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

$$\text{minimize} \quad \|Ax - b\| \tag{6.1}$$

where $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$ are problem data, $x \in \mathbf{R}^n$ is the variable, and $\| \cdot \|$ is a norm on $\mathbf{R}^m$. 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 $b \notin \mathcal{R}(A)$. 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 $a_1, \ldots, a_n \in \mathbf{R}^m$ 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 $y \in \mathbf{R}^m$ is a vector measurement, $x \in \mathbf{R}^n$ is a vector of parameters to be estimated, and $v \in \mathbf{R}^m$ 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

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

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

Geometric interpretation

We consider the subspace $\mathcal{A} = \mathcal{R}(A) \subseteq \mathbf{R}^m$, and a point $b \in \mathbf{R}^m$. 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

$$\begin{array}{ll} \text{minimize} & \|u - b\| \\ \text{subject to} & u \in \mathcal{A}. \end{array}$$

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.

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 $W \in \mathbf{R}^{m\times m}$ 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 ℓ_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

$$f(x) = x^T A^T A x - 2b^T A x + b^T b.$$

A point x minimizes f if and only if

$$\nabla f(x) = 2A^T A x - 2A^T b = 0,$$

i.e., if and only if x satisfies the so-called *normal equations*

$$A^T A x = A^T b,$$

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 ℓ_∞-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

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & -t\mathbf{1} \preceq Ax - b \preceq t\mathbf{1}, \end{array}$$

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

Sum of absolute residuals approximation

When the ℓ_1-norm is used, the norm approximation problem

$$\text{minimize} \quad \|Ax - b\|_1 = |r_1| + \cdots + |r_m|$$

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 ℓ_1-norm approximation problem can be cast as an LP

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T t \\ \text{subject to} & -t \preceq Ax - b \preceq t, \end{array}$$

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

6.1.2 Penalty function approximation

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

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

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 ℓ_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

$$\begin{array}{ll} \text{minimize} & \phi(r_1) + \cdots + \phi(r_m) \\ \text{subject to} & r = Ax - b, \end{array} \tag{6.2}$$

where $\phi : \mathbf{R} \to \mathbf{R}$ is called the (residual) *penalty function*. We assume that ϕ is convex, so the penalty function approximation problem is a convex optimization problem. In many cases, the penalty function ϕ 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.

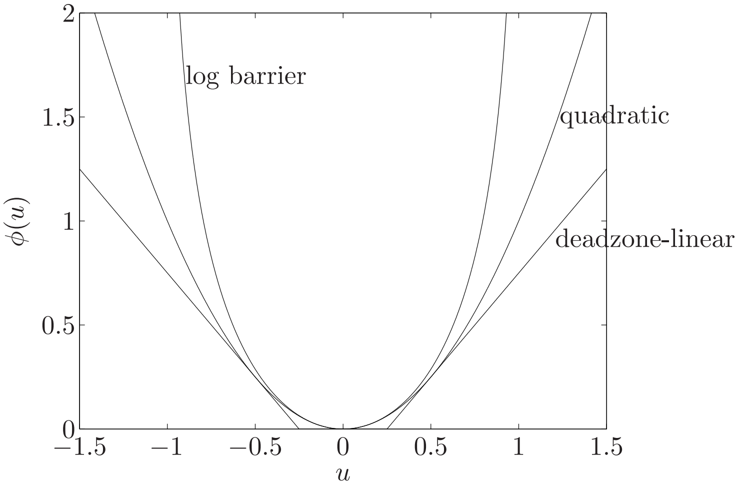

Figure 6.1 Some common penalty functions: the quadratic penalty function $\phi(u) = u^2$, the deadzone-linear penalty function with deadzone width $a = 1/4$, and the log barrier penalty function with limit $a = 1$.

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 ℓ_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 ℓ_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 ϕ 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 ϕ 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 ℓ_1-norm and ℓ_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 ℓ_1-norm approximation puts relatively larger emphasis on small residuals compared to ℓ_2-norm approximation. For large u we have $\phi_2(u) \gg \phi_1(u)$, so ℓ_1-norm approximation puts less weight on large residuals, compared to ℓ_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 ℓ_1-norm approximation problem will tend to have more zero and very small residuals, compared to the ℓ_2-norm approximation solution. In contrast, the ℓ_2-norm solution will tend to have relatively fewer large residuals (since large residuals incur a much larger penalty in ℓ_2-norm approximation than in ℓ_1-norm approximation).

Example

An example will illustrate these ideas. We take a matrix $A \in \mathbf{R}^{100\times 30}$ and vector $b \in \mathbf{R}^{100}$ (chosen at random, but the results are typical), and compute the ℓ_1-norm and ℓ_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 ℓ_1-norm penalty puts the most weight on small residuals and the least weight on large residuals.
- The ℓ_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 ℓ_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 ℓ_1-optimal solution, many residuals are either zero or very small. The ℓ_1-optimal solution also has relatively more large residuals.

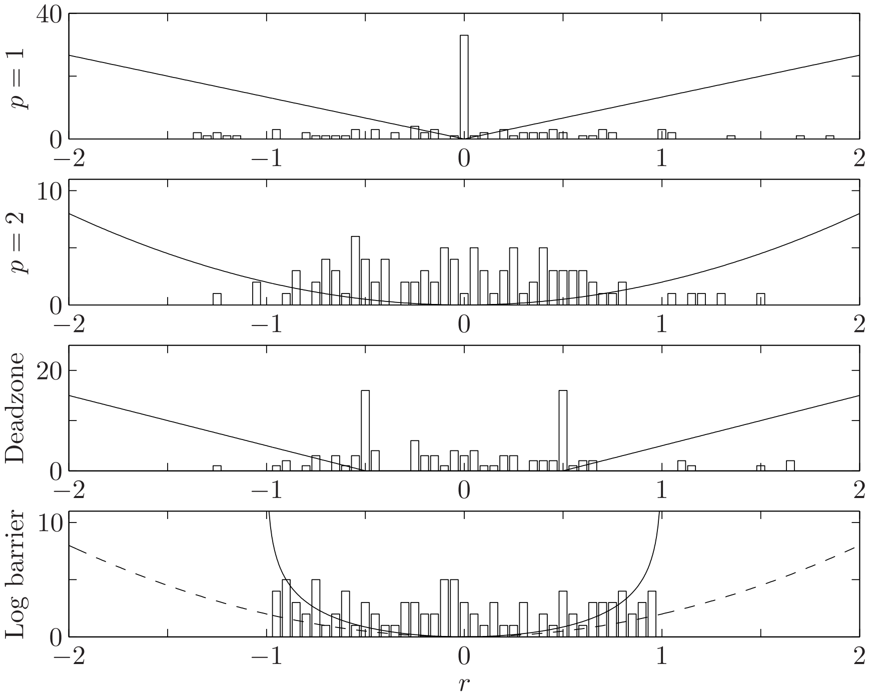

Figure 6.2 Histogram of residual amplitudes for four penalty functions, with the (scaled) penalty functions also shown for reference. For the log barrier plot, the quadratic penalty is also shown, in dashed curve.

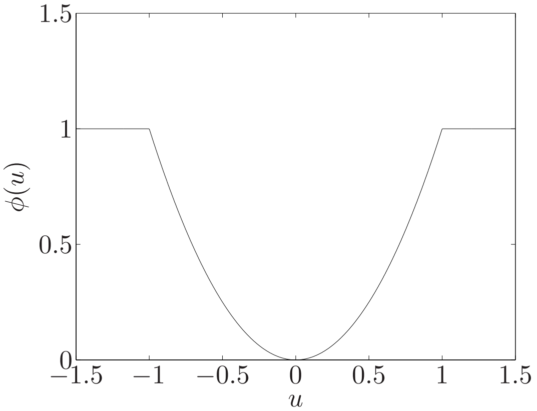

Figure 6.3 A (nonconvex) penalty function that assesses a fixed penalty to residuals larger than a threshold (which in this example is one): $\phi(u) = u^2$ if $|u| \leq 1$ and $\phi(u) = 1$ if $|u| > 1$. As a result, penalty approximation with this function would be relatively insensitive to outliers.

- The ℓ_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 ± 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 ℓ_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

$$\phi(u) = \begin{cases} u^2 & |u| \leq M \\ M^2 & |u| > M, \end{cases} \tag{6.3}$$

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

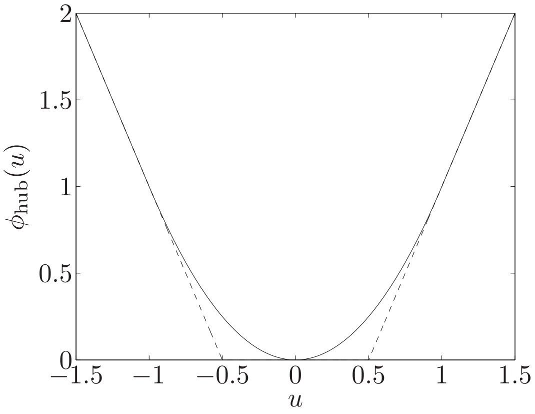

Figure 6.4 The solid line is the robust least-squares or Huber penalty function ϕ_{hub}, with $M = 1$. For $|u| \leq M$ it is quadratic, and for $|u| > M$ it grows linearly.

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 ℓ_1-norm approximation. Another example is the *robust least-squares* or *Huber penalty function*, given by

$$\phi_{\mathrm{hub}}(u) = \begin{cases} u^2 & |u| \leq M \\ M(2|u| - M) & |u| > M, \end{cases} \tag{6.4}$$

shown in figure 6.4. This penalty function agrees with the least-squares penalty function for residuals smaller than M, and then reverts to ℓ_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 α and β are obtained by solving the least-squares problem

$$\text{minimize} \quad \textstyle\sum_{i=1}^{42} (y_i - \alpha - \beta t_i)^2,$$

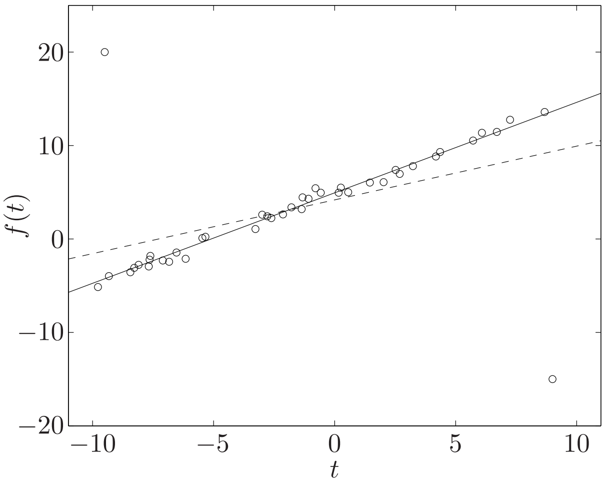

Figure 6.5 The 42 circles show points that can be well approximated by an affine function, except for the two outliers at upper left and lower right. The dashed line is the least-squares fit of a straight line $f(t) = \alpha + \beta t$ to the points, and is rotated away from the main locus of points, 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 α and β. 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

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

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

Since ℓ_1-norm approximation is among the (convex) penalty function approximation methods that are most robust to outliers, ℓ_1-norm approximation is sometimes called *robust estimation* or *robust regression*. The robustness property of ℓ_1-norm estimation can also be understood in a statistical framework; see page 353.

Small residuals and ℓ_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 ℓ_1-norm approximation, tend to produce optimal residuals many of which are very small, or even exactly zero. This means that in ℓ_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 unacceptable 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:

$$\begin{array}{ll} \text{minimize} & \|Ax - b\| \\ \text{subject to} & x \succeq 0. \end{array}$$

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 $l \preceq x \preceq u$, where $l,\ u \in \mathbf{R}^n$ are problem parameters:

$$\begin{array}{ll} \text{minimize} & \|Ax - b\| \\ \text{subject to} & l \preceq x \preceq u. \end{array}$$

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 $x \succeq 0$, $\mathbf{1}^T x = 1$:

$$\begin{array}{ll} \text{minimize} & \|Ax - b\| \\ \text{subject to} & x \succeq 0, \quad \mathbf{1}^T x = 1. \end{array}$$

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:

$$\begin{array}{ll} \text{minimize} & \|Ax - b\| \\ \text{subject to} & \|x - x_0\| \leq d, \end{array}$$

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

$$\begin{array}{ll} \text{minimize} & \|x\| \\ \text{subject to} & Ax = b \end{array} \qquad (6.5)$$

where the data are $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$, the variable is $x \in \mathbf{R}^n$, and $\|\cdot\|$ is a norm on $\mathbf{R}^n$. 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.

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 $Z \in \mathbf{R}^{n\times k}$ 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 $u \in \mathbf{R}^k$. The least-norm problem (6.5) can be expressed as

$$\text{minimize} \quad \|x_0 + Zu\|,$$

with variable $u \in \mathbf{R}^k$, 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 ℓ_2-norm. By squaring the objective we obtain the equivalent problem

$$\begin{array}{ll} \text{minimize} & \|x\|_2^2 \\ \text{subject to} & Ax = b, \end{array}$$

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 $\nu \in \mathbf{R}^m$, the optimality conditions are

$$2x^\star + A^T\nu^\star = 0, \qquad Ax^\star = b,$$

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

$$\nu^\star = -2(AA^T)^{-1}b, \qquad x^\star = A^T(AA^T)^{-1}b.$$

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

Least-penalty problems

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

$$\begin{array}{ll} \text{minimize} & \phi(x_1) + \cdots + \phi(x_n) \\ \text{subject to} & Ax = b, \end{array} \tag{6.6}$$

where $\phi : \mathbf{R} \to \mathbf{R}$ 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 ℓ_1-norm

Recall from the discussion on page 300 that ℓ_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 ℓ_1-norm problem,

$$\begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to} & Ax = b, \end{array}$$

tends to produce a solution x with a large number of components equal to zero. In other words, the least ℓ_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 $\tilde{x} \in \mathbf{R}^m$ 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 $b \notin \mathcal{R}(\tilde{A})$, 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 $b \in \mathcal{R}(\tilde{A})$, 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 ℓ_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 $\|x\|$:

$$\text{minimize (w.r.t. } \mathbf{R}^2_+) \quad (\|Ax - b\|, \|x\|) \,. \tag{6.7}$$

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

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.)

6.3.2 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:

$$\text{minimize} \quad \|Ax - b\| + \gamma\|x\|, \tag{6.8}$$

where $\gamma > 0$ is a problem parameter. As γ 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.*,

$$\text{minimize} \quad \|Ax - b\|^2 + \delta\|x\|^2, \tag{6.9}$$

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:

$$\text{minimize} \quad \|Ax - b\|_2^2 + \delta\|x\|_2^2 = x^T(A^TA + \delta I)x - 2b^TAx + b^Tb. \tag{6.10}$$

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.

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 $x \in \mathbf{R}^n$ 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

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

If Δ is the (tridiagonal, Toeplitz) matrix

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

then Δx 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 $\|Ax - b\|^2$, which might represent a measure of fit, or consistency with experimental data, and the objective $\|\Delta x\|^2$, which is (approximately) the mean-square curvature of the underlying physical parameter. The parameter δ 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 $y_{\rm des}$. We measure output tracking error by the quadratic function

$$J_{\rm track} = \frac{1}{N+1}\sum_{t=0}^{N}(y(t) - y_{\rm des}(t))^2.$$

- *Small input.* The input should not be large. We measure the magnitude of the input by the quadratic function

$$J_{\rm mag} = \frac{1}{N+1}\sum_{t=0}^{N} u(t)^2.$$

- *Small input variations.* The input should not vary rapidly. We measure the magnitude of the input variations by the quadratic function

$$J_{\rm der} = \frac{1}{N}\sum_{t=0}^{N-1}(u(t+1) - u(t))^2.$$

By minimizing a weighted sum

$$J_{\rm track} + \delta J_{\rm der} + \eta J_{\rm mag},$$

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 $y_{\rm des}$), for three values of the regularization parameters δ and η. 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 $J_{\rm track}$ 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.

