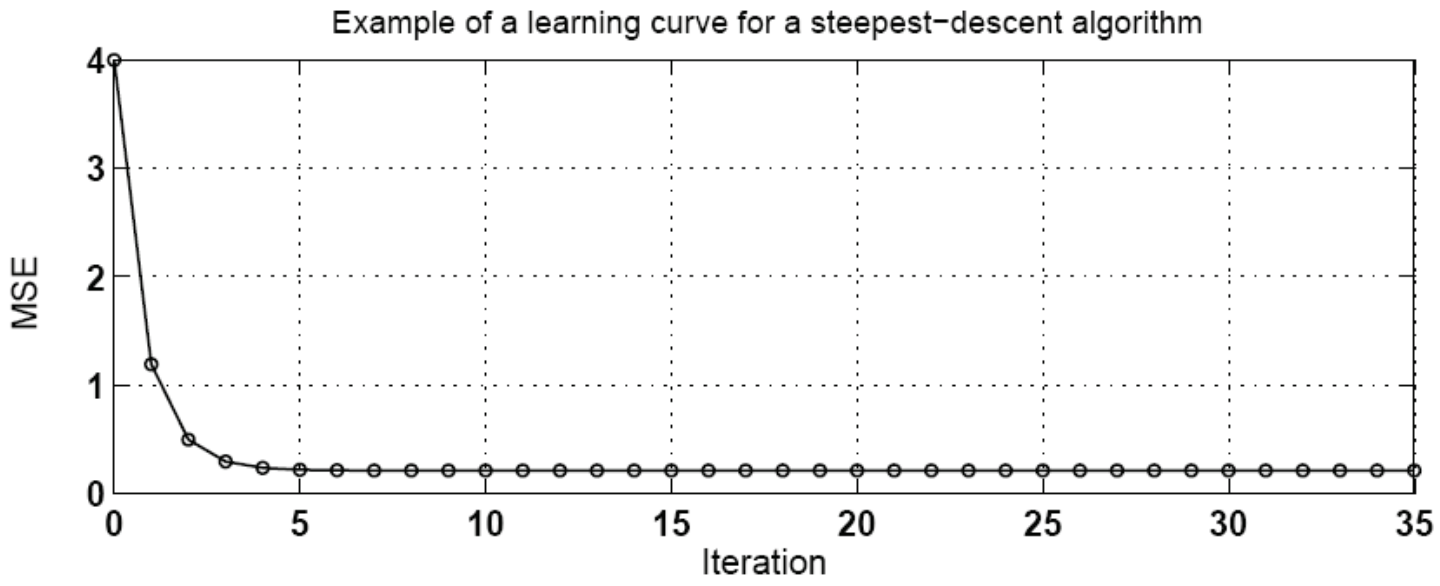

FIGURE 9.4 A typical learning curve $J(i)$ for algorithm (9.1).

9.6 CONTOUR CURVES OF THE ERROR SURFACE

Another useful way to examine the performance of a steepest-descent method is by examining the contours of constant value of its cost function, $J(w)$. These contour curves are more easily characterized if we perform a change of coordinates. For any w, we define $z = U^*(w - w^o)$ or, equivalently,

$$\boxed{w = w^o + Uz} \tag{9.11}$$

where U is obtained from the eigen-decompostion (8.22) of R_u. In other words, we replace the $w-$coordinate system by a $z-$coordinate system. The origin of the new system, $z = 0$, occurs at the point $w = w^o$ in the $w-$coordinate system. Likewise, the first basis vector in the $z-$coordinate system, namely, $z_1 = \text{col}\{1, 0, \ldots, 0\}$, corresponds to the vector $w = w^o + q_1$ in the $w-$coordinate system, where q_1 is the first column of U. This means that the first basis vector in the $z-$domain is obtained by shifting q_1 to w^o in the $w-$domain. A similar construction holds for the other basis vectors of the $z-$domain. This change of basis is illustrated in Fig. 9.5 for the case $M = 2$.

