Chapter 6

Approximation and fitting

6.1 Norm approximation

6.1.1 Basic norm approximation problem

The simplest norm approximation problem is an unconstrained problem of the form

\[
\text{minimize } \| Ax - b \| \tag{6.1}
\]

where \( A \in \mathbb{R}^{m \times n} \) and \( b \in \mathbb{R}^m \) are problem data, \( x \in \mathbb{R}^n \) is the variable, and \( \| \cdot \| \) is a norm on \( \mathbb{R}^m \). A solution of the norm approximation problem is sometimes called an approximate solution of \( Ax \approx b \), in the norm \( \| \cdot \| \). The vector

\[
r = Ax - b
\]

is called the residual for the problem; its components are sometimes called the individual residuals associated with \( x \).

The norm approximation problem (6.1) is a convex problem, and is solvable, i.e., there is always at least one optimal solution. Its optimal value is zero if and only if \( b \in \mathcal{R}(A) \); the problem is more interesting and useful, however, when \( b \not\in \mathcal{R}(A) \). We can assume without loss of generality that the columns of \( A \) are independent; in particular, that \( m \geq n \). When \( m = n \) the optimal point is simply \( A^{-1}b \), so we can assume that \( m > n \).

**Approximation interpretation**

By expressing \( Ax \) as

\[
Ax = x_1 a_1 + \cdots + x_n a_n,
\]

where \( a_1, \ldots, a_n \in \mathbb{R}^m \) are the columns of \( A \), we see that the goal of the norm approximation problem is to fit or approximate the vector \( b \) by a linear combination of the columns of \( A \), as closely as possible, with deviation measured in the norm \( \| \cdot \| \).

The approximation problem is also called the regression problem. In this context the vectors \( a_1, \ldots, a_n \) are called the regressors, and the vector \( x_1 a_1 + \cdots + x_n a_n \),
where $x$ is an optimal solution of the problem, is called the regression of $b$ (onto the regressors).

**Estimation interpretation**

A closely related interpretation of the norm approximation problem arises in the problem of estimating a parameter vector on the basis of an imperfect linear vector measurement. We consider a linear measurement model

$$y = Ax + v,$$

where $y \in \mathbb{R}^m$ is a vector measurement, $x \in \mathbb{R}^n$ is a vector of parameters to be estimated, and $v \in \mathbb{R}^m$ is some measurement error that is unknown, but presumed to be small (in the norm $\| \cdot \|$). The estimation problem is to make a sensible guess as to what $x$ is, given $y$.

If we guess that $x$ has the value $\hat{x}$, then we are implicitly making the guess that $v$ has the value $y - A\hat{x}$. Assuming that smaller values of $v$ (measured by $\| \cdot \|$) are more plausible than larger values, the most plausible guess for $x$ is

$$\hat{x} = \text{argmin}_z \|Az - y\|.$$

(These ideas can be expressed more formally in a statistical framework; see chapter 7.)

**Geometric interpretation**

We consider the subspace $\mathcal{A} = \mathcal{R}(A) \subseteq \mathbb{R}^m$, and a point $b \in \mathbb{R}^m$. A projection of the point $b$ onto the subspace $\mathcal{A}$, in the norm $\| \cdot \|$, is any point in $\mathcal{A}$ that is closest to $b$, i.e., any optimal point for the problem

$$\text{minimize} \quad \|u - b\|$$

subject to $u \in \mathcal{A}$.

Parametrizing an arbitrary element of $\mathcal{R}(A)$ as $u = Ax$, we see that solving the norm approximation problem (6.1) is equivalent to computing a projection of $b$ onto $\mathcal{A}$.

**Design interpretation**

We can interpret the norm approximation problem (6.1) as a problem of optimal design. The $n$ variables $x_1, \ldots, x_n$ are design variables whose values are to be determined. The vector $y = Ax$ gives a vector of $m$ results, which we assume to be linear functions of the design variables $x$. The vector $b$ is a vector of target or desired results. The goal is to choose a vector of design variables that achieves, as closely as possible, the desired results, i.e., $Ax \approx b$. We can interpret the residual vector $r$ as the deviation between the actual results (i.e., $Ax$) and the desired or target results (i.e., $b$). If we measure the quality of a design by the norm of the deviation between the actual results and the desired results, then the norm approximation problem (6.1) is the problem of finding the best design.
6.1 Norm approximation

Weighted norm approximation problems

An extension of the norm approximation problem is the weighted norm approximation problem

\[
\text{minimize} \quad \| W(Ax - b) \|
\]

where the problem data \( W \in \mathbb{R}^{m \times m} \) is called the weighting matrix. The weighting matrix is often diagonal, in which case it gives different relative emphasis to different components of the residual vector \( r = Ax - b \).

The weighted norm problem can be considered as a norm approximation problem with norm \( \| \cdot \| \), and data \( \tilde{A} = WA, \tilde{b} = Wb \), and therefore treated as a standard norm approximation problem (6.1). Alternatively, the weighted norm approximation problem can be considered a norm approximation problem with data \( A \) and \( b \), and the \( W \)-weighted norm defined by

\[
\| z \|_W = \| Wz \|
\]

(assuming here that \( W \) is nonsingular).

Least-squares approximation

The most common norm approximation problem involves the Euclidean or \( \ell_2 \)-norm. By squaring the objective, we obtain an equivalent problem which is called the least-squares approximation problem,

\[
\text{minimize} \quad \| Ax - b \|_2^2 = r_1^2 + r_2^2 + \cdots + r_m^2,
\]

where the objective is the sum of squares of the residuals. This problem can be solved analytically by expressing the objective as the convex quadratic function

\[
f(x) = x^T A^T Ax - 2b^T Ax + b^T b.
\]

A point \( x \) minimizes \( f \) if and only if

\[
\nabla f(x) = 2A^T Ax - 2A^T b = 0,
\]

i.e., if and only if \( x \) satisfies the so-called normal equations

\[
A^T Ax = A^T b,
\]

which always have a solution. Since we assume the columns of \( A \) are independent, the least-squares approximation problem has the unique solution \( x = (A^T A)^{-1} A^T b \).