ℓ_1-norm regularization

Regularization with an ℓ_1-norm can be used as a heuristic for finding a sparse solution. For example, consider the problem

$$\text{minimize} \quad \|Ax - b\|_2 + \gamma\|x\|_1, \qquad (6.11)$$

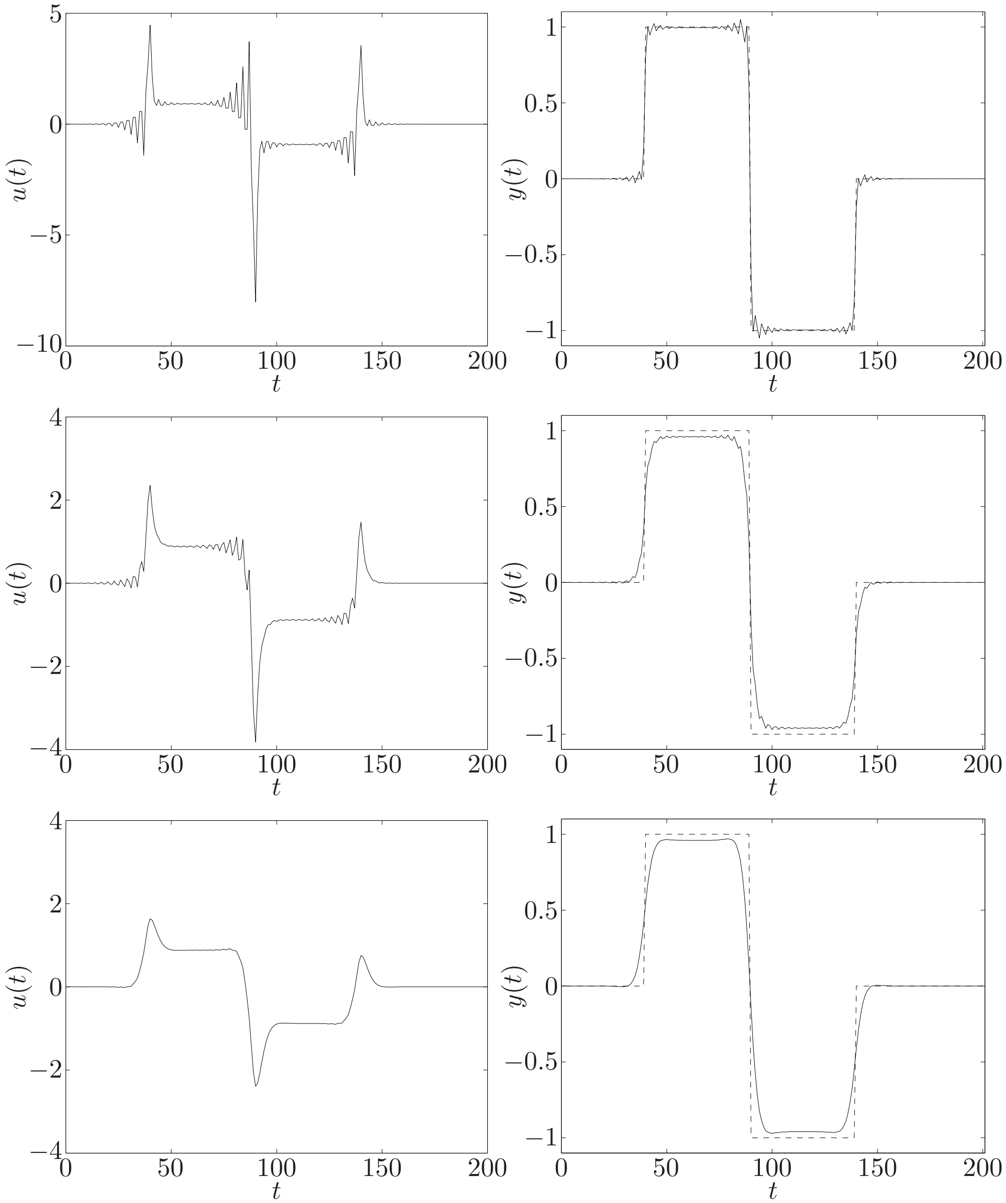

Figure 6.6 Optimal inputs (left) and resulting outputs (right) for three values of the regularization parameters δ (which corresponds to input variation) and η (which corresponds to input magnitude). The dashed line in the righthand plots shows the desired output $y_{\rm des}$. Top row: $\delta = 0$, $\eta = 0.005$; 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 ℓ_1-norm. By varying the parameter γ 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 $\mathbf{card}(x)$ 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 $A \in \mathbf{R}^{m \times n}$, whose columns are potential regressors, and a vector $b \in \mathbf{R}^m$ 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

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2 \\ \text{subject to} & \mathbf{card}(x) \leq k. \end{array}$$

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 γ, finding the smallest value of γ that results in a solution with $\mathbf{card}(x) = k$. 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 $A \in \mathbf{R}^{10 \times 20}$, $x \in \mathbf{R}^{20}$, $b \in \mathbf{R}^{10}$. The circles on the dashed curve are the (globally) Pareto optimal values for the trade-off between $\mathbf{card}(x)$ (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 γ. Note that for $\mathbf{card}(x) = 1$, 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 $x \in \mathbf{R}^n$. 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.)

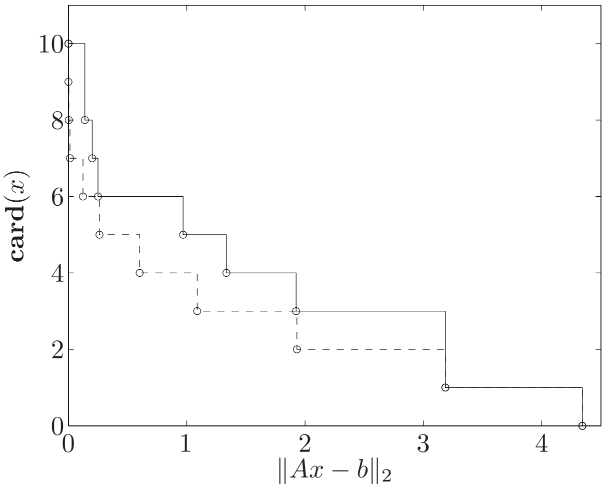

Figure 6.7 Sparse regressor selection with a matrix $A \in \mathbf{R}^{10\times 20}$. The circles on the dashed line are the Pareto optimal values for the trade-off between the residual $\|Ax-b\|_2$ and the number of nonzero elements $\mathbf{card}(x)$. The points indicated by circles on the solid line are obtained via the ℓ_1-norm regularized heuristic.

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_{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_{cor} to produce $\hat{x}$, so the process is also called *smoothing*.

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

$$\text{minimize (w.r.t. } \mathbf{R}^2_+) \quad (\|\hat{x} - x_{\text{cor}}\|_2, \phi(\hat{x})), \tag{6.12}$$

where $\hat{x}$ is the variable and x_{cor} is a problem parameter. The function $\phi : \mathbf{R}^n \to \mathbf{R}$ 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 ℓ_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

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

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

$$D = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix}.$$

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

$$\|\hat{x} - x_{\mathrm{cor}}\|_2^2 + \delta\|D\hat{x}\|_2^2,$$

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

$$\hat{x} = (I + \delta D^T D)^{-1} x_{\mathrm{cor}},$$

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

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

$$\phi_{\mathrm{tv}}(\hat{x}) = \sum_{i=1}^{n-1} |\hat{x}_{i+1} - \hat{x}_i| = \|D\hat{x}\|_1,$$

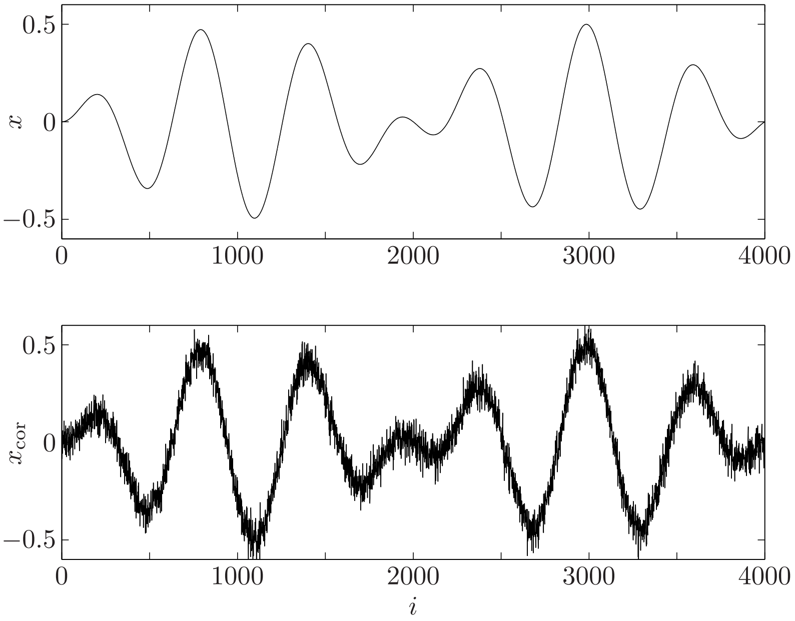

Figure 6.8 *Top:* the original signal $x \in \mathbf{R}^{4000}$. *Bottom:* the corrupted signal x_{cor}.

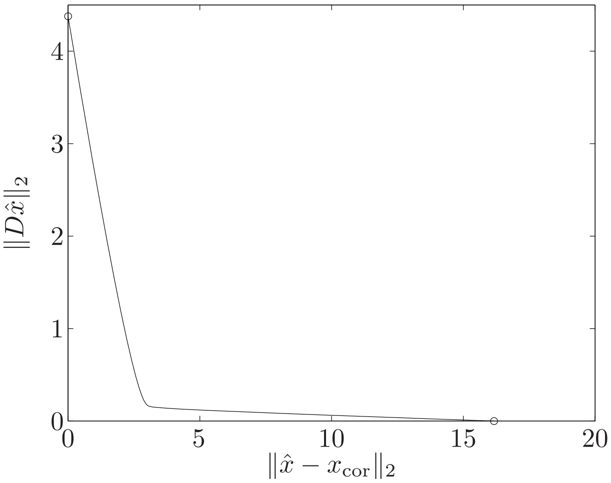

Figure 6.9 Optimal trade-off curve between $\|D\hat{x}\|_2$ and $\|\hat{x} - x_{\mathrm{cor}}\|_2$. The curve has a clear knee near $\|\hat{x} - x_{\mathrm{cor}}\| \approx 3$.

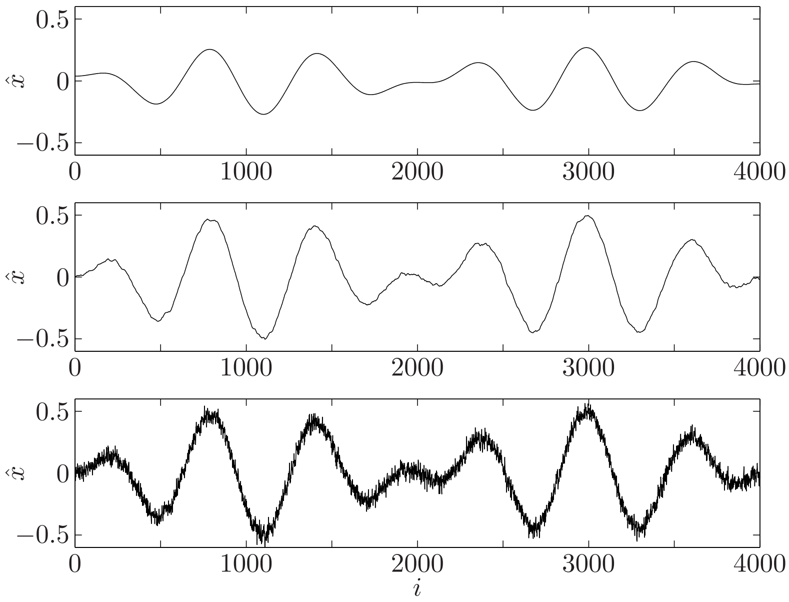

Figure 6.10 Three smoothed or reconstructed signals $\hat{x}$. The top one corresponds to $\|\hat{x} - x_{\rm cor}\|_2 = 8$, the middle one to $\|\hat{x} - x_{\rm cor}\|_2 = 3$, and the bottom one to $\|\hat{x} - x_{\rm cor}\|_2 = 1$.

which is called the *total variation* of $x \in \mathbf{R}^n$. Like the quadratic smoothness measure $\phi_{\rm quad}$, 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 $x \in \mathbf{R}^{2000}$ (in the top plot), and the signal corrupted with noise $x_{\rm cor}$. 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 $\|\hat{x}-x_{\rm cor}\|_2$. 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 $\|\hat{x} - x_{\rm cor}\|_2$. 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.

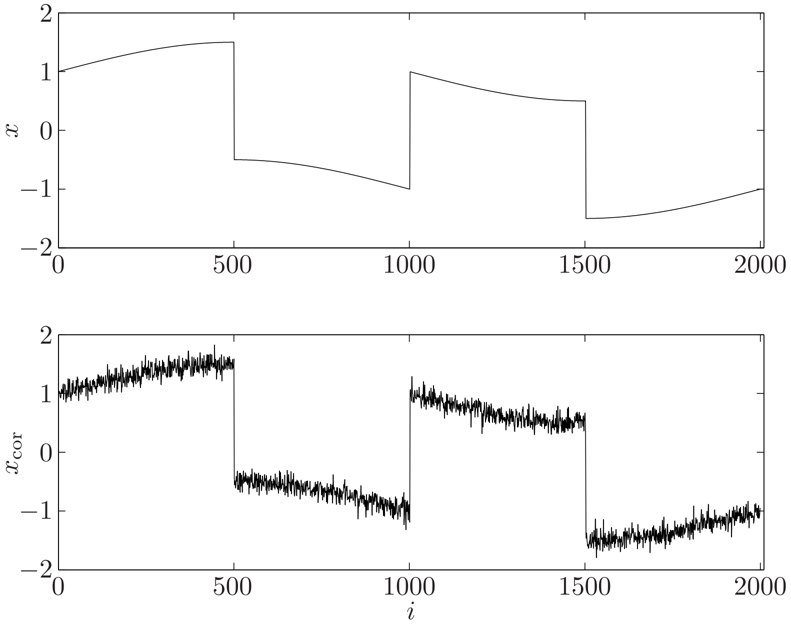

Figure 6.11 A signal $x \in \mathbf{R}^{2000}$, and the corrupted signal $x_{\text{cor}} \in \mathbf{R}^{2000}$. The noise is rapidly varying, and the signal is mostly smooth, with a few rapid variations.