CHANGE OF COORDINAOES

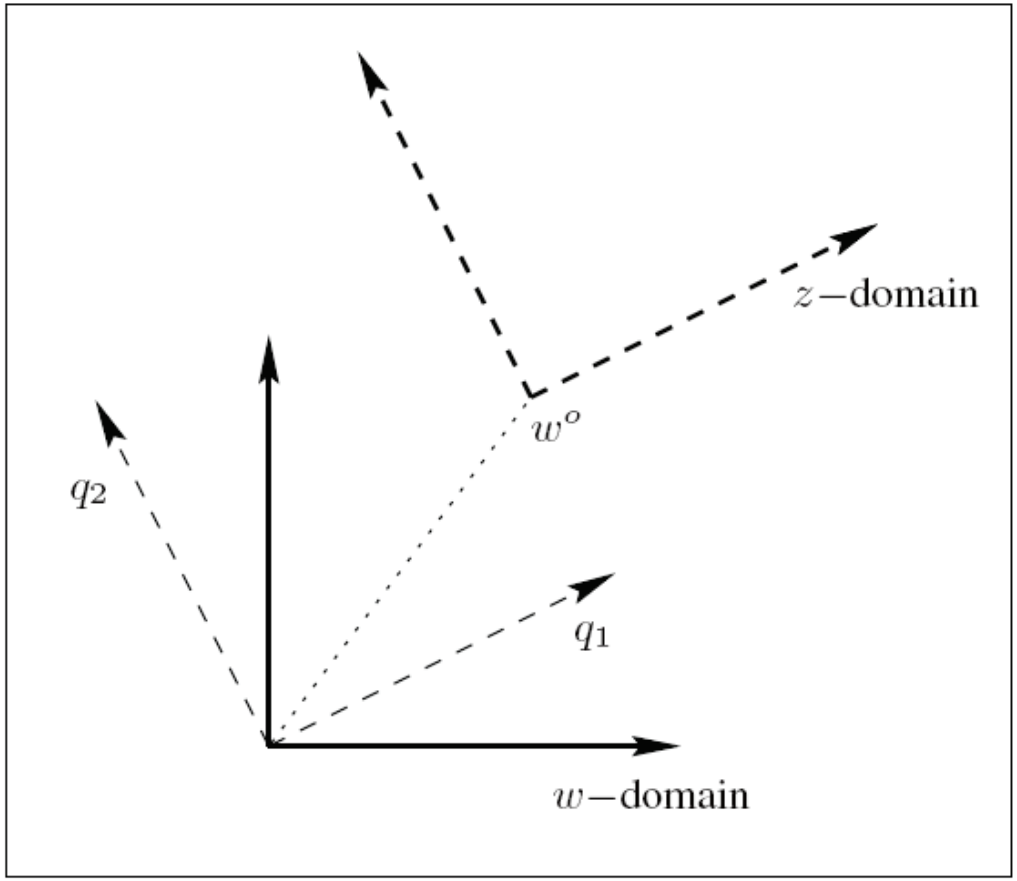

FIGURE 9.5 Change of coordinates from the $w-$domain to the $z-$domain, defined by $z = U^*(w - w^o)$, for the case $M = 2$.

CONTOUR CURVES

Using (9.8), and the eigen-decomposition $R_u = U\Lambda U^*$, we can express the cost function as

$$J(z) \;=\; J_{\min} \;+\; z^*\Lambda z \;=\; J_{\min} \;+\; \sum_{k=1}^{M} \lambda_k |z(k)|^2$$

where the $\{z(k)\}$ denote the entries of z. The contour curves of $J(z)$ (and, correspondingly, of $J(w)$), are the curves for which