Chebyshev or minimax approximation

When the \( \ell_\infty \)-norm is used, the norm approximation problem

\[
\text{minimize} \quad \| Ax - b \|_\infty = \max \{ |r_1|, \ldots, |r_m| \}
\]

is called the Chebyshev approximation problem, or minimax approximation problem, since we are to minimize the maximum (absolute value) residual. The Chebyshev approximation problem can be cast as an LP

\[
\begin{align*}
\text{minimize} & \quad t \\
\text{subject to} & \quad -t \mathbf{1} \leq Ax - b \leq t \mathbf{1},
\end{align*}
\]

with variables \( x \in \mathbb{R}^n \) and \( t \in \mathbb{R} \).
Sum of absolute residuals approximation

When the $\ell_1$-norm is used, the norm approximation problem

$$\minimize \|Ax - b\|_1 = |r_1| + \cdots + |r_m|$$

is called the sum of (absolute) residuals approximation problem, or, in the context of estimation, a robust estimator (for reasons that will be clear soon). Like the Chebyshev approximation problem, the $\ell_1$-norm approximation problem can be cast as an LP

$$\minimize 1^T t
\subjectto -t \preceq Ax - b \preceq t,$$

with variables $x \in \mathbb{R}^n$ and $t \in \mathbb{R}^m$.

6.1.2 Penalty function approximation

In $\ell_p$-norm approximation, for $1 \leq p < \infty$, the objective is

$$\left(|r_1|^p + \cdots + |r_m|^p\right)^{1/p}.$$

As in least-squares problems, we can consider the equivalent problem with objective

$$|r_1|^p + \cdots + |r_m|^p,$$

which is a separable and symmetric function of the residuals. In particular, the objective depends only on the amplitude distribution of the residuals, i.e., the residuals in sorted order.

We will consider a useful generalization of the $\ell_p$-norm approximation problem, in which the objective depends only on the amplitude distribution of the residuals. The penalty function approximation problem has the form

$$\minimize \phi(r_1) + \cdots + \phi(r_m)
\subjectto \quad r = Ax - b,$$

(6.2)

where $\phi : \mathbb{R} \to \mathbb{R}$ is called the (residual) penalty function. We assume that $\phi$ is convex, so the penalty function approximation problem is a convex optimization problem. In many cases, the penalty function $\phi$ is symmetric, nonnegative, and satisfies $\phi(0) = 0$, but we will not use these properties in our analysis.

Interpretation

We can interpret the penalty function approximation problem (6.2) as follows. For the choice $x$, we obtain the approximation $Ax$ of $b$, which has the associated residual vector $r$. A penalty function assesses a cost or penalty for each component of residual, given by $\phi(r_i)$; the total penalty is the sum of the penalties for each residual, i.e., $\phi(r_1) + \cdots + \phi(r_m)$. Different choices of $x$ lead to different resulting residuals, and therefore, different total penalties. In the penalty function approximation problem, we minimize the total penalty incurred by the residuals.
6.1 Norm approximation

Figure 6.1 Some common penalty functions: the quadratic penalty function \( \phi(u) = u^2 \), the deadzone-linear penalty function with deadzone width \( a = 1/4 \), and the log barrier penalty function with limit \( a = 1 \).

Example 6.1 Some common penalty functions and associated approximation problems.

- By taking \( \phi(u) = |u|^p \), where \( p \geq 1 \), the penalty function approximation problem is equivalent to the \( \ell_p \)-norm approximation problem. In particular, the quadratic penalty function \( \phi(u) = u^2 \) yields least-squares or Euclidean norm approximation, and the absolute value penalty function \( \phi(u) = |u| \) yields \( \ell_1 \)-norm approximation.

- The deadzone-linear penalty function (with deadzone width \( a > 0 \)) is given by

\[
\phi(u) = \begin{cases} 
0 & |u| \leq a \\
|u| - a & |u| > a.
\end{cases}
\]

The deadzone-linear function assesses no penalty for residuals smaller than \( a \).

- The log barrier penalty function (with limit \( a > 0 \)) has the form

\[
\phi(u) = \begin{cases} 
-a^2 \log(1 - (u/a)^2) & |u| < a \\
\infty & |u| \geq a.
\end{cases}
\]

The log barrier penalty function assesses an infinite penalty for residuals larger than \( a \).

A deadzone-linear, log barrier, and quadratic penalty function are plotted in figure 6.1. Note that the log barrier function is very close to the quadratic penalty for \( |u/a| \leq 0.25 \) (see exercise 6.1).

Scaling the penalty function by a positive number does not affect the solution of the penalty function approximation problem, since this merely scales the objective
function. But the shape of the penalty function has a large effect on the solution of the penalty function approximation problem. Roughly speaking, \( \phi(u) \) is a measure of our dislike of a residual of value \( u \). If \( \phi \) is very small (or even zero) for small values of \( u \), it means we care very little (or not at all) if residuals have these values. If \( \phi(u) \) grows rapidly as \( u \) becomes large, it means we have a strong dislike for large residuals; if \( \phi \) becomes infinite outside some interval, it means that residuals outside the interval are unacceptable. This simple interpretation gives insight into the solution of a penalty function approximation problem, as well as guidelines for choosing a penalty function.

As an example, let us compare \( \ell_1 \)-norm and \( \ell_2 \)-norm approximation, associated with the penalty functions \( \phi_1(u) = |u| \) and \( \phi_2(u) = u^2 \), respectively. For \( |u| = 1 \), the two penalty functions assign the same penalty. For small \( u \) we have \( \phi_1(u) \gg \phi_2(u) \), so \( \ell_1 \)-norm approximation puts relatively larger emphasis on small residuals compared to \( \ell_2 \)-norm approximation. For large \( u \) we have \( \phi_2(u) \gg \phi_1(u) \), so \( \ell_1 \)-norm approximation puts less weight on large residuals, compared to \( \ell_2 \)-norm approximation. This difference in relative weightings for small and large residuals is reflected in the solutions of the associated approximation problems. The amplitude distribution of the optimal residual for the \( \ell_1 \)-norm approximation problem will tend to have more zero and very small residuals, compared to the \( \ell_2 \)-norm approximation solution. In contrast, the \( \ell_2 \)-norm solution will tend to have relatively fewer large residuals (since large residuals incur a much larger penalty in \( \ell_2 \)-norm approximation than in \( \ell_1 \)-norm approximation).

**Example**

An example will illustrate these ideas. We take a matrix \( A \in \mathbb{R}^{100 \times 30} \) and vector \( b \in \mathbb{R}^{100} \) (chosen at random, but the results are typical), and compute the \( \ell_1 \)-norm and \( \ell_2 \)-norm approximate solutions of \( Ax \approx b \), as well as the penalty function approximations with a deadzone-linear penalty (with \( a = 0.5 \)) and log barrier penalty (with \( a = 1 \)). Figure 6.2 shows the four associated penalty functions, and the amplitude distributions of the optimal residuals for these four penalty approximations. From the plots of the penalty functions we note that

- The \( \ell_1 \)-norm penalty puts the most weight on small residuals and the least weight on large residuals.
- The \( \ell_2 \)-norm penalty puts very small weight on small residuals, but strong weight on large residuals.
- The deadzone-linear penalty function puts no weight on residuals smaller than 0.5, and relatively little weight on large residuals.
- The log barrier penalty puts weight very much like the \( \ell_2 \)-norm penalty for small residuals, but puts very strong weight on residuals larger than around 0.8, and infinite weight on residuals larger than 1.