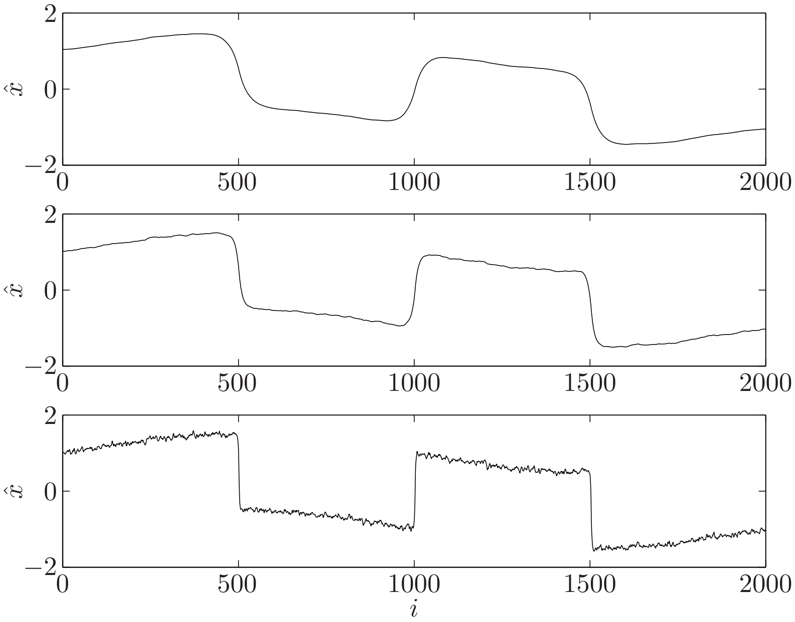

Figure 6.12 Three quadratically smoothed signals $\hat{x}$. The top one corresponds to $\|\hat{x} - x_{\text{cor}}\|_2 = 10$, the middle one to $\|\hat{x} - x_{\text{cor}}\|_2 = 7$, and the bottom one to $\|\hat{x} - x_{\text{cor}}\|_2 = 4$. The top one greatly reduces the noise, but also excessively smooths out the rapid variations in the signal. The bottom smoothed signal does not give enough noise reduction, and still smooths out the rapid variations in the original signal. The middle smoothed signal gives the best compromise, but still smooths out the rapid variations.

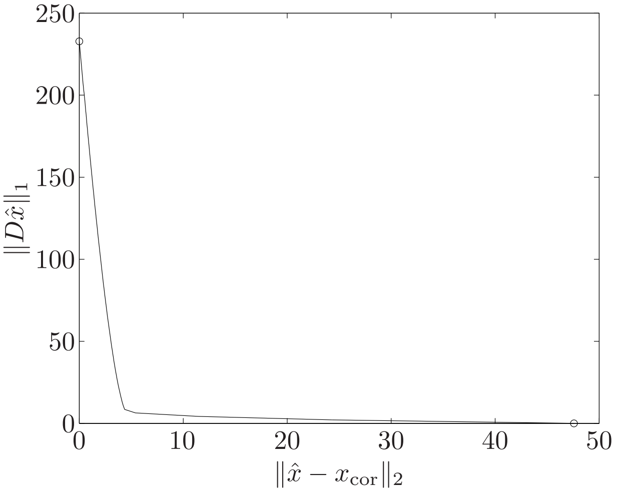

Figure 6.13 Optimal trade-off curve between $\|D\hat{x}\|_1$ and $\|\hat{x} - x_{\text{cor}}\|_2$.

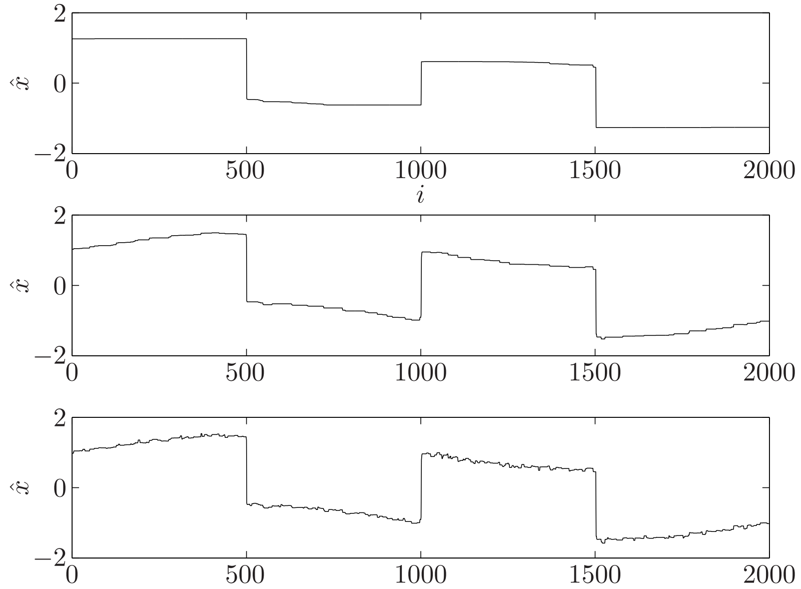

Figure 6.14 Three reconstructed signals $\hat{x}$, using total variation reconstruction. The top one corresponds to $\|D\hat{x}\|_1 = 5$, the middle one to $\|D\hat{x}\|_1 = 8$, and the bottom one to $\|D\hat{x}\|_1 = 10$. The bottom one does not give quite enough noise reduction, while the top one eliminates 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.

6.4 Robust approximation

6.4.1 Stochastic robust approximation

We consider an approximation problem with basic objective $\|Ax-b\|$, but also wish to take into account some uncertainty or possible variation in the data matrix A. (The same ideas can be extended to handle the case where there is uncertainty in both A and b.) In this section we consider some statistical models for the variation in A.

We assume that A is a random variable taking values in $\mathbf{R}^{m\times n}$, with mean $\bar{A}$, so we can describe A as

$$A = \bar{A} + U,$$

where U is a random matrix with zero mean. Here, the constant matrix $\bar{A}$ gives the average value of A, and U describes its statistical variation.

It is natural to use the expected value of $\|Ax-b\|$ as the objective:

$$\text{minimize} \quad \mathbf{E}\,\|Ax-b\|. \tag{6.13}$$

We refer to this problem as the *stochastic robust approximation problem.* It is always a convex optimization problem, but usually not tractable since in most cases it is very difficult to evaluate the objective or its derivatives.

One simple case in which the stochastic robust approximation problem (6.13) can be solved occurs when A assumes only a finite number of values, *i.e.*,

$$\mathbf{prob}(A = A_i) = p_i, \quad i = 1,\ldots,k,$$

where $A_i \in \mathbf{R}^{m\times n}$, $\mathbf{1}^T p = 1$, $p \succeq 0$. In this case the problem (6.13) has the form

$$\text{minimize} \quad p_1\|A_1x-b\| + \cdots + p_k\|A_kx-b\|,$$

which is often called a *sum-of-norms problem.* It can be expressed as

$$\begin{array}{ll} \text{minimize} & p^T t \\ \text{subject to} & \|A_ix-b\| \leq t_i, \quad i=1,\ldots,k, \end{array}$$

where the variables are $x\in\mathbf{R}^n$ and $t\in\mathbf{R}^k$. If the norm is the Euclidean norm, this sum-of-norms problem is an SOCP. If the norm is the ℓ_1- or ℓ_∞-norm, the sum-of-norms problem can be expressed as an LP; see exercise 6.8.

Some variations on the stochastic robust approximation problem (6.13) are tractable. As an example, consider the stochastic robust least-squares problem

$$\text{minimize} \quad \mathbf{E}\,\|Ax-b\|_2^2,$$

where the norm is the Euclidean norm. We can express the objective as

$$\begin{array}{rcl} \mathbf{E}\,\|Ax-b\|_2^2 & = & \mathbf{E}(\bar{A}x-b+Ux)^T(\bar{A}x-b+Ux) \\ & = & (\bar{A}x-b)^T(\bar{A}x-b) + \mathbf{E}\,x^TU^TUx \\ & = & \|\bar{A}x-b\|_2^2 + x^TPx, \end{array}$$

where $P = \mathbf{E}\, U^T U$. Therefore the stochastic robust approximation problem has the form of a regularized least-squares problem

$$\text{minimize} \quad \|\bar{A}x - b\|_2^2 + \|P^{1/2}x\|_2^2,$$

with solution

$$x = (\bar{A}^T\bar{A} + P)^{-1}\bar{A}^T b.$$

This makes perfect sense: when the matrix A is subject to variation, the vector Ax will have more variation the larger x is, and Jensen's inequality tells us that variation in Ax will increase the average value of $\|Ax - b\|_2$. So we need to balance making $\bar{A}x - b$ small with the desire for a small x (to keep the variation in Ax small), which is the essential idea of regularization.

This observation gives us another interpretation of the Tikhonov regularized least-squares problem (6.10), as a robust least-squares problem, taking into account possible variation in the matrix A. The solution of the Tikhonov regularized least-squares problem (6.10) minimizes $\mathbf{E}\,\|(A+U)x - b\|^2$, where U_{ij} are zero mean, uncorrelated random variables, with variance δ/m (and here, A is deterministic).

6.4.2 Worst-case robust approximation

It is also possible to model the variation in the matrix A using a set-based, worst-case approach. We describe the uncertainty by a set of possible values for A:

$$A \in \mathcal{A} \subseteq \mathbf{R}^{m\times n},$$

which we assume is nonempty and bounded. We define the associated *worst-case error* of a candidate approximate solution $x \in \mathbf{R}^n$ as

$$e_{\text{wc}}(x) = \sup\{\|Ax - b\| \mid A \in \mathcal{A}\},$$

which is always a convex function of x. The (worst-case) *robust approximation problem* is to minimize the worst-case error:

$$\text{minimize} \quad e_{\text{wc}}(x) = \sup\{\|Ax - b\| \mid A \in \mathcal{A}\}, \qquad (6.14)$$

where the variable is x, and the problem data are b and the set $\mathcal{A}$. When $\mathcal{A}$ is the singleton $\mathcal{A} = \{A\}$, the robust approximation problem (6.14) reduces to the basic norm approximation problem (6.1). The robust approximation problem is always a convex optimization problem, but its tractability depends on the norm used and the description of the uncertainty set $\mathcal{A}$.

Example 6.5 *Comparison of stochastic and worst-case robust approximation.* To illustrate the difference between the stochastic and worst-case formulations of the robust approximation problem, we consider the least-squares problem

$$\text{minimize} \quad \|A(u)x - b\|_2^2,$$

where $u \in \mathbf{R}$ is an uncertain parameter and $A(u) = A_0 + uA_1$. We consider a specific instance of the problem, with $A(u) \in \mathbf{R}^{20\times 10}$, $\|A_0\| = 10$, $\|A_1\| = 1$, and u

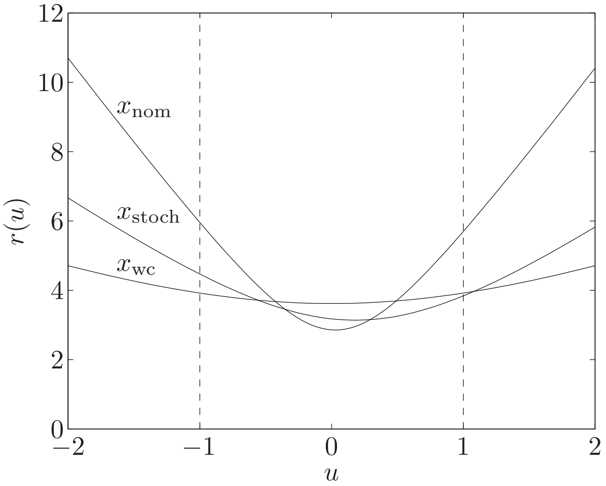

Figure 6.15 The residual $r(u) = \|A(u)x - b\|_2$ as a function of the uncertain parameter u for three approximate solutions x: (1) the nominal least-squares solution $x_{\rm nom}$; (2) the solution of the stochastic robust approximation problem $x_{\rm stoch}$ (assuming u is uniformly distributed on $[-1, 1]$); and (3) the solution of the worst-case robust approximation problem $x_{\rm wc}$, assuming the parameter u lies in the interval $[-1, 1]$. The nominal solution achieves the smallest residual when $u = 0$, but gives much larger residuals as u approaches -1 or 1. The worst-case solution has a larger residual when $u = 0$, but its residuals do not rise much as the parameter u varies over the interval $[-1, 1]$.

in the interval $[-1, 1]$. (So, roughly speaking, the variation in the matrix A is around ±10%.)

We find three approximate solutions:

- *Nominal optimal.* The optimal solution $x_{\rm nom}$ is found, assuming $A(u)$ has its nominal value A_0.
- *Stochastic robust approximation.* We find $x_{\rm stoch}$, which minimizes $\mathbf{E}\,\|A(u)x - b\|_2^2$, assuming the parameter u is uniformly distributed on $[-1, 1]$.
- *Worst-case robust approximation.* We find $x_{\rm wc}$, which minimizes

$$\sup_{-1 \leq u \leq 1} \|A(u)x - b\|_2 = \max\{\|(A_0 - A_1)x - b\|_2, \|(A_0 + A_1)x - b\|_2\}.$$

For each of these three values of x, we plot the residual $r(u) = \|A(u)x - b\|_2$ as a function of the uncertain parameter u, in figure 6.15. These plots show how sensitive an approximate solution can be to variation in the parameter u. The nominal solution achieves the smallest residual when $u = 0$, but is quite sensitive to parameter variation: it gives much larger residuals as u deviates from 0, and approaches -1 or 1. The worst-case solution has a larger residual when $u = 0$, but its residuals do not rise much as u varies over the interval $[-1, 1]$. The stochastic robust approximate solution is in between.

The robust approximation problem (6.14) arises in many contexts and applications. In an estimation setting, the set $\mathcal{A}$ gives our uncertainty in the linear relation between the vector to be estimated and our measurement vector. Sometimes the noise term v in the model $y = Ax + v$ is called *additive noise* or *additive error*, since it is added to the 'ideal' measurement Ax. In contrast, the variation in A is called *multiplicative error*, since it multiplies the variable x.

In an optimal design setting, the variation can represent uncertainty (arising in manufacture, say) of the linear equations that relate the design variables x to the results vector Ax. The robust approximation problem (6.14) is then interpreted as the robust design problem: find design variables x that minimize the worst possible mismatch between Ax and b, over all possible values of A.

Finite set

Here we have $\mathcal{A} = \{A_1, \ldots, A_k\}$, and the robust approximation problem is

$$\text{minimize} \quad \max_{i=1,\ldots,k} \|A_i x - b\|.$$

This problem is equivalent to the robust approximation problem with the polyhedral set $\mathcal{A} = \mathbf{conv}\{A_1, \ldots, A_k\}$:

$$\text{minimize} \quad \sup\left\{\|Ax - b\| \mid A \in \mathbf{conv}\{A_1, \ldots, A_k\}\right\}.$$

We can cast the problem in epigraph form as

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|A_i x - b\| \leq t, \quad i = 1, \ldots, k, \end{array}$$

which can be solved in a variety of ways, depending on the norm used. If the norm is the Euclidean norm, this is an SOCP. If the norm is the ℓ_1- or ℓ_∞-norm, we can express it as an LP.

Norm bound error

Here the uncertainty set $\mathcal{A}$ is a norm ball, $\mathcal{A} = \{\bar{A} + U \mid \|U\| \leq a\}$, where $\|\cdot\|$ is a norm on $\mathbf{R}^{m\times n}$. In this case we have

$$e_{\rm wc}(x) = \sup\{\|\bar{A}x - b + Ux\| \mid \|U\| \leq a\},$$

which must be carefully interpreted since the first norm appearing is on $\mathbf{R}^m$ (and is used to measure the size of the residual) and the second one appearing is on $\mathbf{R}^{m\times n}$ (used to define the norm ball $\mathcal{A}$).