$$J(z) \;=\; \text{a constant} \tag{9.12}$$

for different constant values. Equation (9.12) defines a hyper-ellipse in $M-$dimensions. The hyper-ellipse is centered at w^o and it has M principal axes. The principal axes are, by definition, the lines that pass through the origin and are normal to the hyper-ellipse. For $J(z)$, its principal axes coincide with the basis vectors of the $z-$coordinate system. To see this, note first that the gradient of $J(z)$ with respect to z^* is equal to Λz. Moreover, any line passing through the origin has the form λz for some scalar λ. Therefore, for any such line to be normal to the hyper-ellipse it should satisfy $\lambda z = \Lambda z$. This equality is possible only if λ is an eigenvalue of Λ and z the corresponding eigenvector. But since Λ is diagonal, this conclusion requires z to be one of the basis vectors. Therefore, the basis vectors of the $z-$coordinate system are normal to the hyper-ellipse and, consequently, they are the principal axes of the hyper-ellipse. We therefore find that the eigenvectors of R_u, when shifted to w^o, are the principal axes of the elliptic contours of $J(w)$.

CONTOUR CURVES

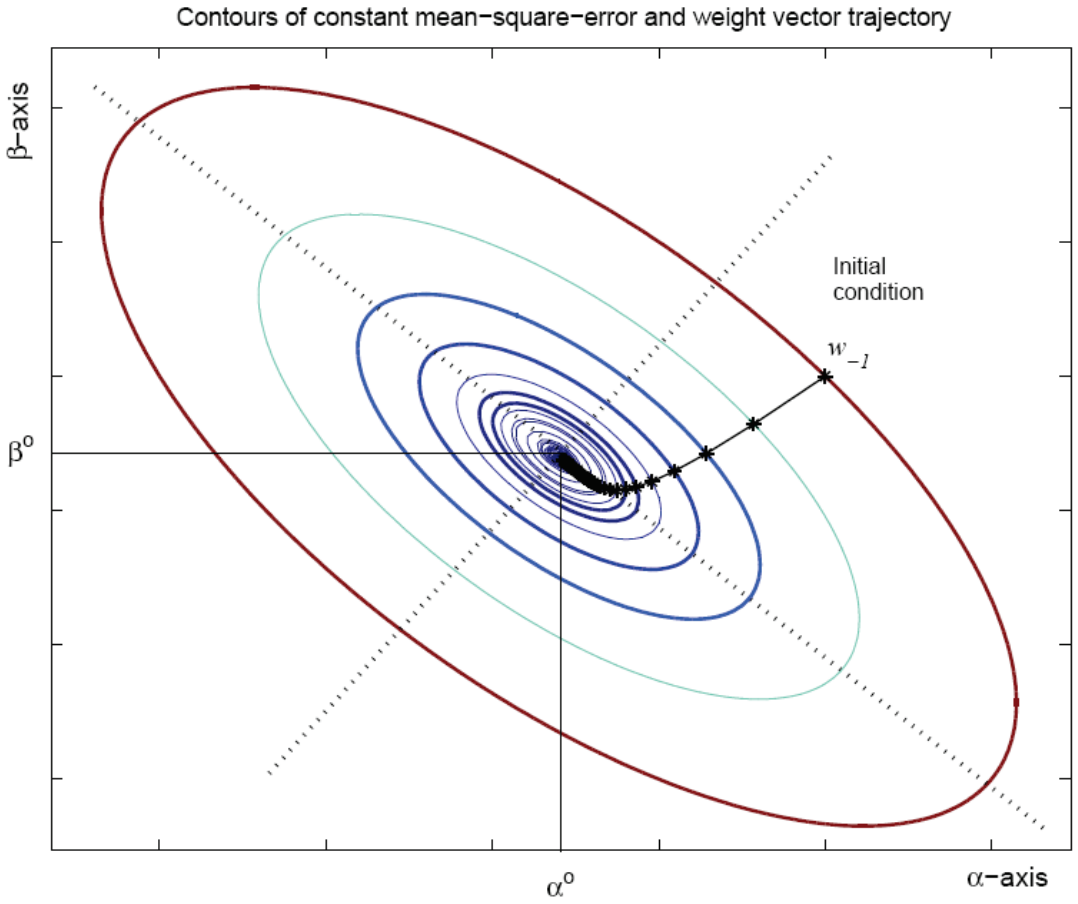

FIGURE 9.6 Elliptic contours of constant mean-square error in two dimensions, where the entries of w are denoted by $w = \text{col}\{\alpha, \beta\}$ and the entries of w^o are $\{\alpha^o, \beta^o\}$. The figure also indicates a typical trajectory starting from some initial condition w_{-1}.