Several features are clear from the amplitude distributions:

- For the \( \ell_1 \)-optimal solution, many residuals are either zero or very small. The \( \ell_1 \)-optimal solution also has relatively more large residuals.
Figure 6.2 Histogram of residual amplitudes for four penalty functions, with the (scaled) penalty functions also shown for reference. For the log barrier plot, the quadratic penalty is also shown, in dashed curve.
Approximation and fitting

Figure 6.3 A (nonconvex) penalty function that assesses a fixed penalty to residuals larger than a threshold (which in this example is one): \( \phi(u) = u^2 \) if \( |u| \leq 1 \) and \( \phi(u) = 1 \) if \( |u| > 1 \). As a result, penalty approximation with this function would be relatively insensitive to outliers.

- The \( \ell_2 \)-norm approximation has many modest residuals, and relatively few larger ones.

- For the deadzone-linear penalty, we see that many residuals have the value \( \pm 0.5 \), right at the edge of the ‘free’ zone, for which no penalty is assessed.

- For the log barrier penalty, we see that no residuals have a magnitude larger than 1, but otherwise the residual distribution is similar to the residual distribution for \( \ell_2 \)-norm approximation.

Sensitivity to outliers or large errors

In the estimation or regression context, an outlier is a measurement \( y_i = a_i^T x + v_i \) for which the noise \( v_i \) is relatively large. This is often associated with faulty data or a flawed measurement. When outliers occur, any estimate of \( x \) will be associated with a residual vector with some large components. Ideally we would like to guess which measurements are outliers, and either remove them from the estimation process or greatly lower their weight in forming the estimate. (We cannot, however, assign zero penalty for very large residuals, because then the optimal point would likely make all residuals large, which yields a total penalty of zero.) This could be accomplished using penalty function approximation, with a penalty function such as

\[
\phi(u) = \begin{cases} 
  u^2 & |u| \leq M \\
  M^2 & |u| > M,
\end{cases}
\]  

shown in figure 6.3. This penalty function agrees with least-squares for any residual smaller than \( M \), but puts a fixed weight on any residual larger than \( M \), no matter how much larger it is. In other words, residuals larger than \( M \) are ignored; they are assumed to be associated with outliers or bad data. Unfortunately, the penalty
6.1 Norm approximation

The solid line is the robust least-squares or Huber penalty function $\phi_{\text{hub}}$, with $M = 1$. For $|u| \leq M$ it is quadratic, and for $|u| > M$ it grows linearly.

Figure 6.4

function (6.3) is not convex, and the associated penalty function approximation problem becomes a hard combinatorial optimization problem.

The sensitivity of a penalty function based estimation method to outliers depends on the (relative) value of the penalty function for large residuals. If we restrict ourselves to convex penalty functions (which result in convex optimization problems), the ones that are least sensitive are those for which $\phi(u)$ grows linearly, i.e., like $|u|$, for large $u$. Penalty functions with this property are sometimes called robust, since the associated penalty function approximation methods are much less sensitive to outliers or large errors than, for example, least-squares.

One obvious example of a robust penalty function is $\phi(u) = |u|$, corresponding to $\ell_1$-norm approximation. Another example is the robust least-squares or Huber penalty function, given by

$$
\phi_{\text{hub}}(u) = \begin{cases} 
  u^2 & \text{if } |u| \leq M \\
  M(2|u| - M) & \text{if } |u| > M,
\end{cases}
$$

shown in figure 6.4. This penalty function agrees with the least-squares penalty function for residuals smaller than $M$, and then reverts to $\ell_1$-like linear growth for larger residuals. The Huber penalty function can be considered a convex approximation of the outlier penalty function (6.3), in the following sense: They agree for $|u| \leq M$, and for $|u| > M$, the Huber penalty function is the convex function closest to the outlier penalty function (6.3).

Example 6.2 Robust regression. Figure 6.5 shows 42 points $(t_i, y_i)$ in a plane, with two obvious outliers (one at the upper left, and one at lower right). The dashed line shows the least-squares approximation of the points by a straight line $f(t) = \alpha + \beta t$. The coefficients $\alpha$ and $\beta$ are obtained by solving the least-squares problem

$$
\text{minimize } \sum_{i=1}^{42} (y_i - \alpha - \beta t_i)^2,
$$
Figure 6.5 The 42 circles show points that can be well approximated by an affine function, except for the two outliers at upper left and lower right. The dashed line is the least-squares fit of a straight line $f(t) = \alpha + \beta t$ to the points, and is rotated away from the main locus of points, toward the outliers. The solid line shows the robust least-squares fit, obtained by minimizing Huber's penalty function with $M = 1$. This gives a far better fit to the non-outlier data.

with variables $\alpha$ and $\beta$. The least-squares approximation is clearly rotated away from the main locus of the points, toward the two outliers.

The solid line shows the robust least-squares approximation, obtained by minimizing the Huber penalty function

$$\text{minimize } \sum_{i=1}^{42} \phi_{\text{hub}}(y_i - \alpha - \beta t_i),$$

with $M = 1$. This approximation is far less affected by the outliers.

Since $\ell_1$-norm approximation is among the (convex) penalty function approximation methods that are most robust to outliers, $\ell_1$-norm approximation is sometimes called robust estimation or robust regression. The robustness property of $\ell_1$-norm estimation can also be understood in a statistical framework; see page 353.

**Small residuals and $\ell_1$-norm approximation**

We can also focus on small residuals. Least-squares approximation puts very small weight on small residuals, since $\phi(u) = u^2$ is very small when $u$ is small. Penalty functions such as the deadzone-linear penalty function put zero weight on small residuals. For penalty functions that are very small for small residuals, we expect the optimal residuals to be small, but not very small. Roughly speaking, there is little or no incentive to drive small residuals smaller.

In contrast, penalty functions that put relatively large weight on small residuals, such as $\phi(u) = |u|$, corresponding to $\ell_1$-norm approximation, tend to produce
optimal residuals many of which are very small, or even exactly zero. This means
that in $\ell_1$-norm approximation, we typically find that many of the equations are
satisfied exactly, i.e., we have $a_i^T x = b_i$ for many $i$. This phenomenon can be seen
in figure 6.2.