This expression for $e_{\rm wc}(x)$ can be simplified in several cases. As an example, let us take the Euclidean norm on $\mathbf{R}^n$ and the associated induced norm on $\mathbf{R}^{m\times n}$, *i.e.*, the maximum singular value. If $\bar{A}x - b \neq 0$ and $x \neq 0$, the supremum in the expression for $e_{\rm wc}(x)$ is attained for $U = auv^T$, with

$$u = \frac{\bar{A}x - b}{\|\bar{A}x - b\|_2}, \qquad v = \frac{x}{\|x\|_2},$$

and the resulting worst-case error is

$$e_{\rm wc}(x) = \|\bar{A}x - b\|_2 + a\|x\|_2.$$

(It is easily verified that this expression is also valid if x or $\bar{A}x - b$ is zero.) The robust approximation problem (6.14) then becomes

$$\text{minimize} \quad \|\bar{A}x - b\|_2 + a\|x\|_2,$$

which is a regularized norm problem, solvable as the SOCP

$$\begin{array}{ll} \text{minimize} & t_1 + at_2 \\ \text{subject to} & \|\bar{A}x - b\|_2 \le t_1, \quad \|x\|_2 \le t_2. \end{array}$$

Since the solution of this problem is the same as the solution of the regularized least-squares problem

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

for some value of the regularization parameter δ, we have another interpretation of the regularized least-squares problem as a worst-case robust approximation problem.

Uncertainty ellipsoids

We can also describe the variation in A by giving an ellipsoid of possible values for each row:

$$\mathcal{A} = \{[a_1 \ \cdots \ a_m]^T \mid a_i \in \mathcal{E}_i, \ i = 1, \dots, m\},$$

where

$$\mathcal{E}_i = \{\bar{a}_i + P_i u \mid \|u\|_2 \le 1\}.$$

The matrix $P_i \in \mathbf{R}^{n \times n}$ describes the variation in a_i. We allow P_i to have a nontrivial nullspace, in order to model the situation when the variation in a_i is restricted to a subspace. As an extreme case, we take $P_i = 0$ if there is no uncertainty in a_i.

With this ellipsoidal uncertainty description, we can give an explicit expression for the worst-case magnitude of each residual:

$$\begin{array}{rcl} \sup\limits_{a_i \in \mathcal{E}_i} |a_i^T x - b_i| & = & \sup\{|\bar{a}_i^T x - b_i + (P_i u)^T x| \mid \|u\|_2 \le 1\} \\ & = & |\bar{a}_i^T x - b_i| + \|P_i^T x\|_2. \end{array}$$

Using this result we can solve several robust approximation problems. For example, the robust ℓ_2-norm approximation problem

$$\text{minimize} \quad e_{\text{wc}}(x) = \sup\{\|Ax - b\|_2 \mid a_i \in \mathcal{E}_i, \ i = 1, \dots, m\}$$

can be reduced to an SOCP, as follows. An explicit expression for the worst-case error is given by

$$e_{\text{wc}}(x) = \left(\sum_{i=1}^m \left(\sup_{a_i \in \mathcal{E}_i} |a_i^T x - b_i| \right)^2 \right)^{1/2} = \left(\sum_{i=1}^m (|\bar{a}_i^T x - b_i| + \|P_i^T x\|_2)^2 \right)^{1/2}.$$

To minimize $e_{\text{wc}}(x)$ we can solve

$$\begin{array}{ll} \text{minimize} & \|t\|_2 \\ \text{subject to} & |\bar{a}_i^T x - b_i| + \|P_i^T x\|_2 \le t_i, \quad i = 1, \dots, m, \end{array}$$

where we introduced new variables $t_1, \ldots, t_m$. This problem can be formulated as

$$\begin{array}{ll} \text{minimize} & \|t\|_2 \\ \text{subject to} & \bar{a}_i^T x - b_i + \|P_i^T x\|_2 \leq t_i, \quad i = 1, \ldots, m \\ & -\bar{a}_i^T x + b_i + \|P_i^T x\|_2 \leq t_i, \quad i = 1, \ldots, m, \end{array}$$

which becomes an SOCP when put in epigraph form.

Norm bounded error with linear structure

As a generalization of the norm bound description $\mathcal{A} = \{\bar{A} + U \mid \|U\| \leq a\}$, we can define $\mathcal{A}$ as the image of a norm ball under an affine transformation:

$$\mathcal{A} = \{\bar{A} + u_1 A_1 + u_2 A_2 + \cdots + u_p A_p \mid \|u\| \leq 1\},$$

where $\|\cdot\|$ is a norm on $\mathbf{R}^p$, and the $p+1$ matrices $\bar{A},\ A_1, \ldots, A_p \in \mathbf{R}^{m \times n}$ are given. The worst-case error can be expressed as

$$\begin{array}{rcl} e_{\text{wc}}(x) & = & \sup\limits_{\|u\| \leq 1} \|(\bar{A} + u_1 A_1 + \cdots + u_p A_p)x - b\| \\ & = & \sup\limits_{\|u\| \leq 1} \|P(x)u + q(x)\|, \end{array}$$

where P and q are defined as

$$P(x) = \left[\begin{array}{cccc} A_1 x & A_2 x & \cdots & A_p x \end{array} \right] \in \mathbf{R}^{m \times p}, \qquad q(x) = \bar{A}x - b \in \mathbf{R}^m.$$

As a first example, we consider the robust Chebyshev approximation problem

$$\text{minimize} \quad e_{\text{wc}}(x) = \sup\nolimits_{\|u\|_\infty \leq 1} \|(\bar{A} + u_1 A_1 + \cdots + u_p A_p)x - b\|_\infty.$$

In this case we can derive an explicit expression for the worst-case error. Let $p_i(x)^T$ denote the ith row of $P(x)$. We have

$$\begin{array}{rcl} e_{\text{wc}}(x) & = & \sup\limits_{\|u\|_\infty \leq 1} \|P(x)u + q(x)\|_\infty \\ & = & \max\limits_{i=1,\ldots,m} \sup\limits_{\|u\|_\infty \leq 1} |p_i(x)^T u + q_i(x)| \\ & = & \max\limits_{i=1,\ldots,m} (\|p_i(x)\|_1 + |q_i(x)|). \end{array}$$

The robust Chebyshev approximation problem can therefore be cast as an LP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & -y_0 \preceq \bar{A}x - b \preceq y_0 \\ & -y_k \preceq A_k x \preceq y_k, \quad k = 1, \ldots, p \\ & y_0 + \sum_{k=1}^p y_k \preceq t\mathbf{1}, \end{array}$$

with variables $x \in \mathbf{R}^n$, $y_k \in \mathbf{R}^m$, $t \in \mathbf{R}$.

As another example, we consider the robust least-squares problem

$$\text{minimize} \quad e_{\text{wc}}(x) = \sup\nolimits_{\|u\|_2 \leq 1} \|(\bar{A} + u_1 A_1 + \cdots + u_p A_p)x - b\|_2.$$

Here we use Lagrange duality to evaluate $e_{\rm wc}$. The worst-case error $e_{\rm wc}(x)$ is the squareroot of the optimal value of the (nonconvex) quadratic optimization problem

$$\begin{array}{ll} \mbox{maximize} & \|P(x)u + q(x)\|_2^2 \\ \mbox{subject to} & u^T u \leq 1, \end{array}$$

with u as variable. The Lagrange dual of this problem can be expressed as the SDP

$$\begin{array}{ll} \mbox{minimize} & t + \lambda \\ \mbox{subject to} & \left[\begin{array}{ccc} I & P(x) & q(x) \\ P(x)^T & \lambda I & 0 \\ q(x)^T & 0 & t \end{array}\right] \succeq 0 \end{array} \tag{6.15}$$

with variables t, $\lambda \in \mathbf{R}$. Moreover, as mentioned in §5.2 and §B.1 (and proved in §B.4), strong duality holds for this pair of primal and dual problems. In other words, for fixed x, we can compute $e_{\rm wc}(x)^2$ by solving the SDP (6.15) with variables t and λ. Optimizing jointly over t, λ, and x is equivalent to minimizing $e_{\rm wc}(x)^2$. We conclude that the robust least-squares problem is equivalent to the SDP (6.15) with x, λ, t as variables.

Example 6.6 *Comparison of worst-case robust, Tikhonov regularized, and nominal least-squares solutions.* We consider an instance of the robust approximation problem

$$\mbox{minimize} \quad \sup_{\|u\|_2 \leq 1} \|(\bar{A} + u_1 A_1 + u_2 A_2)x - b\|_2, \tag{6.16}$$

with dimensions $m = 50$, $n = 20$. The matrix $\bar{A}$ has norm 10, and the two matrices A_1 and A_2 have norm 1, so the variation in the matrix A is, roughly speaking, around 10%. The uncertainty parameters u_1 and u_2 lie in the unit disk in $\mathbf{R}^2$.

We compute the optimal solution of the robust least-squares problem (6.16) $x_{\rm rls}$, as well as the solution of the nominal least-squares problem $x_{\rm ls}$ (*i.e.*, assuming $u = 0$), and also the Tikhonov regularized solution $x_{\rm tik}$, with $\delta = 1$.

To illustrate the sensitivity of each of these approximate solutions to the parameter u, we generate 10^5 parameter vectors, uniformly distributed on the unit disk, and evaluate the residual

$$\|(A_0 + u_1 A_1 + u_2 A_2)x - b\|_2$$

for each parameter value. The distributions of the residuals are shown in figure 6.16.

We can make several observations. First, the residuals of the nominal least-squares solution are widely spread, from a smallest value around 0.52 to a largest value around 4.9. In particular, the least-squares solution is very sensitive to parameter variation. In contrast, both the robust least-squares and Tikhonov regularized solutions exhibit far smaller variation in residual as the uncertainty parameter varies over the unit disk. The robust least-squares solution, for example, achieves a residual between 2.0 and 2.6 for all parameters in the unit disk.

6.5 Function fitting and interpolation

In function fitting problems, we select a member of a finite-dimensional subspace of functions that best fits some given data or requirements. For simplicity we

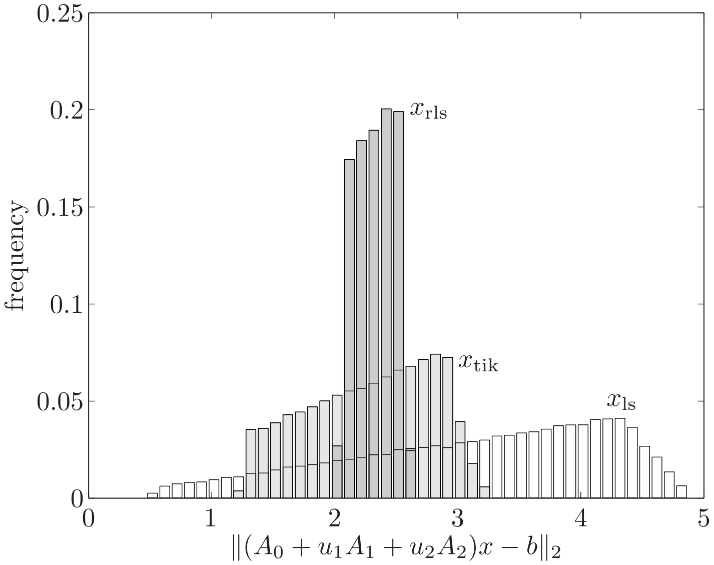

Figure 6.16 Distribution of the residuals for the three solutions of a least-squares problem (6.16): $x_{\rm ls}$, the least-squares solution assuming $u = 0$; $x_{\rm tik}$, the Tikhonov regularized solution with $\delta = 1$; and $x_{\rm rls}$, the robust least-squares solution. The histograms were obtained by generating 10^5 values of the uncertain parameter vector u from a uniform distribution on the unit disk in $\mathbf{R}^2$. The bins have width 0.1.

consider real-valued functions; the ideas are readily extended to handle vector-valued functions as well.

6.5.1 Function families

We consider a family of functions $f_1, \ldots, f_n : \mathbf{R}^k \to \mathbf{R}$, with common domain $\mathbf{dom}\, f_i = D$. With each $x \in \mathbf{R}^n$ we associate the function $f : \mathbf{R}^k \to \mathbf{R}$ given by

$$f(u) = x_1 f_1(u) + \cdots + x_n f_n(u) \tag{6.17}$$

with $\mathbf{dom}\, f = D$. The family $\{f_1, \ldots, f_n\}$ is sometimes called the set of *basis functions* (for the fitting problem) even when the functions are not independent. The vector $x \in \mathbf{R}^n$, which parametrizes the subspace of functions, is our optimization variable, and is sometimes called the *coefficient vector*. The basis functions generate a subspace $\mathcal{F}$ of functions on D.

In many applications the basis functions are specially chosen, using prior knowledge or experience, in order to reasonably model functions of interest with the finite-dimensional subspace of functions. In other cases, more generic function families are used. We describe a few of these below.

Polynomials

One common subspace of functions on $\mathbf{R}$ consists of polynomials of degree less than n. The simplest basis consists of the powers, *i.e.*, $f_i(t) = t^{i-1}$, $i = 1, \ldots, n$. In many applications, the same subspace is described using a different basis, for example, a set of polynomials $f_1, \ldots, f_n$, of degree less than n, that are orthonormal with respect to some positive function (or measure) $\phi : \mathbf{R}^n \to \mathbf{R}_+$, *i.e.*,

$$\int f_i(t) f_j(t) \phi(t)\, dt = \begin{cases} 1 & i = j \\ 0 & i \neq j. \end{cases}$$

Another common basis for polynomials is the *Lagrange basis* $f_1, \ldots, f_n$ associated with distinct points $t_1, \ldots, t_n$, which satisfy

$$f_i(t_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j. \end{cases}$$

We can also consider polynomials on $\mathbf{R}^k$, with a maximum total degree, or a maximum degree for each variable.

As a related example, we have *trigonometric polynomials* of degree less than n, with basis

$$\sin kt, \quad k = 1, \ldots, n-1, \qquad \cos kt, \quad k = 0, \ldots, n-1.$$

Piecewise-linear functions

We start with a *triangularization* of the domain D, which means the following. We have a set of *mesh* or *grid points* $g_1, \ldots, g_n \in \mathbf{R}^k$, and a partition of D into a set of simplexes:

$$D = S_1 \cup \cdots \cup S_m, \qquad \mathbf{int}(S_i \cap S_j) = \emptyset \text{ for } i \neq j.$$

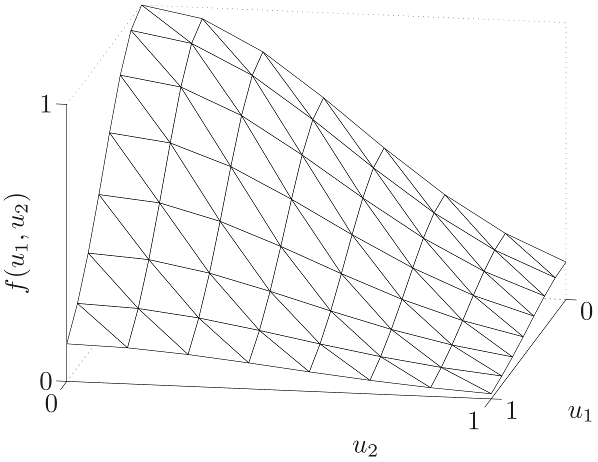

Figure 6.17 A piecewise-linear function of two variables, on the unit square. The triangulation consists of 98 simplexes, and a uniform grid of 64 points in the unit square.

