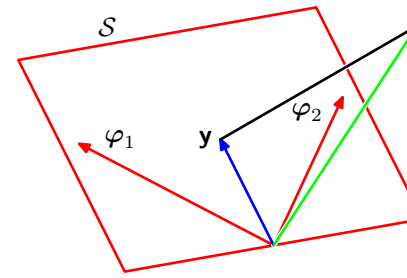

Figure 4.3 Geometrical interpretation of the least-squares solution in an N-dimensional space whose axes are the values of $t_1, \ldots, t_N$. The least-squares regression function is obtained by finding the orthogonal projection of the data vector $\mathbf{t}$ onto the subspace spanned by the basis functions $\phi_j(\mathbf{x})$ in which each basis function is viewed as a vector $\boldsymbol{\varphi}_j$ of length N with elements $\phi_j(\mathbf{x}_n)$.

4.1.4 Geometry of least squares

At this point, it is instructive to consider the geometrical interpretation of the least-squares solution. To do this, we consider an N-dimensional space whose axes are given by the t_n, so that $\mathbf{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$ is a vector in this space. Each basis function $\phi_j(\mathbf{x}_n)$, evaluated at the N data points, can also be represented as a vector in the same space, denoted by $\boldsymbol{\varphi}_j$, as illustrated in Figure 4.3. Note that $\boldsymbol{\varphi}_j$ corresponds to the jth column of $\boldsymbol{\Phi}$, whereas $\boldsymbol{\phi}(\mathbf{x}_n)$ corresponds to the transpose of the nth row of $\boldsymbol{\Phi}$. If the number M of basis functions is smaller than the number N of data points, then the M vectors $\phi_j(\mathbf{x}_n)$ will span a linear subspace $\mathcal{S}$ of dimensionality M. We define $\mathbf{y}$ to be an N-dimensional vector whose nth element is given by $y(\mathbf{x}_n, \mathbf{w})$, where $n = 1, \ldots, N$. Because $\mathbf{y}$ is an arbitrary linear combination of the vectors $\boldsymbol{\varphi}_j$, it can live anywhere in the M-dimensional subspace. The sum-of-squares error (4.11) is then equal (up to a factor of $1/2$) to the squared Euclidean distance between $\mathbf{y}$ and $\mathbf{t}$. Thus, the least-squares solution for $\mathbf{w}$ corresponds to that choice of $\mathbf{y}$ that lies in subspace $\mathcal{S}$ and is closest to $\mathbf{t}$. Intuitively, from Figure 4.3, we anticipate that this solution corresponds to the orthogonal projection of $\mathbf{t}$ onto the subspace $\mathcal{S}$. This is indeed the case, as can easily be verified by noting that the solution for $\mathbf{y}$ is given by $\boldsymbol{\Phi}\mathbf{w}_{\mathrm{ML}}$ and then confirming that this takes the form of an orthogonal projection.

Exercise 4.4

In practice, a direct solution of the normal equations can lead to numerical difficulties when $\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}$ is close to singular. In particular, when two or more of the basis vectors $\boldsymbol{\varphi}_j$ are co-linear, or nearly so, the resulting parameter values can have large magnitudes. Such near degeneracies will not be uncommon when dealing with real data sets. The resulting numerical difficulties can be addressed using the technique of *singular value decomposition*, or *SVD* (Deisenroth, Faisal, and Ong, 2020). Note that the addition of a regularization term ensures that the matrix is non-singular, even in the presence of degeneracies.

4.1.5 Sequential learning

The maximum likelihood solution (4.14) involves processing the entire training set in one go and is known as a *batch* method. This can become computationally costly for large data sets. If the data set is sufficiently large, it may be worthwhile to use *sequential* algorithms, also known as *online* algorithms, in which the data points are considered one at a time and the model parameters updated after each such presentation. Sequential learning is also appropriate for real-time applications in which the data observations arrive in a continuous stream and predictions must be

made before all the data points are seen.

We can obtain a sequential learning algorithm by applying the technique of *stochastic gradient descent*, also known as *sequential gradient descent*, as follows. If the error function comprises a sum over data points $E = \sum_n E_n$, then after presentation of data point n, the stochastic gradient descent algorithm updates the parameter vector $\mathbf{w}$ using

Chapter 7

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n \tag{4.21}$$

where τ denotes the iteration number, and η is a suitably chosen learning rate parameter. The value of $\mathbf{w}$ is initialized to some starting vector $\mathbf{w}^{(0)}$. For the sum-of-squares error function (4.11), this gives

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)\mathrm{T}}\boldsymbol{\phi}_n)\boldsymbol{\phi}_n \tag{4.22}$$

where $\boldsymbol{\phi}_n = \boldsymbol{\phi}(\mathbf{x}_n)$. This is known as the *least-mean-squares* or the *LMS algorithm*.

4.1.6 Regularized least squares

Section 1.2

We have previously introduced the idea of adding a regularization term to an error function to control over-fitting, so that the total error function to be minimized takes the form

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w}) \tag{4.23}$$

where λ is the regularization coefficient that controls the relative importance of the data-dependent error $E_D(\mathbf{w})$ and the regularization term $E_W(\mathbf{w})$. One of the simplest forms of regularizer is given by the sum of the squares of the weight vector elements:

$$E_W(\mathbf{w}) = \frac{1}{2}\sum_j w_j^2 = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}. \tag{4.24}$$

If we also consider the sum-of-squares error function given by

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2, \tag{4.25}$$

then the total error function becomes

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}. \tag{4.26}$$

In statistics, this regularizer provides an example of a *parameter shrinkage* method because it shrinks parameter values towards zero. It has the advantage that the error function remains a quadratic function of $\mathbf{w}$, and so its exact minimizer can be found in closed form. Specifically, setting the gradient of (4.26) with respect to $\mathbf{w}$ to zero and solving for $\mathbf{w}$ as before, we obtain

Exercise 4.6

$$\mathbf{w} = \left(\lambda\mathbf{I} + \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}. \tag{4.27}$$

This represents a simple extension of the least-squares solution (4.14).