6.1.3 Approximation with constraints

It is possible to add constraints to the basic norm approximation problem (6.1). When these constraints are convex, the resulting problem is convex. Constraints arise for a variety of reasons.

- In an approximation problem, constraints can be used to rule out certain un-
  acceptable approximations of the vector $b$, or to ensure that the approximator
  $Ax$ satisfies certain properties.

- In an estimation problem, the constraints arise as prior knowledge of the
  vector $x$ to be estimated, or from prior knowledge of the estimation error $v$.

- Constraints arise in a geometric setting in determining the projection of a
  point $b$ on a set more complicated than a subspace, for example, a cone or
  polyhedron.

Some examples will make these clear.

Nonnegativity constraints on variables

We can add the constraint $x \succeq 0$ to the basic norm approximation problem:

$$\begin{align*}
\text{minimize} & \quad \|Ax - b\| \\
\text{subject to} & \quad x \succeq 0.
\end{align*}$$

In an estimation setting, nonnegativity constraints arise when we estimate a vector
$x$ of parameters known to be nonnegative, e.g., powers, intensities, or rates. The
geometric interpretation is that we are determining the projection of a vector $b$ onto
the cone generated by the columns of $A$. We can also interpret this problem as
approximating $b$ using a nonnegative linear (i.e., conic) combination of the columns
of $A$.

Variable bounds

Here we add the constraint $l \leq x \leq u$, where $l, u \in \mathbb{R}^n$ are problem parameters:

$$\begin{align*}
\text{minimize} & \quad \|Ax - b\| \\
\text{subject to} & \quad l \leq x \leq u.
\end{align*}$$

In an estimation setting, variable bounds arise as prior knowledge of intervals in
which each variable lies. The geometric interpretation is that we are determining
the projection of a vector $b$ onto the image of a box under the linear mapping
induced by $A$. 
Probability distribution

We can impose the constraint that \( x \) satisfy \( x \succeq 0, \ 1^Tx = 1 \):

\[
\begin{align*}
\text{minimize} & \quad \|Ax - b\| \\
\text{subject to} & \quad x \succeq 0, \ 1^Tx = 1.
\end{align*}
\]

This would arise in the estimation of proportions or relative frequencies, which are nonnegative and sum to one. It can also be interpreted as approximating \( b \) by a convex combination of the columns of \( A \). (We will have much more to say about estimating probabilities in §7.2.)

Norm ball constraint

We can add to the basic norm approximation problem the constraint that \( x \) lie in a norm ball:

\[
\begin{align*}
\text{minimize} & \quad \|Ax - b\| \\
\text{subject to} & \quad \|x - x_0\| \leq d,
\end{align*}
\]

where \( x_0 \) and \( d \) are problem parameters. Such a constraint can be added for several reasons.

- In an estimation setting, \( x_0 \) is a prior guess of what the parameter \( x \) is, and \( d \) is the maximum plausible deviation of our estimate from our prior guess. Our estimate of the parameter \( x \) is the value \( \hat{x} \) which best matches the measured data (i.e., minimizes \( \|Az - b\| \) among all plausible candidates (i.e., \( z \) that satisfy \( \|z - x_0\| \leq d \)).

- The constraint \( \|x - x_0\| \leq d \) can denote a trust region. Here the linear relation \( y = Ax \) is only an approximation of some nonlinear relation \( y = f(x) \) that is valid when \( x \) is near some point \( x_0 \), specifically \( \|x - x_0\| \leq d \). The problem is to minimize \( \|Ax - b\| \) but only over those \( x \) for which the model \( y = Ax \) is trusted.

These ideas also come up in the context of regularization; see §6.3.2.

6.2 Least-norm problems

The basic least-norm problem has the form

\[
\begin{align*}
\text{minimize} & \quad \|x\| \\
\text{subject to} & \quad Ax = b
\end{align*}
\]

(6.5)

where the data are \( A \in \mathbb{R}^{m \times n} \) and \( b \in \mathbb{R}^m \), the variable is \( x \in \mathbb{R}^n \), and \( \| \cdot \| \) is a norm on \( \mathbb{R}^n \). A solution of the problem, which always exists if the linear equations \( Ax = b \) have a solution, is called a least-norm solution of \( Ax = b \). The least-norm problem is, of course, a convex optimization problem.

We can assume without loss of generality that the rows of \( A \) are independent, so \( m \leq n \). When \( m = n \), the only feasible point is \( x = A^{-1}b \); the least-norm problem is interesting only when \( m < n \), i.e., when the equation \( Ax = b \) is underdetermined.
6.2 Least-norm problems

Reformulation as norm approximation problem

The least-norm problem (6.5) can be formulated as a norm approximation problem by eliminating the equality constraint. Let \( x_0 \) be any solution of \( Ax = b \), and let \( Z \in \mathbb{R}^{n \times k} \) be a matrix whose columns are a basis for the nullspace of \( A \). The general solution of \( Ax = b \) can then be expressed as \( x_0 + Zu \) where \( u \in \mathbb{R}^k \). The least-norm problem (6.5) can be expressed as

\[
\text{minimize} \quad \|x_0 + Zu\|
\]

with variable \( u \in \mathbb{R}^k \), which is a norm approximation problem. In particular, our analysis and discussion of norm approximation problems applies to least-norm problems as well (when interpreted correctly).

Control or design interpretation

We can interpret the least-norm problem (6.5) as a problem of optimal design or optimal control. The \( n \) variables \( x_1, \ldots, x_n \) are design variables whose values are to be determined. In a control setting, the variables \( x_1, \ldots, x_n \) represent inputs, whose values we are to choose. The vector \( y = Ax \) gives \( m \) attributes or results of the design \( x \), which we assume to be linear functions of the design variables \( x \). The \( m < n \) equations \( Ax = b \) represent \( m \) specifications or requirements on the design. Since \( m < n \), the design is underspecified; there are \( n - m \) degrees of freedom in the design (assuming \( A \) is rank \( m \)).

Among all the designs that satisfy the specifications, the least-norm problem chooses the smallest design, as measured by the norm \( \| \cdot \| \). This can be thought of as the most efficient design, in the sense that it achieves the specifications \( Ax = b \), with the smallest possible \( x \).

Estimation interpretation

We assume that \( x \) is a vector of parameters to be estimated. We have \( m < n \) perfect (noise free) linear measurements, given by \( Ax = b \). Since we have fewer measurements than parameters to estimate, our measurements do not completely determine \( x \). Any parameter vector \( x \) that satisfies \( Ax = b \) is consistent with our measurements.