9.7 ITERATION-DEPENDENT STEP-SIZES

The steepest-descent algorithm of Thm. 8.2 uses a constant step-size μ. In many instances, it may be desirable to vary the value of the step-size in order to obtain better control over the speed of convergence of the algorithm.

Condition for Convergence

Starting with (8.12), the arguments of Sec. 8.2 would still hold if we replace μ by an iteration-dependent positive step-size $\mu(i)$. In this case, recursion (9.1) would be replaced by

$$\boxed{w_i = w_{i-1} + \mu(i)[R_{du} - R_u w_{i-1}], \quad w_{-1} = \text{initial guess}} \tag{9.13}$$

Of course, not every choice of the step-size sequence $\{\mu(i)\}$ will guarantee convergence of w_i to w^o. For example, one might be tempted to extrapolate the arguments of Sec. 8.2 and conclude that by choosing $\mu(i)$ such that $\mu(i) < 2/\lambda_{\max}$ for all i, the weight-error vector $\tilde{w}_i$ will converge to zero. This conclusion is generally *false*.

ITERATION-DEPENDENT STEP-SIZES

Consider for illustration purposes, a scalar recursion of the form $x(i) = a(i)x(i-1)$ for $i \geq 0$. Then

$$x(i) = \left(\prod_{j=0}^{i} a(j) \right) x(-1)$$

If the $\{a(j)\}$ are such that $|a(j)| < 1$ for all finite j, it does *not* necessarily follow that $\prod_{j=0}^{i} a(j) \longrightarrow 0$ as $i \longrightarrow \infty$. That is, the product of infinitely many numbers that are all less than one in magnitude is not necessarily zero (see Prob. III.2). The product would tend to zero if all the $\{a(j)\}$ have their magnitudes *uniformly* bounded away from one, say $|a(j)| < \alpha < 1$ for all j and for some $\alpha > 0$.

The following statement provides one necessary condition on $\mu(i)$ in (9.13) for convergence; the proof is given in Probs. III.9 and III.10. As explained after the statement of the theorem, other conditions are possible.

ITERATION-DEPENDENT STEP-SIZES

Theorem 9.1 (Convergence condition) Given a zero-mean random variable $\boldsymbol{d}$ with variance σ_d^2 and a zero-mean random row vector $\boldsymbol{u}$ with $R_u = \mathsf{E}\,\boldsymbol{u}^*\boldsymbol{u} > 0$, the solution of the linear least-mean-squares estimation problem

$$\min_{w} \; \mathsf{E}\,|\boldsymbol{d} - \boldsymbol{u}w|^2$$

can be obtained recursively as follows. Start with any initial guess w_{-1}, choose a bounded step-size sequence $\mu(i)$ that tends to zero, i.e., $\mu(i) \to 0$, and iterate:

$$w_i = w_{i-1} + \mu(i)[R_{du} - R_u w_{i-1}], \quad i \geq 0$$

Then $w_i \to w^o$ as $i \to \infty$ if, and only if, the step-size sequence satisfies $\sum_{i=0}^{\infty} \mu(i) = \infty$. That is, if and only if, $\{\mu(i)\}$ is a divergent sequence.

CONVERGENCE CONDITION

The result of Thm. 9.1 requires the sequence $\mu(i)$ to tend to zero but not too fast since the sequence has to diverge as well. A typical sequence that satisfies the conditions of the theorem is

$$\mu(i) = \frac{\alpha}{i+\beta}, \quad \alpha > 0, \quad \beta > 0, \quad i \geq 0$$

Other examples are any bounded step-size sequences that satisfy both conditions

$$\sum_{i=0}^{\infty} \mu^2(i) < \infty \qquad \text{and} \qquad \sum_{i=0}^{\infty} \mu(i) = \infty$$

This is because the finite-energy condition on the sequence $\{\mu(i)\}$ guarantees $\mu(i) \to 0$. Still, convergence can occur even if the conditions of the theorem are violated. For example, in Prob. III.4 it is shown that with

$$\lim_{i \to \infty} \mu(i) = \alpha > 0$$

i.e., even with $\mu(i)$ tending to a nonzero limit, but as long as $\alpha < 2/\lambda_{\max}$, then w_i is guaranteed to converge to w^o.

