w_1	w_2	w_3	Measured sag	Predicted sag
1	0	0	0.12	—
0	1	0	0.31	—
0	0	1	0.26	—
0.5	1.1	0.3	0.481	0.479
1.5	0.8	1.2	0.736	0.740

Table 2.1 Loadings on a bridge (first three columns), the associated measured sag at a certain point (fourth column), and the predicted sag using the linear model constructed from the first three experiments (fifth column).

the steel used to construct it. This is always done during the design of a bridge. The vector c can also be *measured* once the bridge is built, using the formula (2.3). We apply the load $w = e_1$, which means that we place a one ton load at the first load position on the bridge, with no load at the other positions. We can then measure the sag, which is c_1. We repeat this experiment, moving the one ton load to positions $2, 3, \ldots, n$, which gives us the coefficients $c_2, \ldots, c_n$. At this point we have the vector c, so we can now *predict* what the sag will be with any other loading. To check our measurements (and linearity of the sag function) we might measure the sag under other more complicated loadings, and in each case compare our prediction (*i.e.*, $c^T w$) with the actual measured sag.

Table 2.1 shows what the results of these experiments might look like, with each row representing an experiment (*i.e.*, placing the loads and measuring the sag). In the last two rows we compare the measured sag and the predicted sag, using the linear function with coefficients found in the first three experiments.

2.2 Taylor approximation

In many applications, scalar-valued functions of n variables, or relations between n variables and a scalar one, can be *approximated* as linear or affine functions. In these cases we sometimes refer to the linear or affine function relating the variables and the scalar variable as a *model*, to remind us that the relation is only an approximation, and not exact.

Differential calculus gives us an organized way to find an approximate affine model. Suppose that $f : \mathbf{R}^n \to \mathbf{R}$ is differentiable, which means that its partial derivatives exist (see §C.1). Let z be an n-vector. The (first-order) *Taylor approximation* of f near (or at) the point z is the function $\hat{f}(x)$ of x defined as

$$\hat{f}(x) = f(z) + \frac{\partial f}{\partial x_1}(z)(x_1 - z_1) + \cdots + \frac{\partial f}{\partial x_n}(z)(x_n - z_n),$$

where $\frac{\partial f}{\partial x_i}(z)$ denotes the partial derivative of f with respect to its ith argument, evaluated at the n-vector z. The hat appearing over f on the left-hand side is a common notational hint that it is an approximation of the function f. (The approximation is named after the mathematician Brook Taylor.)

The first-order Taylor approximation $\hat{f}(x)$ is a very good approximation of $f(x)$ when all x_i are near the associated z_i. Sometimes $\hat{f}$ is written with a second vector argument, as $\hat{f}(x; z)$, to show the point z at which the approximation is developed. The first term in the Taylor approximation is a constant; the other terms can be interpreted as the contributions to the (approximate) change in the function value (from $f(z)$) due to the changes in the components of x (from z).

Evidently $\hat{f}$ is an affine function of x. (It is sometimes called the *linear approximation* of f near z, even though it is in general affine, and not linear.) It can be written compactly using inner product notation as

$$\hat{f}(x) = f(z) + \nabla f(z)^T (x - z), \tag{2.5}$$

where $\nabla f(z)$ is an n-vector, the *gradient of f* (at the point z),

$$\nabla f(z) = \left[\begin{array}{c} \frac{\partial f}{\partial x_1}(z) \\ \vdots \\ \frac{\partial f}{\partial x_n}(z) \end{array}\right]. \tag{2.6}$$

The first term in the Taylor approximation (2.5) is the constant $f(z)$, the value of the function when $x = z$. The second term is the inner product of the gradient of f at z and the *deviation* or *perturbation* of x from z, *i.e.*, $x - z$.

We can express the first-order Taylor approximation as a linear function plus a constant,

$$\hat{f}(x) = \nabla f(z)^T x + (f(z) - \nabla f(z)^T z),$$

but the form (2.5) is perhaps easier to interpret.

The first-order Taylor approximation gives us an organized way to construct an affine approximation of a function $f : \mathbf{R}^n \to \mathbf{R}$, near a given point z, when there is a formula or equation that describes f, and it is differentiable. A simple example, for $n = 1$, is shown in figure 2.3. Over the full x-axis scale shown, the Taylor approximation $\hat{f}$ does not give a good approximation of the function f. But for x near z, the Taylor approximation is very good.

Example. Consider the function $f : \mathbf{R}^2 \to \mathbf{R}$ given by $f(x) = x_1 + \exp(x_2 - x_1)$, which is not linear or affine. To find the Taylor approximation $\hat{f}$ near the point $z = (1, 2)$, we take partial derivatives to obtain

$$\nabla f(z) = \left[\begin{array}{c} 1 - \exp(z_2 - z_1) \\ \exp(z_2 - z_1) \end{array}\right],$$

which evaluates to $(-1.7183, 2.7183)$ at $z = (1, 2)$. The Taylor approximation at $z = (1, 2)$ is then

$$\begin{array}{rcl} \hat{f}(x) & = & 3.7183 + (-1.7183, 2.7183)^T (x - (1, 2)) \\ & = & 3.7183 - 1.7183(x_1 - 1) + 2.7183(x_2 - 2). \end{array}$$

Table 2.2 shows $f(x)$ and $\hat{f}(x)$, and the approximation error $|\hat{f}(x) - f(x)|$, for some values of x relatively near z. We can see that $\hat{f}$ is indeed a very good approximation of f, especially when x is near z.