To make a good guess about what \( x \) is, without taking further measurements, we must use prior information. Suppose our prior information, or assumption, is that \( x \) is more likely to be small (as measured by \( \| \cdot \| \)) than large. The least-norm problem chooses as our estimate of the parameter vector \( x \) the one that is smallest (hence, most plausible) among all parameter vectors that are consistent with the measurements \( Ax = b \). (For a statistical interpretation of the least-norm problem, see page 359.)

Geometric interpretation

We can also give a simple geometric interpretation of the least-norm problem (6.5). The feasible set \( \{ x \mid Ax = b \} \) is affine, and the objective is the distance (measured by the norm \( \| \cdot \| \)) between \( x \) and the point 0. The least-norm problem finds the
point in the affine set with minimum distance to 0, i.e., it determines the projection of the point 0 on the affine set \( \{ x \mid Ax = b \} \).

**Least-squares solution of linear equations**

The most common least-norm problem involves the Euclidean or \( \ell_2 \)-norm. By squaring the objective we obtain the equivalent problem

\[
\begin{align*}
\text{minimize} & \quad \| x \|_2^2 \\
\text{subject to} & \quad Ax = b,
\end{align*}
\]

the unique solution of which is called the *least-squares solution* of the equations \( Ax = b \). Like the least-squares approximation problem, this problem can be solved analytically. Introducing the dual variable \( \nu \in \mathbb{R}^m \), the optimality conditions are

\[
2x^* + A^T \nu^* = 0, \quad Ax^* = b,
\]

which is a pair of linear equations, and readily solved. From the first equation we obtain \( x^* = -(1/2)A^T \nu^* \); substituting this into the second equation we obtain \( -(1/2)AA^T \nu^* = b \), and conclude

\[
\nu^* = -2(AA^T)^{-1}b, \quad x^* = A^T(AA^T)^{-1}b.
\]

(Since \( \text{rank } A = m < n \), the matrix \( AA^T \) is invertible.)

**Least-penalty problems**

A useful variation on the least-norm problem (6.5) is the *least-penalty problem*

\[
\begin{align*}
\text{minimize} & \quad \phi(x_1) + \cdots + \phi(x_n) \\
\text{subject to} & \quad Ax = b, \quad (6.6)
\end{align*}
\]

where \( \phi : \mathbb{R} \to \mathbb{R} \) is convex, nonnegative, and satisfies \( \phi(0) = 0 \). The penalty function value \( \phi(u) \) quantifies our dislike of a component of \( x \) having value \( u \); the least-penalty problem then finds \( x \) that has least total penalty, subject to the constraint \( Ax = b \).

All of the discussion and interpretation of penalty functions in penalty function approximation can be transposed to the least-penalty problem, by substituting the amplitude distribution of \( x \) (in the least-penalty problem) for the amplitude distribution of the residual \( r \) (in the penalty approximation problem).

**Sparse solutions via least \( \ell_1 \)-norm**

Recall from the discussion on page 300 that \( \ell_1 \)-norm approximation gives relatively large weight to small residuals, and therefore results in many optimal residuals small, or even zero. A similar effect occurs in the least-norm context. The least \( \ell_1 \)-norm problem,

\[
\begin{align*}
\text{minimize} & \quad \| x \|_1 \\
\text{subject to} & \quad Ax = b,
\end{align*}
\]

tends to produce a solution \( x \) with a large number of components equal to zero. In other words, the least \( \ell_1 \)-norm problem tends to produce *sparse* solutions of \( Ax = b \), often with \( m \) nonzero components.
It is easy to find solutions of \( Ax = b \) that have only \( m \) nonzero components. Choose any set of \( m \) indices (out of \( 1, \ldots, n \)) which are to be the nonzero components of \( x \). The equation \( Ax = b \) reduces to \( \tilde{A}\tilde{x} = b \), where \( \tilde{A} \) is the \( m \times m \) submatrix of \( A \) obtained by selecting only the chosen columns, and \( \tilde{x} \in \mathbb{R}^m \) is the subvector of \( x \) containing the \( m \) selected components. If \( \tilde{A} \) is nonsingular, then we can take \( \tilde{x} = \tilde{A}^{-1}b \), which gives a feasible solution \( x \) with \( m \) or less nonzero components. If \( \tilde{A} \) is singular and \( b \notin \mathbb{R}(\tilde{A}) \), the equation \( \tilde{A}\tilde{x} = b \) is unsolvable, which means there is no feasible \( x \) with the chosen set of nonzero components. If \( \tilde{A} \) is singular and \( b \in \mathbb{R}(\tilde{A}) \), there is a feasible solution with fewer than \( m \) nonzero components.

This approach can be used to find the smallest \( x \) with \( m \) (or fewer) nonzero entries, but in general requires examining and comparing all \( n!/(m!(n-m)!)) \) choices of \( m \) nonzero coefficients of the \( n \) coefficients in \( x \). Solving the least \( \ell_1 \)-norm problem, on the other hand, gives a good heuristic for finding a sparse, and small, solution of \( Ax = b \).

### 6.3 Regularized approximation

#### 6.3.1 Bi-criterion formulation

In the basic form of regularized approximation, the goal is to find a vector \( x \) that is small (if possible), and also makes the residual \( Ax - b \) small. This is naturally described as a (convex) vector optimization problem with two objectives, \( \|Ax - b\| \) and \( \|x\| \):

\[
\text{minimize (w.r.t. } \mathbb{R}^n_+) \quad (\|Ax - b\|, \|x\|). \tag{6.7}
\]

The two norms can be different: the first, used to measure the size of the residual, is on \( \mathbb{R}^m \); the second, used to measure the size of \( x \), is on \( \mathbb{R}^n \).

The optimal trade-off between the two objectives can be found using several methods. The optimal trade-off curve of \( \|Ax - b\| \) versus \( \|x\| \), which shows how large one of the objectives must be made to have the other one small, can then be plotted. One endpoint of the optimal trade-off curve between \( \|Ax - b\| \) and \( \|x\| \) is easy to describe. The minimum value of \( \|x\| \) is zero, and is achieved only when \( x = 0 \). For this value of \( x \), the residual norm has the value \( \|b\| \).

The other endpoint of the trade-off curve is more complicated to describe. Let \( C \) denote the set of minimizers of \( \|Ax - b\| \) (with no constraint on \( \|x\| \)). Then any minimum norm point in \( C \) is Pareto optimal, corresponding to the other endpoint of the trade-off curve. In other words, Pareto optimal points at this endpoint are given by minimum norm minimizers of \( \|Ax - b\| \). If both norms are Euclidean, this Pareto optimal point is unique, and given by \( x = A^\dagger b \), where \( A^\dagger \) is the pseudo-inverse of \( A \). (See §4.7.6, page 184, and §A.5.4.)
Regularization

Regularization is a common scalarization method used to solve the bi-criterion problem (6.7). One form of regularization is to minimize the weighted sum of the objectives:

$$\text{minimize } \|Ax - b\| + \gamma \|x\|,$$

(6.8)

where $\gamma > 0$ is a problem parameter. As $\gamma$ varies over $(0, \infty)$, the solution of (6.8) traces out the optimal trade-off curve.