9.8 NEWTON'S METHOD

We mentioned in our derivation of the steepest-descent algorithm in Sec. 8.2 that any choice for the search direction of the form (cf. (8.18)):

$$p = -B\ [\nabla_w J(w_{i-1})]^*$$

for any positive-definite matrix B, can be used to enforce the condition

$$\mathsf{Re}\,[\nabla_w J(w_{i-1})p] < 0$$

We chose $B = \mathrm{I}$ in our earlier discussions, which led to the steepest-descent variants of Thms. 8.2 and 9.1. But other choices for B are possible and they lead to different algorithms with different properties.

NEWTON'S METHOD

One useul choice for B for mean-square-error costs is

$$B \equiv \left[\nabla_w^2 J(w_{i-1}\right]^{-1}, \quad \text{where} \quad \nabla_w^2 J(w) \triangleq \nabla_{w^*} \left[\nabla_w J(w)\right]$$

in which case the search direction becomes

$$p = -\left[\nabla_w^2 J(w_{i-1})\right]^{-1} \left[\nabla_w J(w_{i-1})\right]^* \tag{9.17}$$

The resulting steepest-descent recursion (8.12) would be

$$\boxed{w_i = w_{i-1} - \mu \left[\nabla_w^2 J(w_{i-1})\right]^{-1} \left[\nabla_w J(w_{i-1})\right]^*, \quad i \geq 0, \quad w_{-1} = \text{initial guess}} \tag{9.18}$$

This recursive form is known as Newton's method.

For the quadratic cost function $J(w)$ of (8.8), we use (8.14) to find that (9.18) reduces to

$$\boxed{w_i = w_{i-1} + \mu R_u^{-1} \left[R_{du} - R_u w_{i-1}\right]} \tag{9.19}$$

We can examine the properties of this algorithm in much the same way as we did for recursion (9.1). So we shall be brief.

NEWTON'S METHOD

Convergence Properties

Subtracting both sides of (9.19) from w^o, and using the fact that w^o satisfies the normal equations $R_u w^o = R_{du}$, we arrive at the weight-error recursion

$$\tilde{w}_i = (1 - \mu)\tilde{w}_{i-1} \tag{9.20}$$

In contrast to (9.2) and (9.14), we find that the covariance matrix R_u does not appear any longer in (9.20). In particular, convergence is now guaranteed for all step-sizes μ that satisfy $0 < \mu < 2$; a condition that is independent of R_u.

Actually, the choice $\mu = 1$ in (9.20) leads to immediate convergence because $\tilde{w}_i = 0$ with no further iteration. This is a well-known property of Newton's method; the method guarantees convergence in a single iteration to the minimizing argument of a *quadratic* cost function by choosing $\mu = 1$. This fact can also be seen from recursion (9.19), which for $\mu = 1$ collapses to

$$w_i = w_{i-1} + R_u^{-1} R_{du} - w_{i-1} = w_{i-1} + w^o - w_{i-1} = w^o$$

Of course, applying Newton's method (9.19) to the solution of the least-mean-squares estimation problem (8.1) has the same complexity as using (8.4) since both schemes require the inversion of R_u.