Each simplex is the convex hull of $k+1$ grid points, and we require that each grid point is a vertex of any simplex it lies in.

Given a triangularization, we can construct a piecewise-linear (or more precisely, piecewise-affine) function f by assigning function values $f(g_i) = x_i$ to the grid points, and then extending the function affinely on each simplex. The function f can be expressed as (6.17) where the basis functions f_i are affine on each simplex and are defined by the conditions

$$f_i(g_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j. \end{cases}$$

By construction, such a function is continuous.

Figure 6.17 shows an example for $k = 2$.

Piecewise polynomials and splines

The idea of piecewise-affine functions on a triangulated domain is readily extended to piecewise polynomials and other functions.

Piecewise polynomials are defined as polynomials (of some maximum degree) on each simplex of the triangulation, which are continuous, *i.e.*, the polynomials agree at the boundaries between simplexes. By further restricting the piecewise polynomials to have continuous derivatives up to a certain order, we can define various classes of *spline functions*. Figure 6.18 shows an example of a cubic spline, *i.e.*, a piecewise polynomial of degree 3 on $\mathbf{R}$, with continuous first and second derivatives.

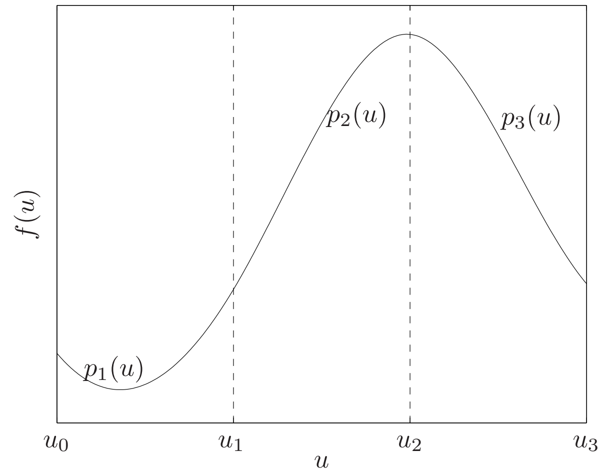

Figure 6.18 *Cubic spline.* A cubic spline is a piecewise polynomial, with continuous first and second derivatives. In this example, the cubic spline f is formed from the three cubic polynomials p_1 (on $[u_0, u_1]$), p_2 (on $[u_1, u_2]$), and p_3 (on $[u_2, u_3]$). Adjacent polynomials have the same function value, and equal first and second derivatives, at the boundary points u_1 and u_2. In this example, the dimension of the family of functions is $n = 6$, since we have 12 polynomial coefficients (4 per cubic polynomial), and 6 equality constraints (3 each at u_1 and u_2).

6.5.2 Constraints

In this section we describe some constraints that can be imposed on the function f, and therefore, on the variable $x \in \mathbf{R}^n$.

Function value interpolation and inequalities

Let v be a point in D. The value of f at v,

$$f(v) = \sum_{i=1}^{n} x_i f_i(v),$$

is a linear function of x. Therefore *interpolation conditions*

$$f(v_j) = z_j, \quad j = 1, \ldots, m,$$

which require the function f to have the values $z_j \in \mathbf{R}$ at specified points $v_j \in D$, form a set of linear equalities in x. More generally, inequalities on the function value at a given point, as in $l \leq f(v) \leq u$, are linear inequalities on the variable x. There are many other interesting convex constraints on f (hence, x) that involve the function values at a finite set of points $v_1, \ldots, v_N$. For example, the Lipschitz constraint

$$|f(v_j) - f(v_k)| \leq L \|v_j - v_k\|, \quad j,\ k = 1, \ldots, m,$$

forms a set of linear inequalities in x.

We can also impose inequalities on the function values at an infinite number of points. As an example, consider the nonnegativity constraint

$$f(u) \geq 0 \text{ for all } u \in D.$$

This is a convex constraint on x (since it is the intersection of an infinite number of halfspaces), but may not lead to a tractable problem except in special cases that exploit the particular structure of the functions. One simple example occurs when the functions are piecewise-linear. In this case, if the function values are nonnegative at the grid points, the function is nonnegative everywhere, so we obtain a simple (finite) set of linear inequalities.

As a less trivial example, consider the case when the functions are polynomials on $\mathbf{R}$, with even maximum degree $2k$ (*i.e.*, $n = 2k + 1$), and $D = \mathbf{R}$. As shown in exercise 2.37, page 65, the nonnegativity constraint

$$p(u) = x_1 + x_2 u + \cdots + x_{2k+1} u^{2k} \geq 0 \quad \text{for all } u \in \mathbf{R},$$

is equivalent to

$$x_i = \sum_{m+n=i+1} Y_{mn}, \quad i = 1, \ldots, 2k + 1, \qquad Y \succeq 0,$$

where $Y \in \mathbf{S}^{k+1}$ is an auxiliary variable.

Derivative constraints

Suppose the basis functions f_i are differentiable at a point $v \in D$. The gradient

$$\nabla f(v) = \sum_{i=1}^{n} x_i \nabla f_i(v),$$

is a linear function of x, so interpolation conditions on the derivative of f at v reduce to linear equality constraints on x. Requiring that the norm of the gradient at v not exceed a given limit,

$$\|\nabla f(v)\| = \left\| \sum_{i=1}^{n} x_i \nabla f_i(v) \right\| \leq M,$$

is a convex constraint on x. The same idea extends to higher derivatives. For example, if f is twice differentiable at v, the requirement that

$$lI \preceq \nabla^2 f(v) \preceq uI$$

is a linear matrix inequality in x, hence convex.

We can also impose constraints on the derivatives at an infinite number of points. For example, we can require that f is monotone:

$$f(u) \geq f(v) \text{ for all } u,\ v \in D,\ u \succeq v.$$

This is a convex constraint in x, but may not lead to a tractable problem except in special cases. When f is piecewise affine, for example, the monotonicity constraint is equivalent to the condition $\nabla f(v) \succeq 0$ inside each of the simplexes. Since the gradient is a linear function of the grid point values, this leads to a simple (finite) set of linear inequalities.

As another example, we can require that the function be convex, *i.e.*, satisfy

$$f((u+v)/2) \leq (f(u) + f(v))/2 \text{ for all } u,\ v \in D$$

(which is enough to ensure convexity when f is continuous). This is a convex constraint, which has a tractable representation in some cases. One obvious example is when f is quadratic, in which case the convexity constraint reduces to the requirement that the quadratic part of f be nonnegative, which is an LMI. Another example in which a convexity constraint leads to a tractable problem is described in more detail in §6.5.5.

Integral constraints

Any linear functional $\mathcal{L}$ on the subspace of functions can be expressed as a linear function of x, *i.e.*, we have $\mathcal{L}(f) = c^T x$. Evaluation of f (or a derivative) at a point is just a special case. As another example, the linear functional

$$\mathcal{L}(f) = \int_D \phi(u) f(u)\, du,$$

where $\phi : \mathbf{R}^k \to \mathbf{R}$, can be expressed as $\mathcal{L}(f) = c^T x$, where

$$c_i = \int_D \phi(u) f_i(u)\, du.$$

Thus, a constraint of the form $\mathcal{L}(f) = a$ is a linear equality constraint on x. One example of such a constraint is the *moment constraint*

$$\int_D t^m f(t)\, dt = a$$

(where $f : \mathbf{R} \to \mathbf{R}$).

6.5.3 Fitting and interpolation problems

Minimum norm function fitting

In a fitting problem, we are given data

$$(u_1, y_1), \quad \ldots, \quad (u_m, y_m)$$

with $u_i \in D$ and $y_i \in \mathbf{R}$, and seek a function $f \in \mathcal{F}$ that matches this data as closely as possible. For example in least-squares fitting we consider the problem

$$\text{minimize} \quad \textstyle\sum_{i=1}^m (f(u_i) - y_i)^2,$$

which is a simple least-squares problem in the variable x. We can add a variety of constraints, for example linear inequalities that must be satisfied by f at various points, constraints on the derivatives of f, monotonicity constraints, or moment constraints.

Example 6.7 *Polynomial fitting.* We are given data $u_1, \ldots, u_m \in \mathbf{R}$ and $v_1, \ldots, v_m \in \mathbf{R}$, and hope to approximately fit a polynomial of the form

$$p(u) = x_1 + x_2 u + \cdots + x_n u^{n-1}$$

to the data. For each x we form the vector of errors,

$$e = (p(u_1) - v_1, \ldots, p(u_m) - v_m).$$

To find the polynomial that minimizes the norm of the error, we solve the norm approximation problem

$$\text{minimize} \quad \|e\| = \|Ax - v\|$$

with variable $x \in \mathbf{R}^n$, where $A_{ij} = u_i^{j-1}$, $i = 1, \ldots, m$, $j = 1, \ldots, n$.

Figure 6.19 shows an example with $m = 40$ data points and $n = 6$ (*i.e.*, polynomials of maximum degree 5), for the ℓ_2- and ℓ_∞-norms.

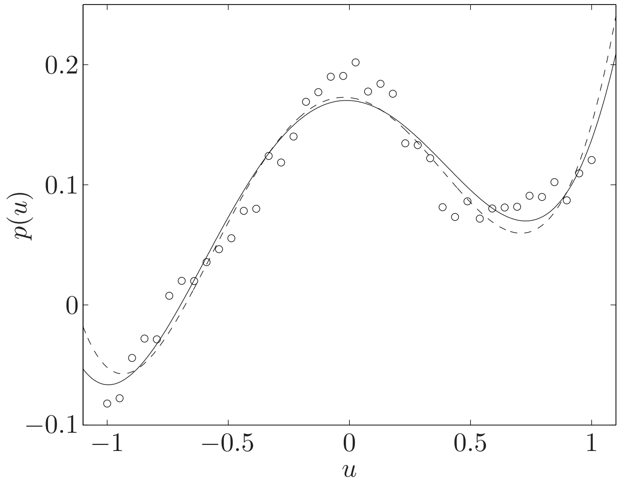

Figure 6.19 Two polynomials of degree 5 that approximate the 40 data points shown as circles. The polynomial shown as a solid line minimizes the ℓ_2-norm of the error; the polynomial shown as a dashed line minimizes the ℓ_∞-norm.

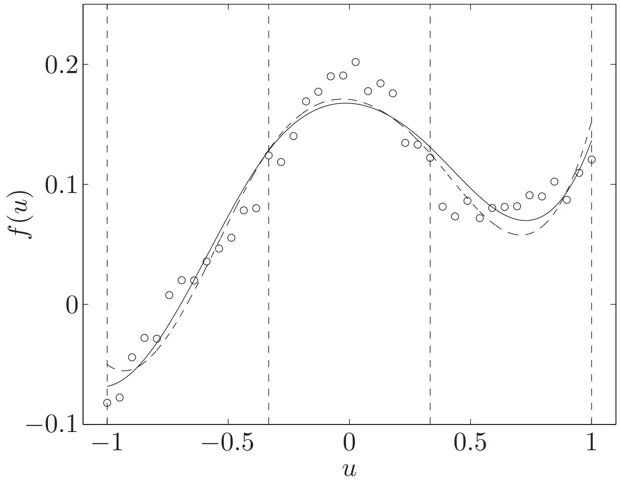

Figure 6.20 Two cubic splines that approximate the 40 data points shown as circles (which are the same as the data in figure 6.19). The spline shown as a solid line minimizes the ℓ_2-norm of the error; the spline shown as a dashed line minimizes the ℓ_∞-norm. As in the polynomial approximation shown in figure 6.19, the dimension of the subspace of fitting functions is 6.

Example 6.8 *Spline fitting.* Figure 6.20 shows the same data as in example 6.7, and two optimal fits with cubic splines. The interval $[-1, 1]$ is divided into three equal intervals, and we consider piecewise polynomials, with maximum degree 3, with continuous first and second derivatives. The dimension of this subspace of functions is 6, the same as the dimension of polynomials with maximum degree 5, considered in example 6.7.

In the simplest forms of function fitting, we have $m \gg n$, *i.e.*, the number of data points is much larger than the dimension of the subspace of functions. Smoothing is accomplished automatically, since all members of the subspace are smooth.

Least-norm interpolation

In another variation of function fitting, we have fewer data points than the dimension of the subspace of functions. In the simplest case, we require that the function we choose must satisfy the interpolation conditions

$$f(u_i) = y_i, \quad i = 1, \ldots, m,$$

which are linear equality constraints on x. Among the functions that satisfy these interpolation conditions, we might seek one that is smoothest, or smallest. These lead to least-norm problems.

In the most general function fitting problem, we can optimize an objective (such as some measure of the error e), subject to a variety of convex constraints that represent our prior knowledge of the underlying function.

Interpolation, extrapolation, and bounding

By evaluating the optimal function fit $\hat{f}$ at a point v not in the original data set, we obtain a guess of what the value of the underlying function is, at the point v. This is called *interpolation* when v is between or near the given data points (*e.g.*, $v \in \mathbf{conv}\{v_1, \ldots, v_m\}$), and *extrapolation* otherwise.

We can also produce an interval in which the value $f(v)$ can lie, by maximizing and minimizing (the linear function) $f(v)$, subject to the constraints. We can use the function fit to help identify faulty data or outliers. Here we might use, for example, an ℓ_1-norm fit, and look for data points with large errors.

6.5.4 Sparse descriptions and basis pursuit

In *basis pursuit*, there is a very large number of basis functions, and the goal is to find a good fit of the given data as a linear combination of a small number of the basis functions. (In this context the function family is linearly dependent, and is sometimes referred to as an *over-complete basis* or *dictionary.*) This is called basis pursuit since we are selecting a much smaller basis, from the given over-complete basis, to model the data.

Thus we seek a function $f \in \mathcal{F}$ that fits the data well,

$$f(u_i) \approx y_i, \quad i = 1, \ldots, m,$$

with a sparse coefficient vector x, *i.e.*, $\mathbf{card}(x)$ small. In this case we refer to

$$f = x_1 f_1 + \cdots + x_n f_n = \sum_{i \in \mathcal{B}} x_i f_i,$$

where $\mathcal{B} = \{i \mid x_i \neq 0\}$ is the set of indices of the chosen basis elements, as a *sparse description* of the data. Mathematically, basis pursuit is the same as the regressor selection problem (see §6.4), but the interpretation (and scale) of the optimization problem are different.

Sparse descriptions and basis pursuit have many uses. They can be used for de-noising or smoothing, or data compression for efficient transmission or storage of a signal. In data compression, the sender and receiver both know the dictionary, or basis elements. To send a signal to the receiver, the sender first finds a sparse representation of the signal, and then sends to the receiver only the nonzero coefficients (to some precision). Using these coefficients, the receiver can reconstruct (an approximation of) the original signal.

One common approach to basis pursuit is the same as the method for regressor selection described in §6.4, and based on ℓ_1-norm regularization as a heuristic for finding sparse descriptions. We first solve the convex problem

$$\text{minimize} \quad \textstyle\sum_{i=1}^m (f(u_i) - y_i)^2 + \gamma \|x\|_1, \tag{6.18}$$

where $\gamma > 0$ is a parameter used to trade off the quality of the fit to the data, and the sparsity of the coefficient vector. The solution of this problem can be used directly, or followed by a refinement step, in which the best fit is found, using the sparsity pattern of the solution of (6.18). In other words, we first solve (6.18), to obtain $\hat{x}$. We then set $\mathcal{B} = \{i \mid \hat{x}_i \neq 0\}$, *i.e.*, the set of indices corresponding to nonzero coefficients. Then we solve the least-squares problem

$$\text{minimize} \quad \textstyle\sum_{i=1}^m (f(u_i) - y_i)^2$$

with variables x_i, $i \in \mathcal{B}$, and $x_i = 0$ for $i \notin \mathcal{B}$.