Another common method of regularization, especially when the Euclidean norm is used, is to minimize the weighted sum of squared norms, i.e.,

$$\text{minimize } \|Ax - b\|^2 + \delta \|x\|^2,$$

(6.9)

for a variety of values of $\delta > 0$.

These regularized approximation problems each solve the bi-criterion problem of making both $\|Ax - b\|$ and $\|x\|$ small, by adding an extra term or penalty associated with the norm of $x$.

Interpretations

Regularization is used in several contexts. In an estimation setting, the extra term penalizing large $\|x\|$ can be interpreted as our prior knowledge that $\|x\|$ is not too large. In an optimal design setting, the extra term adds the cost of using large values of the design variables to the cost of missing the target specifications.

The constraint that $\|x\|$ be small can also reflect a modeling issue. It might be, for example, that $y = Ax$ is only a good approximation of the true relationship $y = f(x)$ between $x$ and $y$. In order to have $f(x) \approx b$, we want $Ax \approx b$, and also need $x$ small in order to ensure that $f(x) \approx Ax$.

We will see in §6.4.1 and §6.4.2 that regularization can be used to take into account variation in the matrix $A$. Roughly speaking, a large $x$ is one for which variation in $A$ causes large variation in $Ax$, and hence should be avoided.

Regularization is also used when the matrix $A$ is square, and the goal is to solve the linear equations $Ax = b$. In cases where $A$ is poorly conditioned, or even singular, regularization gives a compromise between solving the equations (i.e., making $\|Ax - b\|$ zero) and keeping $x$ of reasonable size.

Regularization comes up in a statistical setting; see §7.1.2.

Tikhonov regularization

The most common form of regularization is based on (6.9), with Euclidean norms, which results in a (convex) quadratic optimization problem:

$$\text{minimize } \|Ax - b\|^2 + \delta \|x\|^2 = x^T(A^TA + \delta I)x - 2b^TAx + b^Tb.$$

(6.10)

This Tikhonov regularization problem has the analytical solution

$$x = (A^TA + \delta I)^{-1}A^Tb.$$

Since $A^TA + \delta I \succ 0$ for any $\delta > 0$, the Tikhonov regularized least-squares solution requires no rank (or dimension) assumptions on the matrix $A$. 
6.3 Regularized approximation

Smoothing regularization

The idea of regularization, i.e., adding to the objective a term that penalizes large $x$, can be extended in several ways. In one useful extension we add a regularization term of the form $\|Dx\|$, in place of $\|x\|$. In many applications, the matrix $D$ represents an approximate differentiation or second-order differentiation operator, so $\|Dx\|$ represents a measure of the variation or smoothness of $x$.

For example, suppose that the vector $x \in \mathbb{R}^n$ represents the value of some continuous physical parameter, say, temperature, along the interval $[0, 1]$: $x_i$ is the temperature at the point $i/n$. A simple approximation of the gradient or first derivative of the parameter near $i/n$ is given by $n(x_{i+1} - x_i)$, and a simple approximation of its second derivative is given by the second difference

$$n(n(x_{i+1} - x_i) - n(x_i - x_{i-1})) = n^2(x_{i+1} - 2x_i + x_{i-1}).$$

If $\Delta$ is the (tridiagonal, Toeplitz) matrix

$$\Delta = n^2 \begin{bmatrix}
1 & -2 & 1 & 0 & \cdots & 0 & 0 & 0 \\
0 & 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\
0 & 0 & 1 & -2 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & -2 & 1 & 0 \\
0 & 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\
0 & 0 & 0 & 0 & \cdots & 0 & 1 & -2 & 1
\end{bmatrix} \in \mathbb{R}^{(n-2) \times n},$$

then $\Delta x$ represents an approximation of the second derivative of the parameter, so $\|\Delta x\|_2^2$ represents a measure of the mean-square curvature of the parameter over the interval $[0, 1]$.

The Tikhonov regularized problem

$$\text{minimize} \quad \|Ax - b\|_2^2 + \delta \|\Delta x\|_2^2$$

can be used to trade off the objective $\|Ax - b\|_2^2$, which might represent a measure of fit, or consistency with experimental data, and the objective $\|\Delta x\|_2^2$, which is (approximately) the mean-square curvature of the underlying physical parameter. The parameter $\delta$ is used to control the amount of regularization required, or to plot the optimal trade-off curve of fit versus smoothness.

We can also add several regularization terms. For example, we can add terms associated with smoothness and size, as in

$$\text{minimize} \quad \|Ax - b\|_2^2 + \delta \|\Delta x\|_2^2 + \eta \|x\|_2^2.$$

Here, the parameter $\delta \geq 0$ is used to control the smoothness of the approximate solution, and the parameter $\eta \geq 0$ is used to control its size.

**Example 6.3** Optimal input design. We consider a dynamical system with scalar input sequence $u(0), u(1), \ldots, u(N)$, and scalar output sequence $y(0), y(1), \ldots, y(N)$, related by convolution:

$$y(t) = \sum_{\tau=0}^{t} h(\tau) u(t - \tau), \quad t = 0, 1, \ldots, N.$$
The sequence $h(0), h(1), \ldots, h(N)$ is called the convolution kernel or impulse response of the system.

Our goal is to choose the input sequence $u$ to achieve several goals.

- **Output tracking.** The primary goal is that the output $y$ should track, or follow, a desired target or reference signal $y_{des}$. We measure output tracking error by the quadratic function
  \[
  J_{\text{track}} = \frac{1}{N+1} \sum_{t=0}^{N} (y(t) - y_{des}(t))^2.
  \]

- **Small input.** The input should not be large. We measure the magnitude of the input by the quadratic function
  \[
  J_{\text{mag}} = \frac{1}{N+1} \sum_{t=0}^{N} u(t)^2.
  \]

- **Small input variations.** The input should not vary rapidly. We measure the magnitude of the input variations by the quadratic function
  \[
  J_{\text{der}} = \frac{1}{N} \sum_{t=0}^{N-1} (u(t+1) - u(t))^2.
  \]

By minimizing a weighted sum
\[
J_{\text{track}} + \delta J_{\text{der}} + \eta J_{\text{mag}},
\]
where $\delta > 0$ and $\eta > 0$, we can trade off the three objectives.