NEWTON'S METHOD

Learning Curve

Recursion (9.19) estimates the vector w that minimizes $J(w) = \mathsf{E}\,|\boldsymbol{d} - \boldsymbol{u}w|^2$. It does so by evaluating successive iterates w_i with cost values $J(w_i) = \mathsf{E}\,|\boldsymbol{d} - \boldsymbol{u}w_i|^2$. Since, by choosing $0 < \mu < 2$, we are guaranteed a sequence $\{w_i\}$ that converges to w^o, this same condition on μ guarantees convergence of $J(w_i)$ to the minimum value of $J(w)$, namely (cf. (8.11)),

$$J(w_i) \;\longrightarrow\; J_{\min} = J(w^o) = \sigma_d^2 - R_{ud}R_u^{-1}R_{du} \quad \text{as} \quad i \longrightarrow \infty$$

The decay of $J(w_i)$ to $J_{\min}$ is again monotonic. This can be seen as follows. Using the representation (9.9),

$$J(w_i) = J_{\min} \;+\; \tilde{w}_i^* R_u \tilde{w}_i$$

replacing $\tilde{w}_i$ by Ux_i, where $\tilde{w}_i$ now evolves according to (9.20), and using the eigen-decomposition (8.22) for R_u, we obtain

$$J(w_i) = J_{\min} + \sum_{k=1}^{M} \lambda_k |x_k(i)|^2 \;=\; J_{\min} \;+\; (1-\mu)^{2(i+1)} \sum_{k=1}^{M} \lambda_k \; |x_k(-1)|^2$$

NEWTON'S METHOD

This expression confirms that, under the requirement $0 < \mu < 2$,

$$\lim_{i \to \infty} J(w_i) = J_{\min}$$

irrespective of the initial weight-error vector $\tilde{w}_{-1}$. Moreover, the convergence is both exponential *and* monotonic and, in contrast to the steepest-descent analysis of Sec. 9.5, convergence is now governed by a *single* mode at $(1 - \mu)^2$. Therefore, with Newton's method, we need only associate a single time constant that is equal to (cf. (9.7)):

$$\boxed{\bar{\tau} = -1/2 \ln(1 - \mu)} \quad \text{(iterations)}$$

The value of $\bar{\tau}$ is an approximation for the number of iterations that is needed for $\|\tilde{w}_i\|^2$ to decay by approximately 4.4 dB.

NEWTON'S METHOD

With regards to the contour curves of the error surface, they are still the same hyper-elliptic curves that were described in Sec. 9.6 (after all we are dealing with the same quadratic cost function $J(w)$ from (8.8)). As shown in that section, the principal axes of the contour curves are the eigenvectors of the covariance matrix R_u shifted to the location of w^o. Now, however, the search direction in Newton's method is *not* along the normal direction to the elliptic curves anymore, but along the line connecting w_{i-1} to w^o. To see this, recall that when $\mu = 1$, convergence of Newton's method occurs in a single step, which is only possible if the search direction is along the line connecting w_{i-1} to w^o. When $\mu \neq 1$, we are still moving along the same direction connecting w_{i-1} to w^o but for a shorter distance since from (9.19),

$$w_i = w_{i-1} + \mu(w^o - w_{i-1})$$

REGULARIZATION

Remark 9.1 (Regularization) When the Hessian matrix in (9.18) is close to singular, it is common to employ regularization, in which case Newton's method is sometimes known as the *Levenberg-Marquardt* method and it becomes

$$\boxed{w_i = w_{i-1} - \mu[\epsilon \mathrm{I} + \nabla_w^2 J(w_{i-1})]^{-1} \, [\nabla J(w_{i-1})]^*, \quad i \geq 0, \quad w_{-1} = \text{initial guess}}$$

The difference relative to Newton's recursion (9.17) is the addition of the small positive parameter ϵ. This algorithm can still be interpreted as a steepest-descent method of the form (8.12) with B in (8.18) chosen as

$$B = \left[\epsilon \mathrm{I} + \nabla_w^2 J(w_{i-1})\right]^{-1}$$

More generally, we can employ iteration-dependent step-sizes, $\mu(i)$, and iteration-dependent regularization parameters, $\epsilon(i) > 0$, and write instead

$$\boxed{w_i = w_{i-1} - \mu(i)[\epsilon(i) \mathrm{I} + \nabla_w^2 J(w_{i-1})]^{-1} \, [\nabla J(w_{i-1})]^*, \quad i \geq 0, \quad w_{-1} = \text{initial guess}}$$

$\diamond$