In basis pursuit and sparse description applications it is not uncommon to have a very large dictionary, with n on the order of 10^4 or much more. To be effective, algorithms for solving (6.18) must exploit problem structure, which derives from the structure of the dictionary signals.

Time-frequency analysis via basis pursuit

In this section we illustrate basis pursuit and sparse representation with a simple example. We consider functions (or signals) on $\mathbf{R}$, with the range of interest $[0, 1]$. We think of the independent variable as time, so we use t (instead of u) to denote it.

We first describe the basis functions in the dictionary. Each basis function is a *Gaussian sinusoidal pulse*, or *Gabor function*, with form

$$e^{-(t-\tau)^2/\sigma^2} \cos(\omega t + \phi),$$

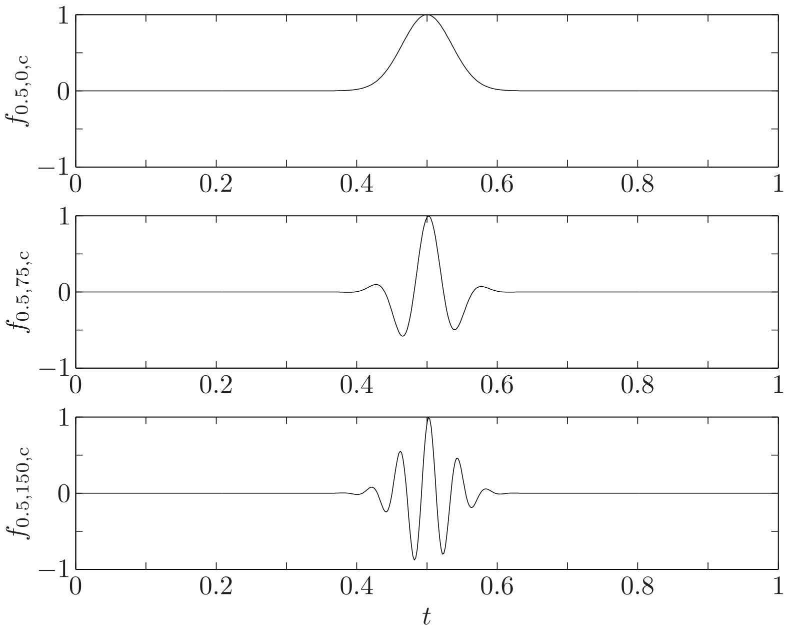

Figure 6.21 Three of the basis elements in the dictionary, all with center time $\tau = 0.5$ and cosine phase. The top signal has frequency $\omega = 0$, the middle one has frequency $\omega = 75$, and the bottom one has frequency $\omega = 150$.

where $\sigma > 0$ gives the width of the pulse, τ is the time of (the center of) the pulse, $\omega \geq 0$ is the frequency, and ϕ is the phase angle. All of the basis functions have width $\sigma = 0.05$. The pulse times and frequencies are

$$\tau = 0.002k, \quad k = 0, \ldots, 500, \qquad \omega = 5k, \quad k = 0, \ldots, 30.$$

For each time τ, there is one basis element with frequency zero (and phase $\phi = 0$), and 2 basis elements (cosine and sine, *i.e.*, phase $\phi = 0$ and $\phi = \pi/2$) for each of 30 remaining frequencies, so all together there are $501 \times 61 = 30561$ basis elements. The basis elements are naturally indexed by time, frequency, and phase (cosine or sine), so we denote them as

$$\begin{array}{lll} f_{\tau,\omega,\mathrm{c}}, & \tau = 0, 0.002, \ldots, 1, & \omega = 0, 5, \ldots, 150, \\ f_{\tau,\omega,\mathrm{s}}, & \tau = 0, 0.002, \ldots, 1, & \omega = 5, \ldots, 150. \end{array}$$

Three of these basis functions (all with time $\tau = 0.5$) are shown in figure 6.21.

Basis pursuit with this dictionary can be thought of as a *time-frequency analysis* of the data. If a basis element $f_{\tau,\omega,\mathrm{c}}$ or $f_{\tau,\omega,\mathrm{s}}$ appears in the sparse representation of a signal (*i.e.*, with a nonzero coefficient), we can interpret this as meaning that the data contains the frequency ω at time τ.

We will use basis pursuit to find a sparse approximation of the signal

$$y(t) = a(t) \sin \theta(t)$$

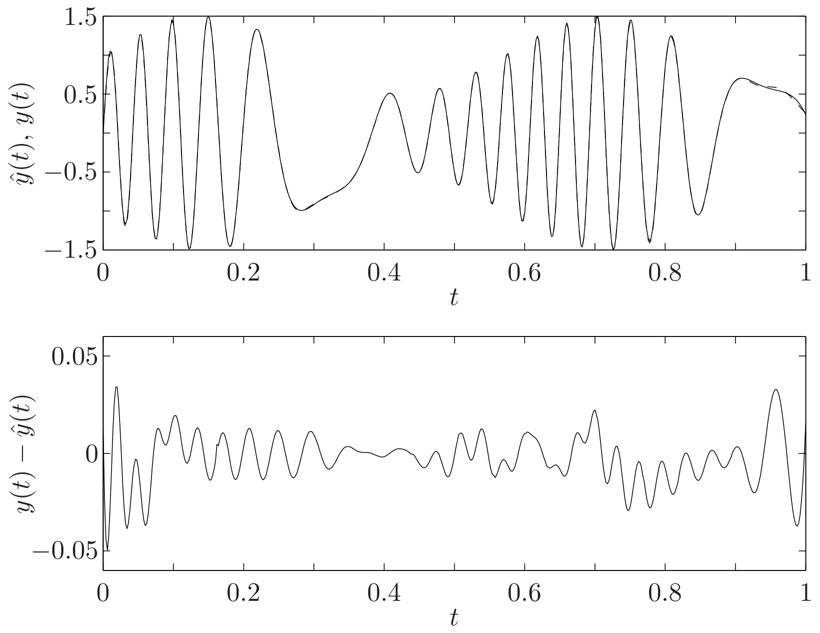

Figure 6.22 *Top.* The original signal (solid line) and approximation $\hat{y}$ obtained by basis pursuit (dashed line) are almost indistinguishable. *Bottom.* The approximation error $y(t) - \hat{y}(t)$, with different vertical scale.

where

$$a(t) = 1 + 0.5\sin(11t), \qquad \theta(t) = 30\sin(5t).$$

(This signal is chosen only because it is simple to describe, and exhibits noticeable changes in its spectral content over time.) We can interpret $a(t)$ as the signal amplitude, and $\theta(t)$ as its total phase. We can also interpret

$$\omega(t) = \left|\frac{d\theta}{dt}\right| = 150|\cos(5t)|$$

as the *instantaneous frequency* of the signal at time t. The data are given as 501 uniformly spaced samples over the interval $[0, 1]$, *i.e.*, we are given 501 pairs (t_k, y_k) with

$$t_k = 0.005k, \quad y_k = y(t_k), \quad k = 0, \ldots, 500.$$

We first solve the ℓ_1-norm regularized least-squares problem (6.18), with $\gamma = 1$. The resulting optimal coefficient vector is very sparse, with only 42 nonzero coefficients out of 30561. We then find the least-squares fit of the original signal using these 42 basis vectors. The result $\hat{y}$ is compared with the original signal y in figure 6.22. The top figure shows the approximated signal (in dashed line) and, almost indistinguishable, the original signal $y(t)$ (in solid line). The bottom figure shows the error $y(t) - \hat{y}(t)$. As is clear from the figure, we have obtained an

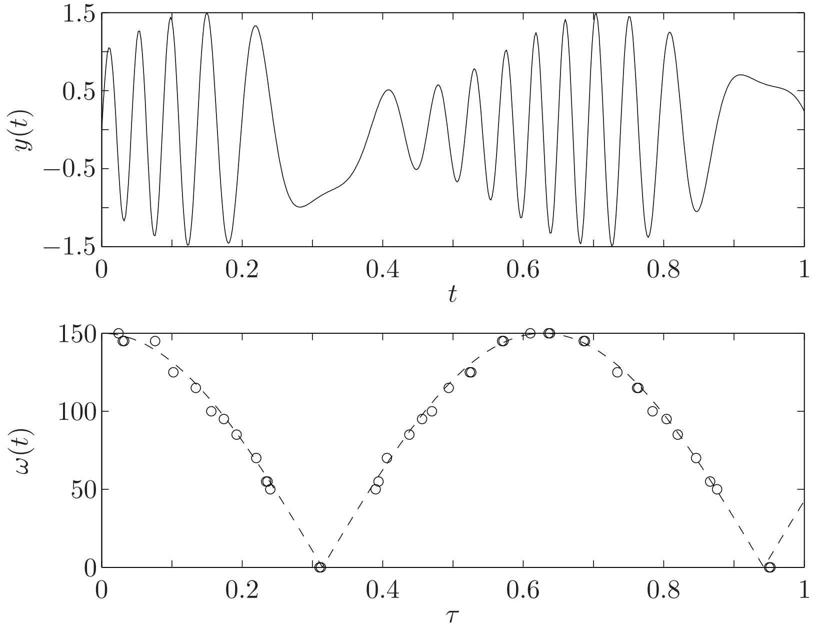

Figure 6.23 *Top: Original signal. Bottom: Time-frequency plot.* The dashed curve shows the instantaneous frequency $\omega(t) = 150|\cos(5t)|$ of the original signal. Each circle corresponds to a chosen basis element in the approximation obtained by basis pursuit. The horizontal axis shows the time index τ, and the vertical axis shows the frequency index ω of the basis element.

approximation $\hat{y}$ with a very good relative fit. The relative error is

$$\frac{(1/501)\sum_{i=1}^{501}(y(t_i) - \hat{y}(t_i))^2}{(1/501)\sum_{i=1}^{501} y(t_i)^2} = 2.6 \cdot 10^{-4}.$$

By plotting the pattern of nonzero coefficients versus time and frequency, we obtain a time-frequency analysis of the original data. Such a plot is shown in figure 6.23, along with the instantaneous frequency. The plot shows that the nonzero components closely track the instantaneous frequency.

6.5.5 Interpolation with convex functions

In some special cases we can solve interpolation problems involving an infinite-dimensional set of functions, using finite-dimensional convex optimization. In this section we describe an example.

We start with the following question: When does there exist a convex function $f : \mathbf{R}^k \to \mathbf{R}$, with $\mathbf{dom}\, f = \mathbf{R}^k$, that satisfies the interpolation conditions

$$f(u_i) = y_i, \quad i = 1, \ldots, m,$$

at given points $u_i \in \mathbf{R}^k$? (Here we do not restrict f to lie in any finite-dimensional subspace of functions.) The answer is: if and only if there exist $g_1, \ldots, g_m$ such that

$$y_j \geq y_i + g_i^T(u_j - u_i), \quad i, \ j = 1, \ldots, m. \tag{6.19}$$

To see this, first suppose that f is convex, $\mathbf{dom}\, f = \mathbf{R}^k$, and $f(u_i) = y_i$, $i = 1, \ldots, m$. At each u_i we can find a vector g_i such that

$$f(z) \geq f(u_i) + g_i^T(z - u_i) \tag{6.20}$$

for all z. If f is differentiable, we can take $g_i = \nabla f(u_i)$; in the more general case, we can construct g_i by finding a supporting hyperplane to $\mathbf{epi}\, f$ at (u_i, y_i). (The vectors g_i are called *subgradients*.) By applying (6.20) to $z = u_j$, we obtain (6.19).

Conversely, suppose $g_1, \ldots, g_m$ satisfy (6.19). Define f as

$$f(z) = \max_{i=1,\ldots,m} (y_i + g_i^T(z - u_i))$$

for all $z \in \mathbf{R}^k$. Clearly, f is a (piecewise-linear) convex function. The inequalities (6.19) imply that $f(u_i) = y_i$, for $i = 1, \ldots, m$.

We can use this result to solve several problems involving interpolation, approximation, or bounding, with convex functions.

Fitting a convex function to given data

Perhaps the simplest application is to compute the least-squares fit of a convex function to given data (u_i, y_i), $i = 1, \ldots, m$:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m (y_i - f(u_i))^2 \\ \text{subject to} & f : \mathbf{R}^k \to \mathbf{R} \text{ is convex}, \quad \mathbf{dom}\, f = \mathbf{R}^k. \end{array}$$

This is an infinite-dimensional problem, since the variable is f, which is in the space of continuous real-valued functions on $\mathbf{R}^k$. Using the result above, we can formulate this problem as

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m (y_i - \hat{y}_i)^2 \\ \text{subject to} & \hat{y}_j \geq \hat{y}_i + g_i^T(u_j - u_i), \quad i, \ j = 1, \ldots, m, \end{array}$$

which is a QP with variables $\hat{y} \in \mathbf{R}^m$ and $g_1, \ldots, g_m \in \mathbf{R}^k$. The optimal value of this problem is zero if and only if the given data can be interpolated by a convex function, *i.e.*, if there is a convex function that satisfies $f(u_i) = y_i$. An example is shown in figure 6.24.

Bounding values of an interpolating convex function

As another simple example, suppose that we are given data (u_i, y_i), $i = 1, \ldots, m$, which can be interpolated by a convex function. We would like to determine the range of possible values of $f(u_0)$, where u_0 is another point in $\mathbf{R}^k$, and f is any convex function that interpolates the given data. To find the smallest possible value of $f(u_0)$ we solve the LP

$$\begin{array}{ll} \text{minimize} & y_0 \\ \text{subject to} & y_j \geq y_i + g_i^T(u_j - u_i), \quad i, \ j = 0, \ldots, m, \end{array}$$

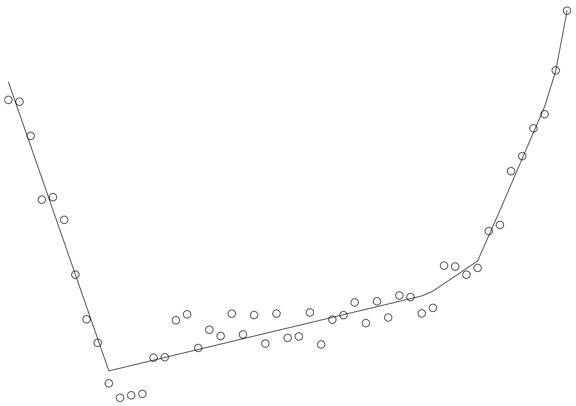

Figure 6.24 Least-squares fit of a convex function to data, shown as circles. The (piecewise-linear) function shown minimizes the sum of squared fitting error, over all convex functions.

which is an LP with variables $y_0 \in \mathbf{R}$, $g_0, \ldots, g_m \in \mathbf{R}^k$. By maximizing y_0 (which is also an LP) we find the largest possible value of $f(u_0)$ for a convex function that interpolates the given data.