Now we consider a specific example, with $N = 200$, and impulse response
\[
h(t) = \frac{1}{9} (0.9)^t (1 - 0.4 \cos(2t)).
\]

Figure 6.6 shows the optimal input, and corresponding output (along with the desired trajectory $y_{des}$), for three values of the regularization parameters $\delta$ and $\eta$. The top row shows the optimal input and corresponding output for $\delta = 0, \eta = 0.005$. In this case we have some regularization for the magnitude of the input, but no regularization for its variation. While the tracking is good (i.e., we have $J_{\text{track}}$ is small), the input required is large, and rapidly varying. The second row corresponds to $\delta = 0, \eta = 0.05$. In this case we have more magnitude regularization, but still no regularization for variation in $u$. The corresponding input is indeed smaller, at the cost of a larger tracking error. The bottom row shows the results for $\delta = 0.3, \eta = 0.05$. In this case we have added some regularization for the variation. The input variation is substantially reduced, with not much increase in output tracking error.

**$\ell_1$-norm regularization**

Regularization with an $\ell_1$-norm can be used as a heuristic for finding a sparse solution. For example, consider the problem
\[
\text{minimize} \quad \|Ax - b\|_2 + \gamma \|x\|_1,
\]
(6.11)
Figure 6.6 Optimal inputs (left) and resulting outputs (right) for three values of the regularization parameters $\delta$ (which corresponds to input variation) and $\eta$ (which corresponds to input magnitude). The dashed line in the righthand plots shows the desired output $y_{\text{des}}$. Top row: $\delta = 0$, $\eta = 0.005$; middle row: $\delta = 0$, $\eta = 0.05$; bottom row: $\delta = 0.3$, $\eta = 0.05$. 
in which the residual is measured with the Euclidean norm and the regularization is done with an $\ell_1$-norm. By varying the parameter $\gamma$ we can sweep out the optimal trade-off curve between $\|Ax - b\|_2$ and $\|x\|_1$, which serves as an approximation of the optimal trade-off curve between $\|Ax - b\|_2$ and the sparsity or cardinality $\text{card}(x)$ of the vector $x$, i.e., the number of nonzero elements. The problem (6.11) can be recast and solved as an SOCP.

Example 6.4 Regressor selection problem. We are given a matrix $A \in \mathbb{R}^{m \times n}$, whose columns are potential regressors, and a vector $b \in \mathbb{R}^m$ that is to be fit by a linear combination of $k < n$ columns of $A$. The problem is to choose the subset of $k$ regressors to be used, and the associated coefficients. We can express this problem as

$$\minimize \quad \|Ax - b\|_2$$
$$\text{subject to} \quad \text{card}(x) \leq k.$$

In general, this is a hard combinatorial problem.

One straightforward approach is to check every possible sparsity pattern in $x$ with $k$ nonzero entries. For a fixed sparsity pattern, we can find the optimal $x$ by solving a least-squares problem, i.e., minimizing $\|\tilde{A}\tilde{x} - b\|_2$, where $\tilde{A}$ denotes the submatrix of $A$ obtained by keeping the columns corresponding to the sparsity pattern, and $\tilde{x}$ is the subvector with the nonzero components of $x$. This is done for each of the $n!/(k!(n-k)!)$ sparsity patterns with $k$ nonzeros.

A good heuristic approach is to solve the problem (6.11) for different values of $\gamma$, finding the smallest value of $\gamma$ that results in a solution with $\text{card}(x) = k$. We then fix this sparsity pattern and find the value of $x$ that minimizes $\|Ax - b\|_2$.

Figure 6.7 illustrates a numerical example with $A \in \mathbb{R}^{10 \times 20}$, $x \in \mathbb{R}^{20}$, $b \in \mathbb{R}^{10}$. The circles on the dashed curve are the (globally) Pareto optimal values for the trade-off between $\text{card}(x)$ (vertical axis) and the residual $\|Ax - b\|_2$ (horizontal axis). For each $k$, the Pareto optimal point was obtained by enumerating all possible sparsity patterns with $k$ nonzero entries, as described above. The circles on the solid curve were obtained with the heuristic approach, by using the sparsity patterns of the solutions of problem (6.11) for different values of $\gamma$. Note that for $\text{card}(x) = 1$, the heuristic method actually finds the global optimum.

This idea will come up again in basis pursuit (§6.5.4).

6.3.3 Reconstruction, smoothing, and de-noising

In this section we describe an important special case of the bi-criterion approximation problem described above, and give some examples showing how different regularization methods perform. In reconstruction problems, we start with a signal represented by a vector $x \in \mathbb{R}^n$. The coefficients $x_i$ correspond to the value of some function of time, evaluated (or sampled, in the language of signal processing) at evenly spaced points. It is usually assumed that the signal does not vary too rapidly, which means that usually, we have $x_i \approx x_{i+1}$. (In this section we consider signals in one dimension, e.g., audio signals, but the same ideas can be applied to signals in two or more dimensions, e.g., images or video.)
The signal $x$ is corrupted by an additive noise $v$:

$$x_{\text{cor}} = x + v.$$  

The noise can be modeled in many different ways, but here we simply assume that it is unknown, small, and, unlike the signal, rapidly varying. The goal is to form an estimate $\hat{x}$ of the original signal $x$, given the corrupted signal $x_{\text{cor}}$. This process is called signal reconstruction (since we are trying to reconstruct the original signal from the corrupted version) or de-noising (since we are trying to remove the noise from the corrupted signal). Most reconstruction methods end up performing some sort of smoothing operation on $x_{\text{cor}}$ to produce $\hat{x}$, so the process is also called smoothing.