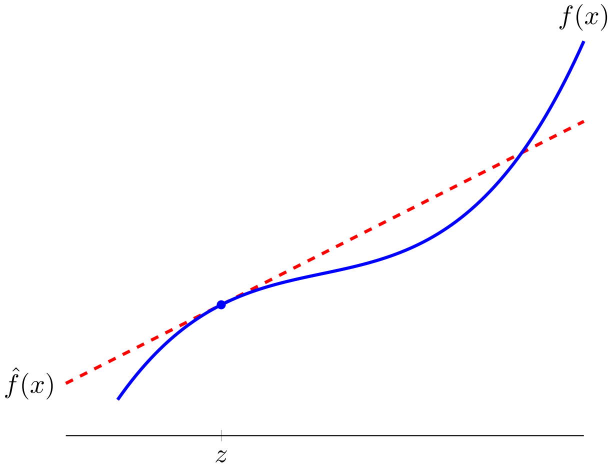

Figure 2.3 A function f of one variable, and the first-order Taylor approximation $\hat{f}(x) = f(z) + f'(z)(x - z)$ at z.

x	$f(x)$	$\hat{f}(x)$	$\lvert\hat{f}(x) - f(x)\rvert$
$(1.00, 2.00)$	3.7183	3.7183	0.0000
$(0.96, 1.98)$	3.7332	3.7326	0.0005
$(1.10, 2.11)$	3.8456	3.8455	0.0001
$(0.85, 2.05)$	4.1701	4.1119	0.0582
$(1.25, 2.41)$	4.4399	4.4032	0.0367

Table 2.2 Some values of x (first column), the function value $f(x)$ (second column), the Taylor approximation $\hat{f}(x)$ (third column), and the error (fourth column).

2.3 Regression model

In this section we describe a very commonly used affine function, especially when the n-vector x represents a feature vector. The affine function of x given by

$$\hat{y} = x^T\beta + v, \tag{2.7}$$

where β is an n-vector and v is a scalar, is called a *regression model*. In this context, the entries of x are called the *regressors*, and $\hat{y}$ is called the *prediction*, since the regression model is typically an approximation or prediction of some true value y, which is called the *dependent variable*, *outcome*, or *label*.

The vector β is called the *weight vector* or *coefficient vector*, and the scalar v is called the *offset* or *intercept* in the regression model. Together, β and v are called the *parameters* in the regression model. (We will see in chapter 13 how the parameters in a regression model can be estimated or guessed, based on some past or known observations of the feature vector x and the associated outcome y.) The symbol $\hat{y}$ is used in the regression model to emphasize that it is an *estimate* or *prediction* of some outcome y.

The entries in the weight vector have a simple interpretation: β_i is the amount by which $\hat{y}$ increases (if $\beta_i > 0$) when feature i increases by one (with all other features the same). If β_i is small, the prediction $\hat{y}$ doesn't depend too strongly on feature i. The offset v is the value of $\hat{y}$ when all features have the value 0.

The regression model is very interpretable when all of the features are Boolean, *i.e.*, have values that are either 0 or 1, which occurs when the features represent which of two outcomes holds. As a simple example consider a regression model for the lifespan of a person in some group, with $x_1 = 0$ if the person is female ($x_1 = 1$ if male), $x_2 = 1$ if the person has type II diabetes, and $x_3 = 1$ if the person smokes cigarettes. In this case, v is the regression model estimate for the lifespan of a female nondiabetic nonsmoker; β_1 is the increase in estimated lifespan if the person is male, β_2 is the increase in estimated lifespan if the person is diabetic, and β_3 is the increase in estimated lifespan if the person smokes cigarettes. (In a model that fits real data, all three of these coefficients would be negative, meaning that they decrease the regression model estimate of lifespan.)

Simplified regression model notation. Vector stacking can be used to lump the weights and offset in the regression model (2.7) into a single parameter vector, which simplifies the regression model notation a bit. We create a new regressor vector $\tilde{x}$, with $n+1$ entries, as $\tilde{x} = (1, x)$. We can think of $\tilde{x}$ as a new feature vector, consisting of all n original features, and one new feature added ($\tilde{x}_1$) at the beginning, which always has the value one. We define the parameter vector $\tilde{\beta} = (v, \beta)$, so the regression model (2.7) has the simple inner product form

$$\hat{y} = x^T\beta + v = \left[\begin{array}{c} 1 \\ x \end{array}\right]^T \left[\begin{array}{c} v \\ \beta \end{array}\right] = \tilde{x}^T\tilde{\beta}. \tag{2.8}$$

Often we omit the tildes, and simply write this as $\hat{y} = x^T\beta$, where we assume that the first feature in x is the constant 1. A feature that always has the value 1 is not particularly informative or interesting, but it does simplify the notation in a regression model.