Interpolation with monotone convex functions

As an extension of convex interpolation, we can consider interpolation with a convex and monotone nondecreasing function. It can be shown that there exists a convex function $f : \mathbf{R}^k \to \mathbf{R}$, with $\mathbf{dom}\, f = \mathbf{R}^k$, that satisfies the interpolation conditions

$$f(u_i) = y_i, \quad i = 1, \ldots, m,$$

and is monotone nondecreasing (*i.e.*, $f(u) \geq f(v)$ whenever $u \succeq v$), if and only if there exist $g_1, \ldots, g_m \in \mathbf{R}^k$, such that

$$g_i \succeq 0, \quad i = 1, \ldots, m, \qquad y_j \geq y_i + g_i^T(u_j - u_i), \quad i,\ j = 1, \ldots, m. \tag{6.21}$$

In other words, we add to the convex interpolation conditions (6.19), the condition that the subgradients g_i are all nonnegative. (See exercise 6.12.)

Bounding consumer preference

As an application, we consider a problem of predicting consumer preferences. We consider different *baskets of goods*, consisting of different amounts of n consumer goods. A goods basket is specified by a vector $x \in [0, 1]^n$ where x_i denotes the amount of consumer good i. We assume the amounts are normalized so that $0 \leq x_i \leq 1$, *i.e.*, $x_i = 0$ is the minimum and $x_i = 1$ is the maximum possible amount of good i. Given two baskets of goods x and $\tilde{x}$, a consumer can either prefer x to $\tilde{x}$, or prefer $\tilde{x}$ to x, or consider x and $\tilde{x}$ equally attractive. We consider one model consumer, whose choices are repeatable.

We model consumer preference in the following way. We assume there is an underlying *utility function* $u : \mathbf{R}^n \to \mathbf{R}$, with domain $[0,1]^n$; $u(x)$ gives a measure of the utility derived by the consumer from the goods basket x. Given a choice between two baskets of goods, the consumer chooses the one that has larger utility, and will be ambivalent when the two baskets have equal utility. It is reasonable to assume that u is monotone nondecreasing. This means that the consumer always prefers to have more of any good, with the amounts of all other goods the same. It is also reasonable to assume that u is concave. This models *satiation*, or decreasing marginal utility as we increase the amount of goods.

Now suppose we are given some consumer preference data, but we do not know the underlying utility function u. Specifically, we have a set of goods baskets $a_1, \ldots, a_m \in [0,1]^n$, and some information about preferences among them:

$$u(a_i) > u(a_j) \text{ for } (i,j) \in \mathcal{P}, \qquad u(a_i) \geq u(a_j) \text{ for } (i,j) \in \mathcal{P}_{\text{weak}}, \tag{6.22}$$

where $\mathcal{P},\ \mathcal{P}_{\text{weak}} \subseteq \{1,\ldots,m\} \times \{1,\ldots,m\}$ are given. Here $\mathcal{P}$ gives the set of known preferences: $(i,j) \in \mathcal{P}$ means that basket a_i is known to be preferred to basket a_j. The set $\mathcal{P}_{\text{weak}}$ gives the set of known weak preferences: $(i,j) \in \mathcal{P}_{\text{weak}}$ means that basket a_i is preferred to basket a_j, or that the two baskets are equally attractive.

We first consider the following question: How can we determine if the given data are consistent, *i.e.*, whether or not there exists a concave nondecreasing utility function u for which (6.22) holds? This is equivalent to solving the feasibility problem

$$\begin{array}{ll} \text{find} & u \\ \text{subject to} & u : \mathbf{R}^n \to \mathbf{R} \text{ concave and nondecreasing} \\ & u(a_i) > u(a_j), \quad (i,j) \in \mathcal{P} \\ & u(a_i) \geq u(a_j), \quad (i,j) \in \mathcal{P}_{\text{weak}}, \end{array} \tag{6.23}$$

with the function u as the (infinite-dimensional) optimization variable. Since the constraints in (6.23) are all homogeneous, we can express the problem in the equivalent form

$$\begin{array}{ll} \text{find} & u \\ \text{subject to} & u : \mathbf{R}^n \to \mathbf{R} \text{ concave and nondecreasing} \\ & u(a_i) \geq u(a_j) + 1, \quad (i,j) \in \mathcal{P} \\ & u(a_i) \geq u(a_j), \quad (i,j) \in \mathcal{P}_{\text{weak}}, \end{array} \tag{6.24}$$

which uses only nonstrict inequalities. (It is clear that if u satisfies (6.24), then it must satisfy (6.23); conversely, if u satisfies (6.23), then it can be scaled to satisfy (6.24).) This problem, in turn, can be cast as a (finite-dimensional) linear programming feasibility problem, using the interpolation result on page 339:

$$\begin{array}{ll} \text{find} & u_1, \ldots, u_m,\ g_1, \ldots, g_m \\ \text{subject to} & g_i \succeq 0, \quad i = 1, \ldots, m \\ & u_j \leq u_i + g_i^T (a_j - a_i), \quad i,j = 1, \ldots, m \\ & u_i \geq u_j + 1, \quad (i,j) \in \mathcal{P} \\ & u_i \geq u_j, \quad (i,j) \in \mathcal{P}_{\text{weak}}. \end{array} \tag{6.25}$$

By solving this linear programming feasibility problem, we can determine whether there exists a concave, nondecreasing utility function that is consistent with the given sets of strict and nonstrict preferences. If (6.25) is feasible, there is at least one such utility function (and indeed, we can construct one that is piecewise-linear, from a feasible $u_1, \ldots, u_m$, $g_1, \ldots, g_m$). If (6.25) is not feasible, we can conclude that there is no concave increasing utility function that is consistent with the given sets of strict and nonstrict preferences.

As an example, suppose that $\mathcal{P}$ and $\mathcal{P}_{\mathrm{weak}}$ are consumer preferences that are known to be consistent with at least one concave increasing utility function. Consider a pair (k, l) that is not in $\mathcal{P}$ or $\mathcal{P}_{\mathrm{weak}}$, *i.e.*, consumer preference between baskets k and l is not known. In some cases we can conclude that a preference holds between basket k and l, even without knowing the underlying preference function. To do this we augment the known preferences (6.22) with the inequality $u(a_k) \leq u(a_l)$, which means that basket l is preferred to basket k, or they are equally attractive. We then solve the feasibility linear program (6.25), including the extra weak preference $u(a_k) \leq u(a_l)$. If the augmented set of preferences is infeasible, it means that any concave nondecreasing utility function that is consistent with the original given consumer preference data must also satisfy $u(a_k) > u(a_l)$. In other words, we can conclude that basket k is preferred to basket l, without knowing the underlying utility function.

Example 6.9 Here we give a simple numerical example that illustrates the discussion above. We consider baskets of two goods (so we can easily plot the goods baskets). To generate the consumer preference data $\mathcal{P}$, we compute 40 random points in $[0, 1]^2$, and then compare them using the utility function

$$u(x_1, x_2) = (1.1x_1^{1/2} + 0.8x_2^{1/2})/1.9.$$

These goods baskets, and a few level curves of the utility function u, are shown in figure 6.25.

We now use the consumer preference data (but not, of course, the true utility function u) to compare each of these 40 goods baskets to the basket $a_0 = (0.5, 0.5)$. For each original basket a_i, we solve the linear programming feasibility problem described above, to see if we can conclude that basket a_0 is preferred to basket a_i. Similarly, we check whether we can conclude that basket a_i is preferred to basket a_0. For each basket a_i, there are three possible outcomes: we can conclude that a_0 is definitely preferred to a_i, that a_i is definitely preferred to a_0, or (if both LP feasibility problems are feasible) that no conclusion is possible. (Here, *definitely preferred* means that the preference holds for any concave nondecreasing utility function that is consistent with the original given data.)

We find that 21 of the baskets are definitely rejected in favor of $(0.5, 0.5)$, and 14 of the baskets are definitely preferred. We cannot make any conclusion, from the consumer preference data, about the remaining 5 baskets. These results are shown in figure 6.26. Note that goods baskets below and to the left of $(0.5, 0.5)$ will definitely be rejected in favor of $(0.5, 0.5)$, using only the monotonicity property of the utility function, and similarly, those points that are above and to the right of $(0.5, 0.5)$ must be preferred. So for these 17 points, there is no need to solve the feasibility LP (6.25). Classifying the 23 points in the other two quadrants, however, requires the concavity assumption, and solving the feasibility LP (6.25).

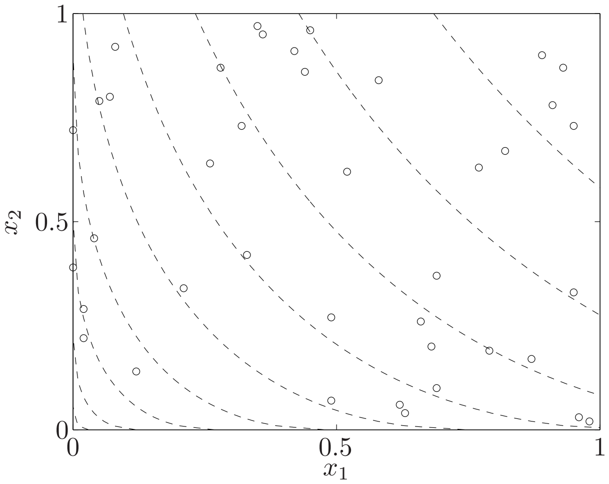

Figure 6.25 Forty goods baskets $a_1, \ldots, a_{40}$, shown as circles. The $0.1, 0.2, \ldots, 0.9$ level curves of the true utility function u are shown as dashed lines. This utility function is used to find the consumer preference data $\mathcal{P}$ among the 40 baskets.

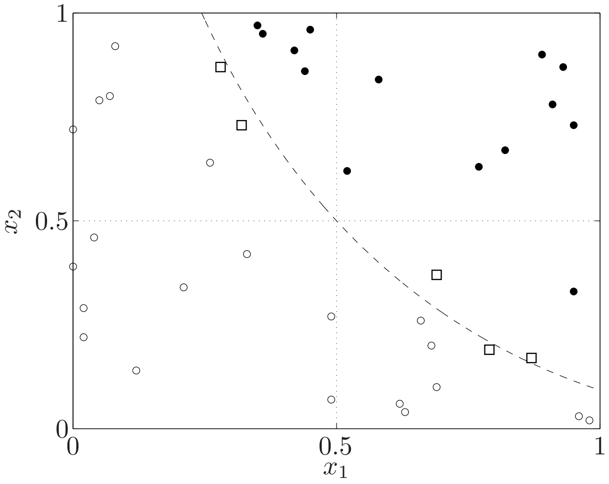

Figure 6.26 Results of consumer preference analysis using the LP (6.25), for a new goods basket $a_0 = (0.5, 0.5)$. The original baskets are displayed as open circles if they are definitely rejected ($u(a_k) < u(a_0)$), as solid black circles if they are definitely preferred ($u(a_k) > u(a_0)$), and as squares when no conclusion can be made. The level curve of the underlying utility function, that passes through $(0.5, 0.5)$, is shown as a dashed curve. The vertical and horizontal lines passing through $(0.5, 0.5)$ divide $[0, 1]^2$ into four quadrants. Points in the upper right quadrant must be preferred to $(0.5, 0.5)$, by the monotonicity assumption on u. Similarly, $(0.5, 0.5)$ must be preferred to the points in the lower left quadrant. For the points in the other two quadrants, the results are not obvious.

Bibliography

The robustness properties of approximations with different penalty functions were analyzed by Huber [Hub64, Hub81], who also proposed the penalty function (6.4). The log-barrier penalty function arises in control theory, where it is applied to the system closed-loop frequency response, and has several names, *e.g.*, *central* $\mathbf{H}_\infty$, or *risk-averse* control; see Boyd and Barratt [BB91] and the references therein.

Regularized approximation is covered in many books, including Tikhonov and Arsenin [TA77] and Hansen [Han98]. Tikhonov regularization is sometimes called *ridge regression* (Golub and Van Loan [GL89, page 564]). Least-squares approximation with ℓ_1-norm regularization is also known under the name *lasso* (Tibshirani [Tib96]). Other least-squares regularization and regressor selection techniques are discussed and compared in Hastie, Tibshirani, and Friedman [HTF01, §3.4].

Total variation denoising was introduced for image reconstruction by Rudin, Osher, and Fatemi [ROF92].

The robust least-squares problem with norm bounded uncertainty (page 321) was introduced by El Ghaoui and Lebret [EL97], and Chandrasekaran, Golub, Gu, and Sayed [CGGS98]. El Ghaoui and Lebret also give the SDP formulation of the robust least-squares problem with structured uncertainty (page 323).

Chen, Donoho, and Saunders [CDS01] discuss basis pursuit via linear programming. They refer to the ℓ_1-norm regularized problem (6.18) as *basis pursuit denoising*. Meyer and Pratt [MP68] is an early paper on the problem of bounding utility functions.

Exercises

Norm approximation and least-norm problems

6.1 *Quadratic bounds for log barrier penalty.* Let $\phi : \mathbf{R} \to \mathbf{R}$ be the log barrier penalty function with limit $a > 0$:

$$\phi(u) = \begin{cases} -a^2 \log(1-(u/a)^2) & |u| < a \\ \infty & \text{otherwise.} \end{cases}$$

Show that if $u \in \mathbf{R}^m$ satisfies $\|u\|_\infty < a$, then

$$\|u\|_2^2 \leq \sum_{i=1}^m \phi(u_i) \leq \frac{\phi(\|u\|_\infty)}{\|u\|_\infty^2}\|u\|_2^2.$$

This means that $\sum_{i=1}^m \phi(u_i)$ is well approximated by $\|u\|_2^2$ if $\|u\|_\infty$ is small compared to a. For example, if $\|u\|_\infty/a = 0.25$, then

$$\|u\|_2^2 \leq \sum_{i=1}^m \phi(u_i) \leq 1.033 \cdot \|u\|_2^2.$$

6.2 *ℓ_1-, ℓ_2-, and ℓ_∞-norm approximation by a constant vector.* What is the solution of the norm approximation problem with one scalar variable $x \in \mathbf{R}$,

$$\text{minimize} \quad \|x\mathbf{1} - b\|,$$

for the ℓ_1-, ℓ_2-, and ℓ_∞-norms?

6.3 Formulate the following approximation problems as LPs, QPs, SOCPs, or SDPs. The problem data are $A \in \mathbf{R}^{m\times n}$ and $b \in \mathbf{R}^m$. The rows of A are denoted a_i^T.

(a) *Deadzone-linear penalty approximation*: minimize $\sum_{i=1}^m \phi(a_i^T x - b_i)$, where

$$\phi(u) = \begin{cases} 0 & |u| \leq a \\ |u| - a & |u| > a, \end{cases}$$

where $a > 0$.

(b) *Log-barrier penalty approximation*: minimize $\sum_{i=1}^m \phi(a_i^T x - b_i)$, where

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

with $a > 0$.

(c) *Huber penalty approximation*: minimize $\sum_{i=1}^m \phi(a_i^T x - b_i)$, where

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

with $M > 0$.

(d) *Log-Chebyshev approximation*: minimize $\max_{i=1,\ldots,m} |\log(a_i^T x) - \log b_i|$. We assume $b \succ 0$. An equivalent convex form is

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & 1/t \leq a_i^T x/b_i \leq t, \quad i = 1, \ldots, m, \end{array}$$

with variables $x \in \mathbf{R}^n$ and $t \in \mathbf{R}$, and domain $\mathbf{R}^n \times \mathbf{R}_{++}$.