One simple formulation of the reconstruction problem is the bi-criterion problem

$$\text{minimize (w.r.t. } R) \quad \left( \|\hat{x} - x_{\text{cor}}\|_2, \phi(\hat{x}) \right),$$  

where $\hat{x}$ is the variable and $x_{\text{cor}}$ is a problem parameter. The function $\phi : R^n \to R$ is convex, and is called the regularization function or smoothing objective. It is meant to measure the roughness, or lack of smoothness, of the estimate $\hat{x}$. The reconstruction problem (6.12) seeks signals that are close (in $\ell_2$-norm) to the corrupted signal, and that are smooth, i.e., for which $\phi(\hat{x})$ is small. The reconstruction problem (6.12) is a convex bi-criterion problem. We can find the Pareto optimal points by scalarization, and solving a (scalar) convex optimization problem.
Quadratic smoothing

The simplest reconstruction method uses the quadratic smoothing function

\[
\phi_{\text{quad}}(x) = \sum_{i=1}^{n-1} (x_{i+1} - x_i)^2 = \|Dx\|_2^2,
\]

where \(D \in \mathbb{R}^{(n-1) \times n}\) is the bidiagonal matrix

\[
D = \begin{bmatrix}
-1 & 1 & 0 & \cdots & 0 & 0 & 0 \\
0 & -1 & 1 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & -1 & 1 & 0 \\
0 & 0 & 0 & \cdots & 0 & -1 & 1
\end{bmatrix}.
\]

We can obtain the optimal trade-off between \(\|\hat{x} - x_{\text{cor}}\|_2\) and \(\|D\hat{x}\|_2\) by minimizing

\[
\|\hat{x} - x_{\text{cor}}\|_2^2 + \delta \|D\hat{x}\|_2^2,
\]

where \(\delta > 0\) parametrizes the optimal trade-off curve. The solution of this quadratic problem,

\[
\hat{x} = (I + \delta D^T D)^{-1} x_{\text{cor}},
\]

can be computed very efficiently since \(I + \delta D^T D\) is tridiagonal; see appendix C.

Quadratic smoothing example

Figure 6.8 shows a signal \(x \in \mathbb{R}^{4000}\) (top) and the corrupted signal \(x_{\text{cor}}\) (bottom). The optimal trade-off curve between the objectives \(\|\hat{x} - x_{\text{cor}}\|_2\) and \(\|D\hat{x}\|_2\) is shown in figure 6.9. The extreme point on the left of the trade-off curve corresponds to \(\hat{x} = x_{\text{cor}}\), and has objective value \(\|Dx_{\text{cor}}\|_2 = 4.4\). The extreme point on the right corresponds to \(\hat{x} = 0\), for which \(\|\hat{x} - x_{\text{cor}}\|_2 = \|x_{\text{cor}}\|_2 = 16.2\). Note the clear knee in the trade-off curve near \(\|\hat{x} - x_{\text{cor}}\|_2 \approx 3\).

Figure 6.10 shows three smoothed signals on the optimal trade-off curve, corresponding to \(\|\hat{x} - x_{\text{cor}}\|_2 = 8\) (top), 3 (middle), and 1 (bottom). Comparing the reconstructed signals with the original signal \(x\), we see that the best reconstruction is obtained for \(\|\hat{x} - x_{\text{cor}}\|_2 = 3\), which corresponds to the knee of the trade-off curve. For higher values of \(\|\hat{x} - x_{\text{cor}}\|_2\), there is too much smoothing; for smaller values there is too little smoothing.

Total variation reconstruction

Simple quadratic smoothing works well as a reconstruction method when the original signal is very smooth, and the noise is rapidly varying. But any rapid variations in the original signal will, obviously, be attenuated or removed by quadratic smoothing. In this section we describe a reconstruction method that can remove much of the noise, while still preserving occasional rapid variations in the original signal. The method is based on the smoothing function

\[
\phi_{tv}(\hat{x}) = \sum_{i=1}^{n-1} |\hat{x}_{i+1} - \hat{x}_i| = \|D\hat{x}\|_1,
\]
6.3 Regularized approximation

Figure 6.8 Top: the original signal $x \in \mathbb{R}^{4000}$. Bottom: the corrupted signal $x_{\text{cor}}$.

Figure 6.9 Optimal trade-off curve between $\|D\hat{x}\|_2$ and $\|\hat{x} - x_{\text{cor}}\|_2$. The curve has a clear knee near $\|\hat{x} - x_{\text{cor}}\| \approx 3$. 
which is called the total variation of $x \in \mathbb{R}^n$. Like the quadratic smoothness measure $\phi_{\text{quad}}$, the total variation function assigns large values to rapidly varying $\hat{x}$. The total variation measure, however, assigns relatively less penalty to large values of $|x_{i+1} - x_i|$.

**Total variation reconstruction example**

Figure 6.11 shows a signal $x \in \mathbb{R}^{2000}$ (in the top plot), and the signal corrupted with noise $x_{\text{cor}}$. The signal is mostly smooth, but has several rapid variations or jumps in value; the noise is rapidly varying.

We first use quadratic smoothing. Figure 6.12 shows three smoothed signals on the optimal trade-off curve between $\|D\hat{x}\|_2$ and $\|\hat{x} - x_{\text{cor}}\|_2$. In the first two signals, the rapid variations in the original signal are also smoothed. In the third signal the steep edges in the signal are better preserved, but there is still a significant amount of noise left.

Now we demonstrate total variation reconstruction. Figure 6.13 shows the optimal trade-off curve between $\|D\hat{x}\|_1$ and $\|\hat{x} - x_{\text{corr}}\|_2$. Figure 6.14 shows the reconstructed signals on the optimal trade-off curve, for $\|D\hat{x}\|_1 = 5$ (top), $\|D\hat{x}\|_1 = 8$ (middle), and $\|D\hat{x}\|_1 = 10$ (bottom). We observe that, unlike quadratic smoothing, total variation reconstruction preserves the sharp transitions in the signal.
Figure 6.11 A signal $x \in \mathbb{R}^{2000}$, and the corrupted signal $x_{\text{cor}} \in \mathbb{R}^{2000}$. The noise is rapidly varying, and the signal is mostly smooth, with a few rapid variations.
Figure 6.12 Three quadratically smoothed signals \( \hat{x} \). The top one corresponds to \( \| \hat{x} - x_{\text{cor}} \|_2 = 10 \), the middle one to \( \| \hat{x} - x_{\text{cor}} \|_2 = 7 \), and the bottom one to \( \| \hat{x} - x_{\text{cor}} \|_2 = 4 \). The top one greatly reduces the noise, but also excessively smoothes out the rapid variations in the signal. The bottom smoothed signal does not give enough noise reduction, and still smoothes out the rapid variations in the original signal. The middle smoothed signal gives the best compromise, but still smooths out the rapid variations.

Figure 6.13 Optimal trade-off curve between \( \| D\hat{x} \|_1 \) and \( \| \hat{x} - x_{\text{cor}} \|_2 \).
Figure 6.14 Three reconstructed signals \( \hat{x} \), using total variation reconstruction. The top one corresponds to \( \|D\hat{x}\|_1 = 5 \), the middle one to \( \|D\hat{x}\|_1 = 8 \), and the bottom one to \( \|D\hat{x}\|_1 = 10 \). The bottom one does not give quite enough noise reduction, while the top one eliminates some of the slowly varying parts of the signal. Note that in total variation reconstruction, unlike quadratic smoothing, the sharp changes in the signal are preserved.