(e) Minimizing the sum of the largest k residuals:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^k |r|_{[i]} \\ \text{subject to} & r = Ax - b, \end{array}$$

where $|r|_{[1]} \geq |r|_{[2]} \geq \cdots \geq |r|_{[m]}$ are the numbers $|r_1|$, $|r_2|$, $\ldots$, $|r_m|$ sorted in decreasing order. (For $k = 1$, this reduces to ℓ_∞-norm approximation; for $k = m$, it reduces to ℓ_1-norm approximation.) *Hint.* See exercise 5.19.

6.4 *A differentiable approximation of ℓ_1-norm approximation.* The function $\phi(u) = (u^2+\epsilon)^{1/2}$, with parameter $\epsilon > 0$, is sometimes used as a differentiable approximation of the absolute value function $|u|$. To approximately solve the ℓ_1-norm approximation problem

$$\text{minimize} \quad \|Ax - b\|_1, \tag{6.26}$$

where $A \in \mathbf{R}^{m \times n}$, we solve instead the problem

$$\text{minimize} \quad \sum_{i=1}^m \phi(a_i^T x - b_i), \tag{6.27}$$

where a_i^T is the ith row of A. We assume $\mathbf{rank}\, A = n$.

Let $p^\star$ denote the optimal value of the ℓ_1-norm approximation problem (6.26). Let $\hat{x}$ denote the optimal solution of the approximate problem (6.27), and let $\hat{r}$ denote the associated residual, $\hat{r} = A\hat{x} - b$.

(a) Show that $p^\star \geq \sum_{i=1}^m \hat{r}_i^2/(\hat{r}_i^2 + \epsilon)^{1/2}$.

(b) Show that

$$\|A\hat{x} - b\|_1 \leq p^\star + \sum_{i=1}^m |\hat{r}_i| \left(1 - \frac{|\hat{r}_i|}{(\hat{r}_i^2 + \epsilon)^{1/2}}\right).$$

(By evaluating the righthand side after computing $\hat{x}$, we obtain a bound on how suboptimal $\hat{x}$ is for the ℓ_1-norm approximation problem.)

6.5 *Minimum length approximation.* Consider the problem

$$\begin{array}{ll} \text{minimize} & \text{length}(x) \\ \text{subject to} & \|Ax - b\| \leq \epsilon, \end{array}$$

where $\text{length}(x) = \min\{k \mid x_i = 0 \text{ for } i > k\}$. The problem variable is $x \in \mathbf{R}^n$; the problem parameters are $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^m$, and $\epsilon > 0$. In a regression context, we are asked to find the minimum number of columns of A, taken in order, that can approximate the vector b within ϵ.

Show that this is a quasiconvex optimization problem.

6.6 *Duals of some penalty function approximation problems.* Derive a Lagrange dual for the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \phi(r_i) \\ \text{subject to} & r = Ax - b, \end{array}$$

for the following penalty functions $\phi : \mathbf{R} \to \mathbf{R}$. The variables are $x \in \mathbf{R}^n$, $r \in \mathbf{R}^m$.

(a) *Deadzone-linear penalty* (with deadzone width $a = 1$),

$$\phi(u) = \left\{ \begin{array}{ll} 0 & |u| \leq 1 \\ |u| - 1 & |u| > 1. \end{array} \right.$$

(b) *Huber penalty* (with $M = 1$),

$$\phi(u) = \left\{ \begin{array}{ll} u^2 & |u| \leq 1 \\ 2|u| - 1 & |u| > 1. \end{array} \right.$$

(c) *Log-barrier* (with limit $a = 1$),

$$\phi(u) = -\log(1 - u^2), \qquad \mathbf{dom}\,\phi = (-1, 1).$$

(d) *Relative deviation from one,*

$$\phi(u) = \max\{u, 1/u\} = \begin{cases} u & u \geq 1 \\ 1/u & u \leq 1, \end{cases}$$

with $\mathbf{dom}\,\phi = \mathbf{R}_{++}$.

Regularization and robust approximation

6.7 *Bi-criterion optimization with Euclidean norms.* We consider the bi-criterion optimization problem

$$\text{minimize (w.r.t. } \mathbf{R}^2_+\text{)} \quad (\|Ax - b\|_2^2, \|x\|_2^2),$$

where $A \in \mathbf{R}^{m \times n}$ has rank r, and $b \in \mathbf{R}^m$. Show how to find the solution of each of the following problems from the singular value decomposition of A,

$$A = U\,\mathbf{diag}(\sigma)V^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

(see §A.5.4).

(a) *Tikhonov regularization:* minimize $\|Ax - b\|_2^2 + \delta\|x\|_2^2$.

(b) Minimize $\|Ax - b\|_2^2$ subject to $\|x\|_2^2 = \gamma$.

(c) Maximize $\|Ax - b\|_2^2$ subject to $\|x\|_2^2 = \gamma$.

Here δ and γ are positive parameters.

Your results provide efficient methods for computing the optimal trade-off curve and the set of achievable values of the bi-criterion problem.

6.8 Formulate the following robust approximation problems as LPs, QPs, SOCPs, or SDPs. For each subproblem, consider the ℓ_1-, ℓ_2-, and the ℓ_∞-norms.

(a) *Stochastic robust approximation with a finite set of parameter values*, *i.e.*, the sum-of-norms problem

$$\text{minimize} \quad \textstyle\sum_{i=1}^{k} p_i\|A_i x - b\|$$

where $p \succeq 0$ and $\mathbf{1}^T p = 1$. (See §6.4.1.)

(b) *Worst-case robust approximation with coefficient bounds:*

$$\text{minimize} \quad \sup_{A \in \mathcal{A}} \|Ax - b\|$$

where

$$\mathcal{A} = \{A \in \mathbf{R}^{m \times n} \mid l_{ij} \leq a_{ij} \leq u_{ij},\ i = 1, \ldots, m,\ j = 1, \ldots, n\}.$$

Here the uncertainty set is described by giving upper and lower bounds for the components of A. We assume $l_{ij} < u_{ij}$.

(c) *Worst-case robust approximation with polyhedral uncertainty:*

$$\text{minimize} \quad \sup_{A \in \mathcal{A}} \|Ax - b\|$$

where

$$\mathcal{A} = \{[a_1 \ \cdots \ a_m]^T \mid C_i a_i \preceq d_i,\ i = 1, \ldots, m\}.$$

The uncertainty is described by giving a polyhedron $\mathcal{P}_i = \{a_i \mid C_i a_i \preceq d_i\}$ of possible values for each row. The parameters $C_i \in \mathbf{R}^{p_i \times n}$, $d_i \in \mathbf{R}^{p_i}$, $i = 1, \ldots, m$, are given. We assume that the polyhedra $\mathcal{P}_i$ are nonempty and bounded.

Function fitting and interpolation

6.9 *Minimax rational function fitting.* Show that the following problem is quasiconvex:

$$\text{minimize} \quad \max_{i=1,\ldots,k} \left| \frac{p(t_i)}{q(t_i)} - y_i \right|$$

where

$$p(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_m t^m, \qquad q(t) = 1 + b_1 t + \cdots + b_n t^n,$$

and the domain of the objective function is defined as

$$D = \{(a,b) \in \mathbf{R}^{m+1} \times \mathbf{R}^n \mid q(t) > 0, \ \alpha \leq t \leq \beta\}.$$

In this problem we fit a rational function $p(t)/q(t)$ to given data, while constraining the denominator polynomial to be positive on the interval $[\alpha, \beta]$. The optimization variables are the numerator and denominator coefficients a_i, b_i. The interpolation points $t_i \in [\alpha, \beta]$, and desired function values y_i, $i = 1, \ldots, k$, are given.

6.10 *Fitting data with a concave nonnegative nondecreasing quadratic function.* We are given the data

$$x_1, \ldots, x_N \in \mathbf{R}^n, \qquad y_1, \ldots, y_N \in \mathbf{R},$$

and wish to fit a quadratic function of the form

$$f(x) = (1/2)x^T P x + q^T x + r,$$

where $P \in \mathbf{S}^n$, $q \in \mathbf{R}^n$, and $r \in \mathbf{R}$ are the parameters in the model (and, therefore, the variables in the fitting problem).

Our model will be used only on the box $\mathcal{B} = \{x \in \mathbf{R}^n \mid l \preceq x \preceq u\}$. You can assume that $l \prec u$, and that the given data points x_i are in this box.

We will use the simple sum of squared errors objective,

$$\sum_{i=1}^{N} (f(x_i) - y_i)^2,$$

as the criterion for the fit. We also impose several constraints on the function f. First, it must be concave. Second, it must be nonnegative on $\mathcal{B}$, *i.e.*, $f(z) \geq 0$ for all $z \in \mathcal{B}$. Third, f must be nondecreasing on $\mathcal{B}$, *i.e.*, whenever z, $\tilde{z} \in \mathcal{B}$ satisfy $z \preceq \tilde{z}$, we have $f(z) \leq f(\tilde{z})$.

Show how to formulate this fitting problem as a convex problem. Simplify your formulation as much as you can.

6.11 *Least-squares direction interpolation.* Suppose $F_1, \ldots, F_n : \mathbf{R}^k \to \mathbf{R}^p$, and we form the linear combination $F : \mathbf{R}^k \to \mathbf{R}^p$,

$$F(u) = x_1 F_1(u) + \cdots + x_n F_n(u),$$

where x is the variable in the interpolation problem.

In this problem we require that $\angle(F(v_j), q_j) = 0$, $j = 1, \ldots, m$, where q_j are given vectors in $\mathbf{R}^p$, which we assume satisfy $\|q_j\|_2 = 1$. In other words, we require the direction of F to take on specified values at the points v_j. To ensure that $F(v_j)$ is not zero (which makes the angle undefined), we impose the minimum length constraints $\|F(v_j)\|_2 \geq \epsilon$, $j = 1, \ldots, m$, where $\epsilon > 0$ is given.

Show how to find x that minimizes $\|x\|^2$, and satisfies the direction (and minimum length) conditions above, using convex optimization.

6.12 *Interpolation with monotone functions.* A function $f : \mathbf{R}^k \to \mathbf{R}$ is monotone nondecreasing (with respect to $\mathbf{R}^k_+$) if $f(u) \geq f(v)$ whenever $u \succeq v$.

(a) Show that there exists a monotone nondecreasing function $f : \mathbf{R}^k \to \mathbf{R}$, that satisfies $f(u_i) = y_i$ for $i = 1, \dots, m$, if and only if

$$y_i \geq y_j \text{ whenever } u_i \succeq u_j, \quad i,\ j = 1, \dots, m.$$

(b) Show that there exists a convex monotone nondecreasing function $f : \mathbf{R}^k \to \mathbf{R}$, with $\mathbf{dom}\, f = \mathbf{R}^k$, that satisfies $f(u_i) = y_i$ for $i = 1, \dots, m$, if and only if there exist $g_i \in \mathbf{R}^k$, $i = 1, \dots, m$, such that

$$g_i \succeq 0, \quad i = 1, \dots, m, \qquad y_j \geq y_i + g_i^T(u_j - u_i), \quad i,\ j = 1, \dots, m.$$

6.13 *Interpolation with quasiconvex functions.* Show that there exists a quasiconvex function $f : \mathbf{R}^k \to \mathbf{R}$, that satisfies $f(u_i) = y_i$ for $i = 1, \dots, m$, if and only if there exist $g_i \in \mathbf{R}^k$, $i = 1, \dots, m$, such that

$$g_i^T(u_j - u_i) \leq -1 \text{ whenever } y_j < y_i, \quad i,\ j = 1, \dots, m.$$

6.14 [Nes00] *Interpolation with positive-real functions.* Suppose $z_1, \dots, z_n \in \mathbf{C}$ are n distinct points with $|z_i| > 1$. We define $K_{\rm np}$ as the set of vectors $y \in \mathbf{C}^n$ for which there exists a function $f : \mathbf{C} \to \mathbf{C}$ that satisfies the following conditions.

- f is *positive-real*, which means it is analytic outside the unit circle (*i.e.*, for $|z| > 1$), and its real part is nonnegative outside the unit circle ($\Re f(z) \geq 0$ for $|z| > 1$).
- f satisfies the *interpolation conditions*

$$f(z_1) = y_1, \qquad f(z_2) = y_2, \qquad \dots, \qquad f(z_n) = y_n.$$

If we denote the set of positive-real functions as $\mathcal{F}$, then we can express $K_{\rm np}$ as

$$K_{\rm np} = \{y \in \mathbf{C}^n \mid \exists f \in \mathcal{F},\ y_k = f(z_k),\ k = 1, \dots, n\}.$$

(a) It can be shown that f is positive-real if and only if there exists a nondecreasing function ρ such that for all z with $|z| > 1$,

$$f(z) = i\Im f(\infty) + \int_0^{2\pi} \frac{e^{i\theta} + z^{-1}}{e^{i\theta} - z^{-1}}\, d\rho(\theta),$$

where $i = \sqrt{-1}$ (see [KN77, page 389]). Use this representation to show that $K_{\rm np}$ is a closed convex cone.

(b) We will use the inner product $\Re(x^H y)$ between vectors $x, y \in \mathbf{C}^n$, where x^H denotes the complex conjugate transpose of x. Show that the dual cone of $K_{\rm np}$ is given by

$$K_{\rm np}^* = \left\{ x \in \mathbf{C}^n \;\middle|\; \Im(\mathbf{1}^T x) = 0,\ \Re\left(\sum_{l=1}^n x_l \frac{e^{-i\theta} + \bar{z}_l^{-1}}{e^{-i\theta} - \bar{z}_l^{-1}} \right) \geq 0\ \forall \theta \in [0, 2\pi] \right\}.$$

(c) Show that

$$K_{\rm np}^* = \left\{ x \in \mathbf{C}^n \;\middle|\; \exists Q \in \mathbf{H}_+^n,\ x_l = \sum_{k=1}^n \frac{Q_{kl}}{1 - z_k^{-1}\bar{z}_l^{-1}},\ l = 1, \dots, n \right\}$$

where $\mathbf{H}_+^n$ denotes the set of positive semidefinite Hermitian matrices of size $n \times n$. Use the following result (known as *Riesz-Fejér theorem*; see [KN77, page 60]). A function of the form

$$\sum_{k=0}^n (y_k e^{-ik\theta} + \bar{y}_k e^{ik\theta})$$

is nonnegative for all θ if and only if there exist $a_0, \dots, a_n \in \mathbf{C}$ such that

$$\sum_{k=0}^n (y_k e^{-ik\theta} + \bar{y}_k e^{ik\theta}) = \left| \sum_{k=0}^n a_k e^{ik\theta} \right|^2.$$

(d) Show that $K_{\text{np}} = \{y \in \mathbf{C}^n \mid P(y) \succeq 0\}$ where $P(y) \in \mathbf{H}^n$ is defined as

$$P(y)_{kl} = \frac{y_k + \overline{y}_l}{1 - z_k^{-1}\bar{z}_l^{-1}}, \quad l, k = 1, \ldots, n.$$

The matrix $P(y)$ is called the *Nevanlinna-Pick matrix* associated with the points z_k, y_k.

Hint. As we noted in part (a), K_{np} is a closed convex cone, so $K_{\text{np}} = K_{\text{np}}^{**}$.

(e) As an application, pose the following problem as a convex optimization problem:

$$\begin{array}{ll} \text{minimize} & \sum_{k=1}^n |f(z_k) - w_k|^2 \\ \text{subject to} & f \in \mathcal{F}. \end{array}$$

The problem data are n points z_k with $|z_k| > 1$ and n complex numbers $w_1, \ldots,$ w_n. We optimize over all positive-real functions f.